BTS 2 - Services Informatiques aux Organisations

BTS SIO SISR – Chapitre 3 : Configuration de serveurs et d'applications avec Ansible

1. Installation d'Ansible

La commande **nano /etc/hosts** permet de modifier la résolution locale (associer 127.0.1.1 au nom Ansible).

```
 GNU nano 8.4                    /etc/hosts *
127.0.0.1       localhost
127.0.1.1       Ansible

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

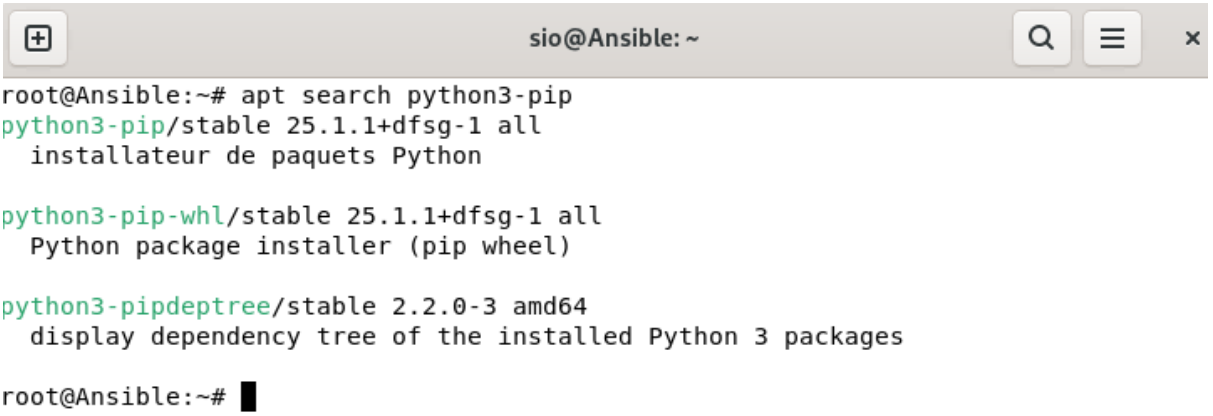
La commande **nano /etc/hostname** permet de changer le nom d'hôte de la machine en Ansible.

```
 GNU nano 8.4                   /etc/hostname *
Ansible
```

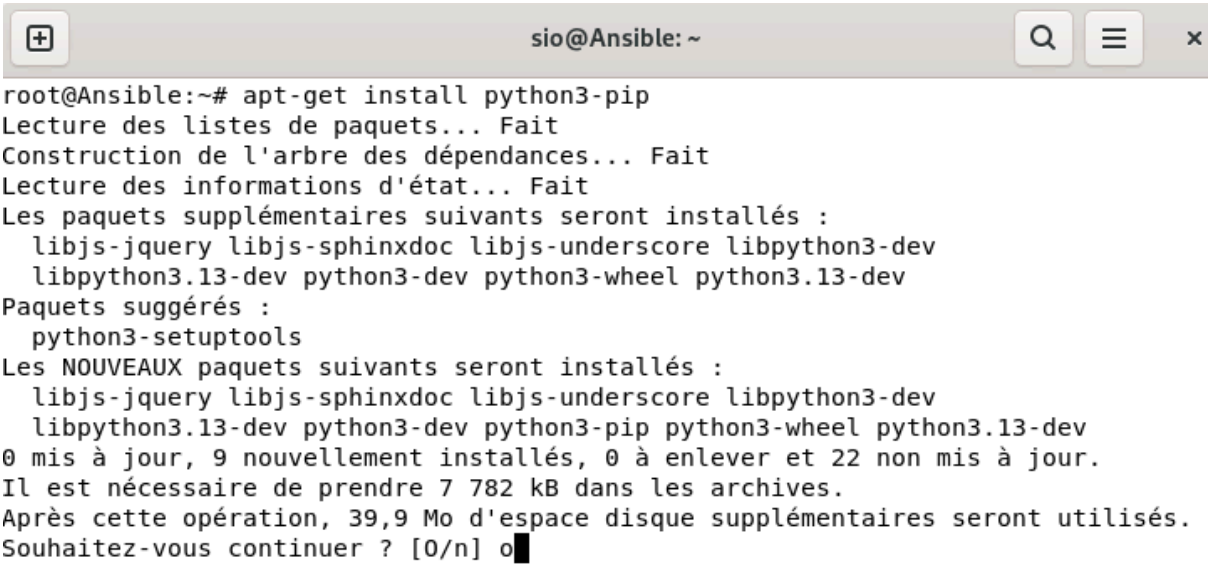
La commande **apt-get update** permet de mettre à jour la liste des paquets disponibles.

