

# Inst. Serv. Rés.

**Serv-182**

---

**SAE 2.03**

---



**Nom : BOULOUIHA Yassir, RABAN Théo**

*Informatique 1<sup>ière</sup> Année*

# Table des matières :

<b>Introduction</b>	<b>2</b>
<b>Information distribution Linux:</b>	<b>3</b>
Nom et version de la distribution :	3
Site de référence Debian:	3
Date fin de vie de la distribution Debian 11 bullseye:	3
<b>Liste des commandes utilisées</b>	<b>4</b>
<b>1 Mission : création d'un compte par étudiant·e</b>	<b>4</b>
<b>2 Mission : création du compte enseignant</b>	<b>6</b>
<b>3 Mission : installation du logiciel cowsay</b>	<b>7</b>
<b>Conclusion</b>	<b>7</b>

## Introduction

Voici le compte rendu de notre première mission concernant la SAE 2.03. Les objectifs de cette semaine étaient de se connecter à un serveur attribué, de déterminer sa distribution linux (en l'occurrence Debian) et ensuite de réaliser les missions fournies. Notre première et deuxième mission consistait à créer nos profils utilisateurs (étudiants : Théo et Yassir) mais aussi à créer un profil enseignant.

Ces profils utilisateurs devaient répondre à plusieurs contraintes, ils devaient notamment appartenir au groupe sudoers et être munis d'un mot de passe.

Notre dernière mission consistait à installer un logiciel qui permet de faire parler une vache (ou autre animal) en ASCII art.

## Information distribution Linux:

Comme précédemment explicité dans l'introduction notre distribution linux est Debian.

### **Nom et version de la distribution :**

Obtenu grâce à la commande: cat /etc/os-release  
 Debian GNU/Linux 11 (bullseye)

### **Site de référence Debian:**

Voici le site de référence de Debian :

<https://www.debian.org>

Voici le site de référence support de Debian :

<https://www.debian.org/support>

### **Date fin de vie de la distribution Debian 11 bullseye:**

<https://wiki.debian.org/LTS>

Version	support architecture	schedule
<b>Previous LTS Releases</b>		
Debian 6 "Squeeze"	i386 and amd64	2nd June 2014 to 29th of February 2016
Debian 7 "Wheezy"	i386, amd64, armel and armhf	26th April 2016 to 31st May 2018
Debian 8 "Jessie"	i386, amd64, armel and armhf	17th June 2018 to June 30, 2020
Debian 9 "Stretch"	i386, amd64, armel, armhf and arm64	July 6, 2020 to June 30, 2022
<b>Current LTS Release(s)</b>		
Debian 10 "Buster"	i386, amd64, armhf and arm64	August 1st, 2022 to June 30th, 2024
<b>Future LTS Release(s)</b>		
Debian 11 "Bullseye"	i386, amd64, armhf and arm64	August 15th, 2024 to June 30th, 2026
Debian 12 "Bookworm"	i386, amd64, armhf and arm64	June 11th, 2026 to June 30th, 2028

Legend: End of life Supported by LTS team Supported by security and release teams

La date de fin de vie de la distribution est le 30 juin 2026.

## Liste des commandes utilisées

### **1 Mission : création d'un compte par étudiant·e**

Pour créer un compte initialement nous avions utilisé la commande useradd, cependant cette commande ne créait pas de home directory (répertoire d'accueil) pour nos utilisateurs récemment créés, nous avons donc plutôt utilisé la commande "adduser" de cette façon :

**\$sudo adduser nom\_utilisateur**

Cette commande nous demande un mot de passe après l'exécution, ainsi que des informations complémentaires optionnelles (nom, numéro de téléphone etc...).

```
yassir@SAE-2324-DEBIAN-04:~$ sudo adduser toto
[sudo] Mot de passe de yassir :
Ajout de l'utilisateur « toto » ...
Ajout du nouveau groupe « toto » (1004) ...
Ajout du nouvel utilisateur « toto » (1005) avec le groupe « toto » ...
Création du répertoire personnel « /home/toto » ...
Copie des fichiers depuis « /etc/skel » ...
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd: password updated successfully
Changing the user information for toto
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Cette information est-elle correcte ? [0/n]0
yassir@SAE-2324-DEBIAN-04:~$
```

Après avoir créé les deux utilisateurs on les a ajoutés au groupe sudoers avec la commande :

**\$sudo adduser nom\_utilisateur sudo**

Cette commande permet de donner des permissions spéciales aux utilisateurs, qui leur permettent de faire un peu pres tout, à condition d'être en mode exécution privilégiée (en préfixant une commande avec sudo ou en basculant en mode sudo directement avec sudo su - )

Pour créer nos fichiers texte de présentation WELCOME.txt dans les dossiers d'accueil de nos utilisateurs respectifs :

**\$touch WELCOME.txt  
\$nano WELCOME.txt**

La commande touch permet de créer un fichier de type ordinaire, et la commande nano permet quant à elle de créer ou modifier (en fonction si le fichier en paramètre existe déjà ou non) un fichier sur le terminal.

Enfin pour cette partie nous avons désactivé le compte étudiant à l'aide de la commande :

**sudo passwd -l etudiant**

L'option -l de la commande passwd permet de bloquer le mot de passe de l'utilisateur passé en paramètre dans notre cas etudiant. Cela aura pour incidence l'impossibilité pour l'utilisateur etudiant de se connecter étant donné qu'il ne peut pas utiliser son mot de passe.

On a également utilisé des commandes comme groupdel ou userdel, pour supprimer des utilisateurs, et des groupes que l'on ne voulait plus.

Nous avons également en parallèle utilisé des commandes vues en S1 telles que :

**ls** : afin de lister le contenu d'un fichier de type répertoire

**pwd** : afin d'afficher le chemin absolu du répertoire de travail actuel

**cat** : affiche le contenu d'un fichier de type ordinaire à l'écran

**tree** (installé car non présent de base) : permet d'afficher une représentation graphique de l'arborescence.

**echo** : permet d'afficher à l'écran un message, et avec une redirection (**echo "texte" > ./monfichiertexte.txt**) d'écrire dans un fichier texte.

## **2 Mission : création du compte enseignant**

Pour créer un compte enseignant nous avons utilisé la même commande que pour la création des anciens comptes :

**\$sudo adduser enseignant**

Suite à cela on a créé le dossier caché .ssh dans le home directory de enseignant puis copié la clé fournie dans le fichier authorized\_keys dans ce même dossier (.ssh) à l'aide de ces commandes :

```
$sudo cd /home/enseignant  
$sudo mkdir .ssh  
$ls -a  
$sudo echo "cle_publique" > ./ssh/authorized_keys
```

On a utilisé l'option -a sur le ls afin de pouvoir afficher tous les fichiers même ceux qui sont cachés notamment .ssh .

## **3 Mission : installation du logiciel cowsay**

Nous avons dû dans un premier temps installer le logiciel, pour cela nous avons dû utiliser la commande suivante:

**sudo apt-get install cowsay**

La partie "sudo" de cette commande permet de passer en super administrateur de manière temporaire pour pouvoir poursuivre l'exécution du reste de la commande qui permet d'installer le package de cowsay.

**cowsay ....**

Cette commande permet de faire parler la vache en lui faisant dire quelque chose (dans notre cas le quelque chose est représenté par les points).

## Conclusion

Ces premières missions nous ont permis de nous faire réviser les commandes Unix vues au premier semestre mais également de nous faire découvrir certains aspects de l'administration des systèmes Unix. En effet, nous avons appris à repérer la distribution Linux d'un serveur (Linux) et aussi d'y repérer le site officiel de la distribution du serveur.

Nous avons ensuite découvert comment créer des profils utilisateur sudoers en utilisant la commande "adduser", nous savons également maintenant comment passer en mode super utilisateur pour exécuter des commandes grâce à "sudo".

De plus, cette première semaine de mission, nous a appris un installer des "packages" sous unix Debian grâce à la commande "apt-get install" ainsi nous avons pu installer "tree" (qui permet de visualiser les arborescences de fichiers à partir de l'emplacement de l'utilisateur dans cette même arborescence) et cette commande nous a aussi permis d'installer le logiciel cowsay sur notre serveur.

Finalement, nous pouvons dire que cette première semaine de SAE 2.03 nous a été très instructive et nous a permis de commencer à monter en compétence dans l'administration des systèmes Linux.

