Ansible

Cybersécurité et Automatisation avec Ansible

Job 1 – Prise en main et premières automatisations

Objectif

Ce premier job a pour but d'initier l'étudiant à **Ansible**, en installant les bases nécessaires pour automatiser des tâches sur un parc de machines Linux. À la fin de ce job, l'étudiant saura :

- Installer Ansible sur une machine de contrôle.
- Établir une connexion SSH sans mot de passe.
- Structurer un inventaire de serveurs.
- Lancer ses premières commandes et playbooks simples.

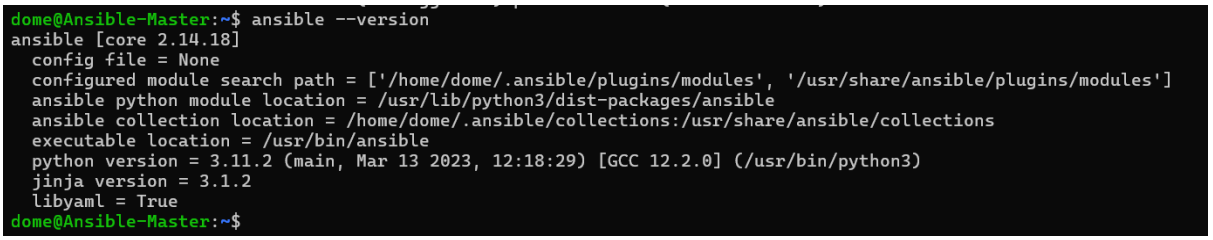
1. Installation d'Ansible

Sur une machine sous **Debian/Ubuntu** (ex : `Ubuntu Server` ou `Debian`), exécuter les commandes suivantes :

```
sudo apt update
sudo apt install ansible -y
```

Vérification

```
ansible --version
```



Sortie attendue :

```
ansible [core 2.x.x]
  config file = /etc/ansible/ansible.cfg
  ...
```

Gestion des droits sudo pour Ansible

Pour permettre à Ansible d'exécuter des tâches nécessitant des privilèges administrateurs (comme le redémarrage de services ou l'installation de paquets), il est nécessaire que l'utilisateur distant puisse utiliser `sudo` **sans saisie de mot de passe**. Cette configuration est essentielle pour garantir une automatisation fluide.

La méthode recommandée consiste à créer un fichier dédié dans le répertoire `/etc/sudoers.d/`, ce qui permet d'éviter de modifier directement le fichier principal `/etc/sudoers`.

Voici les étapes réalisées :

1. Connexion à chaque machine cible en SSH.

Création d'un fichier spécifique, par exemple `/etc/sudoers.d/ansible_user` :

```
echo "dome ALL=(ALL) NOPASSWD: ALL" | sudo tee
/etc/sudoers.d/ansible_user
```

2. Remarques :
    - `dome` correspond à l'utilisateur utilisé par Ansible pour se connecter.
    - Le mot-clé `NOPASSWD` autorise l'exécution de `sudo` sans demander de mot de passe.
    - Il est recommandé d'utiliser `visudo -c` pour vérifier la syntaxe, ou de créer ce fichier avec `sudo visudo -f /etc/sudoers.d/ansible_user`.

Attribution des bons droits :
```
sudo chmod 440 /etc/sudoers.d/ansible_user
```

3. Cette configuration est essentielle dans les environnements d'automatisation pour éviter les blocages lors de l'exécution des playbooks. Elle ne doit être utilisée que dans un environnement de test ou bien sécurisé.

2. Configuration SSH (connexion sans mot de passe)

Permettre à Ansible de se connecter aux machines cibles sans mot de passe, via une clé SSH.

1. **Générer la clé SSH** sur la machine de contrôle :

```
ssh-keygen -t rsa -b 4096
```

Appuyer sur **Entrée** à chaque question pour garder les valeurs par défaut.

2. **Copier la clé publique sur chaque machine cible** :

```
ssh-copy-id user@ip_de_la_machine_cible
```

Répéter pour chaque machine cible.

```
dome@Ansible-Master:~$ ssh-copy-id dome@192.168.40.158
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/dome/.ssh/id_rsa.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
dome@192.168.40.158's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'dome@192.168.40.158'"
and check to make sure that only the key(s) you wanted were added.
```

```
dome@Ansible-Master:~$ ssh-copy-id dome@192.168.40.156
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/dome/.ssh/id_rsa.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
dome@192.168.40.156's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'dome@192.168.40.156'"
and check to make sure that only the key(s) you wanted were added.

dome@Ansible-Master:~$
```

Vérification

```
ssh user@ip_de_la_machine_cible
```

```
dome@Ansible-Master:~$ ssh dome@192.168.40.158
Linux Ansible-Worker1 6.1.0-21-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.90-1 (2024-05-03) 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: Tue Jul 22 10:28:56 2025 from 192.168.40.1
dome@Ansible-Worker1:~$ exit
déconnexion
Connection to 192.168.40.158 closed.
dome@Ansible-Master:~$ ssh dome@192.168.40.156
Linux Ansible-Worker2 6.1.0-21-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.90-1 (2024-05-03) 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: Tue Jul 22 10:29:34 2025 from 192.168.40.1
dome@Ansible-Worker2:~$ exit
déconnexion
Connection to 192.168.40.156 closed.
dome@Ansible-Master:~$
```

Cela doit **se connecter sans demander de mot de passe**.

3. Création de l'inventaire `inventory.ini`

Organiser les machines dans des groupes pour les gérer plus facilement.

Exemple de fichier `inventory.ini`

```
[webservers]
192.168.1.10
192.168.1.11

[dbservers]
192.168.1.20

[all_servers:children]
webservers
dbservers
```

Ici le fichier adapté ma config du moment

---

4. Tests initiaux avec le module ping

Tester la communication avec les machines.

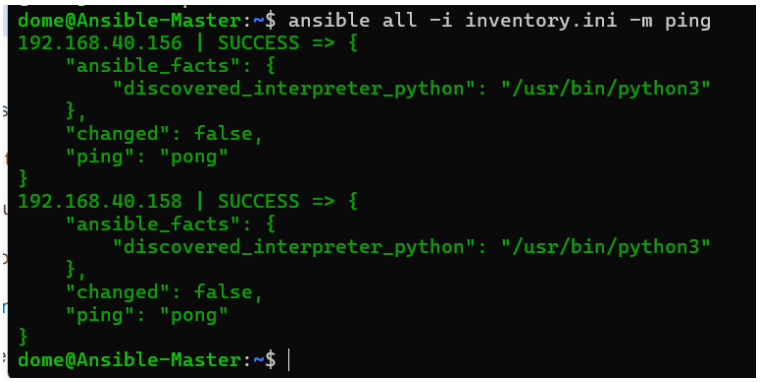
Commande

```
ansible all -i inventory.ini -m ping
```

Sortie attendue

Pour chaque machine :

```
192.168.1.10 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```



5. Premiers playbooks

5.1 Vérifier l'état d'un service (ex: ssh)

playbook `check_service.yml`

```yaml
- hosts: all
  become: true
  tasks:
    - name: Vérifier si le service SSH est actif
      ansible.builtin.service:
        name: ssh
        state: started
```

Commande

```
ansible-playbook -i inventory.ini check_service.yml
```

```
dome@Ansible-Master:~$ ansible-playbook -i inventory.ini check_service.yml

PLAY [all] *********************************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.40.156]
ok: [192.168.40.158]

TASK [Vérifier si le service SSH est actif] ************************************
ok: [192.168.40.158]
ok: [192.168.40.156]

PLAY RECAP *********************************************************************
192.168.40.156             : ok=2    changed=0    unreachable=0    failed=0    skipped=0
    rescued=0    ignored=0
192.168.40.158             : ok=2    changed=0    unreachable=0    failed=0    skipped=0
    rescued=0    ignored=0

dome@Ansible-Master:~$
```

```
dome@Ansible-Master:~$ ansible-inventory -i inventory.ini --list
{
    "_meta": {
        "hostvars": {}
    },
    "all": {
        "children": [
            "ungrouped",
            "all_servers"
        ]
    },
    "all_servers": {
        "children": [
            "workers"
        ]
    },
    "workers": {
        "hosts": [
            "192.168.40.156",
            "192.168.40.158"
        ]
    }
}
```

5.2 Copier un fichier simple

playbook `copy_file.yml`

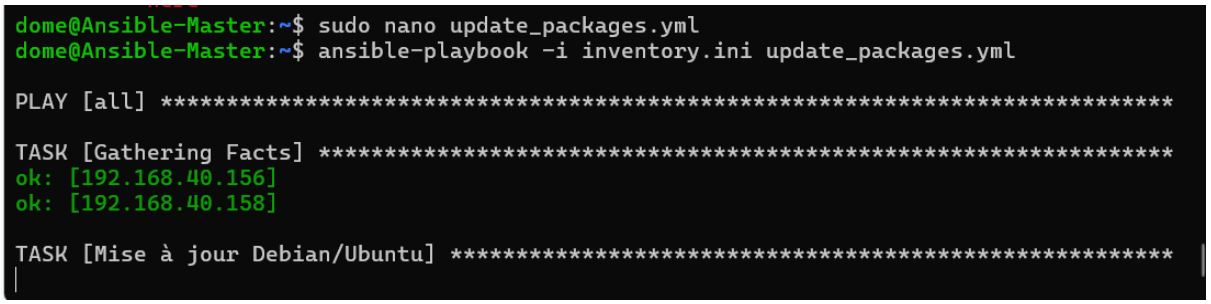
```
- hosts: all
  tasks:
    - name: Copier un fichier de test
      copy:
        src: /home/ansible_admin/test.txt
        dest: test.txt
```

```
dome@Ansible-Master:~$ ansible-playbook -i inventory.ini copy_file.yml

PLAY [all] *********************************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.40.156]
ok: [192.168.40.158]

TASK [Copier un fichier de test] ***********************************************
changed: [192.168.40.158]
changed: [192.168.40.156]

PLAY RECAP *********************************************************************
192.168.40.156             : ok=2    changed=1    unreachable=0    failed=0    skipped=0
    rescued=0    ignored=0
192.168.40.158             : ok=2    changed=1    unreachable=0    failed=0    skipped=0
    rescued=0    ignored=0

dome@Ansible-Master:~$
```

5.3 Mettre à jour tous les paquets

playbook `update_packages.yml`

```
- hosts: all
  become: true
  tasks:
    - name: Mise à jour Debian/Ubuntu
      apt:
        update_cache: yes
        upgrade: dist
      when: ansible_facts['os_family'] == "Debian"