```
sio@Ansible: ~
root@Ansible:~# apt-get update
Atteint : 1 http://deb.debian.org/debian trixie InRelease
Réception de : 2 http://security.debian.org/debian-security trixie-security InRe
lease [43,4 kB]
Réception de : 3 http://deb.debian.org/debian trixie-updates InRelease [47,3 kB]
Réception de : 4 http://security.debian.org/debian-security trixie-security/main
 Sources [110 kB]
Réception de : 5 http://security.debian.org/debian-security trixie-security/main
 amd64 Packages [80,7 kB]
Réception de : 6 http://security.debian.org/debian-security trixie-security/main
 Translation-en [50,9 kB]
333 ko réceptionnés en 0s (748 ko/s)
Lecture des listes de paquets... Fait
root@Ansible:~#
```

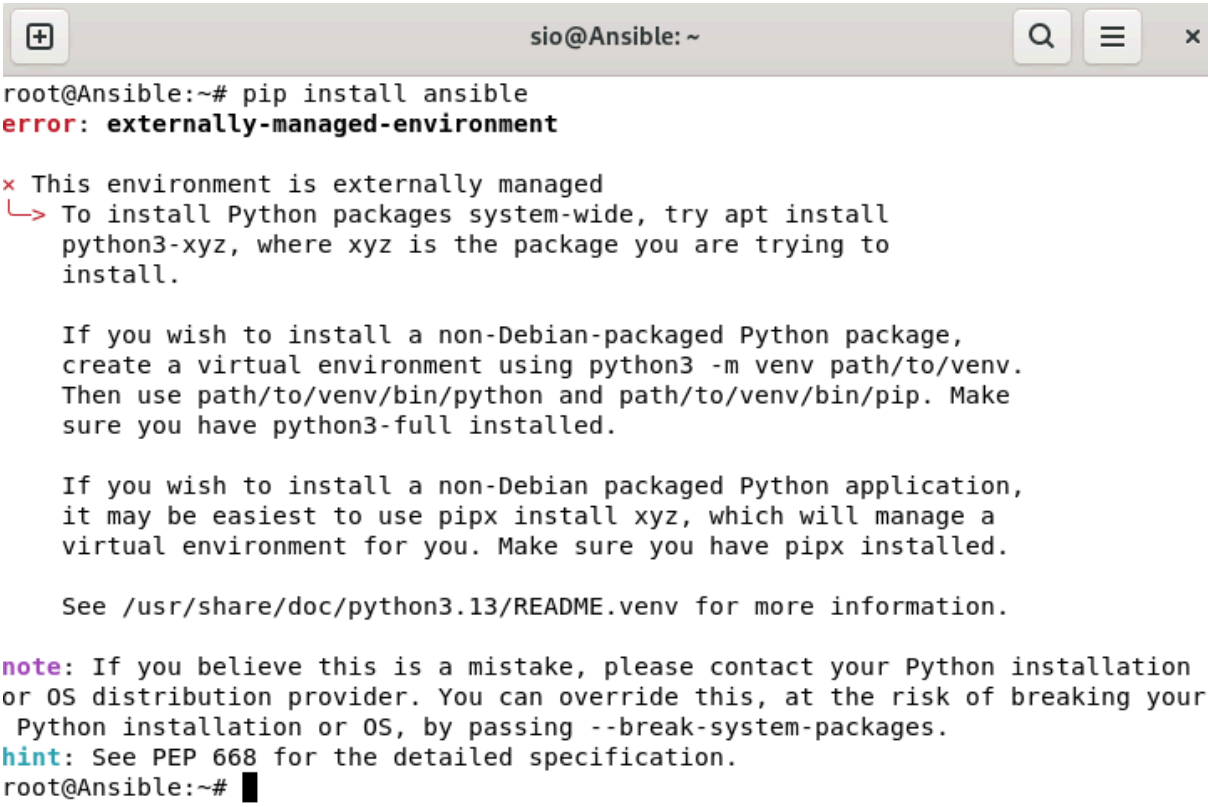
La commande **apt-get install python3-pip** permet d'installer pip pour Python 3 (utile pour installer Ansible via Python).



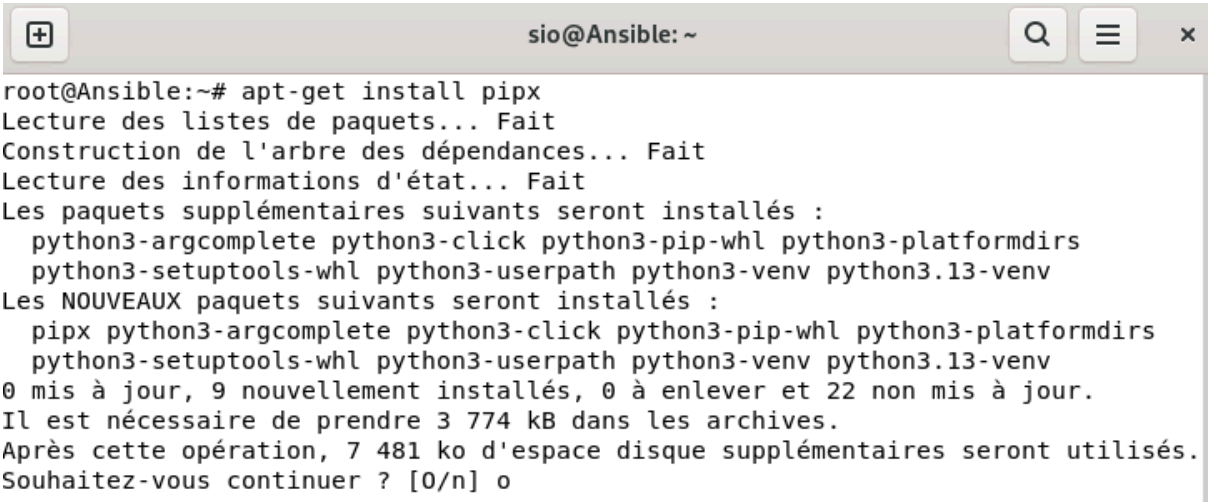
La commande **apt-get install python3-pip** permet d'installer pip pour Python 3, afin de pouvoir installer des outils Python comme Ansible.



La commande **pip install ansible** permet d'installer Ansible via pip, mais ici elle échoue car l'environnement Python est géré par Debian ("externally-managed-environment").



La commande **apt-get install pipx** permet d'installer pipx pour installer des outils Python dans un environnement isolé.



La commande **pipx ensurepath** permet d'ajouter le dossier pipx au PATH pour rendre les commandes accessibles.


```
root@Ansible:~# pipx ensurepath
Success! Added /root/.local/bin to the PATH environment variable.

