HESTIM
ENGINEERING & BUSINESS SCHOOL

Ingénierie Informatique, Intelligence Artificielle

Et Cybersécurité

Sujet :

Conception et Administration d'un Serveur Ubuntu : Sécurisation via Sudo et Supervision avec Cockpit

Réalisé par :
Oklin Ghislain TOURE

Membre de jury :
Pr.Azeddine KHIAT

Année universitaire
2025/2026

Table des matières

Liste des Figures

Introduction

Dans le cadre de l'administration de serveurs Linux, la gestion sécurisée des accès et la surveillance constante des ressources sont deux piliers fondamentaux pour garantir la stabilité et la sécurité d'une infrastructure. L'utilisation systématique du compte « root » présente des risques importants, d'où la nécessité de mettre en place une gestion fine des privilèges.
Ce TP a pour objectif la mise en service et la configuration d'un serveur Ubuntu hébergé sur une instance AWS EC2. Le travail réalisé s'articule autour de trois axes principaux :

1. **La gestion des utilisateurs** : création d'un utilisateur dédié nommé adminuser et attribution de droits d'administration via la commande sudo, permettant ainsi de restreindre l'utilisation du compte super-utilisateur.

2. **L'installation de Cockpit** : mise en place de cette interface web graphique open source qui permet de centraliser la gestion du serveur (monitoring CPU/RAM, gestion des services et des journaux système).

3. **La configuration réseau** : ajustement des règles de sécurité (Security Groups) sur la console AWS pour autoriser les flux sur le port 9090, nécessaire à l'accès distant de l'interface d'administration.

À travers ces étapes, nous verrons comment transformer une instance cloud brute en un serveur administrable de manière intuitive et sécurisée.

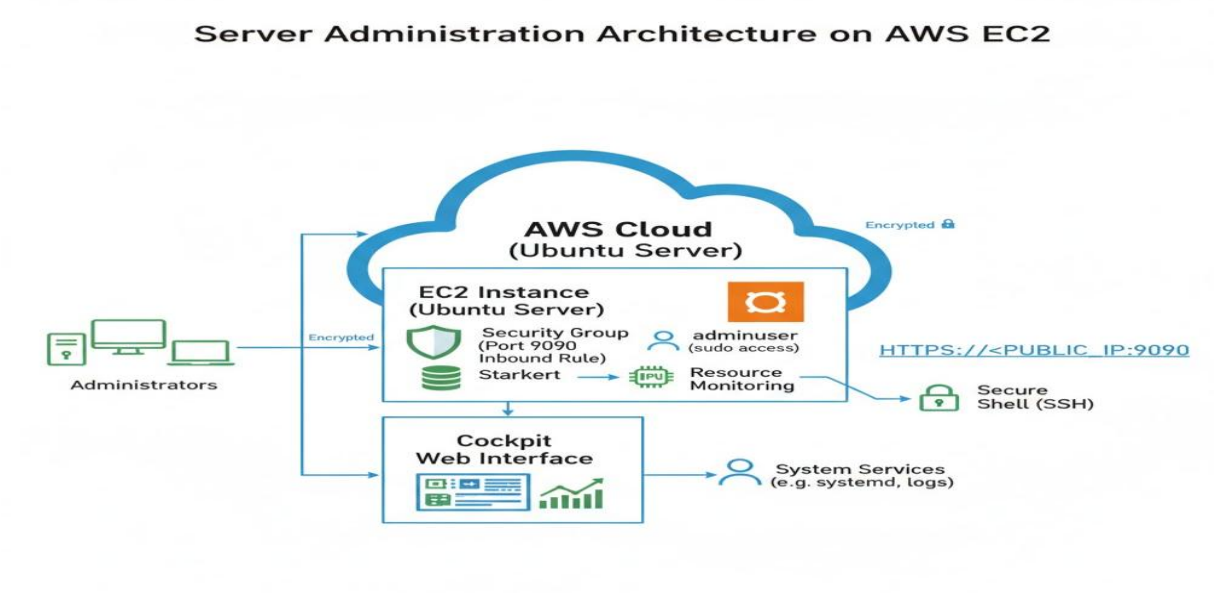

Figure 1:Architecture logique de l'administration du serveur Ubuntu sur AWS

1 Création et Configuration de l'Instance AWS EC2

Avant toute manipulation technique sur le système d'exploitation, nous devons déployer une machine virtuelle (instance) sur l'infrastructure AWS.

1.1 Choix de l'Image Système (AMI)

Pour ce TP, nous avons sélectionné une image **Ubuntu Server** (version Noble 24.04 LTS). Ce choix garantit un environnement Linux stable et moderne, compatible avec les dernières versions de Cockpit.

1.2 Paramétrage des ressources

- **Type d'instance** : Nous avons opté pour une **t2.micro**. Ce type d'instance fait partie de l'offre gratuite (Free Tier) d'AWS et offre des ressources suffisantes (1 vCPU, 1 Go de RAM) pour les tâches d'administration de base et le monitoring.