    - name: Mise à jour CentOS/RHEL/Rocky
      yum:
        name: '*'
        state: latest
      when: ansible_facts['os_family'] == "RedHat"
```



Job 2 – Hardening des Systèmes

Objectif pédagogique

Ce Job a pour but d'automatiser le durcissement de la configuration de systèmes Linux à l'aide d'Ansible. Il s'agit d'appliquer des bonnes pratiques de sécurité pour :

- Renforcer la gestion des utilisateurs.
- Sécuriser l'accès SSH.
- Protéger l'accès réseau via un pare-feu.
- Réduire la surface d'attaque en désactivant les services inutiles.

1. Gestion Sécurisée des Utilisateurs et Groupes

Objectif

Supprimer les comptes à risque, et créer un utilisateur administrateur dédié à Ansible avec des droits sudo sécurisés.

1. **Suppression des utilisateurs inutiles** :
    - Exemple : `games`, `irc`, `lp`, `news`, etc.

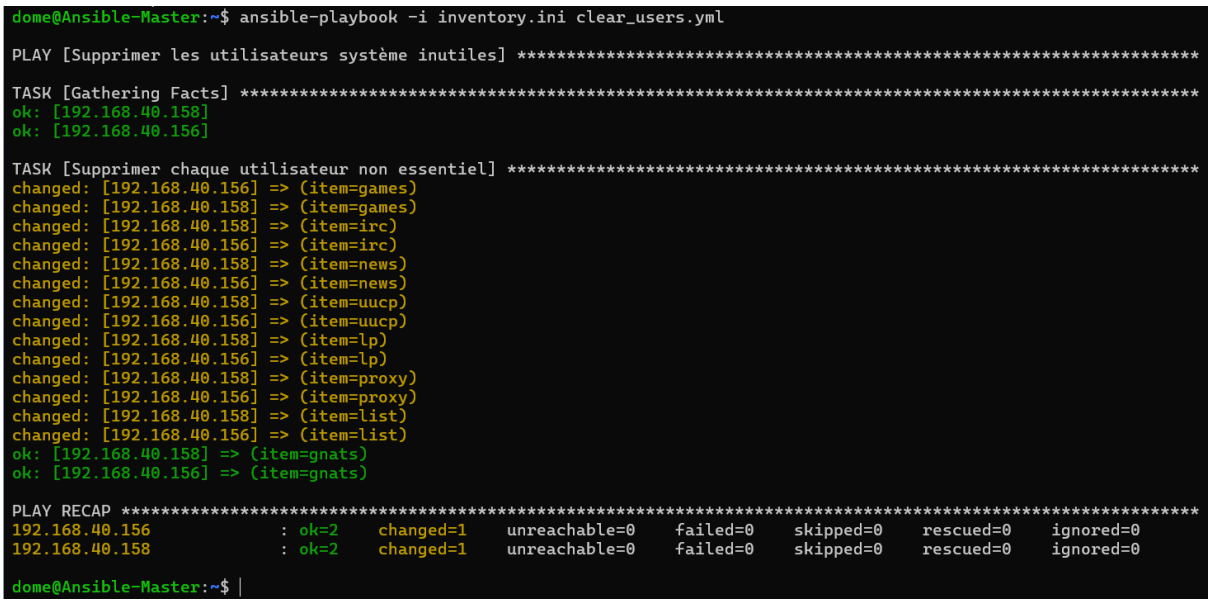
    Playbook (extrait) :

```
- name: Supprimer les utilisateurs système inutiles
  hosts: all
  become: yes
  vars:
    users_to_remove:
      - games
      - irc
      - news
      - uucp
      - lp
      - proxy
      - list
      - gnats
   state: absent
    ignore_errors: yes
```

    Vérifier les comptes restants :
    `getent passwd | cut -d: -f1`
    Vérifier les groupes :
    `getent group`
    S'assurer qu'aucun service critique n'est cassé :
    `sudo systemctl status`

Bonnes pratiques:

- Toujours garder au minimum : root + ansible_admin (ou équivalent).
- Ne pas supprimer les comptes standards utilisés par le système.
- Tester le playbook sur une seule machine avant de le lancer sur toute l'infrastructure.



2. Création de l'utilisateur `ansible_admin` avec mot de passe haché :

Générer un mot de passe chiffré avec SHA-512

Si ce n'est pas encore fait, installe l'outil :

```
sudo apt install whois
```

Puis exécute la commande :

```
mkpasswd --method=SHA-512
```

Entre par exemple : `Ansible2025!`

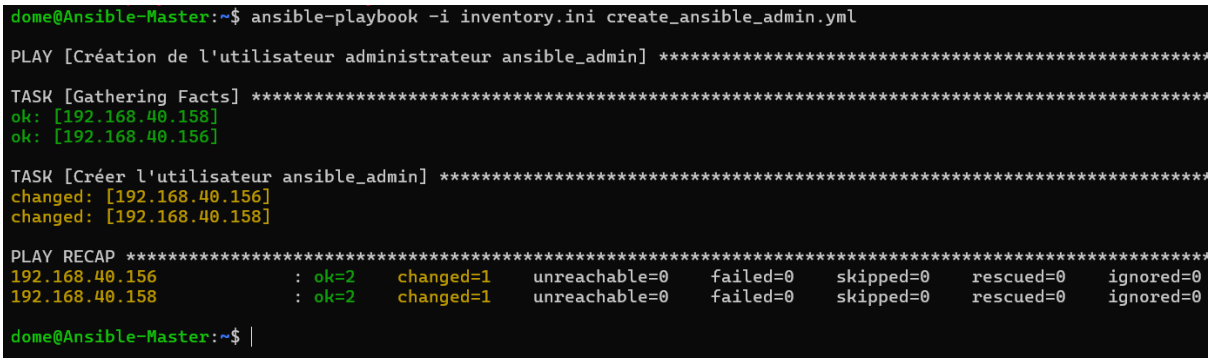
On obtiens une sortie comme :

```
$6$h3LpFDFf$SbRkXcYhTqffL9H9B1CgEFq7...0tqY
```

Copier ce **hash** (c'est lui que tu vas utiliser dans le playbook)

Playbook (extrait) :

```
---
- name: Création de l'utilisateur administrateur ansible_admin
  hosts: all
  become: yes
  vars:
    ansible_admin_password:
"$6$h3LpFDFf$SbRkXcYhTqffL9H9B1CgEFq7...0tqY"

  tasks:
    - name: Créer l'utilisateur ansible_admin
      user:
        name: ansible_admin
        comment: "Utilisateur dédié à Ansible"
        shell: /bin/bash
        password: "{{ ansible_admin_password }}"
        groups: "{{ 'sudo' if ansible_facts['os_family'] == 'Debian'
else 'wheel' }}"
        append: yes
```



Objectif : Limiter l'accès SSH pour réduire les risques d'intrusion.

Procédure

1. Modifier `/etc/ssh/sshd_config` avec Ansible :
    - Désactiver root login
    - Désactiver mot de passe
    - Changer le port par défaut
    - Désactiver le X11Forwarding

Playbook (extrait) :

```
- name: Durcir la configuration SSH
  hosts: all
  become: yes
  tasks:
    - name: Appliquer configuration SSH
      lineinfile:
        path: /etc/ssh/sshd_config
        regexp: '^#?PermitRootLogin'
        line: 'PermitRootLogin no'
      notify: Redémarrer SSH

    - name: Désactiver authentification par mot de passe
      lineinfile:
        path: /etc/ssh/sshd_config
        regexp: '^#?PasswordAuthentication'
        line: 'PasswordAuthentication no'
      notify: Redémarrer SSH

    - name: Modifier le port SSH
      lineinfile:
        path: /etc/ssh/sshd_config
        regexp: '^#?Port'
        line: 'Port 2002'
      notify: Redémarrer SSH

    - name: Désactiver X11Forwarding
      lineinfile:
        path: /etc/ssh/sshd_config
        regexp: '^#?X11Forwarding'
        line: 'X11Forwarding no'
      notify: Redémarrer SSH

  handlers:
    - name: Redémarrer SSH
      service:
        name: sshd
        state: restarted
```

```
dome@Ansible-Master:~$ ansible-playbook -i inventory.ini ssh_config.yml

PLAY [Durcir la configuration SSH] *********************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.40.156]
ok: [192.168.40.158]

TASK [Appliquer configuration SSH] *********************************************************************************
ok: [192.168.40.158]
ok: [192.168.40.156]

TASK [Désactiver authentification par mot de passe] ****************************************************************
ok: [192.168.40.156]
ok: [192.168.40.158]

TASK [Modifier le port SSH] ****************************************************************************************
ok: [192.168.40.156]
ok: [192.168.40.158]

TASK [Désactiver X11Forwarding] ************************************************************************************
ok: [192.168.40.156]
ok: [192.168.40.158]

PLAY RECAP *********************************************************************************************************
192.168.40.156             : ok=5    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.40.158             : ok=5    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

2. Tester la nouvelle connexion :

   `ssh -p 2002 ansible_admin@ip_du_serveur`

```
dome@Ansible-Master:~$ ssh -p 2002 ansible_admin@192.168.40.158
Linux Ansible-Worker1 6.1.0-37-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.140-1 (2025-05-22) 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: Wed Jul 23 15:04:26 2025 from 192.168.40.157
ansible_admin@Ansible-Worker1:~$ exit
déconnexion
Connection to 192.168.40.158 closed.
dome@Ansible-Master:~$ ssh -p 2002 ansible_admin@192.168.40.156
Linux Ansible-Worker2 6.1.0-37-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.140-1 (2025-05-22) 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.
```

---

4. Configuration du Pare-feu

Objectif

Activer le pare-feu UFW, bloquer le trafic par défaut, autoriser les ports nécessaires `2002`.

Procédure

Modifier l’inventaire Ansible pour utiliser le port 2002

Ouvre ton fichier `inventory.ini` et remplace par :

ini

CopierModifier

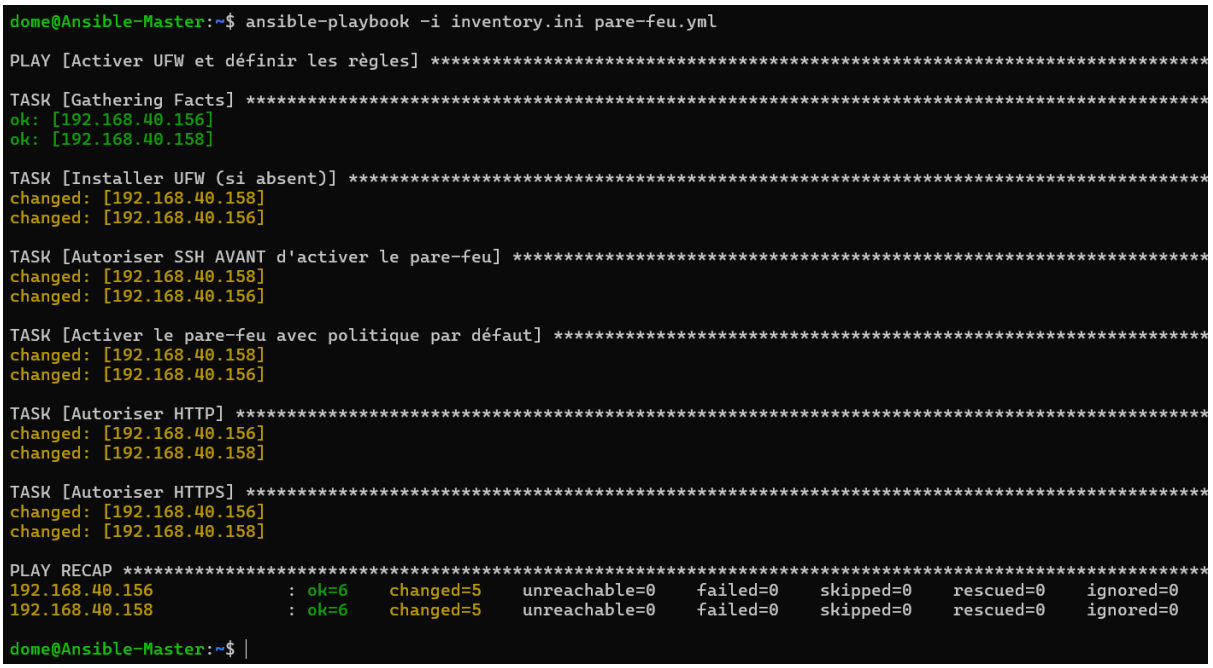
```
[all]

192.168.40.156 ansible_port=2002 ansible_user=ansible_admin

192.168.40.158 ansible_port=2002 ansible_user=ansible_admin
```

Vérifie aussi que la clé SSH est bien déployée sur ces machines pour `ansible_admin`.