Consider adding shell completions for pipx. Run 'pipx completions' for
instructions.

You will need to open a new terminal or re-login for the PATH changes to take
effect. Alternatively, you can source your shell's config file with e.g.
'source ~/.bashrc'.

Otherwise pipx is ready to go! ✨ 🌟 ✨
root@Ansible:~#
```

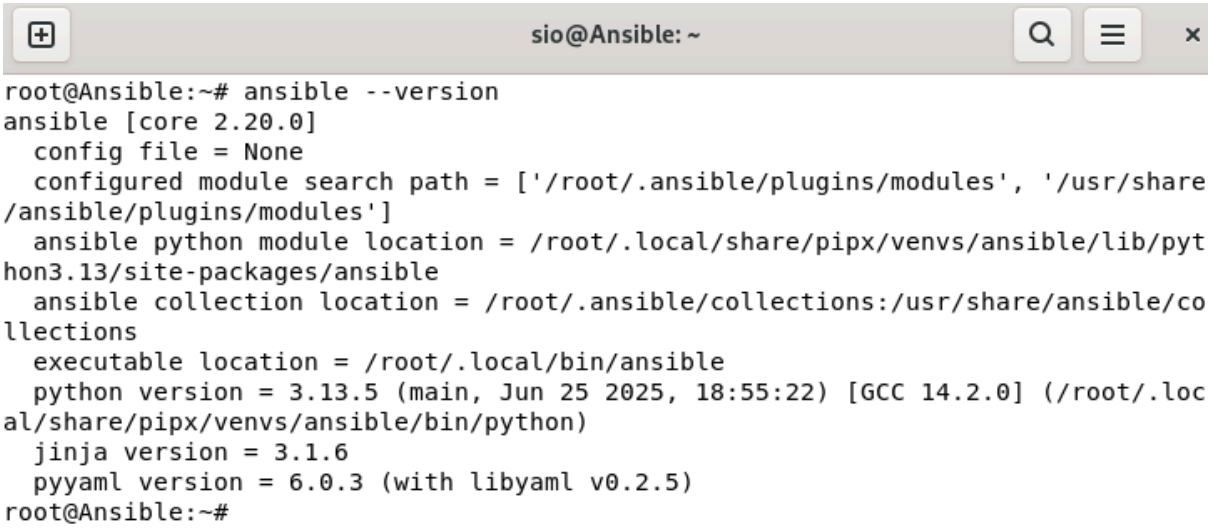
La commande **pipx ensurepath --global** permet d'ajouter pipx au PATH global (si nécessaire).

```
root@Ansible:~# pipx ensurepath --global
/usr/local/bin is already in PATH.

⚠️ All pipx binary directories have been appended to PATH. If you are sure
you want to proceed, try again with the '--force' flag.

Otherwise pipx is ready to go! ✨ 🌟 ✨
root@Ansible:~#
```

La commande **pipx install --include-deps ansible** permet d'installer Ansible via pipx avec ses dépendances, de façon propre.

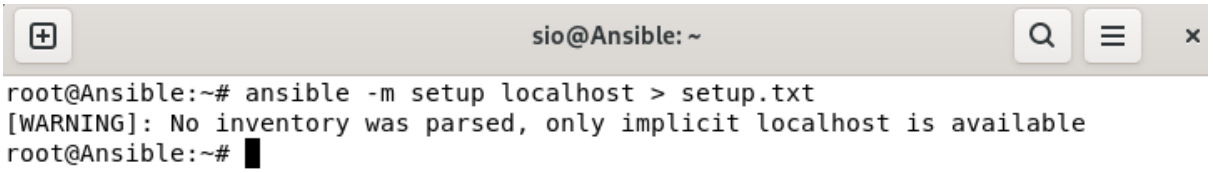
```
root@Ansible:~# pipx install --include-deps ansible
  installed package ansible 13.0.0, installed using Python 3.13.5
  These apps are now globally available
    - ansible
    - ansible-community
    - ansible-config
    - ansible-console
    - ansible-doc
    - ansible-galaxy
    - ansible-inventory
    - ansible-playbook
    - ansible-pull
    - ansible-test
    - ansible-vault
⚠️ Note: '/root/.local/bin' is not on your PATH environment variable. These
    apps will not be globally accessible until your PATH is updated. Run `pipx
    ensurepath` to automatically add it, or manually modify your PATH in your
    shell's config file (e.g. ~/.bashrc).
done! ✨ 🌟 ✨
root@Ansible:~#
```

La commande **ansible --version** permet de vérifier la version et confirmer qu'Ansible est bien installé.



2. Premier module avec Ansible (setup)

La commande **ansible -m setup localhost > setup.txt** permet d'exécuter le module setup sur localhost pour récupérer les informations système et les enregistrer dans setup.txt.



La commande cat setup.txt permet d'afficher le contenu du fichier setup.txt, avec les facts Ansible récupérés sur la machine (IP, matériel, etc.).