- **Stockage** : Utilisation d'un volume **de 8 Go** pour le système de fichiers.

1.3 Configuration de la Sécurité (Pare-feu)

Lors de la création, nous définissons un **Security Group** (groupe de sécurité). Initialement, le port **22 (SSH)** ainsi que les ports **443 (HTTPS)** et **80 (HTTP)** sont ouvert pour permettre l'accès à distance en ligne de commande.

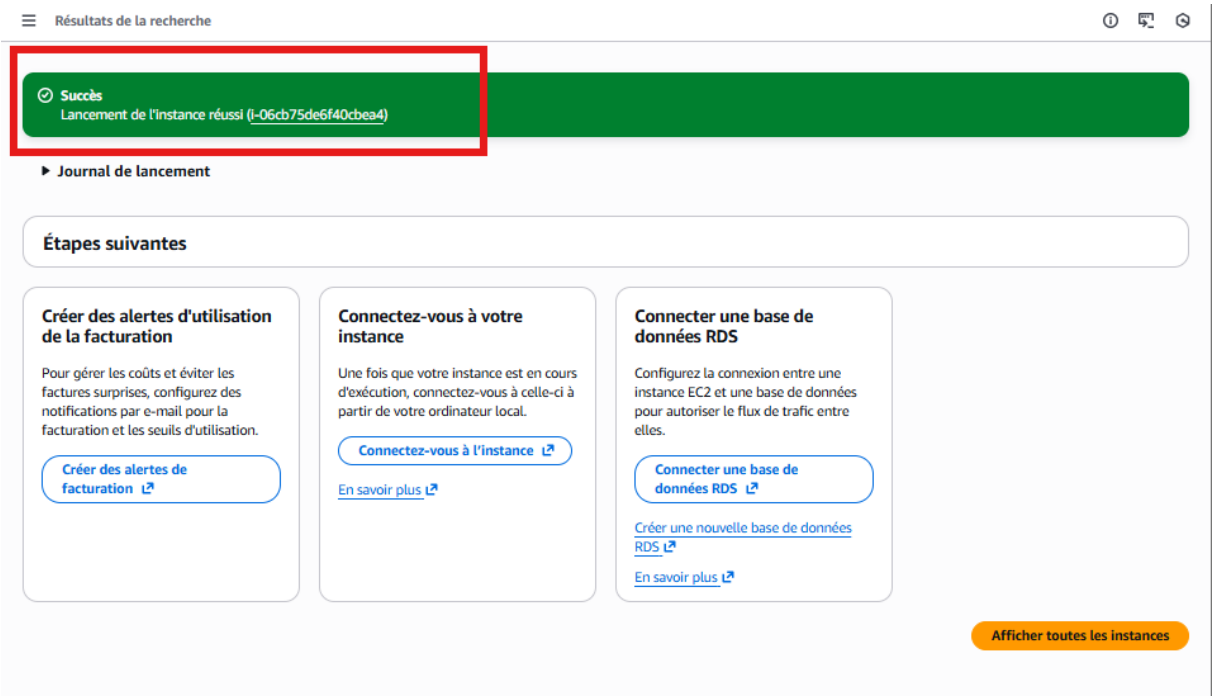

Figure 2: Création Instance VM_Linux_Cockpit

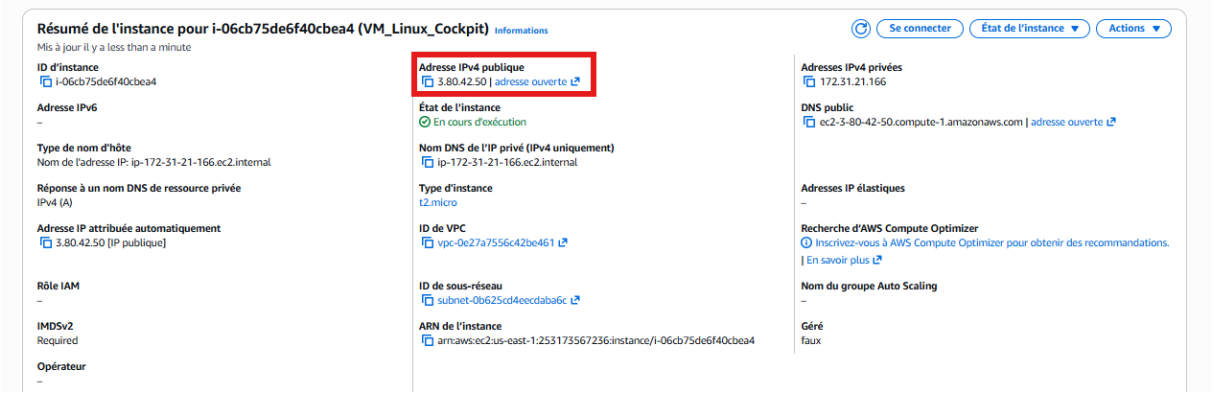

Figure 3:Résumé de l'instance sur la console AWS montrant l'ID de l'instance et l'Adresse IPv4 publique.