# Inst. Serv. Rés.

**Serv-182**

---

**SAE 2.03**

---



**Nom : BOULOUIHA Yassir, RABAN Théo**

*Informatique 1<sup>ière</sup> Année*

# Table des matières :

<b>Introduction</b>	<b>2</b>
<b>Mission 1</b>	<b>2</b>
<b>Mission 2</b>	<b>4</b>
<b>Mission 3</b>	<b>4</b>
<b>Mission 4</b>	<b>6</b>
<b>Mission 5</b>	<b>6</b>
<b>Conclusion</b>	<b>7</b>

## Introduction

Les missions de cette semaine ont pour objectif de nous faire travailler et comprendre les fichiers utilisés lors de la création d'un nouvel utilisateur mais également de nous faire manipuler et modifier ces fichiers dans l'objectif de personnaliser l'espace utilisateur initialement défini.

Les missions vont nous permettre de monter en compétences notamment dans la gestion des comptes utilisateurs mais également de manière plus généralisée grâce à la découverte de nouvelles commandes Unix.

## Mission 1

### **.profile:**

Ce fichier est source par chaque shell de connexion (login) et par le premier shell non-login interactif. Il est utilisé pour configurer l'environnement global de l'utilisateur.

### **.bashrc:**

Ce fichier est source par chaque shell interactif non-login (bash lancé par un utilisateur). Il est utilisé pour configurer l'environnement de travail de l'utilisateur.

### **.bash\_logout:**

Ce fichier est exécuté lorsqu'un utilisateur quitte un shell de connexion interactif. Son utilité est limitée, car il n'est pas toujours appelé (déconnexion via SSH, fermeture du terminal sans déconnexion).

nom des fichiers	Rôle Précis
.profile	<ul style="list-style-type: none"> <li>● Définir des variables d'environnement globales.</li> <li>● Définir des alias de commandes globales.</li> <li>● Charger des fonctions et des scripts de configuration système.</li> </ul>
bashrc	<ul style="list-style-type: none"> <li>● Définir des alias de commandes.</li> <li>● Définir des variables d'environnement.</li> <li>● Charger des fonctions et des scripts.</li> <li>● Personnaliser l'invite de commande.</li> </ul>
.bash_logout	<ul style="list-style-type: none"> <li>● Exécuter des actions de nettoyage lors de la déconnexion, comme la suppression de fichiers temporaires.</li> <li>● Définir des messages de déconnexion personnalisés.</li> </ul>

## Mission 2

2.1) Pour faire cela nous avons ajouté dans le fichier caché .bashrc les lignes suivantes :

```
echo "Bienvenue dans votre Serveur Debian!"
```

Le fichier .bashrc étant exécuté à chaque connexion de l'utilisateur ce message sera donc affiché lors de chaque connexion des nouveaux utilisateurs.

```
export EDITOR=/usr/bin/nano
```

Cette commande permet de changer l'éditeur de base pour la crontab en renseignant le chemin absolu pour trouver l'éditeur choisi dans notre cas nano.

```
alias contenu='ls -la'  
alias dns-restart='sudo systemctl restart systemd-resolved'  
alias vache='cowsay -d MEEEEUUUUUUUHHH'
```

Ces commandes permettent de définir des alias, c'est des mots clés qui pourront ensuite être rentrés dans le terminal et qui exécuteront les commandes qui sont de l'autre côté du égal.

```
#creation rep IUT et definition du rep de travail
export USERNAME=$(whoami)
export IUT_DIR="/home/$USERNAME/IUT/"
if [ ! -d $IUT_DIR ]
then
    mkdir $IUT_DIR
fi
cd $IUT_DIR
```

Ce script permet de créer un rep IUT dans le répertoire de chaque nouvel utilisateur et de le définir en tant que répertoire de travail lors de sa connexion.

## Mission 3

Pour réaliser cette mission nous avons dans un premier temps créé un répertoire personnel intitulé Consignes dans lequel nous avons mis un fichier de type ordinaire ayant pour nom README.txt. Nous avons placé le répertoire "consignes" dans /etc/skel afin qu'il soit copié sur tous les espaces personnels des prochains utilisateurs qui seront créés.

Dans un deuxième temps nous avons pu concevoir et placer le script de sauvegarde automatique commenté ci-dessous dans le fichier ".bashrc".

```
#Script backup

# Répertoire source à sauvegarder
export source_dir="/home/$USERNAME"

# Répertoire de destination pour les sauvegardes
export backup_dir="/sauvegardes/$USERNAME"

if [ ! -d $backup_dir ]
then
    mkdir "$backup_dir"
fi

# Créer un nom de répertoire unique pour la sauvegarde
export backup_dir_name="backup_$(date +%d-%m-%Y_%H-%M-%S)"

# Chemin complet du répertoire de sauvegarde
export full_backup_dir="$backup_dir/$backup_dir_name"

# Copier le répertoire source vers le répertoire de sauvegarde
cp -r "$source_dir" "$full_backup_dir"

# Afficher un message de confirmation
if [ -d "$full_backup_dir" ]
then
    echo "Sauvegarde terminée : $full_backup_dir"
fi
```

Pour arriver à la faire exécuter le script de sauvegarde automatique présent dans .bashrc à la connexion de l'utilisateur il a fallu modifier les droits du répertoire "/sauvegarde" en ajoutant les droits en écriture pour tous **"sudo chmod a+w /sauvegardes"**.

Le fichier README.txt fournit des instructions utiles pour guider les nouveaux utilisateurs, selon le message inscrit dedans.

Le script backup.sh permet aux utilisateurs de sauvegarder automatiquement leur répertoire personnel. Cela garantit la sécurité des données en cas de perte de fichiers ou de dysfonctionnement du système. En automatisant ce processus, les utilisateurs n'ont pas à se soucier de faire des sauvegardes manuelles régulières, ce qui peut souvent être oublié ou négligé.

## Mission 4

Voici la capture d'écran de la création de l'utilisateur skeleton, comme vous pouvez le constater la sauvegarde a bien été effectuée.

```
theo@SAE-2324-DEBIAN-04:~$ sudo adduser skeleton
Ajout de l'utilisateur « skeleton » ...
Ajout du nouveau groupe « skeleton » (1005) ...
Ajout du nouvel utilisateur « skeleton » (1006) avec le groupe « skeleton » ...
Création du répertoire personnel « /home/skeleton » ...
Copie des fichiers depuis « /etc/skel » ...
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd: password updated successfully
Changing the user information for skeleton
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Cette information est-elle correcte ? [0/n]
theo@SAE-2324-DEBIAN-04:~$ su skeleton
Mot de passe :
Bienvenue dans votre Serveur Debian !
Sauvegarde terminée : /sauvegardes/skeleton/backup_01-03-2024_16-08-54
```

Le fichier répertoire consignes est bien copié dans l'espace personnel de l'utilisateur créé.

```
skeleton@SAE-2324-DEBIAN-04:~/IUT$ tree /home/skeleton
/home/skeleton
└── consignes
    └── README.txt
IUT
```

Voici ici le backup des actions effectuées par l'utilisateur que nous avons créé précédemment.

```
skeleton@SAE-2324-DEBIAN-04:~/IUT$ tree /sauvegardes
/sauvegardes
└── skeleton
    ├── backup_01-03-2024_14-12-33
    │   └── consignes
    │       └── README.txt
    └── IUT
    └── backup_01-03-2024_16-08-54
        └── consignes
            └── README.txt
IUT
```

Cette mission nous a permis de confirmer que les actions précédemment effectuées ont bien étées effectives sur l'utilisateur de test skeleton.

## Mission 5

### **Quelle est la différence entre « /etc/skel/ » et le répertoire personnel d'un utilisateur (~) ?**

Le répertoire /etc/skel/ est utilisé comme modèle pour créer le répertoire personnel d'un nouvel utilisateur. Tout ce qui est contenu dans ce répertoire sera copié dans le répertoire personnel de chaque nouvel utilisateur lors de leur création. Tandis que le répertoire “~” est le répertoire spécifique à chaque utilisateur. Il contient les fichiers et répertoires propres à cet utilisateur, y compris ses paramètres, ses fichiers personnels, ses scripts, etc.

### **Pourquoi est-il important de personnaliser le profil d'un nouvel utilisateur sur un système Linux ?**

La personnalisation du profil d'un nouvel utilisateur permet d'adapter les droits par défauts, les consignes du serveur en fonction de l'administrateur du serveur, des exigences en termes de sécurité et du contexte d'application de ce serveur.