```
"ansible_apparmor": {
    "status": "enabled"
},
"ansible_architecture": "x86_64",
"ansible_bios_date": "12/01/2006",
"ansible_bios_vendor": "innotek GmbH",
"ansible_bios_version": "VirtualBox",
"ansible_board_asset_tag": "NA",
"ansible_board_name": "VirtualBox",
"ansible_board_serial": "0",
"ansible_board_vendor": "Oracle Corporation",
"ansible_board_version": "1.2",
"ansible_chassis_asset_tag": "NA",
"ansible_chassis_serial": "NA",
"ansible_chassis_vendor": "Oracle Corporation",
"ansible_chassis_version": "NA",
"ansible_cmdline": {
    "BOOT_IMAGE": "/boot/vmlinuz-6.12.48+deb13-amd64",
    "quiet": true,
    "ro": true,
    "root": "UUID=9ccbd479-6936-430a-8f70-1ed16e8b90e6"
},
"ansible_date_time": {
    "date": "2025-12-03",
    "day": "03",
    "epoch": "1764772054",
    "epoch_int": "1764772054",
    "hour": "15",
    "iso8601": "2025-12-03T14:27:34Z",
    "iso8601_basic": "20251203T152734286655",
    "iso8601_basic_short": "20251203T152734",
    "iso8601_micro": "2025-12-03T14:27:34.286655Z",
    "minute": "27",
    "month": "12",
```

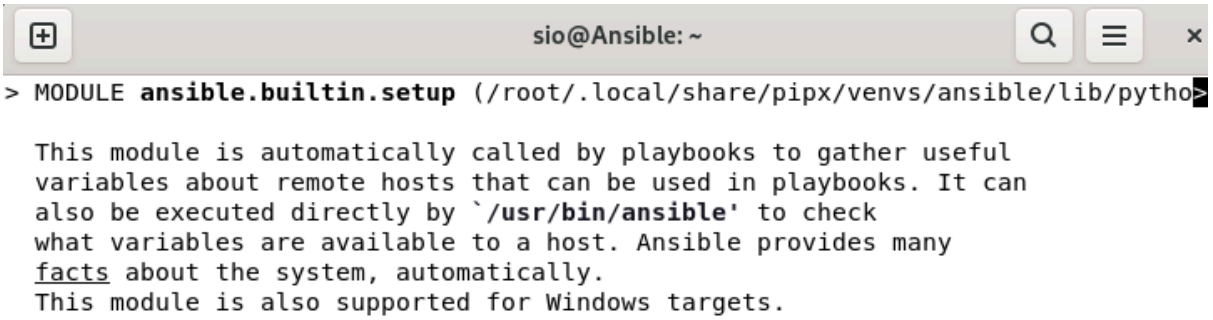
sio@Ansible: ~

```
    "iso8601_basic": "20251203T152734286655",
    "iso8601_basic_short": "20251203T152734",
    "iso8601_micro": "2025-12-03T14:27:34.286655Z",
    "minute": "27",
    "month": "12",
    "second": "34",
    "time": "15:27:34",
    "tz": "CET",
    "tz_dst": "CEST",
    "tz_offset": "+0100",
    "weekday": "mercredi",
    "weekday_number": "3",
    "weeknumber": "48",
    "year": "2025"
},
```

```
"ansible_default_ipv4": {
    "address": "10.0.2.15",
    "alias": "enp0s3",
    "broadcast": "10.0.2.255",
    "gateway": "10.0.2.2",
    "interface": "enp0s3",
    "macaddress": "08:00:27:2f:7a:90",
    "mtu": 1500,
    "netmask": "255.255.255.0",
    "network": "10.0.2.0",
    "prefix": "24",
    "type": "ether"
},
"ansible_default_ipv6": {
    "address": "fd17:625c:f037:2:b2ef:cab4:e8c9:4823",
    "gateway": "fe80::2",
    "interface": "enp0s3",
    "macaddress": "08:00:27:2f:7a:90",
    "mtu": 1500,
    "prefix": "64",
    "scope": "global",
    "type": "ether"
},
"ansible_device_links": {
    "ids": {
        "sda": [
            "ata-VBOX_HARDDISK_VB9eaff554-ae91a163"
        ],
        "sda1": [
            "ata-VBOX_HARDDISK_VB9eaff554-ae91a163-part1"
```


3. Aide et documentation oficielle

La commande **ansible-doc ansible.builtin.setup** permet d'afficher la documentation du module setup (rôle, options et facts collectés).



```
OPTIONS (red indicates it is required):

   fact_path  Path used for local ansible facts (`*.fact')
              - files in this dir will be run (if executable) and
              their results be added to `ansible_local'
              facts. If a file is not executable it is read instead.
              File/results format can be JSON or INI-format. The
              default `fact_path' can be specified in
              `ansible.cfg' for when setup is automatically
              called as part of `gather_facts'. NOTE - For
              windows clients, the results will be added to a variable
              named after the local file (without extension suffix),
              rather than `ansible_local'.
              Since Ansible 2.1, Windows hosts can use `fact_path'.
              Make sure that this path exists on the target host.
              Files in this path MUST be PowerShell scripts
              `.ps1' which outputs an object. This object
              will be formatted by Ansible as json so the script
              should be outputting a raw hashtable, array, or other
              primitive object.
         default: /etc/ansible/facts.d
         type: path

   filter  If supplied, only return facts that match one of the
           shell-style (fnmatch) pattern. An empty list basically
           means 'no filter'. As of Ansible 2.11, the type has changed
           from string to list and the default has became an empty
           list. A simple string is still accepted and works as a
           single pattern. The behaviour prior to Ansible 2.11
           remains.
         default: []
         elements: str
         type: list