2 Mise en place de l'environnement et gestion des accès

Cette phase initiale consiste à établir une communication sécurisée avec l'instance déployée sur AWS et à configurer un utilisateur administrateur pour sécuriser les opérations futures.

2.1 Connexion au serveur distant

La première étape est l'accès au serveur via le protocole SSH. Nous utilisons une clé privée (tp-ubuntu-key.pem) pour authentifier la connexion vers l'utilisateur par défaut ubuntu.

- **Commande de connexion** : ssh -i "tp-ubuntu-key.pem" ubuntu@ec2-3-80-42-50.compute-1.amazonaws.com
- **Vérification** : Une fois connecté, le prompt affiche ubuntu@ip-172-31-21-166.

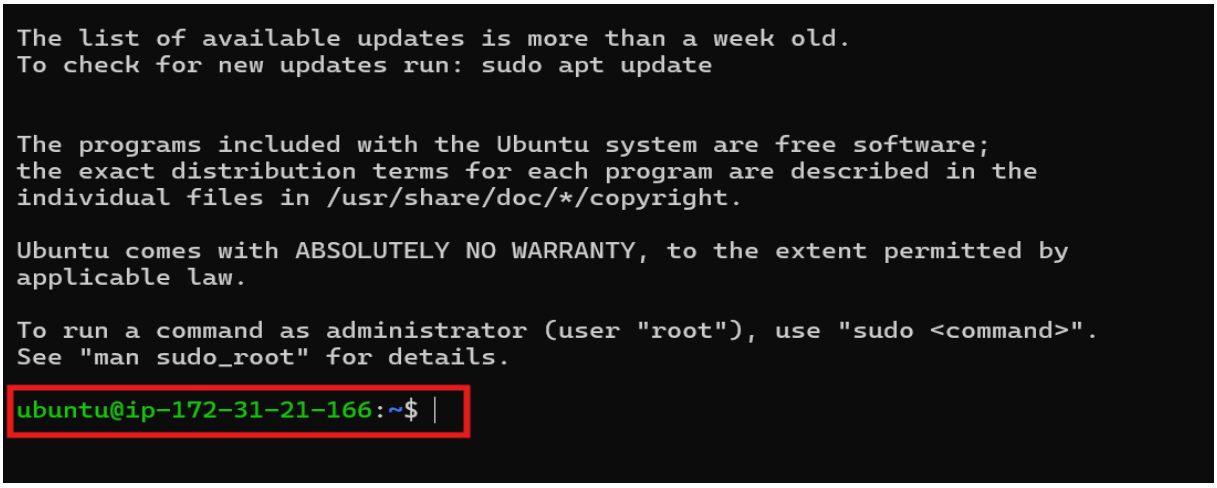

Figure 4:Terminal montrant le prompt après la connexion réussie.

2.2 Création de l'utilisateur administrateur

Pour respecter les principes de sécurité, nous créons un utilisateur dédié nommé adminuser au lieu d'utiliser le compte générique.

1. **Ajout de l'utilisateur** : La commande sudo adduser adminuser initie la création.
2. **Configuration** : Le système demande de définir un mot de passe et de renseigner des informations facultatives (Nom, bureau, etc.) que l'on peut ignorer en appuyant sur **Entrée**.

```
ubuntu@ip-172-31-21-166:~$ sudo adduser adminuser
info: Adding user `adminuser' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `adminuser' (1001) ...
info: Adding new user `adminuser' (1001) with group `adminuser (1001)' ...
info: Creating home directory `/home/adminuser' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for adminuser
Enter the new value, or press ENTER for the default
        Full Name []: adminuser
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
info: Adding new user `adminuser' to supplemental / extra groups `users' ...
info: Adding user `adminuser' to group `users' ...
ubuntu@ip-172-31-21-166:~$
```

Figure 5:Création du nouvel utilisateur.

2.3 Attribution des droits Root (Sudo)

Par défaut, le nouvel utilisateur n'a pas les privilèges pour modifier le système. Nous devons lui accorder des droits via le groupe sudo.

- **Commande d'attribution** : sudo usermod -aG sudo adminuser.
- **Vérification des privilèges** : En utilisant la commande id, on vérifie que l'utilisateur appartient bien au groupe 27(sudo).
- **Test de basculement** : On change d'identité avec su adminuser puis on teste la capacité d'administration avec sudo apt update.

```
Is the information correct? [Y/n] Y
info: Adding new user `adminuser' to supplemental / extra groups `users' ...
info: Adding user `adminuser' to group `users' ...
ubuntu@ip-172-31-21-166:~$ sudo usermod -aG sudo adminuser
ubuntu@ip-172-31-21-166:~$ id
uid=1000(ubuntu) gid=1000(ubuntu) groups=1000(ubuntu),4(adm),24(cdrom),27(sudo),30(dip),105(lxd)
ubuntu@ip-172-31-21-166:~$ su adminuser
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

adminuser@ip-172-31-21-166:/home/ubuntu$ sudo apt update
[sudo] password for adminuser:
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1410 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1717 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [317 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [16.0 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1525 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [312 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [386 kB]
```

Figure 6:Résultat de la commande id montrant l'appartenance au groupe sudo.