### **Pourquoi est-il important de tester les configurations personnalisées en tant que nouvel utilisateur avant de les déployer à grande échelle ?**

Tester les configurations personnalisées en tant que nouvel utilisateur permet de s'assurer de leur fonctionnement. Les tester avant d'accueillir plusieurs personnes sur le serveur permet également de limiter l'impact sur les utilisateurs du serveur linux avant qu'il soit ouvert à d'autres utilisateurs.

## Conclusion

Cette deuxième semaine de mission nous a été très instructive et nous a permis d'acquérir des compétences en personnalisation d'espace utilisateurs et gestion des systèmes Unix de façon plus globale.

En effet, nous connaissons désormais l'utilité du répertoire `/etc/skel/` et l'utilité des différents fichiers qu'il contient permettant de définir des droits et des actions à la connexion des utilisateurs ajoutés.

Enfin nous pouvons dire que nous avons trouvé les missions de cette semaine plus difficiles que celles de la semaine précédente.

# Inst. Serv. Rés.

**Serv-182**

---

**SAE 2.03**

---



**Nom : BOULOUIHA Yassir, RABAN Théo**

*Informatique 1<sup>ière</sup> Année*

# Table des matières :

<b>Introduction</b>	<b>2</b>
<b>Résumé des modifications effectuées sur votre site internet :</b>	<b>3</b>
Structure et organisation:	3
Design et couleurs:	3
Eléments du site:	3
Responsive design:	4
Objectifs:	4
Résultat:	4
<b>Aperçu du Site web amélioré</b>	<b>5</b>
Accueil (index.html)	5
Page evenements (evenements.php)	5
Nouvelle page Statique évenements (evenements.html)	6
Nouvelle page de Contact (contact.html)	6
Nouvelle page Observations (observations.html)	7
<b>Conclusion</b>	<b>7</b>

## Introduction

Après avoir installé apache2, php7.4, postgreSQL sur notre serveur, nous avons mis en place le site qui nous a été confié par l'association des OVNI d'occitanie. En plus d'avoir fait fonctionner leur site sur notre serveur en se souciant des compatibilités, nous avons modifié le site internet pour l'améliorer et le rendre par la même occasion plus attractif. Nous allons donc vous présenter les modifications effectuées.

## Résumé des modifications effectuées sur votre site internet :

### Structure et organisation:

- Création d'une structure de base avec un en-tête, une barre de navigation, un contenu principal et un pied de page.
- Mise en place d'une page HTML et de plusieurs pages CSS pour une meilleure organisation et évolutivité.

### Design et couleurs:

- Application d'une réinitialisation des styles par défaut pour une base propre.
- Définition d'une couleur de fond sombre pour le site.
- Choix de couleurs pour l'en-tête, la navigation, le contenu et le pied de page.
- Ajout d'une image d'arrière-plan plein écran.

### Éléments du site:

- Stylisation de l'en-tête avec un fond et du texte centrés.
- Centrage des liens dans la barre de navigation et ajout d'une couleur de survol.
- Ajout d'une marge et d'un fond blanc au contenu principal pour le mettre en valeur.
- Insertion de liens vers d'autres pages du site.
- Personnalisation du pied de page avec une couleur et du texte centrés.

### **Responsive design:**

- Application de la propriété "display: flex" pour garantir l'adaptabilité du site à tous les appareils (ordinateurs, tablettes, téléphones).

### **Objectifs:**

- Améliorer l'esthétique et la clarté du site.
- Faciliter la navigation et la lecture du contenu.
- Assurer une expérience utilisateur optimale sur tous les supports.

### **Résultat:**

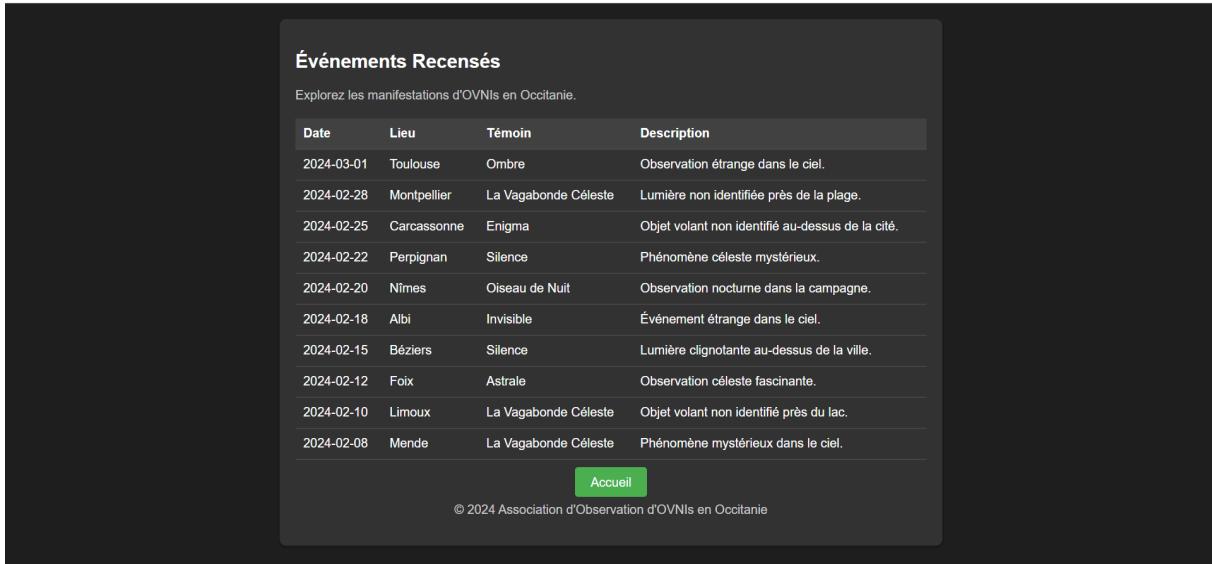
Un site internet moderne, responsive et facile à utiliser.

## Aperçu du Site web amélioré

### Accueil (index.html)



### Page evenements (evenements.php)



**Événements Recensés**

Explorez les manifestations d'OVNIls en Occitanie.

Date	Lieu	Témoin	Description
2024-03-01	Toulouse	Ombre	Observation étrange dans le ciel.
2024-02-28	Montpellier	La Vagabonde Céleste	Lumière non identifiée près de la plage.
2024-02-25	Carcassonne	Enigma	Objet volant non identifié au-dessus de la cité.
2024-02-22	Perpignan	Silence	Phénomène céleste mystérieux.
2024-02-20	Nîmes	Oiseau de Nuit	Observation nocturne dans la campagne.
2024-02-18	Albi	Invisible	Événement étrange dans le ciel.
2024-02-15	Béziers	Silence	Lumière clignotante au-dessus de la ville.
2024-02-12	Foix	Astrale	Observation céleste fascinante.
2024-02-10	Limoux	La Vagabonde Céleste	Objet volant non identifié près du lac.
2024-02-08	Mende	La Vagabonde Céleste	Phénomène mystérieux dans le ciel.

[Accueil](#)

© 2024 Association d'Observation d'OVNIls en Occitanie

## Nouvelle page Statique événements (evenements.html)



## Nouvelle page de Contact (contact.html)



## Nouvelle page Observations (observations.html)



## Conclusion

En résumé, ces modifications ont été apportées pour créer une mise en page cohérente, avec un fond d'écran plutôt neutre mais permettant de rendre les pages design, des en-têtes stylisés, une barre de navigation centrée, des sections de contenu dans des cadres blancs et des pieds de page stylisés sur toutes les pages du site.

# Inst. Serv. Rés.

**Serv-182**

---

**SAE 2.03**

---



**Nom : BOULOUIHA Yassir, RABAN Théo**

*Informatique 1<sup>ière</sup> Année*

# Table des matières :

<b>Introduction</b>	<b>2</b>
<b>Mission 1</b>	<b>2</b>
<b>Mission 2</b>	<b>6</b>
<b>Mission 3</b>	<b>8</b>
<b>Conclusion :</b>	<b>10</b>

## Introduction

Les missions de cette semaine ont pour objectifs de tenir compte des enjeux de la cybersécurité mais également de comprendre le fonctionnement de la génération des clefs SSH et de leur utilisation.