Playbook (extrait) :

```
- name: Activer UFW et définir les règles
  hosts: all
  become: yes
  tasks:
    - name: Installer UFW (si absent)
      apt:
        name: ufw
        state: present
        update_cache: yes

    - name: Autoriser SSH AVANT d'activer le pare-feu
      ufw:
        rule: allow
        port: 2002
        proto: tcp

    - name: Activer le pare-feu avec politique par défaut
      ufw:
        state: enabled
        policy: deny
        direction: incoming

    - name: Autoriser HTTP
```

```
    ufw:
      rule: allow
      port: 80
      proto: tcp

  - name: Autoriser HTTPS
    ufw:
      rule: allow
      port: 443
      proto: tcp
```



Vérifications

- Utiliser `ansible all -a "sudo ufw status"` pour confirmer le pare-feu.
- Vérifier l'accès SSH sur le nouveau port : `ssh -p 2002`.

---

5. Désactivation des Services Inutiles

Objectif

Désactiver les services non essentiels (comme `cups`, `avahi-daemon`) pour limiter les points d'entrée.

Procédure

Playbook (extrait) :

```
- name: Désactiver les services inutiles
  hosts: all
  become: yes
  tasks:
    - name: Masquer cups
      systemd:
        name: cups
        enabled: no
        state: stopped
        masked: yes
      ignore_errors: yes

    - name: Masquer avahi-daemon
      systemd:
        name: avahi-daemon
        enabled: no
        state: stopped
        masked: yes
      ignore_errors: yes
```

6. Méthode utilisée

- Liste dynamique de services à auditer (`cups`, `avahi-daemon`, etc.)
- Vérification de l'existence du service avec `stat`
- Désactivation conditionnelle avec le module `systemd`
- Ignorer les services absents pour un playbook portable et sans échec

```
dome@Ansible-Master:~$ ansible-playbook -i inventory.ini stop_unnecessary_serveces.yml

PLAY [Désactiver et masquer les services inutiles] ********************************************************************
*

TASK [Gathering Facts] ************************************************************************************************
*
ok: [192.168.40.156]
ok: [192.168.40.158]

TASK [Vérifier si le service {{ item }} existe] ***********************************************************************
*
ok: [192.168.40.156] => (item=cups)
ok: [192.168.40.158] => (item=cups)
ok: [192.168.40.156] => (item=avahi-daemon)
ok: [192.168.40.158] => (item=avahi-daemon)

TASK [Désactiver et masquer {{ item.item }}] **************************************************************************
*
skipping: [192.168.40.156] => (item={'changed': False, 'stat': {'exists': False}, 'invocation': {'module_args': {'path'
: '/lib/systemd/system/cups.service', 'follow': False, 'get_md5': False, 'get_checksum': True, 'get_mime': True, 'get_a
ttributes': True, 'checksum_algorithm': 'sha1'}}, 'failed': False, 'item': 'cups', 'ansible_loop_var': 'item'})
skipping: [192.168.40.156] => (item={'changed': False, 'stat': {'exists': False}, 'invocation': {'module_args': {'path'
: '/lib/systemd/system/avahi-daemon.service', 'follow': False, 'get_md5': False, 'get_checksum': True, 'get_mime': True
, 'get_attributes': True, 'checksum_algorithm': 'sha1'}}, 'failed': False, 'item': 'avahi-daemon', 'ansible_loop_var':
'item'})
skipping: [192.168.40.156]
skipping: [192.168.40.158] => (item={'changed': False, 'stat': {'exists': False}, 'invocation': {'module_args': {'path'
: '/lib/systemd/system/cups.service', 'follow': False, 'get_md5': False, 'get_checksum': True, 'get_mime': True, 'get_a
ttributes': True, 'checksum_algorithm': 'sha1'}}, 'failed': False, 'item': 'cups', 'ansible_loop_var': 'item'})
skipping: [192.168.40.158] => (item={'changed': False, 'stat': {'exists': False}, 'invocation': {'module_args': {'path'
: '/lib/systemd/system/avahi-daemon.service', 'follow': False, 'get_md5': False, 'get_checksum': True, 'get_mime': True
, 'get_attributes': True, 'checksum_algorithm': 'sha1'}}, 'failed': False, 'item': 'avahi-daemon', 'ansible_loop_var':
'item'})
skipping: [192.168.40.158]

PLAY RECAP ************************************************************************************************************
*
192.168.40.156             : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.40.158             : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

Vérifications

Confirmer que les services inutiles sont désactivés : `systemctl status cups`.

Job 03 – Surveillance, Logs et Conformité

L'objectif de ce Job est d'apprendre à :

- Déployer un outil de gestion de logs avec Ansible (Filebeat).
- Mettre en place un audit de conformité basique.
- Appliquer des paramètres de sécurité sur le noyau et les mots de passe.
- Organiser son travail avec les rôles Ansible.

Chaque tâche permet d'approfondir l'automatisation de la sécurité, en ajoutant un niveau de contrôle et de traçabilité aux serveurs.

1. Déploiement de Filebeat pour la Gestion des Logs

Objectif

Installer Filebeat sur tous les serveurs et le configurer pour :

- Activer les modules `system` et `auditd` (si applicable).

- Simuler l'envoi des logs vers une destination fictive.

Étapes détaillées

1. Créer un fichier de rôle `roles/filebeat/tasks/main.yml`.
2. Gérer l'installation du paquet Filebeat.
3. Déployer une configuration `filebeat.yml` via un template Jinja2.
4. Activer les modules `system` et `auditd`.
5. Démarrer et activer le service `filebeat`.

Commandes (exemples)

```
- name: Installer et configurer Filebeat
  hosts: all
  become: true
  tasks:

    - name: Installer gpg si nécessaire
      package:
        name: gnupg
        state: present

    - name: Ajouter la clé GPG d'Elastic
      apt_key:
        url: https://artifacts.elastic.co/GPG-KEY-elasticsearch
        state: present

    - name: Ajouter le dépôt Filebeat
      apt_repository:
        repo: 'deb https://artifacts.elastic.co/packages/7.x/apt
stable main'
        state: present
        filename: elastic-7.x

    - name: Mettre à jour les paquets
      apt:
        update_cache: yes
        cache_valid_time: 3600
```

```
    - name: Installer Filebeat
      apt:
        name: filebeat
        state: present

    - name: Copier le fichier de configuration Filebeat
      template:
        src: filebeat.yml.j2
        dest: /etc/filebeat/filebeat.yml
      notify: Redémarrer Filebeat

    - name: Activer les modules system et auditd
      command: filebeat modules enable system auditd
      args:
        creates: /etc/filebeat/modules.d/system.yml

    - name: Activer et démarrer le service Filebeat
      systemd:
        name: filebeat
        enabled: yes
        state: started

  handlers:
    - name: Redémarrer Filebeat
      systemd:
        name: filebeat
        state: restarted
```

Sortie attendue

- Filebeat installé et démarré.
- Modules activés visibles via `filebeat modules list`.

```
dome@Ansible-Master:~/playbook$ ansible-playbook -i inventory.ini filebeat.yml

PLAY [Installer et configurer Filebeat] ****************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.40.158]
ok: [192.168.40.156]

TASK [Installer gpg si nécessaire] *********************************************************************************
ok: [192.168.40.158]
ok: [192.168.40.156]

TASK [Ajouter la clé GPG d'Elastic] ********************************************************************************
ok: [192.168.40.158]
ok: [192.168.40.156]

TASK [Ajouter le dépôt Filebeat] ***********************************************************************************
ok: [192.168.40.156]
ok: [192.168.40.158]

TASK [Mettre à jour les paquets] ***********************************************************************************
ok: [192.168.40.156]
ok: [192.168.40.158]

TASK [Installer Filebeat] ******************************************************************************************
ok: [192.168.40.156]
ok: [192.168.40.158]
```

```
TASK [Copier le fichier de configuration Filebeat] *****************************************************************
changed: [192.168.40.156]
changed: [192.168.40.158]

TASK [Activer les modules system et auditd] *************************************************************************
changed: [192.168.40.158]
changed: [192.168.40.156]

TASK [Activer et démarrer le service Filebeat] *********************************************************************
changed: [192.168.40.158]
changed: [192.168.40.156]

RUNNING HANDLER [Redémarrer Filebeat] ******************************************************************************
changed: [192.168.40.156]
changed: [192.168.40.158]

PLAY RECAP *********************************************************************************************************
192.168.40.156             : ok=10   changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.40.158             : ok=10   changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Un template Jinja2 (`filebeat.yml.j2`) a été utilisé pour générer dynamiquement le fichier de configuration de Filebeat. Ce fichier doit être placé dans le dossier `templates/` du projet Ansible. Le module `template` utilisé dans le playbook copie ce fichier vers `/etc/filebeat/filebeat.yml` sur les hôtes cibles.

2. Audit de Conformité Basique

Objectif

Vérifier :

- L'existence et les permissions des fichiers critiques.
- L'absence de mots de passe vides.
- L'absence de répertoires avec permissions `777`.

Étapes détaillées

1. Vérifier les permissions de `/etc/passwd`, `/etc/shadow`, `/etc/sudoers`, `/etc/ssh/sshd_config`.

2. Lister les utilisateurs ayant des mots de passe vides.

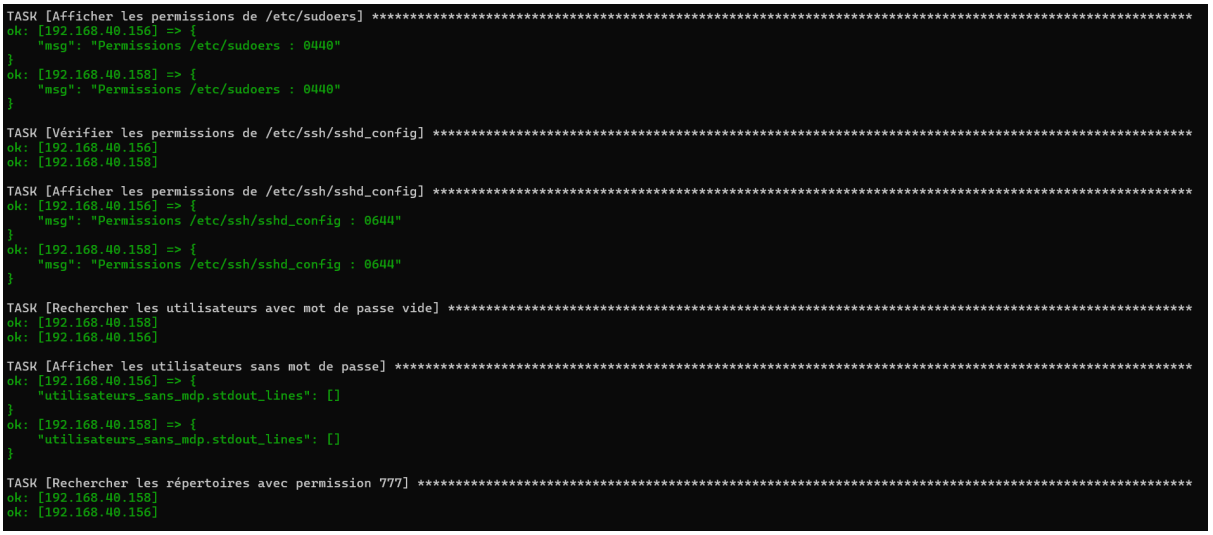
3. Rechercher des répertoires `777`.