2.4 Test de reconnexion SSH

Lors de la tentative de reconnexion directe avec adminuser, une erreur de sécurité survient : **"Permission denied (publickey)"**.

2.4.1 Analyse de l'erreur

Cette erreur est normale car, contrairement à l'utilisateur ubuntu, le dossier personnel de adminuser ne contient pas encore la clé publique nécessaire pour autoriser une connexion SSH sans mot de passe depuis notre machine locale.

2.4.2 Procédure de resolution

Pour permettre la connexion de adminuser, nous avons dû copier les clés autorisées depuis le compte ubuntu :

1. **Création du répertoire sécurisé** : sudo mkdir -p /home/adminuser/.ssh.
2. **Copie de la clé** : sudo cp /home/ubuntu/.ssh/authorized_keys /home/adminuser/.ssh/.
3. **Attribution des droits** : Nous avons rendu adminuser propriétaire de ses fichiers et restreint les permissions pour garantir la sécurité (chmod 700 pour le dossier et 600 pour le fichier).

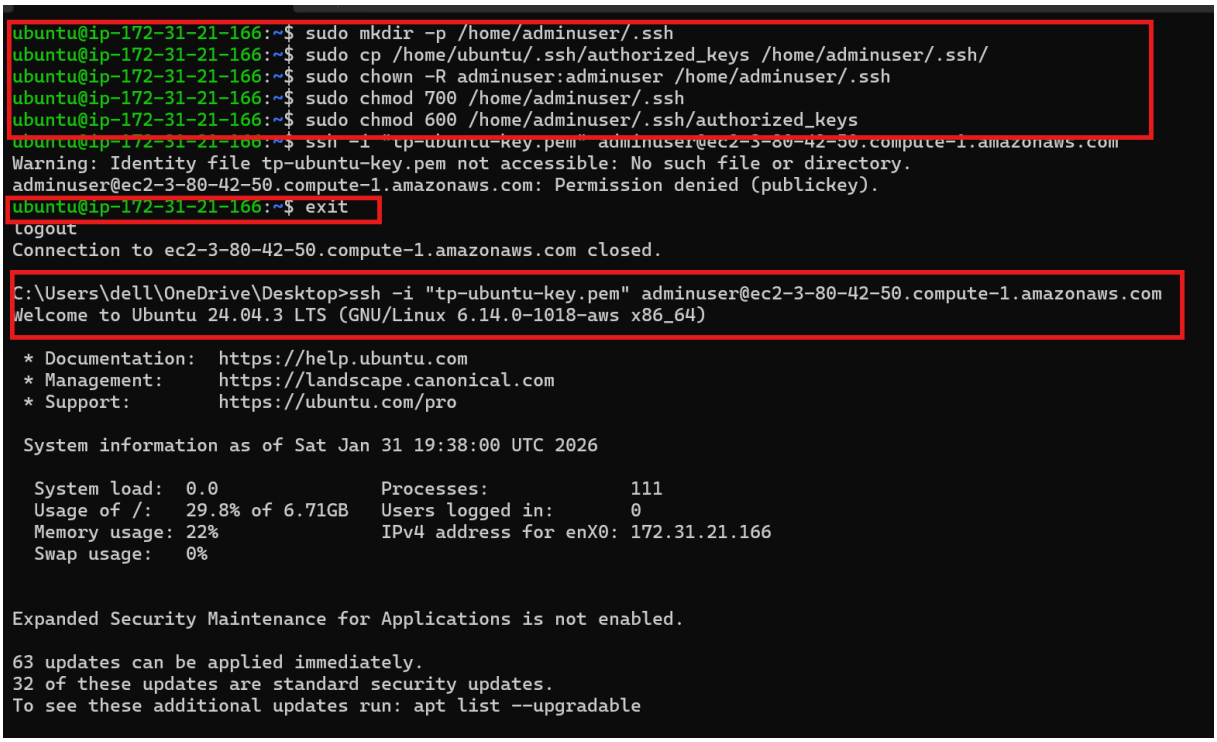

Figure 7:Commandes mkdir, cp, chown et chmod suivies de la connexion réussie.

2.4.3 Validation finale de l'acces

Une fois la clé configurée, la connexion avec adminuser est opérationnelle. L'utilisateur peut désormais s'authentifier et utiliser ses privilèges sudo pour mettre à jour le système avant l'installation de Cockpit.

```
C:\Users\dell\OneDrive\Desktop>ssh -i "tp-ubuntu-key.pem" adminuser@ec2-3-80-42-50.compute-1.amazonaws.com
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1018-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sat Jan 31 20:13:36 UTC 2026

  System load:  0.0               Processes:             112
  Usage of /:   29.8% of 6.71GB   Users logged in:       1
  Memory usage: 22%               IPv4 address for enX0: 172.31.21.166
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

63 updates can be applied immediately.
32 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Sat Jan 31 19:38:01 2026 from 102.103.249.209
adminuser@ip-172-31-21-166:~$ sudo apt update
[sudo] password for adminuser:
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
```

Figure 8:Mise a jour depuis adminuser

3 Installation et configuration de l'interface Cockpit

L'objectif de cette partie est d'installer l'outil de gestion graphique et de rendre le serveur accessible via un navigateur web.