## Mission 1

### Nom et version de la distribution :

Obtenu grâce à la commande: cat /etc/os-release  
 Debian GNU/Linux 11 (bullseye)

### Site de référence Debian:

Voici le site de référence de Debian :

<https://www.debian.org>

Voici le site de référence support de Debian :

<https://www.debian.org/support>

### Date fin de vie de la distribution Debian 11 bullseye:

<https://wiki.debian.org/LTS>

Version	support architecture	schedule
<b>Previous LTS Releases</b>		
Debian 6 "Squeeze"	i386 and amd64	2nd June 2014 to 29th of February 2016
Debian 7 "Wheezy"	i386, amd64, armel and armhf	26th April 2016 to 31st May 2018
Debian 8 "Jessie"	i386, amd64, armel and armhf	17th June 2018 to June 30, 2020
Debian 9 "Stretch"	i386, amd64, armel, armhf and arm64	July 6, 2020 to June 30, 2022
<b>Current LTS Release(s)</b>		
Debian 10 "Buster"	i386, amd64, armhf and arm64	August 1st, 2022 to June 30th, 2024
<b>Future LTS Release(s)</b>		
Debian 11 "Bullseye"	i386, amd64, armhf and arm64	August 15th, 2024 to June 30th, 2026
Debian 12 "Bookworm"	i386, amd64, armhf and arm64	June 11th, 2026 to June 30th, 2028

Legend: End of life Supported by LTS team Supported by security and release teams

La date de fin de vie de la distribution est le 30 juin 2026.

La commande utilisé est **apt list --upgradable | tail -n +2 | wc -l** qui permet de compter le nombre de lignes correspondant chacune à un paquets qui peut être mis à jour, le tout est compté par wc-l (compte le nombre de lignes), ce qui permet d'obtenir le nombre de paquets pouvant être mis à jour. La partie de la commande comportant tail -n +2 permet de sauter la première ligne qui n'est pas un paquet.

```
theo@SAE-2324-DEBIAN-04:~$ apt list --upgradable
En train de lister... Fait
base-files/oldstable 11.1+deb11u9 amd64 [pouvant être mis à jour depuis : 11.1+deb11u8]
bind9-dnsutils/oldstable-security 1:9.16.48-1 amd64 [pouvant être mis à jour depuis : 1:9.16.44-1~deb11u1]
bind9-host/oldstable-security 1:9.16.48-1 amd64 [pouvant être mis à jour depuis : 1:9.16.44-1~deb11u1]
bind9-libs/oldstable-security 1:9.16.48-1 amd64 [pouvant être mis à jour depuis : 1:9.16.44-1~deb11u1]
distro-info-data/oldstable 0.51+deb11u5 all [pouvant être mis à jour depuis : 0.51+deb11u4]
libc-bin/oldstable 2.31-13+deb11u8 amd64 [pouvant être mis à jour depuis : 2.31-13+deb11u7]
libc-l10n/oldstable 2.31-13+deb11u8 all [pouvant être mis à jour depuis : 2.31-13+deb11u7]
libc6/oldstable 2.31-13+deb11u8 amd64 [pouvant être mis à jour depuis : 2.31-13+deb11u7]
libglib2.0-0/oldstable 2.66.8-1+deb11u1 amd64 [pouvant être mis à jour depuis : 2.66.8-1]
libglib2.0-data/oldstable 2.66.8-1+deb11u1 all [pouvant être mis à jour depuis : 2.66.8-1]
libgnutls30/oldstable 3.7.1-5+deb11u4 amd64 [pouvant être mis à jour depuis : 3.7.1-5+deb11u3]
libnftables/oldstable 0.9.8-3.1+deb11u2 amd64 [pouvant être mis à jour depuis : 0.9.8-3.1+deb11u1]
libperl5.32/oldstable 5.32.1-4+deb11u3 amd64 [pouvant être mis à jour depuis : 5.32.1-4+deb11u2]
libpq5/oldstable-security 13.14-0+deb11u1 amd64 [pouvant être mis à jour depuis : 13.13-0+deb11u1]
linux-image-amd64/oldstable 5.10.209-2 amd64 [pouvant être mis à jour depuis : 5.10.158-2]
locales/oldstable 2.31-13+deb11u8 all [pouvant être mis à jour depuis : 2.31-13+deb11u7]
nftables/oldstable 0.9.8-3.1+deb11u2 amd64 [pouvant être mis à jour depuis : 0.9.8-3.1+deb11u1]
perl-base/oldstable 5.32.1-4+deb11u3 amd64 [pouvant être mis à jour depuis : 5.32.1-4+deb11u2]
perl-modules-5.32/oldstable 5.32.1-4+deb11u3 all [pouvant être mis à jour depuis : 5.32.1-4+deb11u2]
perl/oldstable 5.32.1-4+deb11u3 amd64 [pouvant être mis à jour depuis : 5.32.1-4+deb11u2]
postgresql-13/oldstable-security 13.14-0+deb11u1 amd64 [pouvant être mis à jour depuis : 13.13-0+deb11u1]
postgresql-client-13/oldstable-security 13.14-0+deb11u1 amd64 [pouvant être mis à jour depuis : 13.13-0+deb11u1]
tar/oldstable 1.34+dfsg-1+deb11u1 amd64 [pouvant être mis à jour depuis : 1.34+dfsg-1]
tzdata/oldstable 2024a-0+deb11u1 all [pouvant être mis à jour depuis : 2021a-1+deb11u11]
theo@SAE-2324-DEBIAN-04:~$ apt list --upgradable | tail -n +2 | wc -l

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.
```

24

Pour afficher le nombre de paquets installés nous avons utilisé la commande suivante **dpkg -l**, pour pouvoir les dénombrés plus facilement nous avons utilisé la commande suivante : **dpkg -l | grep '^ii' | wc -l** cette commande permet de compter le nombre de ligne retourné par dpkg -l (chaque ligne correspond à un paquet) le **grep '^ii'** permet de prendre les lignes commençant uniquement par "ii", ce sont les lignes correspondant aux paquets installés. En effet, **dpkg -l** retourne quelques lignes au début qui ne sont pas des paquets.

```
theo@SAE-2324-DEBIAN-04:~$ dpkg -l | grep '^ii' | wc -l
383
```

Grâce à la commande **top** nous pouvons connaître en temps réel les processus qui tournent sur notre serveur mais également permet d'en le nombre.

```
theo@SAE-2324-DEBIAN-04:~$ top
top - 10:28:14 up 29 days, 18:32, 2 users, load average: 0,00, 0,00, 0,00
Tâches: 147 total, 1 en cours, 146 en veille, 0 arrêté, 0 zombie
%Cpu(s): 0,3 ut, 0,3 sy, 0,0 ni, 99,3 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 1983,1 total, 995,1 libr, 140,5 util, 847,6 tmap/cache
MiB Éch : 975,0 total, 975,0 libr, 0,0 util. 1644,3 dispo Mem

PID UTIL. PR NI VIRT RES SHR S %CPU %MEM TEMPS+ COM.
148138 root 20 0 0 0 0 I 0,3 0,0 2:21.49 kworker/0:0-events
159492 theo 20 0 10268 3944 3360 R 0,3 0,2 0:00.72 top
  1 root 20 0 166944 10728 7892 S 0,0 0,5 1:18.19 systemd
  2 root 20 0 0 0 0 S 0,0 0,0 0:00.61 kthreadd
  3 root 0 -20 0 0 0 I 0,0 0,0 0:00.00 rcu_gp
  4 root 0 -20 0 0 0 I 0,0 0,0 0:00.00 rcu_par_gp
  6 root 0 -20 0 0 0 I 0,0 0,0 0:00.00 kworker/0:0H-events_highpri
  8 root 0 -20 0 0 0 I 0,0 0,0 0:00.00 mm_percpu_wq
  9 root 20 0 0 0 0 S 0,0 0,0 0:00.00 rcu_tasks_rude_
10 root 20 0 0 0 0 S 0,0 0,0 0:00.00 rcu_tasks_trace
11 root 20 0 0 0 0 S 0,0 0,0 0:04.89 ksoftirqd/0
12 root 20 0 0 0 0 I 0,0 0,0 2:22.97 rcu_sched
13 root rt 0 0 0 0 S 0,0 0,0 0:07.95 migration/0
15 root 20 0 0 0 0 S 0,0 0,0 0:00.00 cpuhp/0
17 root 20 0 0 0 0 S 0,0 0,0 0:00.00 kdevtmpfs
18 root 0 -20 0 0 0 I 0,0 0,0 0:00.00 netns
19 root 20 0 0 0 0 S 0,0 0,0 0:00.00 kauditd
20 root 20 0 0 0 0 S 0,0 0,0 0:02.70 khungtaskd
21 root 20 0 0 0 0 S 0,0 0,0 0:00.00 oom_reaper
22 root 0 -20 0 0 0 I 0,0 0,0 0:00.00 writeback
23 root 20 0 0 0 0 S 0,0 0,0 0:56.03 kcompactd0
24 root 25 5 0 0 0 S 0,0 0,0 0:00.00 ksmd
25 root 39 19 0 0 0 S 0,0 0,0 0:28.43 khugepaged
43 root 0 -20 0 0 0 I 0,0 0,0 0:00.00 kintegrityd
44 root 0 -20 0 0 0 I 0,0 0,0 0:00.00 kblockd
45 root 0 -20 0 0 0 I 0,0 0,0 0:00.00 blkcg_punt_bio
46 root 0 -20 0 0 0 I 0,0 0,0 0:00.00 edac-poller
47 root 0 -20 0 0 0 I 0,0 0,0 0:00.00 devfreq_wq
48 root 0 -20 0 0 0 I 0,0 0,0 0:06.47 kworker/0:1H-kblockd
49 root 20 0 0 0 0 S 0,0 0,0 0:00.00 kswapd0
50 root 0 -20 0 0 0 I 0,0 0,0 0:00.00 kthrotld
51 root -51 0 0 0 0 S 0,0 0,0 0:00.00 irq/24-pciehp
52 root -51 0 0 0 0 S 0,0 0,0 0:00.00 irq/25-pciehp
53 root -51 0 0 0 0 S 0,0 0,0 0:00.00 irq/26-pciehp
54 root -51 0 0 0 0 S 0,0 0,0 0:00.00 irq/27-pciehp
55 root -51 0 0 0 0 S 0,0 0,0 0:00.00 irq/28-pciehp
56 root -51 0 0 0 0 S 0,0 0,0 0:00.00 irq/29-pciehp
```