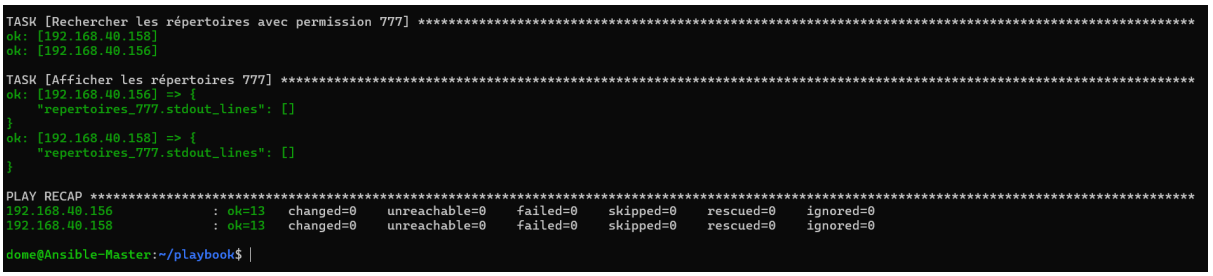
Exemple de tâches

```
- name: Audit de conformité de base
  hosts: all
  become: true
  tasks:

    - name: Vérifier les permissions de /etc/passwd
      stat:
        path: /etc/passwd
      register: passwd_stat

    - name: Afficher les permissions de /etc/passwd
      debug:
        msg: "Permissions /etc/passwd : {{ passwd_stat.stat.mode }}"

    - name: Vérifier les permissions de /etc/shadow
      stat:
        path: /etc/shadow
      register: shadow_stat

    - name: Afficher les permissions de /etc/shadow
      debug:
        msg: "Permissions /etc/shadow : {{ shadow_stat.stat.mode }}"

    - name: Vérifier les permissions de /etc/sudoers
      stat:
        path: /etc/sudoers
      register: sudoers_stat

    - name: Afficher les permissions de /etc/sudoers
      debug:
        msg: "Permissions /etc/sudoers : {{ sudoers_stat.stat.mode
}}"
```

```yaml
    - name: Vérifier les permissions de /etc/ssh/sshd_config
      stat:
        path: /etc/ssh/sshd_config
      register: sshd_stat

    - name: Afficher les permissions de /etc/ssh/sshd_config
      debug:
        msg: "Permissions /etc/ssh/sshd_config : {{
sshd_stat.stat.mode }}"

    - name: Rechercher les utilisateurs avec mot de passe vide
      shell: "awk -F: '($2 == \"\") {print $1}' /etc/shadow"
      register: utilisateurs_sans_mdp
      changed_when: false

    - name: Afficher les utilisateurs sans mot de passe
      debug:
        var: utilisateurs_sans_mdp.stdout_lines

    - name: Rechercher les répertoires avec permission 777
      shell: "find / -type d -perm 0777 2>/dev/null"
      register: repertoires_777
      changed_when: false

    - name: Afficher les répertoires 777
      debug:
        var: repertoires_777.stdout_lines
```

```
dome@Ansible-Master:~/playbook$ ansible-playbook -i inventory.ini audit_conformite.yml

PLAY [Audit de conformité de base] ********************************************************************************************************************************

TASK [Gathering Facts] ********************************************************************************************************************************************
ok: [192.168.40.156]
ok: [192.168.40.158]

TASK [Vérifier les permissions de /etc/passwd] ********************************************************************************************************************
ok: [192.168.40.158]
ok: [192.168.40.156]

TASK [Afficher les permissions de /etc/passwd] ********************************************************************************************************************
ok: [192.168.40.156] => {
    "msg": "Permissions /etc/passwd : 0644"
}
ok: [192.168.40.158] => {
    "msg": "Permissions /etc/passwd : 0644"
}

TASK [Vérifier les permissions de /etc/shadow] ********************************************************************************************************************
ok: [192.168.40.156]
ok: [192.168.40.158]

TASK [Afficher les permissions de /etc/shadow] ********************************************************************************************************************
ok: [192.168.40.156] => {
    "msg": "Permissions /etc/shadow : 0640"
}
ok: [192.168.40.158] => {
    "msg": "Permissions /etc/shadow : 0640"
}

TASK [Vérifier les permissions de /etc/sudoers] *******************************************************************************************************************
ok: [192.168.40.156]
ok: [192.168.40.158]
```





Sortie attendue

Lors de l'exécution, Ansible renverra pour chaque tâche :

- Le mode (permissions) des fichiers critiques, sous forme numérique (ex: `0644`).
- Une liste d'utilisateurs (ou vide) n'ayant pas de mot de passe.
- Les chemins des répertoires avec permissions `777`, s'ils existent.

3. Durcissement du Noyau et des Politiques de Mot de Passe

Objectif

- Appliquer des paramètres de sécurité réseau dans `/etc/sysctl.conf`.
- Modifier les politiques de mot de passe via PAM ou `login.defs`.

Étapes détaillées

1. Ajouter des paramètres de type `net.ipv4.tcp_syncookies = 1`.

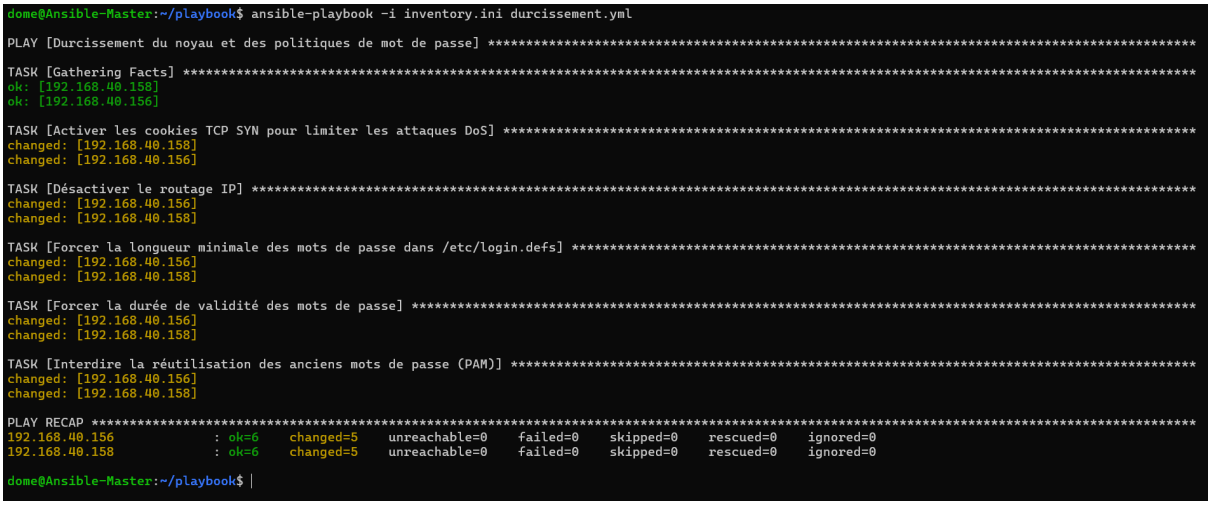
2. Appliquer avec la commande `sysctl -p`.

3. Modifier les fichiers PAM pour la complexité et l'historique.

Exemple

```
- name: Durcissement du noyau et des politiques de mot de passe
  hosts: all
  become: true
  tasks:

    - name: Activer les cookies TCP SYN pour limiter les attaques
DoS
      sysctl:
        name: net.ipv4.tcp_syncookies
        value: "1"
        state: present
        reload: yes

    - name: Désactiver le routage IP
      sysctl:
        name: net.ipv4.ip_forward
        value: "0"
        state: present
        reload: yes

    - name: Forcer la longueur minimale des mots de passe dans
/etc/login.defs
      lineinfile:
        path: /etc/login.defs
        regexp: '^PASS_MIN_LEN'
        line: 'PASS_MIN_LEN   12'
        state: present

    - name: Forcer la durée de validité des mots de passe
      lineinfile:
        path: /etc/login.defs
        regexp: '^PASS_MAX_DAYS'
        line: 'PASS_MAX_DAYS   90'
        state: present
```

```
    - name: Interdire la réutilisation des anciens mots de passe
(PAM)
      lineinfile:
        path: /etc/pam.d/common-password
        regexp: '^password\s+required\s+pam_unix.so'
        line: 'password required pam_unix.so remember=5 use_authtok
sha512 shadow'
        state: present
        backup: yes
        create: no
        insertafter: EOF
```



Détails importants

- **`sysctl`** applique immédiatement les paramètres réseau de sécurité.
- **`lineinfile`** assure que les valeurs dans `/etc/login.defs` soient forcées (longueur, validité).
- **PAM** (via `common-password`) empêche la réutilisation des 5 derniers mots de passe.

⚠️ Si la distribution n'utilise pas `/etc/pam.d/common-password` (cas de certaines RedHat/CentOS), il faut adapter vers `/etc/pam.d/system-auth`.

Sortie attendue

À l'exécution :

- Ansible retournera les changements appliqués sur les paramètres `sysctl`.
- Les lignes seront ajoutées ou remplacées dans les fichiers `/etc/login.defs` et `/etc/pam.d/common-password`.

4. Introduction aux Rôles Ansible

Objectif

Organiser les tâches en créant un rôle réutilisable, par exemple `hardening_base`.

Structure à créer :

Créer le dossier `roles` manuellement
Dans dossier `~/playbook`, tape :

```
mkdir roles
```

Vérifier les droits
Assure-toi que tu es bien propriétaire :

```
ls -ld roles
```

```
dome@Ansible-Master:~/playbook$ sudo mkdir roles
dome@Ansible-Master:~/playbook$ ls -ld roles
drwxr-xr-x 2 root root 4096 24 juil. 10:40 roles
```

La sortie attendue doit être similaire à :

```
drwxr-xr-x 2 dome dome 4096 ... roles
```

Sinon, corriger avec :

```
sudo chown -R dome:dome roles
```

```
dome@Ansible-Master:~/playbook$ sudo chown -R dome:dome roles
dome@Ansible-Master:~/playbook$ ansible-galaxy init roles/hardening_base
- Role roles/hardening_base was created successfully
dome@Ansible-Master:~/playbook$
```

Étapes

1. Générer un rôle :

```
ansible-galaxy init roles/hardening_base
```

2. Déplacer les tâches des exercices précédents dans :
    - `roles/hardening_base/tasks/main.yml`
    - `roles/hardening_base/templates/`
    - `roles/hardening_base/defaults/main.yml`