3.1 Installation du service Cockpit

Cockpit est une interface web open source qui permet de surveiller l'état du serveur (CPU, RAM, disques) et de gérer les services sans utiliser la ligne de commande.

1. **Mise à jour des dépôts** : Avant l'installation, nous mettons à jour la liste des paquets avec sudo apt update.

```
ubuntu@ip-172-31-21-166:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
69 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-21-166:~$
```

Figure 9:Mise a jour sudo apt update

2. **Installation** : Nous installons le paquet principal avec la commande sudo apt install cockpit -y. Cette opération installe également des dépendances nécessaires comme cockpit-bridge, cockpit-networkmanager et cockpit-storaged.

```
ubuntu@ip-172-31-21-166:~$ sudo apt install cockpit -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  cockpit-bridge cockpit-networkmanager cockpit-packagekit
  cockpit-storaged cockpit-system cockpit-ws cracklib-runtime
  dconf-gsettings-backend dconf-service dns-root-data
  dnsmasq-base glib-networking glib-networking-common
  glib-networking-services gsettings-desktop-schemas
  libbluetooth3 libcrack2 libdconf1 libndp0 libnm0
  libpcsclite1 libproxy1v5 libpwquality-common
  libpwquality-tools libpwquality1 libteamdctl0
  network-manager network-manager-pptp ppp pptp-linux
  session-migration wamerican wireless-regdb wpasupplicant
Suggested packages:
  cockpit-doc cockpit-pcp cockpit-sosreport xdg-utils
```

Figure 10:Installation Cockpit

3. **Activation du service** : Pour que Cockpit démarre automatiquement, nous utilisons la commande : sudo systemctl enable --now cockpit.socket.

```
ubuntu@ip-172-31-21-166:~$ sudo systemctl enable --now cockpit.socket
ubuntu@ip-172-31-21-166:~$ sudo systemctl status cockpit.socket
● cockpit.socket - Cockpit Web Service Socket
     Loaded: loaded (/usr/lib/systemd/system/cockpit.socket; en>
     Active: active (listening) since Sat 2026-01-31 20:22:38 U>
   Triggers: ● cockpit.service
       Docs: man:cockpit-ws(8)
     Listen: [::]:9090 (Stream)
      Tasks: 0 (limit: 1121)
     Memory: 248.0K (peak: 1.7M)
        CPU: 16ms
     CGroup: /system.slice/cockpit.socket

Jan 31 20:22:38 ip-172-31-21-166 systemd[1]: Starting cockpit.s>
Jan 31 20:22:38 ip-172-31-21-166 systemd[1]: Listening on cockp>
lines 1-13/13 (END)...skipping...
● cockpit.socket - Cockpit Web Service Socket
     Loaded: loaded (/usr/lib/systemd/system/cockpit.socket; enabled; preset: enabled)
     Active: active (listening) since Sat 2026-01-31 20:22:38 UTC; 1min 45s ago
   Triggers: ● cockpit.service
       Docs: man:cockpit-ws(8)
     Listen: [::]:9090 (Stream)
      Tasks: 0 (limit: 1121)
     Memory: 248.0K (peak: 1.7M)
        CPU: 16ms
     CGroup: /system.slice/cockpit.socket

Jan 31 20:22:38 ip-172-31-21-166 systemd[1]: Starting cockpit.socket - Cockpit Web Service Socket...
Jan 31 20:22:38 ip-172-31-21-166 systemd[1]: Listening on cockpit.socket - Cockpit Web Service Socket.
~
```

Figure 11:Résultat de sudo systemctl status cockpit.socket avec la mention "active (listening)" sur le port 9090.

3.2 Configuration du Pare-feu (Security Group) sur AWS

Par défaut, AWS bloque toutes les connexions entrantes sauf le port 22 (SSH). Le service Cockpit utilisant le port **9090**, il est impératif d'ouvrir ce flux dans la console AWS.

- **Procédure** :
    1. Accéder à la console **EC2** et sélectionner l'instance.
    2. Aller dans l'onglet **Sécurité** puis cliquer sur le **Security Group**.