   gather_subset  If supplied, restrict the additional facts collected
:
```

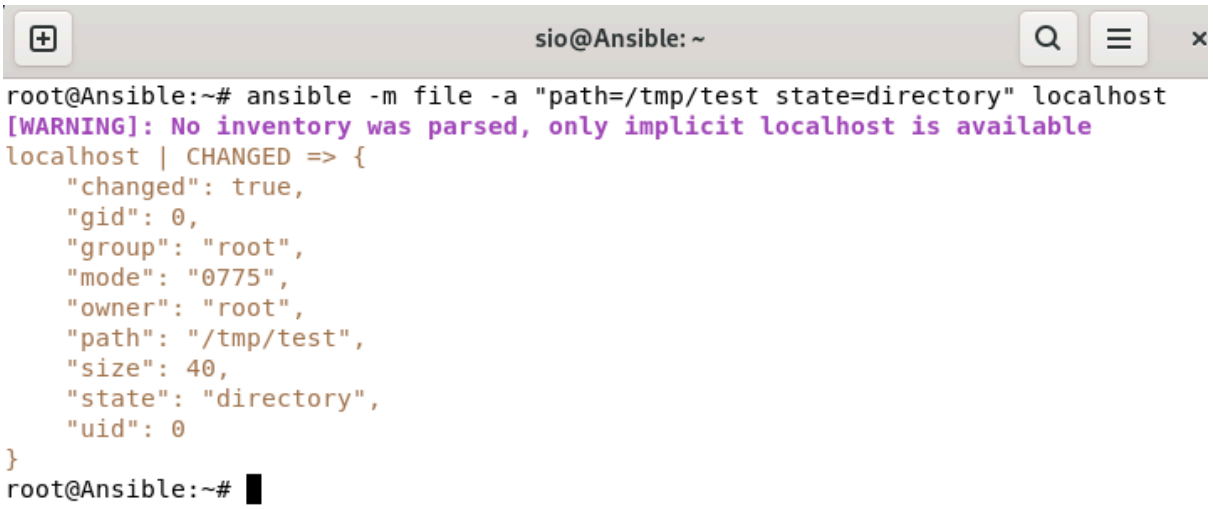
La commande **ansible -m setup -a gather_subset=min localhost** permet de récupérer un minimum d'informations système (facts) sur localhost en limitant la collecte au strict nécessaire.