L'utilisateur svcmgr a été créé grâce à la commande

**sudo adduser svcmgr** sudo permet de passer en super utilisateur pour pouvoir effectuer la commande, adduser permet simplement d'ajouter l'utilisateur spécifié "svcmgr".

Nous avons utilisé la commande **sudo visudo** pour effectuer les modifications sur les droits des utilisateurs.

Voici une capture d'écran de des deux lignes entrées pour permettre à SVCMGR d'effectuer les actions requises. Seules les deux dernières lignes de la capture d'écran ont été ajoutées au fichier des droits sudoers.

```
root    ALL=(ALL:ALL) ALL
etudiant    ALL=(ALL:ALL) ALL
svcmgr  ALL=(ALL) NOPASSWD: /bin/systemctl start apache2, /bin/systemctl stop apache2, /bin/systemctl restart apache2
svcmgr  ALL=(ALL) NOPASSWD: /bin/systemctl start postgresql, /bin/systemctl stop postgresql, /bin/systemctl restart postgresql
```

## Mission 2

Pour créer le groupe smrtacc et le compte utilisateur smrtusr :

```
yassir@SAE-2324-DEBIAN-04:~$ sudo groupadd smrtacc
sudo adduser smrtusr --ingroup smrtacc
```

Verification de l'association du groupe smrtacc sur smrtusr :

```
yassir@SAE-2324-DEBIAN-04:~$ groups smrtusr
smrtusr : smrtacc
```

Création de la clé ssh :

```
yassir@SAE-2324-DEBIAN-04:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/yassir/.ssh/id_rsa):
Created directory '/home/yassir/.ssh'.
```

Copie de ma clé publique sur le répertoire des clés autorisées de smrtusr :

```
yassir@SAE-2324-DEBIAN-04:~$ ssh-copy-id srmtusr@127.0.0.1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/yassir/.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
srmtusr@127.0.0.1's password:
```

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'srmtusr@127.0.0.1'"  
 and check to make sure that only the key(s) you wanted were added.

Connexion sur srmtusr à l'aide d'uniquement à clé privée :

```
yassir@SAE-2324-DEBIAN-04:~$ ssh srmtusr@127.0.0.1
Linux SAE-2324-DEBIAN-04 5.10.0-20-amd64 #1 SMP Debian 5.10.158-2 (2022-12-13) 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.

Bienvenue dans votre Serveur Debian !

Sauvegarde terminée : /sauvegardes/srmtusr/backup\_27-03-2024\_10-34-22  
 srmtusr@SAE-2324-DEBIAN-04:~/IUT\$

J'ai ouvert le fichier sshd\_config avec la commande :

**sudo nano /etc/ssh/sshd\_config**

Ensuite j'ai ajouté les lignes suivantes dans le fichier de configuration :

```
Match Group srmtacc
    PasswordAuthentication no
```

L'ajout de ces instructions permet d'empêcher l'authentification par  
 mot de passe pour les connexions ssh au groupe srmtacc.

Test :

```
C:\Windows\System32>ssh srmtusr@192.168.224.182
The authenticity of host '192.168.224.182 (192.168.224.182)' can't be established.
ECDSA key fingerprint is SHA256:eqXQwcCLMDuFWyf2fintVWfcS0411gwBUXr20yaMBms.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.224.182' (ECDSA) to the list of known hosts.
srmtusr@192.168.224.182: Permission denied (publickey).
C:\Windows\System32>
```

## Mission 3

Pour afficher uniquement les informations du serveur web via curl:

```
yassir@SAE-2324-DEBIAN-04:~$ curl -I localhost
HTTP/1.1 200 OK
Date: Thu, 28 Mar 2024 12:05:17 GMT
Server: Apache/2.4.56 (Debian)
Last-Modified: Thu, 07 Mar 2024 13:25:09 GMT
ETag: "60b-61312012542be"
Accept-Ranges: bytes
Content-Length: 1547
Vary: Accept-Encoding
Content-Type: text/html
```

Ici on peut donc voir la version du serveur web apache mais pas celle du PHP qui est censée être précédée par "X-Powered By" d'après nos recherches, une des hypothèses serait que la version du PHP de base n'est pas affichée pour des soucis de sécurité comme pour l'affichage des erreurs par exemple.

Voici le résultat après avoir utilisé la commande curl -I localhost:

```
theo@SAE-2324-DEBIAN-04:~$ curl -I localhost
HTTP/1.1 200 OK
Date: Thu, 28 Mar 2024 12:15:38 GMT
Server: Apache/2.4.56 (Debian)
Last-Modified: Thu, 07 Mar 2024 13:25:09 GMT
ETag: "60b-61312012542be"
Accept-Ranges: bytes
Content-Length: 1547
Vary: Accept-Encoding
Content-Type: text/html
```

Voici les deux lignes permettant de masquer les informations relatives à la version du serveur apache :

- ServerTokens défini sur Prod, ce qui masquera la version du serveur.
- ServerSignature pour la définir sur Off, qui permet de masquer la signature du serveur dans les réponses HTTP.

```
# Include the virtual host configurations:  
IncludeOptional sites-enabled/*.conf  
  
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet  
ServerTokens Prod  
ServerSignature Off
```

Voici l'affichage des informations après avoir ajouté les deux lignes dans le fichier apache2.conf :

```
theo@SAE-2324-DEBIAN-04:~$ curl -I localhost  
HTTP/1.1 200 OK  
Date: Thu, 28 Mar 2024 12:41:19 GMT  
Server: Apache  
Last-Modified: Thu, 07 Mar 2024 13:25:09 GMT  
ETag: "60b-61312012542be"  
Accept-Ranges: bytes  
Content-Length: 1547  
Vary: Accept-Encoding  
Content-Type: text/html
```

## Conclusion :

Ces missions nous ont permis d'approfondir nos connaissances sur les connexions SSH, mais également de nous faire acquérir des compétences sur le plan de la cybersécurité notamment en cachant les informations relatives à la version de notre serveur apache.

# Inst. Serv. Rés.

**Serv-182**

---

**SAE 2.03**

---



**Nom : BOULOUIHA Yassir, RABAN Théo**

*Informatique 1<sup>ière</sup> Année*

# Table des matières :

<b>Introduction</b>	<b>2</b>
<b>Détail Mission 1</b>	<b>2</b>
<b>Détail Mission 2</b>	<b>7</b>
<b>Partie 1</b>	<b>7</b>
Fonctionnement de Ansible	7
Inventaire :	8
Playbooks :	8
Résumé	12
Idempotence	12
Définition d'idempotence :	12
Application à Ansible :	12
Résumé	12
<b>Lien Gitlab :</b>	<b>13</b>
<b>Conclusion :</b>	<b>13</b>

En rouge les parties mises à jour

## Introduction

Les missions de cette semaine ont pour objectifs de nous faire faire apprendre à utiliser Ansible, un logiciel d'automatisation que l'on doit configurer au préalable.

### Détail Mission 1

On commence par installer les dépendances:

```
theo@SAE-2324-DEBIAN-04:~$ sudo apt-get install software-properties-common
Lecture des listes de paquets ... Fait
Construction de l'arbre des dépendances ... Fait
Lecture des informations d'état ... Fait
Les paquets suivants ont été installés automatiquement et ne sont plus nécessaires :
  ethtool javascript-common libdrm-common libdrm2 libexpat1-dev libjs-jquery
  libjs-sphinxdoc libjs-underscore libmspack0 libpython3-dev libpython3.9
  libpython3.9-dev libxmlsec1 libxmlsec1-openssl net-tools python-pip-whl
  python3.9-dev whois zerofree zlib1g-dev
Veuillez utiliser « sudo apt autoremove » pour les supprimer.
Les paquets supplémentaires suivants seront installés :
  gir1.2-glib-2.0 gir1.2-packagekitglib-1.0 libappstream4 libdw1
  libgirepository-1.0-1 libglib2.0-0 libglib2.0-bin libgstreamer1.0-0
  libpackagekit-glib2-18 libpolkit-agent-1-0 libpolkit-gobject-1-0 libstemmer0d
  libunwind8 libyaml-0-2 lsb-release packagekit packagekit-tools policykit-1 python3
  python3-apt python3-dbus python3-distro-info python3-gi python3-pycurl
  python3-software-properties unattended-upgrades
Paquets suggérés :
  gstreamer1.0-tools appstream python3-doc python3-tk python3-venv python3-apt-dbg
  python-apt-doc python-dbus-doc python3-dbus-dbg libcurl4-gnutls-dev
  python-pycurl-doc python3-pycurl-dbg bsd-mailx default-mta | mail-transport-agent
  needrestart powermgmt-base
Les NOUVEAUX paquets suivants seront installés :
  gir1.2-glib-2.0 gir1.2-packagekitglib-1.0 libappstream4 libdw1
  libgirepository-1.0-1 libglib2.0-bin libgstreamer1.0-0 libpackagekit-glib2-18
  libpolkit-agent-1-0 libpolkit-gobject-1-0 libstemmer0d libunwind8 libyaml-0-2
  lsb-release packagekit packagekit-tools policykit-1 python3 python3-apt
  python3-dbus python3-distro-info python3-gi python3-pycurl
  python3-software-properties software-properties-common unattended-upgrades
```

On installe ensuite ansible :

```
theo@SAE-2324-DEBIAN-04:~$ sudo apt-get install ansible
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances ... Fait
Lecture des informations d'état ... Fait
Les paquets suivants ont été installés automatiquement et ne sont plus nécessaires
  ethtool javascript-common libdrm-common libdrm2 libexpat1-dev libjs-jquery
  libjs-sphinxdoc libjs-underscore libmspack0 libpython3-dev libpython3.9
  libpython3.9-dev libxmlsec1 libxmlsec1-openssl net-tools python-pip-whl
  python3.9-dev whois zerofree zlib1g-dev
Veuillez utiliser « sudo apt autoremove » pour les supprimer.
Les paquets supplémentaires suivants seront installés :
  ieee-data python3-argcomplete python3-certifi python3-cffi-backend python3-chardet
  python3-cryptography python3-distutils python3-dnspython python3-httplib2
  python3-idna python3-jinja2 python3-jmespath python3-kerberos python3-lib2to3
  python3-libcloud python3-lockfile python3-markupsafe python3-netaddr
  python3-ntlm-auth python3-packaging python3-pkg-resources python3-pycryptodome
  python3-pyparsing python3-requests python3-requests-kerberos python3-requests-ntlm
  python3-requests-toolbelt python3-selinux python3-simplejson python3-six
  python3-urllib3 python3-winrm python3-xmldict python3-yaml
Paquets suggérés :
  python-cryptography-doc python3-cryptography-vectors python3-sniffio python3-tri
  python-jinja2-doc python-lockfile-doc ipython3 python-netaddr-docs
  python3-setuptools python-pyparsing-doc python3-openssl python3-socks
  python-requests-doc
Les NOUVEAUX paquets suivants seront installés :
  ansible ieee-data python3-argcomplete python3-certifi python3-cffi-backend
  python3-chardet python3-cryptography python3-distutils python3-dnspython
  python3-httplib2 python3-idna python3-jinja2 python3-jmespath python3-kerberos
  python3-lib2to3 python3-libcloud python3-lockfile python3-markupsafe
  python3-netaddr python3-ntlm-auth python3-packaging python3-pkg-resources
  python3-pycryptodome python3-pyparsing python3-requests python3-requests-kerberos
  python3-requests-ntlm python3-requests-toolbelt python3-selinux python3-simplejson
  python3-six python3-urllib3 python3-winrm python3-xmldict python3-yaml
0 mis à jour, 35 nouvellement installés, 0 à enlever et 22 non mis à jour.
Il est nécessaire de prendre 33,5 Mo dans les archives.
Après cette opération, 283 Mo d'espace disque supplémentaires seront utilisés.
```

Ensuite nous devons installer pip sur l'environnement python du serveur:

```
theo@SAE-2324-DEBIAN-04:~$ sudo apt install python3-pip
Lecture des listes de paquets ... Fait
Construction de l'arbre des dépendances ... Fait
Lecture des informations d'état... Fait
Les paquets suivants ont été installés automatiquement et ne sont plus nécessaires :
  ethtool libdrm-common libdrm2 libmspack0 libxmlsec1 libxmlsec1-openssl net-tools
    whois zerofree
Veuillez utiliser « sudo apt autoremove » pour les supprimer.
Les paquets supplémentaires suivants seront installés :
  python3-dev python3-setuptools python3-wheel
Paquets suggérés :
  python-setuptools-doc
Les NOUVEAUX paquets suivants seront installés :
  python3-dev python3-pip python3-setuptools python3-wheel
0 mis à jour, 4 nouvellement installés, 0 à enlever et 22 non mis à jour.
Il est nécessaire de prendre 753 ko dans les archives.
```

```
theo@SAE-2324-DEBIAN-04:~$ python3 -m pip install --user ansible-core
Collecting ansible-core
  Downloading ansible_core-2.15.11-py3-none-any.whl (2.3 MB)
    |████████| 2.3 MB 1.8 MB/s
Requirement already satisfied: packaging in /usr/lib/python3/dist-packages (from ansible-core) (20.9)
Requirement already satisfied: cryptography in /usr/lib/python3/dist-packages (from ansible-core) (3.3.2)
Collecting importlib-resources<5.1, ≥5.0
  Using cached importlib_resources-5.0.7-py3-none-any.whl (24 kB)
Collecting jinja2≥3.0.0
  Using cached Jinja2-3.1.3-py3-none-any.whl (133 kB)
Requirement already satisfied: PyYAML≥5.1 in /usr/lib/python3/dist-packages (from ansible-core) (5.3.1)
Collecting resolvelib<1.1.0, ≥0.5.3
  Using cached resolvelib-1.0.1-py2.py3-none-any.whl (17 kB)
Collecting MarkupSafe≥2.0
  Using cached MarkupSafe-2.1.5-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (25 kB)
Installing collected packages: MarkupSafe, resolvelib, jinja2, importlib-resources, ansible-core
```

On installe ensuite pip.py

```
theo@SAE-2324-DEBIAN-04:~$ curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
% Total    % Received % Xferd  Average Speed   Time     Time      Current
          Dload  Upload Total   Spent    Left  Speed
100 2574k  100 2574k    0     0  5931k      0 --:--:-- --:--:-- --:--:-- 5917k
```

```
theo@SAE-2324-DEBIAN-04:~$ python3 get-pip.py --user
Collecting pip
  Downloading pip-24.0-py3-none-any.whl.metadata (3.6 kB)
  Downloading pip-24.0-py3-none-any.whl (2.1 MB)
    2.1/2.1 MB 12.1 MB/s eta 0:00:00
Installing collected packages: pip
  WARNING: The scripts pip, pip3 and pip3.9 are installed in '/home/theo/.local/bin' which is not on PATH.
H.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed pip-24.0
```

On peut donc désormais installer venv:

```
theo@SAE-2324-DEBIAN-04:~$ sudo apt install python3 python3-pip python3-venv
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
python3-pip est déjà la version la plus récente (20.3.4-4+deb11u1).
python3 est déjà la version la plus récente (3.9.2-3).
Les paquets suivants ont été installés automatiquement et ne sont plus nécessaires :
  ethtool libdrm-common libdrm2 libmspack0 libxmlsec1 libxmlsec1-openssl net-tools whois zerofree
Veuillez utiliser « sudo apt autoremove » pour les supprimer.
Les paquets supplémentaires suivants seront installés :
  python3.9-venv
Les NOUVEAUX paquets suivants seront installés :
  python3-venv python3.9-venv
0 mis à jour, 2 nouvellement installés, 0 à enlever et 22 non mis à jour.
Il est nécessaire de prendre 6 584 o dans les archives.
Après cette opération, 33,8 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [0/n] 0
Réception de :1 http://deb.debian.org/debian bullseye/main amd64 python3.9-venv amd64 3.9.2-1 [5 396 B]
Réception de :2 http://deb.debian.org/debian bullseye/main amd64 python3-venv amd64 3.9.2-3 [1 188 B]
6 584 o réceptionnés en 0s (39,5 ko/s)
Sélection du paquet python3.9-venv précédemment désélectionné.
(Lecture de la base de données... 78757 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de ... /python3.9-venv_3.9.2-1_amd64.deb ...
Dépaquetage de python3.9-venv (3.9.2-1) ...
Sélection du paquet python3-venv précédemment désélectionné.
Préparation du dépaquetage de ... /python3-venv_3.9.2-3_amd64.deb ...
Dépaquetage de python3-venv (3.9.2-3) ...
Paramétrage de python3.9-venv (3.9.2-1) ...
Paramétrage de python3-venv (3.9.2-3) ...
```

On installe maintenant sshpass:

```
theo@SAE-2324-DEBIAN-04:~$ sudo apt -y install sshpass
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
sshpass est déjà la version la plus récente (1.09-1+b1).
Les paquets suivants ont été installés automatiquement et ne sont plus nécessaires :
  ethtool libdrm-common libdrm2 libmspack0 libxmlsec1 libxmlsec1-openssl net-tools whois zerofree
Veuillez utiliser « sudo apt autoremove » pour les supprimer.
0 mis à jour, 0 nouvellement installés, 0 à enlever et 22 non mis à jour.
```

Maintenant il faut créer et activer l'environnement virtuel grâce à venv:

```
theo@SAE-2324-DEBIAN-04:~$ sudo python3 -m venv envsae
theo@SAE-2324-DEBIAN-04:~$ ls
Ansible_conf  commands.txt  envsae  get-pip.py  project  WELCOME.txt
theo@SAE-2324-DEBIAN-04:~$
```

```
theo@SAE-2324-DEBIAN-04:~$ source envsae/bin/activate
(envsae) theo@SAE-2324-DEBIAN-04:~$
```

L'environnement virtuel a bien été activé car (envsae) apparaît devant mon nom d'utilisateur dans le terminal.

Il faut désormais installer ansible sur l'environnement virtuel :

```
(envsae) theo@SAE-2324-DEBIAN-04:~$ sudo pip install ansible
Collecting ansible
  Downloading ansible-8.7.0-py3-none-any.whl (48.4 MB)
    |████████| 48.4 MB 43.5 MB/s
Collecting ansible-core>=2.15.7
  Downloading ansible_core-2.15.11-py3-none-any.whl (2.3 MB)
    |████████| 2.3 MB 32.7 MB/s
Collecting jinja2>=3.0.0
  Downloading Jinja2-3.1.3-py3-none-any.whl (133 kB)
    |████████| 133 kB 46.6 MB/s
Collecting resolvelib<1.1.0, >=0.5.3
  Downloading resolvelib-1.0.1-py2.py3-none-any.whl (17 kB)
```

```
(envsae) theo@SAE-2324-DEBIAN-04:~$ ansible --version
ansible [core 2.15.11]
  config file = None
  configured module search path = ['/home/theo/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /home/theo/.local/lib/python3.9/site-packages/ansible
  ansible collection location = /home/theo/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/local/bin/ansible
  python version = 3.9.2 (default, Feb 28 2021, 17:03:44) [GCC 10.2.1 20210110] (/usr/bin/python3)
  jinja version = 3.1.3
  libyaml = True
```

Ansible a bien été installé sur l'espace virtuel.

## Détail Mission 2

Pour créer un utilisateur sur le serveur cible et faire en sorte qu'il soit sudoer nous avons fait usage des commandes suivantes (retrouvées grâce à un history):

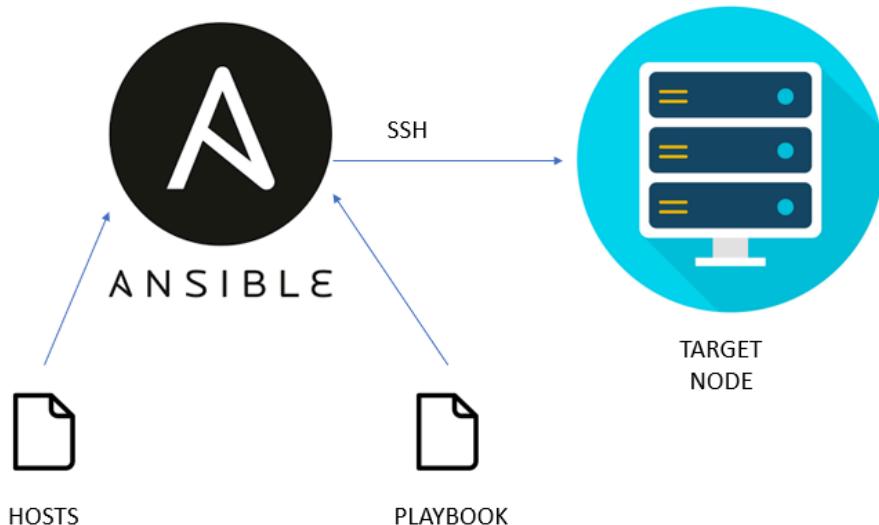
```
432 sudo adduser AnsibleHost
433 sudo adduser ansible_host
```

## Partie 1

Pour tester Ansible nous avons dû le mettre en place pour le mettre en place, nous avons créé un environnement virtuel python afin d'y installer ansible dessus (le client), puis nous avons créé un utilisateur avec les permissions sudo (ansible\_host) qui servira quant à lui de serveur à paramétrer via ansible.

Pour réussir à établir une connexion entre le client ansible (installé sur un environnement virtuel python sur l'utilisateur théo), afin que la connexion entre notre client ansible et notre serveur à configurer puisse être établie on a généré une paire de clés ssh de chaque côté et ajouté dans le authorized\_keys du serveur à configurer (serveur cible) cela a permis de se connecter en ssh à l'aide de la paire de clés sur notre client Ansible, et donc de pousser les configurations que l'on voulait sur le serveur.

### **Fonctionnement de Ansible**



Ansible est une plateforme de gestion de configuration et d'orchestration qui fonctionne selon une architecture sans agent. Cela signifie qu'aucun logiciel supplémentaire n'est requis sur les nœuds (serveurs) gérés. Ansible utilise le protocole SSH pour se connecter et exécuter des commandes sur les machines distantes.

#### **Inventaire :**

Ansible utilise un inventaire pour définir les nœuds qu'il doit gérer. Cet inventaire peut être statique (fichier texte) ou dynamique (généré à partir de sources comme des services de cloud ou des bases de données). Dans notre cas on a utilisé un inventaire statique (inventaire.ini). Ce fichier représente les hôtes sur lesquels nous pousserons nos modifications, on y renseigne l'utilisateur, l'hôte ainsi que des informations supplémentaires dans ce cas ci le ansible\_become\_method par exemple qui permet d'exécuter des commandes sudo à distance.

Voilà à quoi ressemble inventaire.ini :