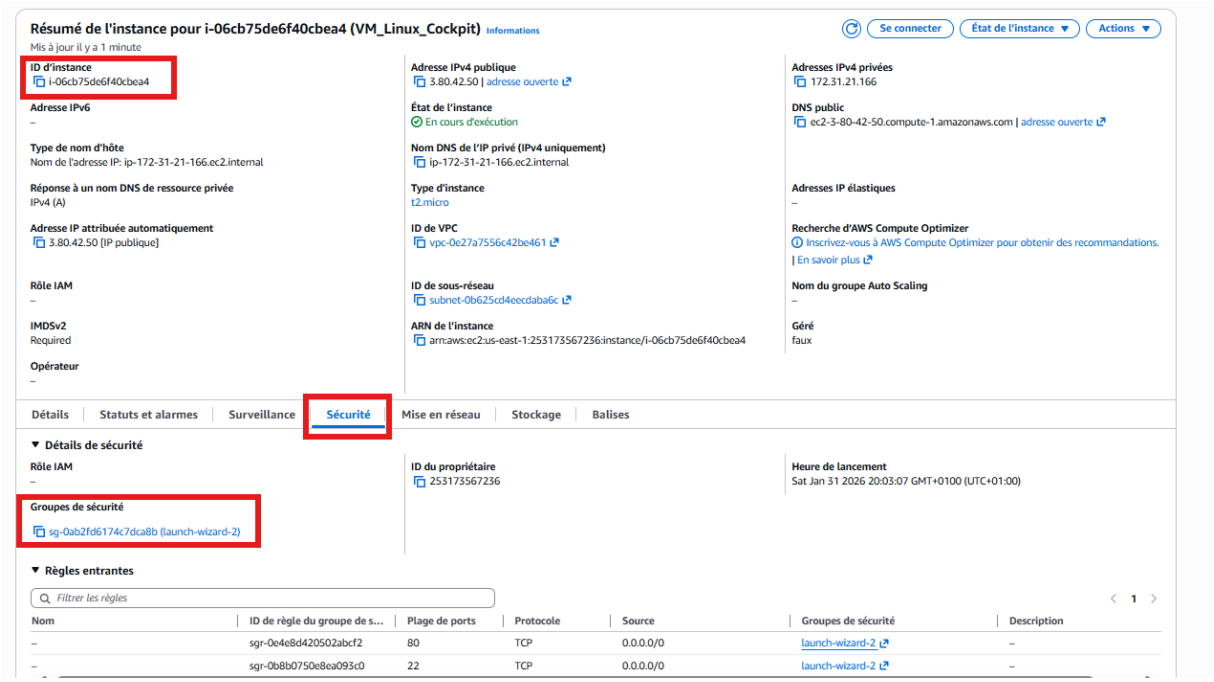

Figure 12:Résumé Instance

3. Cliquer sur **Modifier les règles entrantes**.

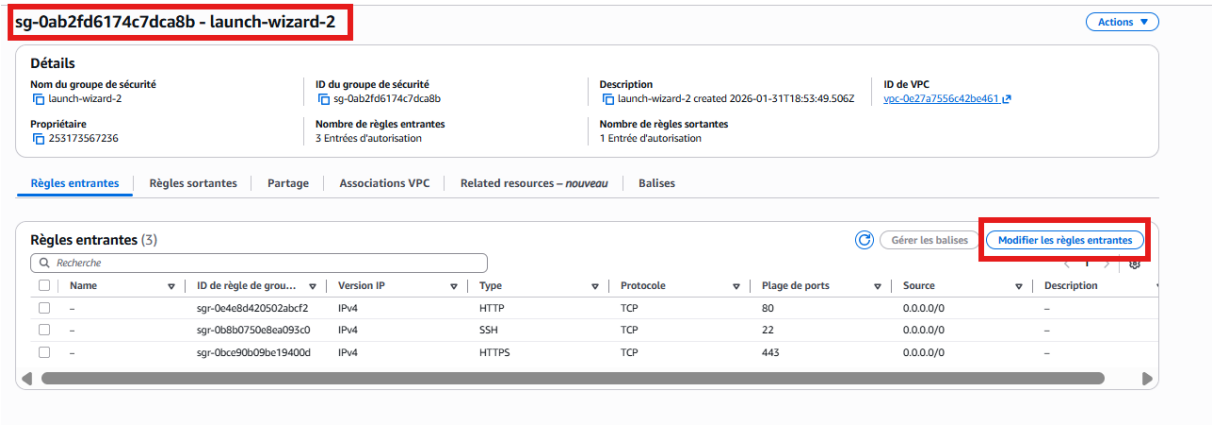

Figure 13: Regles Entrantes de l'instance

4. Ajout règle : Type **Custom TCP**, Port **9090**, Source **0.0.0.0/0**.

Figure 14:Ajout nouvelle Regle port 9090

3.3 Premier accès à l'interface Web

Une fois le port ouvert, l'interface est accessible via l'adresse IP publique du serveur.