Copier les tâches dans roles/hardening_base/tasks/main.yml

```
---
# Tâches d'audit de conformité

- name: Vérifier les permissions de /etc/passwd
  stat:
    path: /etc/passwd
  register: passwd_stat

- name: Afficher les permissions de /etc/passwd
  debug:
    msg: "Permissions /etc/passwd : {{ passwd_stat.stat.mode }}"
```

```
- name: Vérifier les permissions de /etc/shadow
  stat:
    path: /etc/shadow
  register: shadow_stat

- name: Afficher les permissions de /etc/shadow
  debug:
    msg: "Permissions /etc/shadow : {{ shadow_stat.stat.mode }}"

- name: Vérifier les permissions de /etc/sudoers
  stat:
    path: /etc/sudoers
  register: sudoers_stat

- name: Afficher les permissions de /etc/sudoers
  debug:
    msg: "Permissions /etc/sudoers : {{ sudoers_stat.stat.mode }}"

- name: Vérifier les permissions de /etc/ssh/sshd_config
  stat:
    path: /etc/ssh/sshd_config
  register: sshd_stat

- name: Afficher les permissions de /etc/ssh/sshd_config
  debug:
    msg: "Permissions /etc/ssh/sshd_config : {{ sshd_stat.stat.mode
}}"

- name: Rechercher les utilisateurs avec mot de passe vide
  shell: "awk -F: '($2 == \"\") {print $1}' /etc/shadow"
  register: utilisateurs_sans_mdp
  changed_when: false

- name: Afficher les utilisateurs sans mot de passe
  debug:
    var: utilisateurs_sans_mdp.stdout_lines

- name: Rechercher les répertoires avec permission 777
  shell: "find / -type d -perm 0777 2>/dev/null"
  register: repertoires_777
```

```
  changed_when: false

- name: Afficher les répertoires 777
  debug:
    var: repertoires_777.stdout_lines

# Tâches de durcissement du noyau (sysctl)

- name: Activer la protection contre les attaques SYN
  sysctl:
    name: net.ipv4.tcp_syncookies
    value: "1"
    state: present
    reload: yes

- name: Désactiver le routage IP (ip_forward)
  sysctl:
    name: net.ipv4.ip_forward
    value: "0"
    state: present
    reload: yes

# Tâches de durcissement des politiques de mot de passe

- name: Forcer la longueur minimale du mot de passe
  lineinfile:
    path: /etc/login.defs
    regexp: '^PASS_MIN_LEN'
    line: 'PASS_MIN_LEN   12'
    state: present

- name: Définir la durée maximale de validité du mot de passe
  lineinfile:
    path: /etc/login.defs
    regexp: '^PASS_MAX_DAYS'
    line: 'PASS_MAX_DAYS   90'
    state: present

- name: Empêcher la réutilisation des anciens mots de passe (PAM)
  lineinfile:
    path: /etc/pam.d/common-password
```

```
    regexp: '^password\s+required\s+pam_unix.so'
    line: 'password required pam_unix.so remember=5 use_authtok
sha512 shadow'
    state: present
    backup: yes
    insertafter: EOF
```

Créer un playbook pour utiliser ce rôle

Créer un fichier apply_hardening.yml :

```
nano apply_hardening.yml
```

Et insèrer :

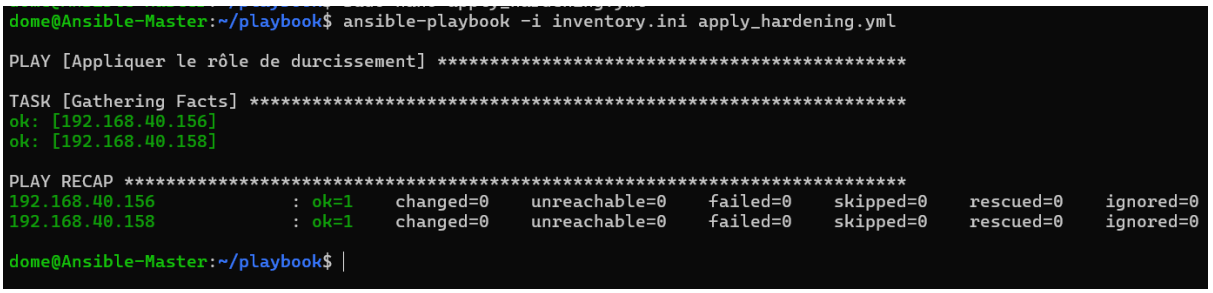
```
- name: Appliquer le rôle de durcissement
  hosts: all
  become: true

  roles:
    - hardening_base
```

Lancer le rôle

```
ansible-playbook -i inventory.ini apply_hardening.yml
```

Résultat attendu

- Tous les audits et durcissements sont effectués en une seule exécution propre.
- Le rôle est réutilisable dans tous les futurs projets.



Job 04 – Détection, Réponse à Incident et Cas d'Usage Avancés

Objectif

Ce Job a pour objectif de simuler des mécanismes de détection d'intrusion, de réponse automatisée à un incident, de gestion sécurisée de secrets et d'extraction d'informations système via Ansible. Il initie à des usages avancés de la cybersécurité opérationnelle.

1. Déploiement d'un IDS/IPS simple (Snort ou Suricata)

Installer un système de détection d'intrusion réseau (IDS) sur une ou plusieurs machines, configuré pour surveiller une interface réseau.

Étapes détaillées

1. **Choix de l'outil** : Nous utilisons `Suricata`, libre et robuste.
2. **Installation du paquet** via le gestionnaire de paquets.
3. **Configuration minimale** pour écouter l'interface `eth0`.
4. **Activation du service**.

Exemple de Playbook : `deploy_suricata.yml`

```
---
- name: Déploiement de Suricata IDS
  hosts: all
  become: true
  gather_facts: true

  tasks:
    - name: Installer Suricata
      apt:
        name: suricata
        state: present
        update_cache: yes

    - name: Détecter l'interface par défaut
      set_fact:
        main_interface: "{{ ansible_default_ipv4.interface }}"
```

```yaml
- name: Configurer Suricata pour écouter sur eth0
  lineinfile:
    path: /etc/suricata/suricata.yaml
    regexp: '^ *interface:'
    line: '  interface: {{ main_interface }}'
    insertafter: '^ *af-packet:'

- name: Redémarrer Suricata
  service:
    name: suricata
    state: restarted
    enabled: true
```

```
dome@Ansible-Master:~/playbook$ sudo nano templates/suricata.yaml.j2
dome@Ansible-Master:~/playbook$ ansible-playbook -i inventory.ini deploy_suricata.yml

PLAY [Déploiement de Suricata avec template] ***********************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************************************************
ok: [192.168.40.156]
ok: [192.168.40.158]

TASK [Installer Suricata] ******************************************************************************************************************************************************
ok: [192.168.40.158]
ok: [192.168.40.156]

TASK [Détecter l'interface réseau par défaut] **********************************************************************************************************************************
ok: [192.168.40.156]
ok: [192.168.40.158]

TASK [Appliquer le fichier de configuration depuis un template] ****************************************************************************************************************
changed: [192.168.40.156]
changed: [192.168.40.158]

TASK [Tester la configuration Suricata] ****************************************************************************************************************************************
ok: [192.168.40.156]
ok: [192.168.40.158]

TASK [Afficher le résultat du test] ********************************************************************************************************************************************
ok: [192.168.40.156] => {
    "suricata_test.stdout_lines": [
        "24/7/2025 -- 14:28:44 - <Info> - Running suricata under test mode",
        "24/7/2025 -- 14:28:44 - <Notice> - This is Suricata version 6.0.10 RELEASE running in SYSTEM mode",
        "24/7/2025 -- 14:28:44 - <Info> - CPUs/cores online: 1",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol http enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol tls enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol dcerpc enable status not set, so enabling by defaul
```

```
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol ftp enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol ssh enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol ssh enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol smtp enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol dns enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol dns enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol modbus enable status not set, so enabling by defaul
t. This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol enip enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol enip enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol dnp3 enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol nfs enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol nfs enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol ntp enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol tftp enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol ikev2 enable status not set, so enabling by default
. This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol krb5 enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol krb5 enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol dhcp enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol snmp enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
```

```
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol snmp enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol sip enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol rfb enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol mqtt enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol rdp enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol http2 enable status not set, so enabling by default
. This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol http2 enable status not set, so enabling by default
. This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol imap enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Info> - No 'host-mode': suricata is in IDS mode, using default setting 'sniffer-only'",
        "24/7/2025 -- 14:28:45 - <Warning> - [ERRCODE: SC_WARN_NO_STATS_LOGGERS(261)] - stats are enabled but no loggers are active",
        "24/7/2025 -- 14:28:45 - <Info> - No signatures supplied.",
        "24/7/2025 -- 14:28:45 - <Notice> - Configuration provided was successfully loaded. Exiting.",
        "24/7/2025 -- 14:28:45 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol http enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:45 - <Info> - cleaning up signature grouping structure... complete"
    ]
}
ok: [192.168.40.158] => {
    "suricata_test.stdout_lines": [
        "24/7/2025 -- 14:28:44 - <Info> - Running suricata under test mode",
        "24/7/2025 -- 14:28:44 - <Notice> - This is Suricata version 6.0.10 RELEASE running in SYSTEM mode",
        "24/7/2025 -- 14:28:44 - <Info> - CPUs/cores online: 1",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol http enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol tls enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol dcerpc enable status not set, so enabling by defaul
t. This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol dcerpc enable status not set, so enabling by defaul
t. This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol smb enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol ftp enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol ftp enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol ssh enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol ssh enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol smtp enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol dns enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol dns enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol modbus enable status not set, so enabling by defaul
t. This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol enip enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol enip enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol dnp3 enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol nfs enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol nfs enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol ntp enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol tftp enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol ikev2 enable status not set, so enabling by default
. This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol krb5 enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol krb5 enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol dhcp enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol snmp enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol sip enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol krb5 enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol krb5 enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol dhcp enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol snmp enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol sip enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol rfb enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol mqtt enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol rdp enable status not set, so enabling by default.
This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol http2 enable status not set, so enabling by default
. This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol http2 enable status not set, so enabling by default
. This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol imap enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:44 - <Info> - No 'host-mode': suricata is in IDS mode, using default setting 'sniffer-only'",
        "24/7/2025 -- 14:28:45 - <Warning> - [ERRCODE: SC_WARN_NO_STATS_LOGGERS(261)] - stats are enabled but no loggers are active",
        "24/7/2025 -- 14:28:45 - <Info> - No signatures supplied.",
        "24/7/2025 -- 14:28:45 - <Notice> - Configuration provided was successfully loaded. Exiting.",
        "24/7/2025 -- 14:28:45 - <Warning> - [ERRCODE: SC_ERR_CONF_YAML_ERROR(242)] - App-Layer protocol http enable status not set, so enabling by default.
 This behavior will change in Suricata 7, so please update your config. See ticket #4744 for more details.",
        "24/7/2025 -- 14:28:45 - <Info> - cleaning up signature grouping structure... complete"
    ]
}

TASK [Redémarrer Suricata (si config valide)] ***********************************************************************************************************
changed: [192.168.40.158]
changed: [192.168.40.156]

PLAY RECAP **********************************************************************************************************************************************
192.168.40.156             : ok=7    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.40.158             : ok=7    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Des nombreux avertissements comme :

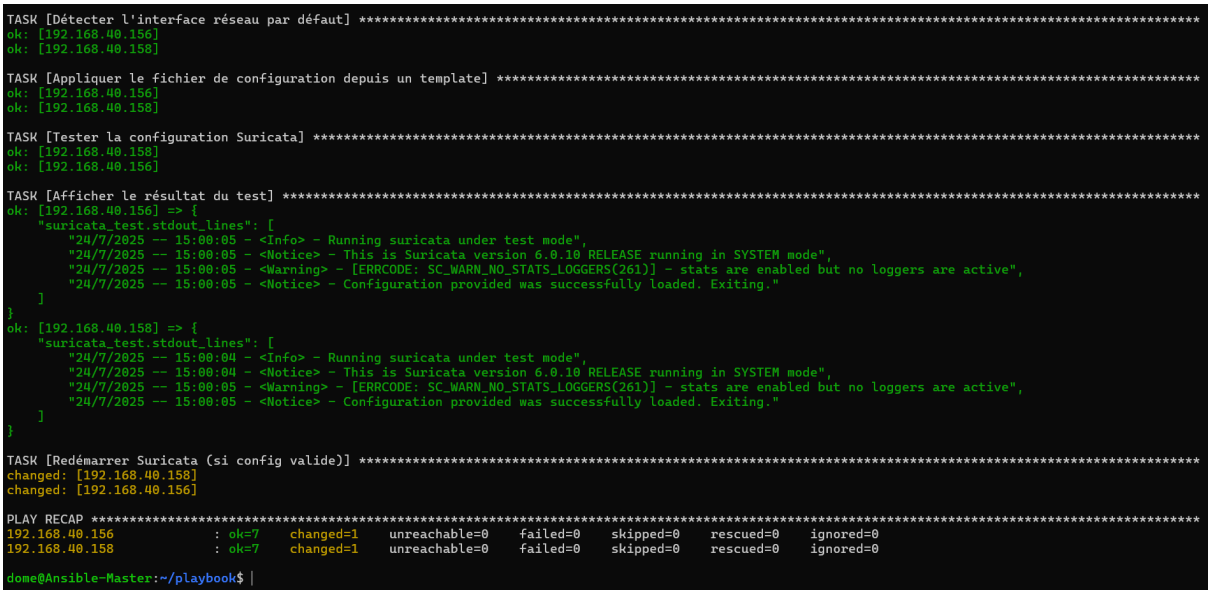
`App-Layer protocol http enable status not set, so enabling by default.`

Ce que cela signifie :

- Suricata active certains protocoles automatiquement, mais cela ne sera **plus automatique dans la version 7**.
- Ces messages sont **informatifs**, pas bloquants.
- Tu peux ignorer ces avertissements **dans le cadre d'un test IDS minimal**.

Pour aller plus loin, enrichir le fichier `suricata.yaml.j2` avec des sections comme :