```
GNU nano 5.4                                         inventory.ini
[servers]
serveur_cible ansible_host=192.168.224.182 ansible_user=ansible_host ansible_become=true ansible_become_method=sudo ansible_become_password=[REDACTED]
```

### Playbooks :

Les playbooks sont des fichiers YAML qui décrivent les tâches à exécuter par Ansible sur les nœuds gérés. Ils contiennent des directives telles que les hôtes cibles, les tâches à effectuer, les variables, etc. Les playbooks peuvent être très simples, ne faisant qu'une seule tâche, ou complexes, orchestrant des déploiements entiers. Dans notre cas, le playbooks contient un script plutôt basique qui permet d'ajouter un groupe, un utilisateur et déployer un contenu sur le serveur.

Voici-dessous l'ancien Script permettant de mettre en oeuvre les configurations requises mais pas de la bonne manière en utilisant copy et le texte contenu dans la variable template\_content :

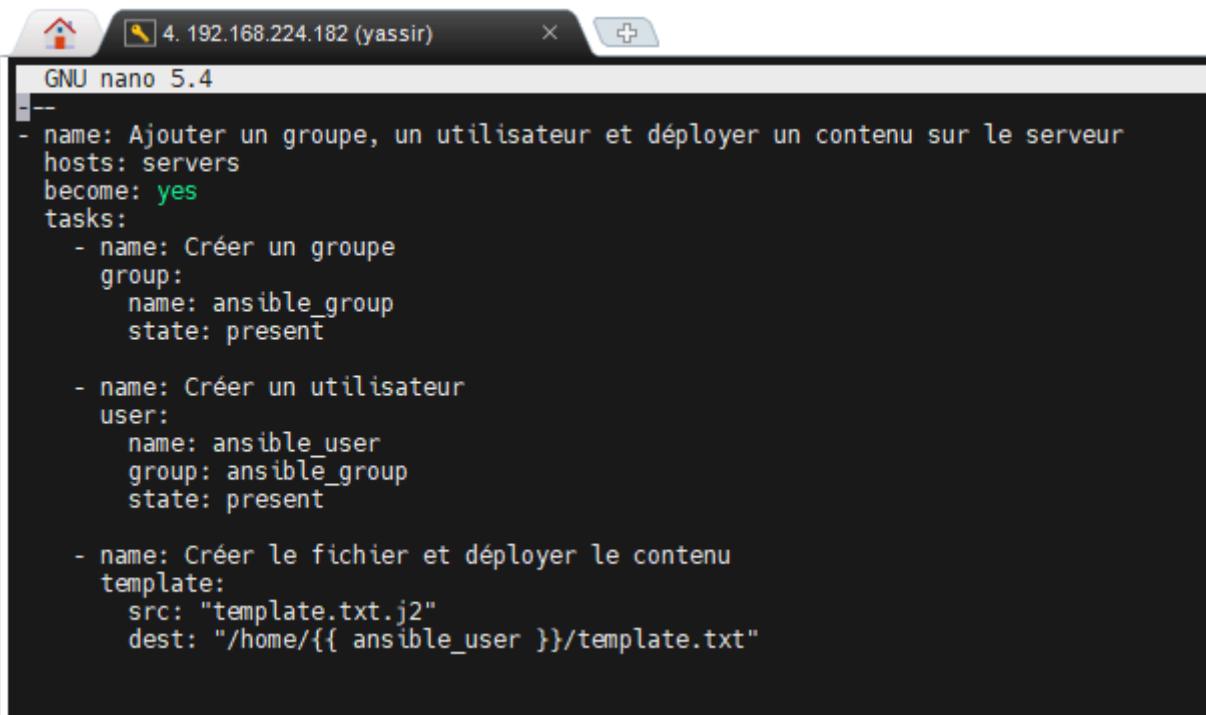
```
GNU nano 5.4
---
- name: Ajouter un groupe, un utilisateur et déployer un contenu sur le serveur
  hosts: servers
  become: yes
  vars:
    template_content: |
      Template Proposée
      Une autre ligne de la template
      Une dernière ligne pour montrer que c'est possible d'écrire un texte structuré
  tasks:
    - name: Créer un groupe
      group:
        name: ansible_group
        state: present

    - name: Créer un utilisateur
      user:
        name: ansible_user
        group: ansible_group
        state: present

    - name: Créer le fichier et déployer le contenu
      copy:
        content: "{{ template_content }}"
        dest: "/home/{{ ansible_user }}/template.txt"
```

Le nouveau script correct, en utilisant template fait à peu près pareil sauf que le contenu du template (fichier à déployer) sera stocké directement en tant que fichier de façon à ce qu'il soit accessible par notre script, ainsi nous n'avons plus besoin des variables créées en haut, et nous pouvons utiliser template avec la mention source ou l'on indiquera le chemin vers notre fichier template.

Voici le nouveau script :

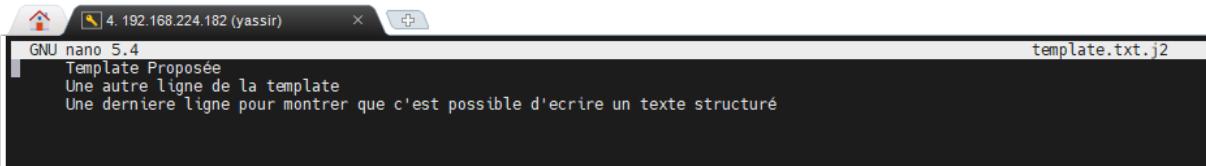


```
GNU nano 5.4
---
- name: Ajouter un groupe, un utilisateur et déployer un contenu sur le serveur
  hosts: servers
  become: yes
  tasks:
    - name: Créer un groupe
      group:
        name: ansible_group
        state: present

    - name: Créer un utilisateur
      user:
        name: ansible_user
        group: ansible_group
        state: present

    - name: Créer le fichier et déployer le contenu
      template:
        src: "template.txt.j2"
        dest: "/home/{{ ansible_user }}/template.txt"
```

Le nouveau fichier template utilisé :



```
Template Proposée
Une autre ligne de la template
Une dernière ligne pour montrer que c'est possible d'écrire un texte structuré
```

Test avec ces modifications du bon fonctionnement du script :

```
theo@SAE-2324-DEBIAN-04:~/Ansible_conf$ ansible-playbook -i inventory.ini script.yml
/ PLAY [Ajouter un groupe, un utilisateur et déployer un \
\ contenu sur le serveur]
  \ \ ^__^
    (oo)\_____
     (__)\       )\/\
      ||----w |
      ||     ||

< TASK [Gathering Facts] >
  \ \ ^__^
    (oo)\_____
     (__)\       )\/\
      ||----w |
      ||     ||

ok: [serveur_cible]
< TASK [Créer un groupe] >
  \ \ ^__^
    (oo)\_____
     (__)\       )\/\
      ||----w |
      ||     ||

ok: [serveur_cible]
< TASK [Créer un utilisateur] >
  \ \ ^__^
    (oo)\_____
     (__)\       )\/\
      ||----w |
      ||     ||

ok: [serveur_cible]
< TASK [Créer le fichier et déployer le contenu] >
  \ \ ^__^
    (oo)\_____
     (__)\       )\/\
      ||----w |
      ||     ||

changed: [serveur_cible]
< PLAY RECAP >
  \ \ ^__^
    (oo)\_____
     (__)\       )\/\
      ||----w |
      ||     ||

serveur_cible : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
theo@SAE-2324-DEBIAN-04:~/Ansible_conf$
```

## Résumé

En résumé, Ansible simplifie la gestion et l'automatisation des configurations système et des déploiements d'applications à travers une architecture sans agent, des playbooks déclaratifs et une grande flexibilité.

## ***Idempotence***

### **Définition d'idempotence :**

En informatique, une opération est dite "idempotente" si son effet reste le même même si elle est appliquée plusieurs fois. En d'autres termes, répéter une opération identique plusieurs fois ne change pas le résultat après la première application.

### **Application à Ansible :**

Dans le contexte d'Ansible, cela signifie que lorsqu'on exécute un playbook plusieurs fois sur un ensemble de nœuds, le système reste dans un état cohérent, peu importe le nombre de fois que le playbook est exécuté. Par exemple, si un playbook installe un logiciel sur un serveur, le playbook peut être exécuté plusieurs fois, mais si le logiciel est déjà installé, Ansible ne tentera pas de le réinstaller à moins que nous l'ayons explicitement spécifié.

### **Résumé**

En résumé, l'idempotence est un principe clé dans Ansible qui garantit que les opérations de gestion de configuration peuvent être répétées de manière fiable sans causer d'effets indésirables, ce qui améliore la stabilité et la prédictibilité des opérations système.

## **Lien Gitlab :**

[https://gitlab.iut-blagnac.fr/traban/semaine5\\_sae2.03](https://gitlab.iut-blagnac.fr/traban/semaine5_sae2.03)

## Conclusion :

Ces missions nous ont permis de comprendre des principes essentiels d'automatisation de certaines tâches grâce à ansible. En effet, nous avons découvert comment configurer un inventaire ansible et comment utiliser des scripts .yml.