- **URL de connexion** : https:// 3.80.42.50:9090

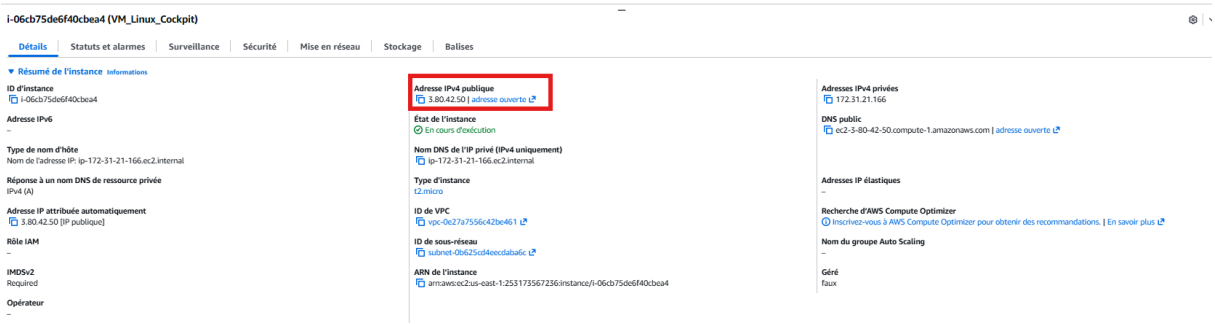

Figure 15:Résumé Instance montrant Adresse IPv4

- **Authentification** : Nous utilisons les identifiants de l'utilisateur créé précédemment : adminuser.