```
sio@Ansible: ~
root@Ansible:~# ansible -m setup -a gather_subset=min localhost
[WARNING]: No inventory was parsed, only implicit localhost is available
localhost | SUCCESS => {
    "ansible_facts": {
        "ansible_apparmor": {
            "status": "enabled"
        },
        "ansible_architecture": "x86_64",
        "ansible_cmdline": {
            "BOOT_IMAGE": "/boot/vmlinuz-6.12.48+deb13-amd64",
            "quiet": true,
            "ro": true,
            "root": "UUID=9ccbd479-6936-430a-8f70-1ed16e8b90e6"
```

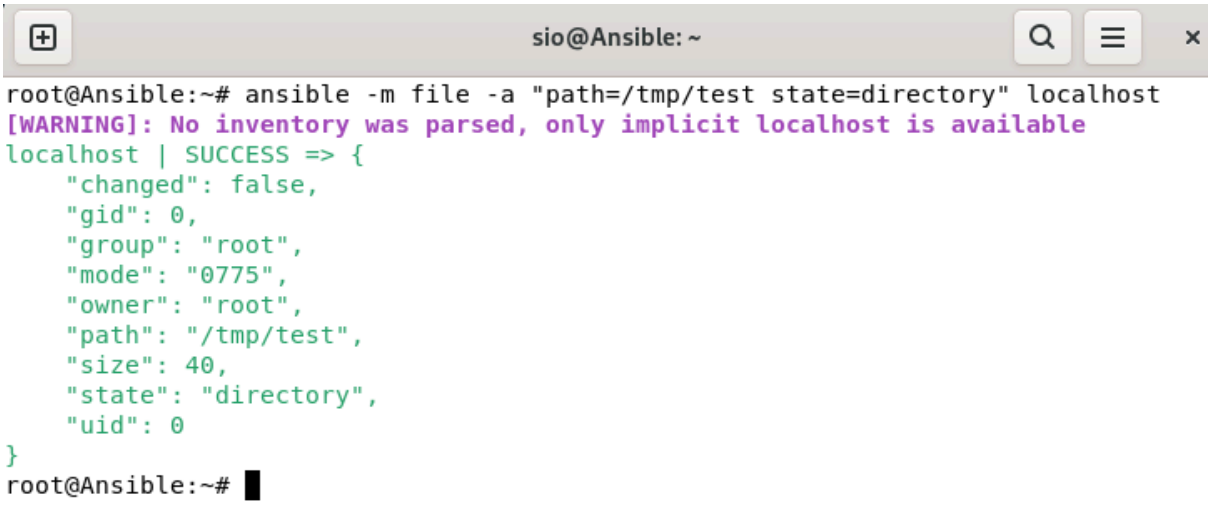
```
        },
        "ansible_date_time": {
            "date": "2025-12-03",
            "day": "03",
            "epoch": "1764772287",
            "epoch_int": "1764772287",
            "hour": "15",
            "iso8601": "2025-12-03T14:31:27Z",
            "iso8601_basic": "20251203T153127237499",
            "iso8601_basic_short": "20251203T153127",
            "iso8601_micro": "2025-12-03T14:31:27.237499Z",
            "minute": "31",
            "month": "12",
            "second": "27",
            "time": "15:31:27",
            "tz": "CET",
            "tz_dst": "CEST",
            "tz_offset": "+0100",
            "weekday": "mercredi",
            "weekday_number": "3",
            "weeknumber": "48",
            "year": "2025"
        },
        "ansible_distribution": "Debian",
        "ansible_distribution_file_parsed": true,