```
app-layer:
  protocols:
    http:
      enabled: yes
    tls:
      enabled: yes
    dns:
      enabled: yes
    ...
```



Vérification

`sudo systemctl status suricata`

```
ansible_admin@Ansible-Worker1:~$  sudo systemctl status suricata
[sudo] Mot de passe de ansible_admin :
● suricata.service - Suricata IDS/IDP daemon
     Loaded: loaded (/lib/systemd/system/suricata.service; enabled; preset: enabled)
     Active: active (running) since Thu 2025-07-24 14:28:46 CEST; 3min 55s ago
       Docs: man:suricata(8)
             man:suricatasc(8)
             https://suricata-ids.org/docs/
    Process: 13222 ExecStart=/usr/bin/suricata -D --af-packet -c /etc/suricata/surica
   Main PID: 13223 (Suricata-Main)
      Tasks: 4 (limit: 2258)
     Memory: 25.9M
        CPU: 10.608s
     CGroup: /system.slice/suricata.service
             └─13223 /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml

juil. 24 14:28:46 Ansible-Worker1 suricata[13222]: 24/7/2025 -- 14:28:46 - <Warning>
juil. 24 14:28:46 Ansible-Worker1 suricata[13222]: 24/7/2025 -- 14:28:46 - <Warning>
juil. 24 14:28:46 Ansible-Worker1 suricata[13222]: 24/7/2025 -- 14:28:46 - <Warning>
juil. 24 14:28:46 Ansible-Worker1 suricata[13222]: 24/7/2025 -- 14:28:46 - <Warning>
juil. 24 14:28:46 Ansible-Worker1 suricata[13222]: 24/7/2025 -- 14:28:46 - <Warning>
juil. 24 14:28:46 Ansible-Worker1 suricata[13222]: 24/7/2025 -- 14:28:46 - <Warning>
juil. 24 14:28:46 Ansible-Worker1 suricata[13222]: 24/7/2025 -- 14:28:46 - <Info> - F
juil. 24 14:28:46 Ansible-Worker1 suricata[13222]: 24/7/2025 -- 14:28:46 - <Info> - F
juil. 24 14:28:46 Ansible-Worker1 systemd[1]: suricata.service: Can't open PID file /
juil. 24 14:28:46 Ansible-Worker1 systemd[1]: Started suricata.service - Suricata IDS
```

```
ansible_admin@Ansible-Worker2:~$ sudo systemctl status suricata
● suricata.service - Suricata IDS/IDP daemon
     Loaded: loaded (/lib/systemd/system/suricata.service; enabled; preset: enabled)
     Active: active (running) since Thu 2025-07-24 14:28:47 CEST; 3min 14s ago
       Docs: man:suricata(8)
             man:suricatasc(8)
             https://suricata-ids.org/docs/
    Process: 13118 ExecStart=/usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.
   Main PID: 13119 (Suricata-Main)
      Tasks: 4 (limit: 2258)
     Memory: 25.9M
        CPU: 8.722s
     CGroup: /system.slice/suricata.service
             └─13119 /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --p

juil. 24 14:28:46 Ansible-Worker2 suricata[13118]: 24/7/2025 -- 14:28:46 - <Warning> - [
juil. 24 14:28:46 Ansible-Worker2 suricata[13118]: 24/7/2025 -- 14:28:46 - <Warning> - [
juil. 24 14:28:46 Ansible-Worker2 suricata[13118]: 24/7/2025 -- 14:28:46 - <Warning> - [
juil. 24 14:28:46 Ansible-Worker2 suricata[13118]: 24/7/2025 -- 14:28:46 - <Warning> - [
juil. 24 14:28:46 Ansible-Worker2 suricata[13118]: 24/7/2025 -- 14:28:46 - <Warning> - [
juil. 24 14:28:46 Ansible-Worker2 suricata[13118]: 24/7/2025 -- 14:28:46 - <Warning> - [
juil. 24 14:28:46 Ansible-Worker2 suricata[13118]: 24/7/2025 -- 14:28:46 - <Info> - Foun
juil. 24 14:28:46 Ansible-Worker2 suricata[13118]: 24/7/2025 -- 14:28:46 - <Info> - Foun
juil. 24 14:28:47 Ansible-Worker2 systemd[1]: suricata.service: Can't open PID file /run
juil. 24 14:28:47 Ansible-Worker2 systemd[1]: Started suricata.service - Suricata IDS/ID
```

Vérification

```
sudo systemctl status suricata
```

2. Réponse à incident simulée

Objectif

Simuler la détection d'un fichier malveillant et automatiser une réponse sécurisée.

Étapes

1. **Création manuelle** du fichier `/tmp/malicious_script.sh`.
2. **Détection** de ce fichier avec Ansible.
3. **Réaction automatique** :
    - Arrêt d'un service compromis (ex: nginx).
    - Déplacement du fichier vers `/root/quarantine/`.
    - Collecte de logs + processus.
    - Écriture d'un log d'incident sur le contrôleur.

Exemple de Playbook : `incident_response.yml`

```
---
- name: Réponse à incident simulée
  hosts: all
  become: true
  vars:
    suspicious_file: "/tmp/malicious_script.sh"
    quarantine_dir: "/root/quarantine"

  tasks:
    - name: Vérifier la présence du fichier suspect
      stat:
        path: "{{ suspicious_file }}"
      register: file_stat

    - name: Arrêter nginx si fichier suspect détecté
      service:
        name: nginx
        state: stopped
```

```
  when: file_stat.stat.exists

- name: Créer le répertoire de quarantaine
  file:
    path: "{{ quarantine_dir }}"
    state: directory
    mode: '0700'
  when: file_stat.stat.exists

- name: Déplacer le fichier vers la quarantaine
  command: mv {{ suspicious_file }} {{ quarantine_dir }}/
  when: file_stat.stat.exists

- name: Collecter les processus actifs
  shell: ps aux
  register: process_list
  when: file_stat.stat.exists

- name: Collecter les derniers logs
  shell: journalctl -n 50
  register: log_output
  when: file_stat.stat.exists

- name: Écrire un log local sur le contrôleur
  delegate_to: localhost
  copy:
    content: |
      Incident détecté !
      --- Processus :
      {{ process_list.stdout }}

      --- Logs :
      {{ log_output.stdout }}
    dest: "./incident_logs/{{ inventory_hostname }}.log"
  when: file_stat.stat.exists
```

```
dome@Ansible-Master:~/playbook$ ansible-playbook -i inventory.ini incident_response.yml

PLAY [Réponse à incident simulée] ***********************************************************************

TASK [Gathering Facts] **********************************************************************************
ok: [192.168.40.158]
ok: [192.168.40.156]

TASK [Vérifier la présence du fichier suspect] **********************************************************
ok: [192.168.40.158]
ok: [192.168.40.156]

TASK [Arrêter nginx si fichier suspect détecté] *********************************************************
skipping: [192.168.40.156]
skipping: [192.168.40.158]

TASK [Créer le répertoire de quarantaine] ***************************************************************
skipping: [192.168.40.156]
skipping: [192.168.40.158]

TASK [Déplacer le fichier vers la quarantaine] **********************************************************
skipping: [192.168.40.156]
skipping: [192.168.40.158]

TASK [Collecter les processus actifs] *******************************************************************
skipping: [192.168.40.156]
skipping: [192.168.40.158]

TASK [Collecter les derniers logs] **********************************************************************
skipping: [192.168.40.156]
skipping: [192.168.40.158]

TASK [Écrire un log local sur le contrôleur] ************************************************************
skipping: [192.168.40.156]
skipping: [192.168.40.158]

PLAY RECAP **********************************************************************************************
192.168.40.156             : ok=2    changed=0    unreachable=0    failed=0    skipped=6    rescued=0    ignored=0
192.168.40.158             : ok=2    changed=0    unreachable=0    failed=0    skipped=6    rescued=0    ignored=0

dome@Ansible-Master:~/playbook$
```

Vérification

- Fichier déplacé dans `/root/quarantine`
- Log créé sur le contrôleur dans `./incident_logs/`

3. Utilisation d'Ansible Vault

Objectif

Protéger des données sensibles dans les playbooks.

Étapes

1. **Créer un fichier chiffré** contenant une variable sensible.
2. **L'utiliser dans un playbook.**

Exemple

```
ansible-vault create secrets.yml
```

Contenu :

```
api_key: "super-secret-key-123"
```

Utilisation dans un playbook :

```
---
- name: Exemple Vault
  hosts: localhost
  vars_files:
    - secrets.yml

  tasks:
    - name: Afficher une clé API chiffrée
      debug:
        msg: "La clé est : {{ api_key }}"
```

Commande d'exécution

```
ansible-playbook example_vault.yml --ask-vault-pass
```

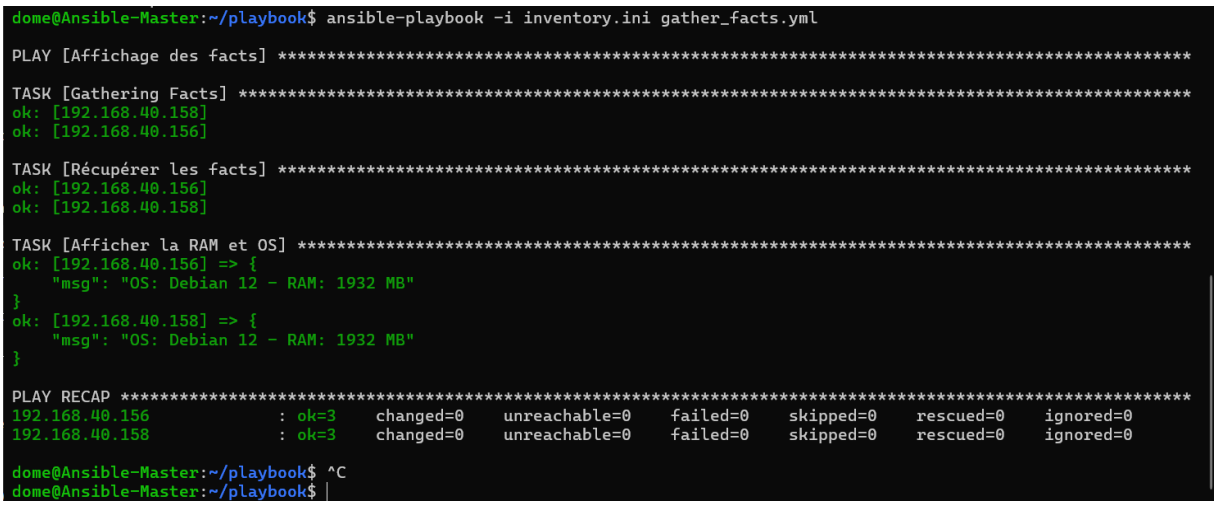
4. Gestion des Facts

Objectif

Collecter automatiquement des informations système (RAM, OS, etc.).

Exemple de Playbook : `gather_facts.yml`