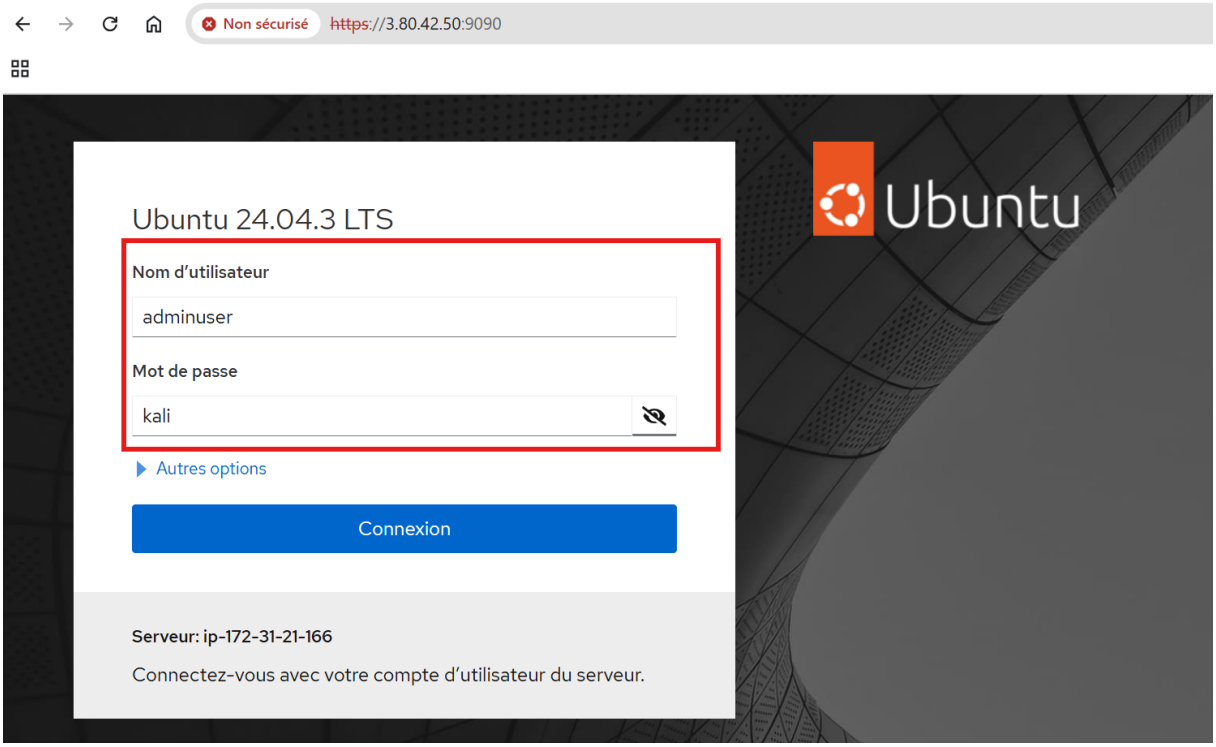

Figure 16:Page de connexion de Ubuntu

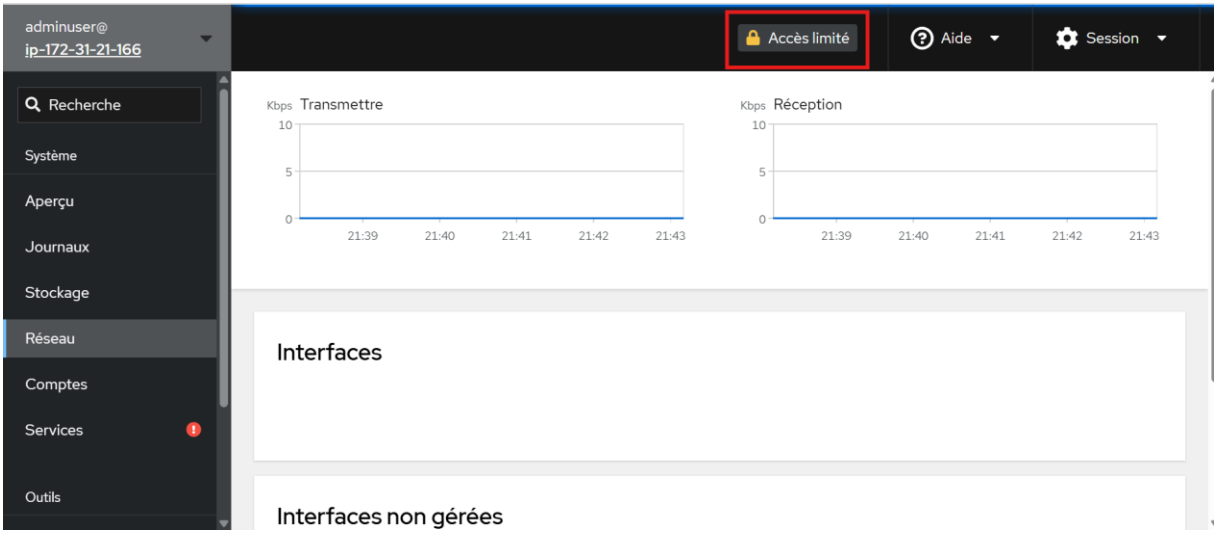

Figure 17:Tableau de bord principal montrant les graphiques de performance

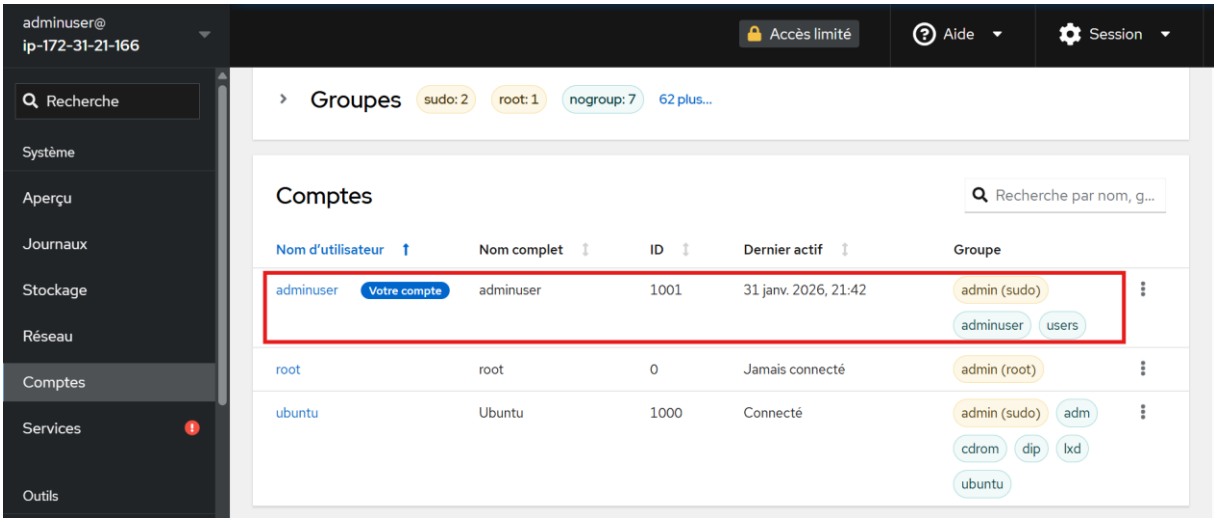

Figure 18:tableau de bord principal montrant les comptes

Conclusion

La réalisation de ce TP a permis de mettre en œuvre les étapes fondamentales de la sécurisation et de la gestion d'un serveur Linux dans un environnement Cloud (AWS).
D'une part, nous avons renforcé la sécurité du système en créant un utilisateur dédié (adminuser) doté de privilèges restreints via sudo. L'analyse de l'erreur lors de la reconnexion SSH a été un point clé du travail, soulignant l'importance de la gestion des clés publiques pour l'authentification sécurisée.

D'autre part, l'installation de l'interface Cockpit a démontré qu'il est possible de simplifier considérablement l'administration système. Cet outil offre une visibilité immédiate sur les ressources critiques (CPU, RAM, logs) et permet d'effectuer des tâches complexes, comme la gestion des comptes ou des services, via une interface web intuitive.

Enfin, la configuration des groupes de sécurité AWS pour l'ouverture du port 9090 nous a rappelé que l'administration d'un serveur ne se limite pas au système d'exploitation, mais inclut également la maîtrise de l'infrastructure réseau qui l'héberge.

En conclusion, la combinaison d'une gestion rigoureuse des accès en ligne de commande et d'un outil de supervision graphique comme Cockpit constitue une base solide pour tout administrateur système souhaitant allier sécurité et efficacité opérationnelle.