        "ansible_distribution_file_path": "/etc/os-release",
        "ansible_distribution_file_variety": "Debian",
        "ansible_distribution_major_version": "13",
        "ansible_distribution_minor_version": "1",
        "ansible_distribution_release": "trixie",
        "ansible_distribution_version": "13.1",
        "ansible_dns": {
```


4. Idempotence : exemple avec gestion des répertoires

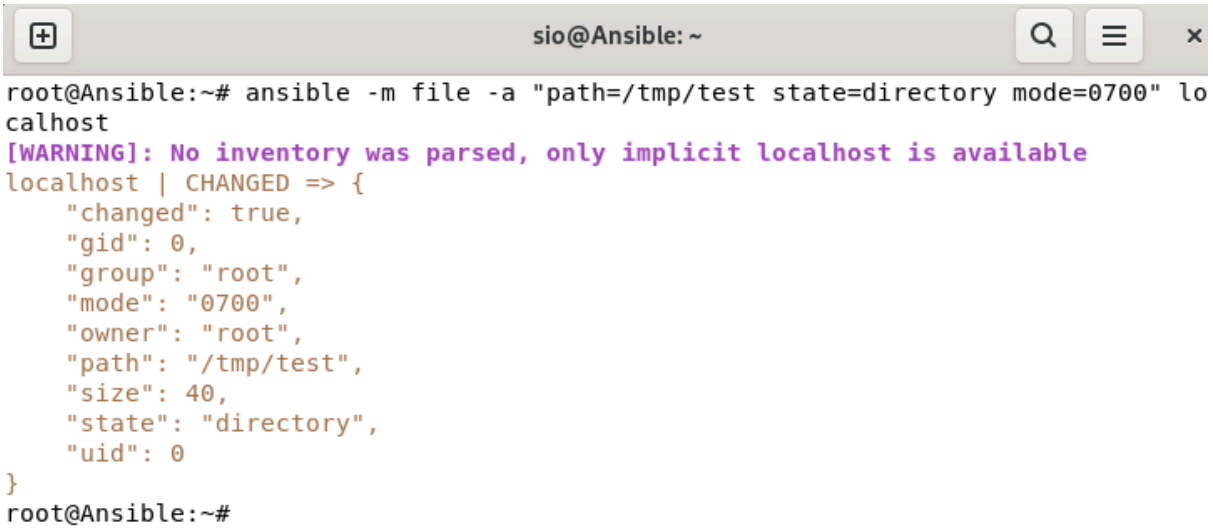
La commande **ansible -m file -a "path=/tmp/test state=directory" localhost** permet de créer le dossier /tmp/test sur localhost via le module file.



La commande **ansible -m file -a "path=/tmp/test state=directory"** localhost permet de vérifier/créer le dossier /tmp/test (si changed: false, il existait déjà).



La commande **ansible -m file -a "path=/tmp/test state=directory mode=0700" localhost** permet de modifier les permissions du dossier en 0700.

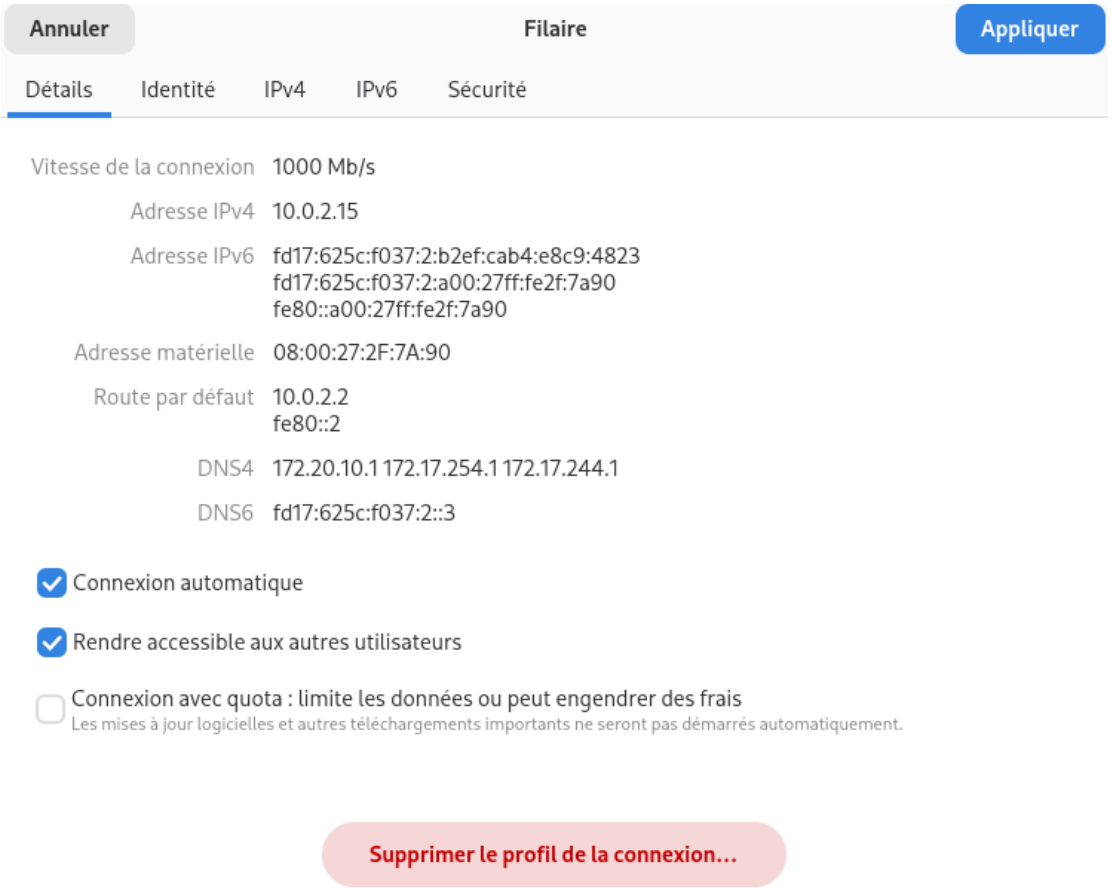


La commande **ls -ld /tmp/test/** permet de vérifier les droits appliqués sur le dossier /tmp/test.

5. Création d'un fichier d'inventaire

La commande **apt-get install kate** permet d'installer l'éditeur de texte Kate (interface graphique) pour modifier des fichiers plus facilement.

Vérification de la configuration réseau de la VM Ansible (mode réseau interne LAN) :



La commande **nano /etc/hosts** permet de définir les correspondances nom ↔ IP (ex. 192.168.3.4 → Ansible, 192.168.2.2 → Web_1).