```
---
- name: Affichage des facts
  hosts: all

  tasks:
    - name: Récupérer les facts
      setup:

    - name: Afficher la RAM et OS
      debug:
        msg: "OS: {{ ansible_distribution }} {{
ansible_distribution_version }} - RAM: {{ ansible_memtotal_mb }} MB"
```



Job 5 – Scénario, Optimisation et Bilan

Objectif

Ce dernier job a pour but de consolider toutes les compétences acquises en Ansible, depuis la sécurisation des serveurs jusqu'au déploiement complet d'une application web simple. Il vise également à structurer proprement le projet, optimiser les playbooks et produire un rapport final exploitable.

1– Déploiement sécurisé d'une application web avec Nginx

Objectif

Déployer une **page web HTML statique** sur un serveur Linux en respectant des **normes de sécurité strictes** :

- Serveur durci (SSH, pare-feu, etc.)
- Serveur HTTP Nginx avec HTTPS (certificat auto-signé)
- Logging intégré avec Filebeat

1. Créer un rôle `webserver_secure`

```
roles/
└── webserver_secure/
    ├── tasks/
    │   ├── main.yml
    ├── templates/
    │   ├── nginx.conf.j2
    │   ├── index.html.j2
    ├── files/
    │   └── certs/ (certificat + clé auto-signée)
    └── handlers/
        └── main.yml
```

Dans ton projet Ansible, crée un rôle dédié appelé webserver_secure

```
ansible-galaxy init roles/webserver_secure
```

```
dome@Ansible-Master:~/playbook$ ansible-galaxy init roles/webserver_secure
- Role roles/webserver_secure was created successfully
dome@Ansible-Master:~/playbook$
```

Créer le certificat SSL auto-signé (localement)

Sur ta machine de contrôle (pas via Ansible, juste pour préparer les fichiers) :

```
mkdir -p roles/webserver_secure/files/certs

openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
  -keyout roles/webserver_secure/files/certs/key.pem \
  -out roles/webserver_secure/files/certs/cert.pem \
  -subj "/CN=localhost"
```

```
dome@Ansible-Master:~/playbook$ mkdir -p roles/webserver_secure/files/certs
dome@Ansible-Master:~/playbook$ openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
  -keyout roles/webserver_secure/files/certs/key.pem \
  -out roles/webserver_secure/files/certs/cert.pem \
  -subj "/CN=localhost"
.....+...+....+...+..+....+.....+.....+.........+....+.+..+...++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++.+.+...+...+...+..........+.........+....+..+.+..+...+....+.........+..+.......+..+.+..+...+....+...+..++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++.......+..+.....+.......+..+....+..............+.....+.+
......+..+.+..........+.+.........+.....+.+...+..+...+......++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++
..............+.....+.+.....++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*......+......++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++*.......+.+..+.+...........+.........+.+........+...+....+........
......+.........+...+.......+...+..+.......+..................+......+......+...+..+......+.............+.........+..+.+
..+......+.+......+...+.....................+.+...+..+..............+.............+.....+...+....+........+.+..+........
....+.+..............+......+...+..+.............................+...+...+..+....+.....+...+..........+........+...+....
..+.........++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
-----
dome@Ansible-Master:~/playbook$
```

2. Tâches à inclure dans le rôle

```
# roles/webserver_secure/tasks/main.yml
- name: Appliquer les rôles de hardening
  import_role:
```

```
    name: hardening_base

- name: Installer Nginx
  apt:
    name: nginx
    state: present
  become: yes

- name: Copier la page HTML statique
  template:
    src: index.html.j2
    dest: /var/www/html/index.html
    mode: '0644'

- name: Copier la configuration Nginx sécurisée
  template:
    src: nginx.conf.j2
    dest: /etc/nginx/sites-available/default
  notify: Redémarrer Nginx

- name: Copier le certificat SSL
  copy:
    src: certs/
    dest: /etc/nginx/certs/
    mode: '0644'
    owner: root
    group: root
  notify: Redémarrer Nginx

- name: S'assurer que Filebeat collecte les logs Nginx
  lineinfile:
    path: /etc/filebeat/filebeat.yml
    line: "- /var/log/nginx/*.log"
    insertafter: "^paths:"
  notify: Redémarrer Filebeat
```

3. Handlers

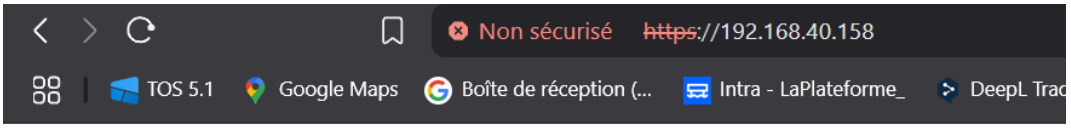
```
# roles/webserver_secure/handlers/main.yml
- name: Redémarrer Nginx
  service:
    name: nginx
```

```
    state: restarted

- name: Redémarrer Filebeat
  service:
    name: filebeat
    state: restarted
```

```
dome@Ansible-Master:~/playbook$ ansible-playbook -i inventory.ini deploy_web.yml

PLAY [Déploiement web sécurisé] ************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.40.158]

TASK [webserver_secure : Assurer l'installation de Nginx] **********************************************************
changed: [192.168.40.158]

TASK [webserver_secure : Copier la page HTML] **********************************************************************
changed: [192.168.40.158]

TASK [webserver_secure : Copier la configuration Nginx] ************************************************************
changed: [192.168.40.158]

TASK [webserver_secure : Créer le répertoire des certificats SSL] **************************************************
changed: [192.168.40.158]

TASK [webserver_secure : Copier les certificats SSL] ***************************************************************
changed: [192.168.40.158]

TASK [webserver_secure : Activer et démarrer Nginx] ****************************************************************
changed: [192.168.40.158]

RUNNING HANDLER [webserver_secure : Redémarrer Nginx] **************************************************************
changed: [192.168.40.158]

PLAY RECAP *********************************************************************************************************
192.168.40.158             : ok=8    changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

dome@Ansible-Master:~/playbook$
```

Vérifications attendues

- Accès à `https://IP_serveur` depuis un navigateur sans erreur de chargement (accepter le certificat auto-signé).



Page depoyee automatiquement via Ansible

- `curl -k https://localhost` renvoie la page HTML.

```
dome@Ansible-Master:~/playbook$ curl -k https://192.168.40.158
<!DOCTYPE html>
<html>
<head>
  <title>Bienvenue</title>
</head>
<body>
  <h1>Page déployée automatiquement via Ansible 🚀</h1>
</body>
</html>
dome@Ansible-Master:~/playbook$
```

- `journalctl -u nginx` ne montre aucune erreur.

```
dome@Ansible-Master:~/playbook$ sudo cat ./incident_logs/
cat: ./incident_logs/: Aucun fichier ou dossier de ce type
```

- `filebeat` tourne et les logs nginx sont dans les fichiers simulés.

!! A ce stade filebeat t'es pas encore configuré pour gérer les logs de nginx, le capture d'écran ci dessous montre les services actifs:

```
ansible_admin@Ansible-Worker1:~$ sudo filebeat modules list
[sudo] Mot de passe de ansible_admin :
Enabled:
auditd
system

Disabled:
activemq
apache
aws
awsfargate
azure
barracuda
```

1.1 - Configuration Filebeat pour enregistrer les logs nginx

Créer dossier roles dans répertoire `playbook/`

```
mkdir -p roles
```

Puis préparez le fichier
`touch deploy_web.yml`

Créer un rôle structuré : webserver_secure

`ansible-galaxy init roles/webserver_secure`

Résultat attendu :

```
- Role roles/webserver_secure was created successfully
```

Créer un template de config Nginx

Dans roles/webserver_secure/templates/nginx.conf.j2 :

```
mkdir -p roles/webserver_secure/templates
nano roles/webserver_secure/templates/nginx.conf.j2
```

Colle :

```
server {
    listen 443 ssl;
    server_name localhost;

    ssl_certificate /etc/nginx/certs/cert.pem;
    ssl_certificate_key /etc/nginx/certs/key.pem;

    root /var/www/html;
    index index.html;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

Écrire les tâches du rôle

Dans `roles/webserver_secure/tasks/main.yml` :

```
nano roles/webserver_secure/tasks/main.yml
```

Ajouter :

```
- name: Assurer l'installation de Nginx

  apt:

    name: nginx

    state: present
```

```
    update_cache: yes
  become: yes

- name: Copier la page HTML
  template:
    src: index.html.j2
    dest: /var/www/html/index.html
    mode: '0644'

- name: Copier la configuration Nginx
  template:
    src: nginx.conf.j2
    dest: /etc/nginx/sites-available/default
  notify: Redémarrer Nginx

- name: Créer le répertoire des certificats SSL
  file:
    path: /etc/nginx/certs
    state: directory
    mode: '0755'

- name: Copier les certificats SSL
  copy:
    src: certs/
```

```
    dest: /etc/nginx/certs/
    mode: '0644'
  notify: Redémarrer Nginx
- name: Activer et démarrer Nginx
  service:
    name: nginx
    state: started
    enabled: yes
```

Ajouter un handler

Dans `roles/webserver_secure/handlers/main.yml` :

```
nano roles/webserver_secure/handlers/main.yml
```

Ajouter :

```
- name: Redémarrer nginx
  service:
    name: nginx
    state: restarted
```

Créer un playbook principal

Dans `deploy_web.yml` :

```
nano deploy_web.yml
```

Ajouter :

```
- name: Déploiement complet web + monitoring
  hosts: webservers
  become: yes

  roles:
    - webserver_secure
```

```
    - filebeat  # rôle du job 3
```

Activer le module nginx de Filebeat

Dans ton **rôle Filebeat** (ou dans un playbook séparé), active le module `nginx` :

```
- name: Activer le module nginx
  command: filebeat modules enable nginx
  args:
    creates: /etc/filebeat/modules.d/nginx.yml
```

```
    - name: Activer les modules system et auditd
      command: filebeat modules enable system auditd
      args:
        creates: /etc/filebeat/modules.d/system.yml

    - name: Activer le module nginx
      command: filebeat modules enable nginx
      args:
        creates: /etc/filebeat/modules.d/nginx.yml
```

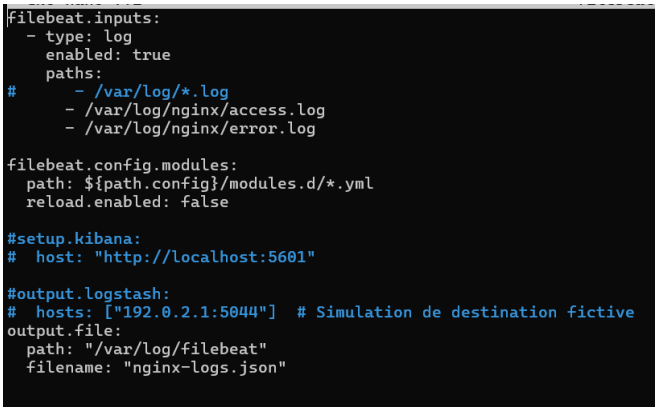
S'assurer que Filebeat lit les bons fichiers de log

Modifier le fichier `filebeat.yml` (via un template Jinja2) :

```
filebeat.inputs:
  - type: log
    enabled: true
    paths:
#      - /var/log/*.log
      - /var/log/nginx/access.log
      - /var/log/nginx/error.log
filebeat.config.modules:
  path: ${path.config}/modules.d/*.yml
  reload.enabled: false
```

```
#setup.kibana:

#  host: "http://localhost:5601"


#output.logstash:

#  hosts: ["192.0.2.1:5044"]  # Simulation de destination fictive

output.file:

  path: "/var/log/filebeat"

  filename: "nginx-logs.json"
```

Cela **n'envoie pas les logs à un serveur distant**, mais les enregistre dans un fichier local (format JSON), pour **simulation**.