La commande **systemctl restart sshd** permet de redémarrer le service SSH pour appliquer la configuration.

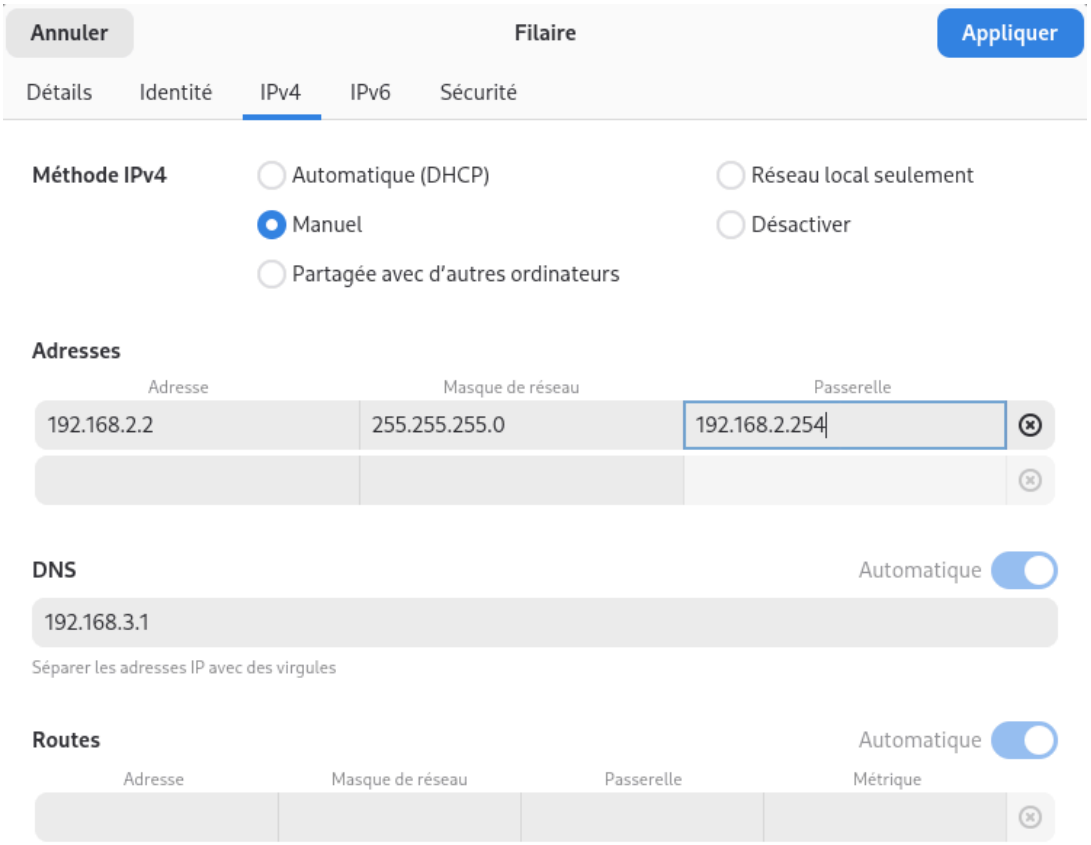
La commande **nano /etc/hostname** permet de changer le nom d'hôte de la machine (ex. Web_1).

La commande **nano /etc/hosts** permet d'associer l'IP 192.168.2.2 au nom Web_1 pour la résolution locale.

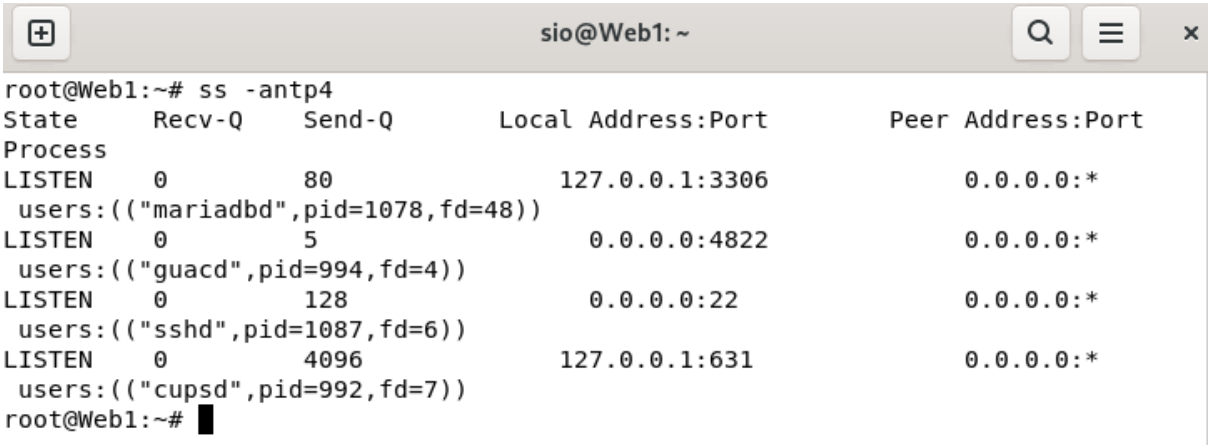
La commande **nano /etc/hostname** permet de définir le nom d'hôte de la machine en Web_1.

Les VM **web-1** et **bdd-1** sont des clones de Debian 13, configurés en réseau interne DMZ (192.168.2.0/24).

- VM web-1 :



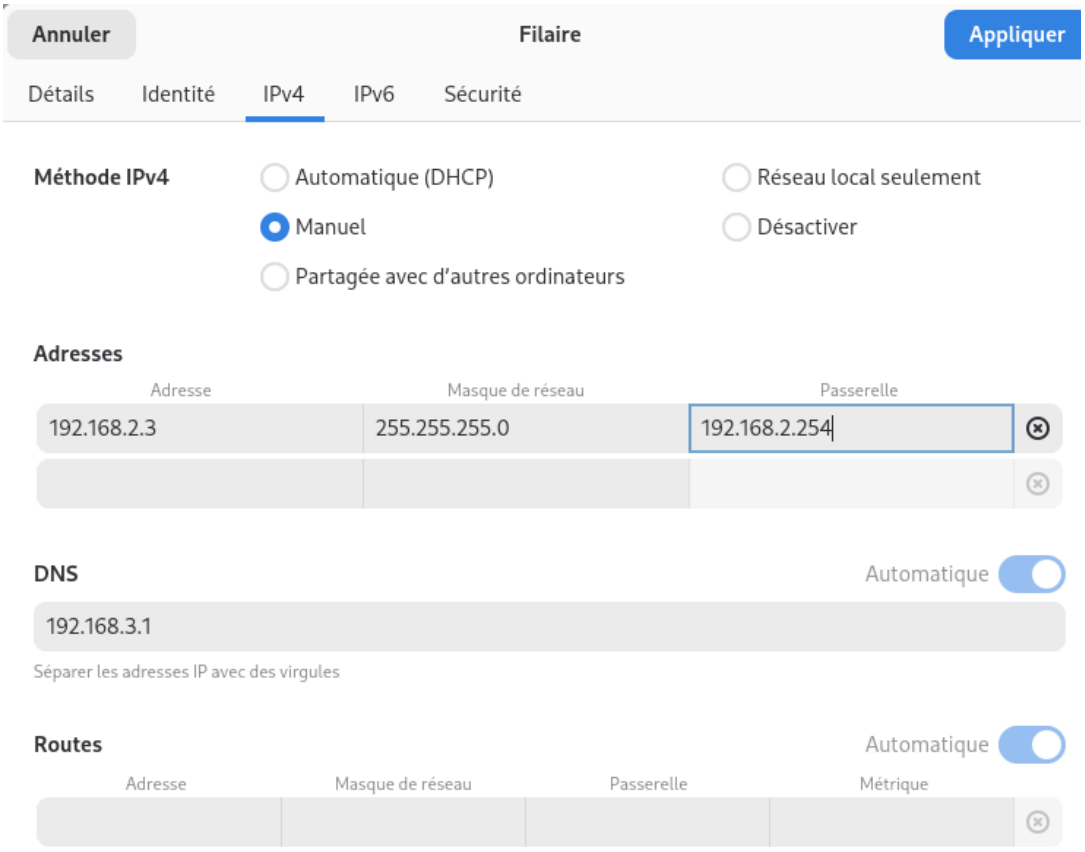
La commande **ss -antp4** permet d'afficher les ports TCP IPv4 en écoute et les processus associés (ex. SSH, MariaDB, guacd).



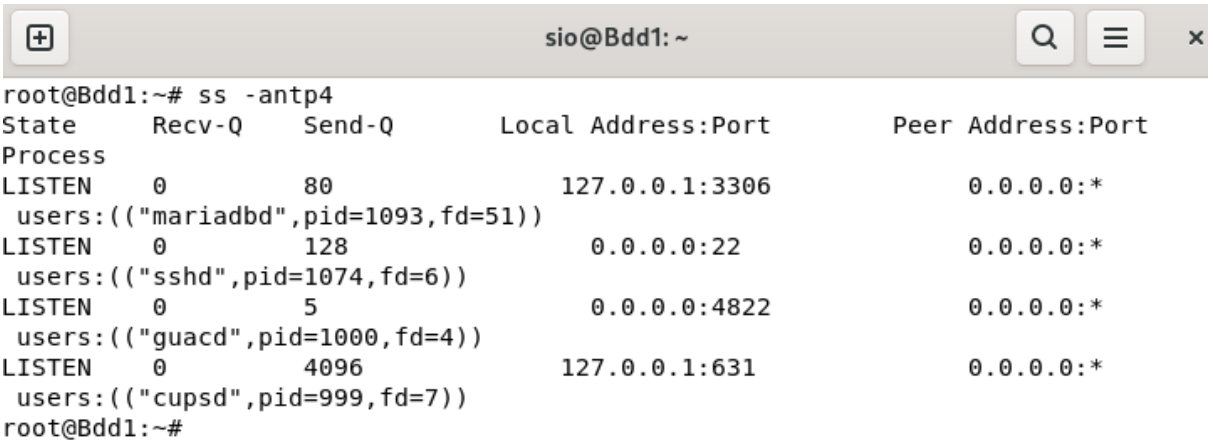
La commande **nano /etc/ssh/sshd_config** permet de modifier la configuration du serveur SSH (ex. autoriser root avec PermitRootLogin yes).

La commande nano **/etc/hosts** permet d'associer l'IP 192.168.2.3 au nom Bdd_1 pour la résolution locale.

- VM bdd-1 :



La commande **ss -antp4** permet d'afficher les ports TCP IPv4 en écoute et les processus associés (ex. SSH, MariaDB, guacd).



La commande **nano /etc/ssh/sshd_config** permet de modifier la configuration du serveur SSH (ex. autoriser root avec PermitRootLogin yes).

La commande **systemctl restart sshd** permet de redémarrer le service SSH pour appliquer les changements de configuration.