Etape 3 – Créer le répertoire de simulation

Ajouter cette tâche dans ton rôle Filebeat (dans `roles/filebeat/tasks/main.yml`) **avant** le déploiement du template :

```
- name: Créer le répertoire local pour simulation des logs Filebeat
  file:
    path: /var/log/filebeat
    state: directory
    mode: '0755'
```

Toujours dans le même rôle (ou playbook), tu peux ajouter :

```
- name: Déployer le fichier de configuration filebeat.yml
  template:
    src: filebeat.yml.j2
    dest: /etc/filebeat/filebeat.yml
    owner: root
    group: root
    mode: '0644'
  notify: Redémarrer Filebeat
```

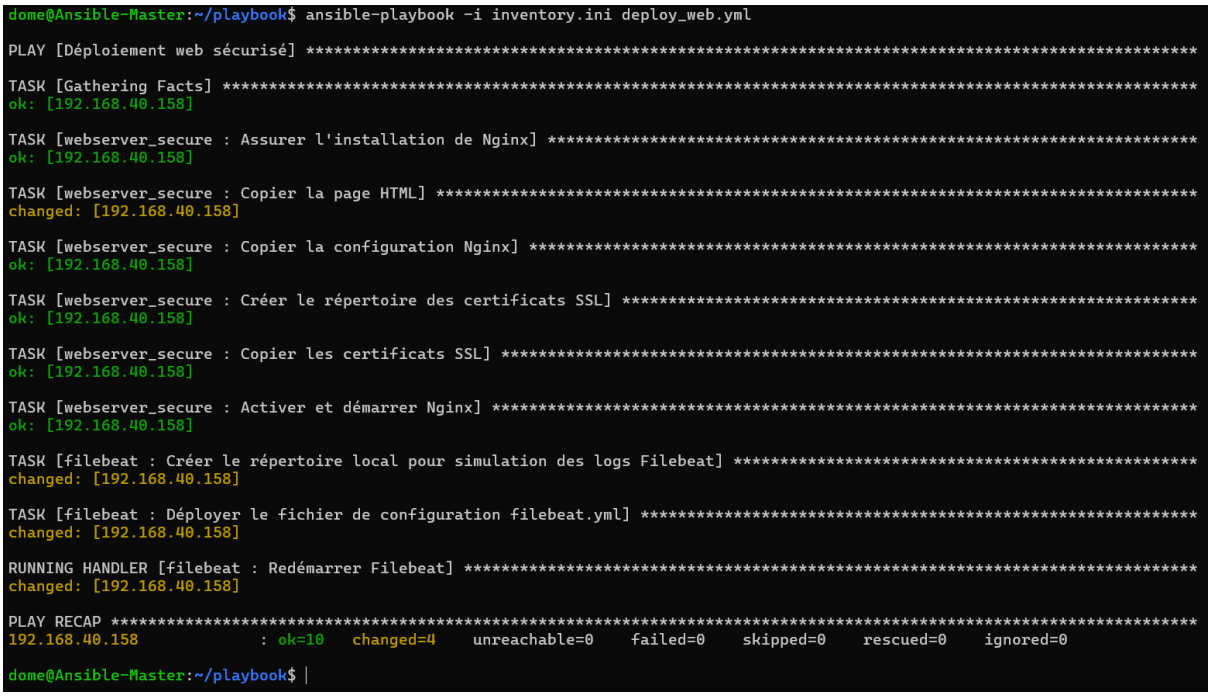
Étape 4 – Activer le module `nginx` dans filebit.yml

Ajouter une commande :

```
- name: Activer le module nginx
  command: filebeat modules enable nginx
  args:
    creates: /etc/filebeat/modules.d/nginx.yml
```

Lancer a nouveau la config du server web:

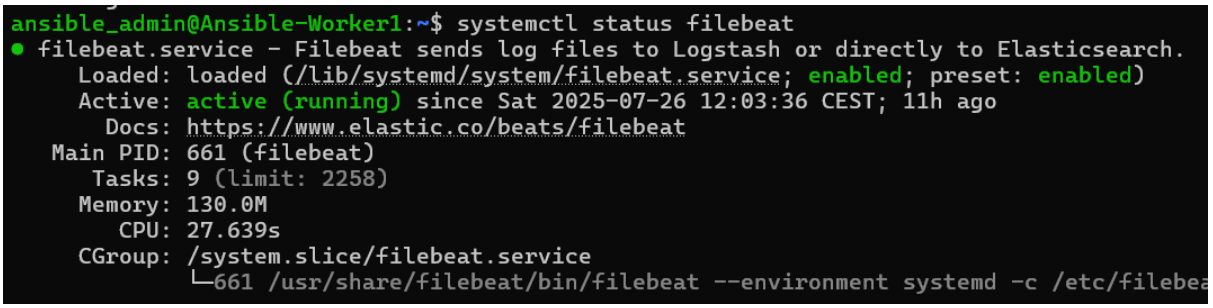
`ansible-playbook -i inventory.ini deploy_web.yml`



Vérifications finales

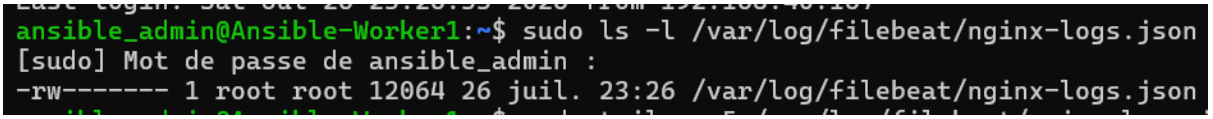
1. Filebeat est actif :

```
systemctl status filebeat
```



2. Le fichier de log simulé existe :

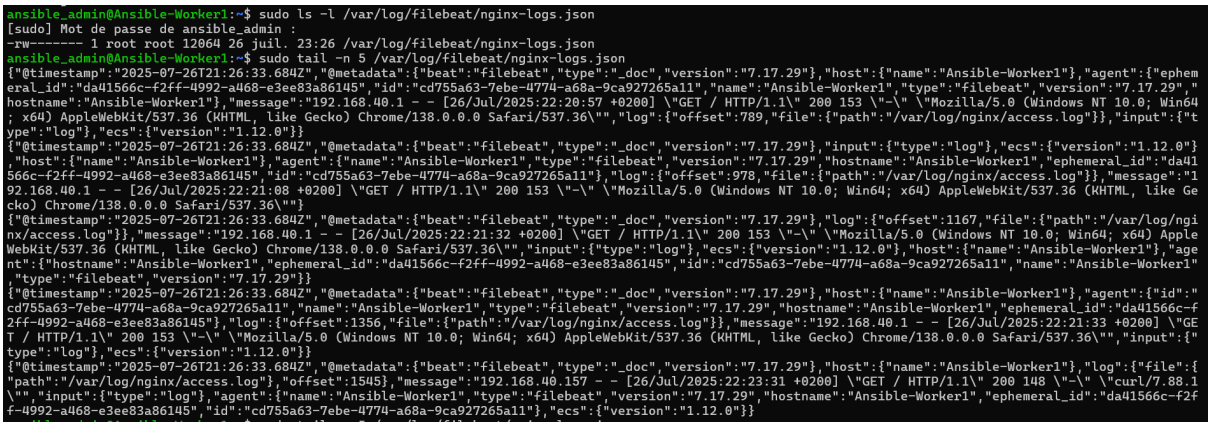
```
ls -l /var/log/filebeat/nginx-logs.json
```



3. Il contient des logs :

```
tail -n 5 /var/log/filebeat/nginx-logs.json
```

Lignes JSON contenant des accès à Nginx



Exemples

Bloc avec conditions :

```
- block:
    - name: Installer les paquets nécessaires
      apt:
        name: "{{ item }}"
        state: present
      loop:
        - nginx
        - filebeat
  when: ansible_os_family == "Debian"
```

Template Jinja2 pour `/etc/nginx/sites-available/default` :

```
server {
    listen 443 ssl;
    server_name localhost;

    ssl_certificate     /etc/nginx/certs/cert.pem;
    ssl_certificate_key /etc/nginx/certs/key.pem;

    root /var/www/html;
    index index.html;
}
```

2 – Optimisation des playbooks

Objectif : Rendre le projet plus maintenable, modulaire et lisible

- Utiliser **des rôles**
- Grouper les tâches dans des **blocs**
- Centraliser les variables dans `group_vars`
- Employer **des templates Jinja2**
- Mettre en place **des handlers**

Créer les variables personnalisées

Dans roles/webserver_secure/vars/main.yml :

nano roles/webserver_secure/vars/main.yml

Ajouter :

```
nginx_listen_port: 443
ssl_cert_path: /etc/nginx/certs/cert.pem
ssl_key_path: /etc/nginx/certs/key.pem
web_root: /var/www/html
```

Créer un template de config Nginx

Dans roles/webserver_secure/templates/nginx.conf.j2 :

```
mkdir -p roles/webserver_secure/templates
nano roles/webserver_secure/templates/nginx.conf.j2
```

Colle :

```
server {

    listen {{ nginx_listen_port }} ssl;
    server_name localhost;

    ssl_certificate {{ ssl_cert_path }};
    ssl_certificate_key {{ ssl_key_path }};

    root {{ web_root }};
    index index.html;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

Écrire les tâches du rôle

Dans `roles/webserver_secure/tasks/main.yml` :

```
nano roles/webserver_secure/tasks/main.yml
```

Ajouter :

```
- name: Installer nginx
  apt:
    name: nginx
    state: present
```

```
    update_cache: yes
  become: yes

- name: Créer le dossier de certificats
  file:
    path: /etc/nginx/certs
    state: directory
    mode: '0755'

- name: Copier les certificats auto-signés
  copy:
    src: certs/
    dest: /etc/nginx/certs/
    mode: '0644'
  notify: Redémarrer nginx

- name: Déployer la page HTML
  template:
    src: index.html.j2
    dest: "{{ web_root }}/index.html"
    mode: '0644'

- name: Déployer la config nginx
  template:
    src: nginx.conf.j2
    dest: /etc/nginx/sites-available/default
    mode: '0644'
  notify: Redémarrer nginx

- name: Activer nginx
  service:
    name: nginx
    state: started
    enabled: true
```

Résumé global

Bonne pratique	Mise en œuvre	Dossier/fichier concerné
Utiliser des rôles	`ansible-galaxy init roles/mon_role`	`roles/`
Blocs `block`	Grouper plusieurs tâches conditionnelles	`tasks/main.yml`
Centraliser dans group_vars	`group_vars/webservers.yml`	`group_vars/`
Templates Jinja2	`nginx.conf.j2`, `index.html.j2`	`templates/`
Handlers	`Redémarrer nginx` si fichier changé	`handlers/main.yml` + `notify:` dans tâches