



# Dossier Technique

## Projet 10 - Automatisation avec Ansible

Introduction.....	2
Contexte.....	2
Environnement technique.....	4
Infrastructure.....	4
Réseau.....	4
Machines.....	4
Prérequis Configuration Machines.....	5
- OPNSense.....	5
- Mise en place Ansible.....	6
- Fonctionnement.....	6
- Architecture Ansible.....	6
- Inventaire.....	8
- Ansible Playbook.....	8
- Contexte.....	8
- Configuration SNMP.....	8
- Ajout d'host sur Centreon.....	8
- Gestion des mots de passe.....	8
- Ressources.....	8

# Introduction

## Contexte

Au sein de diverses infrastructures, il est très recommandé d'utiliser des solutions d'automatisation/industrialisation des actions.

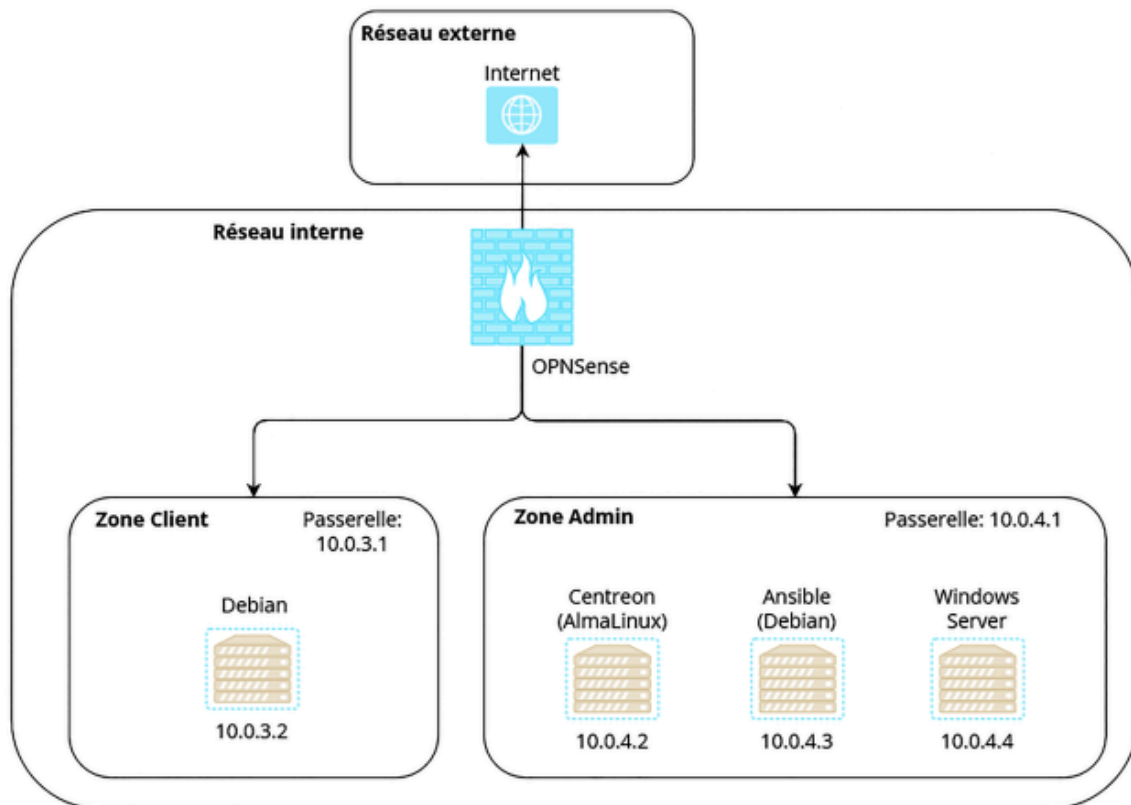
Pour la réalisation de ce projet nous allons utiliser Ansible qui est un logiciel d'automatisation informatique en ligne de commande open source écrit en Python.

Il peut configurer des systèmes, déployer des logiciels et orchestrer des flux de travaux avancés pour prendre en charge le déploiement d'applications, les mises à jour du système, etc.

Les principaux atouts d'Ansible sont la simplicité et la facilité d'utilisation. Il met également l'accent sur la sécurité et la fiabilité, avec un minimum de pièces mobiles. Il utilise OpenSSH pour le transport.

# Environnement technique

## Infrastructure



## Réseau

Le Netmask sera égale a 255.255.255.0 pour tous les réseau

le DNS sera égale à 8.8.8.8

## Machines

Nous allons utiliser 5 machines principale :

- OPNSense pour définir notre réseau interne
- Windows Server 2019 qui aura le SNMP configuré par Ansible
- Linux Debian qui aura le SNMP configuré par Ansible
- Centreon AlmaLinux pour la supervision
- Ansible Debian pour automatiser les tâches.

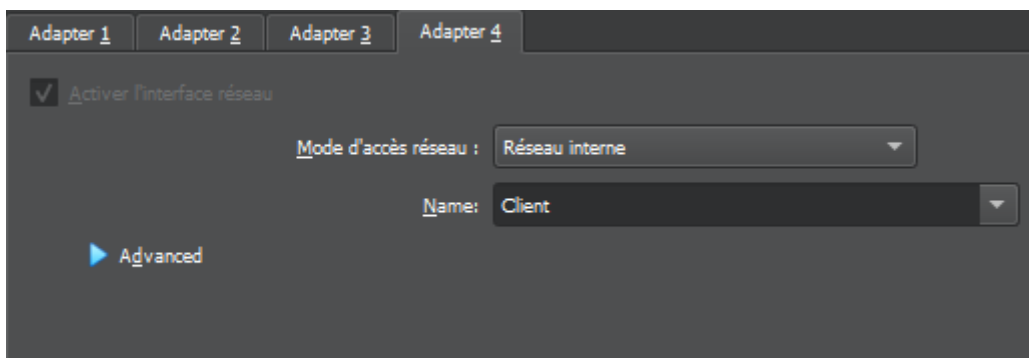
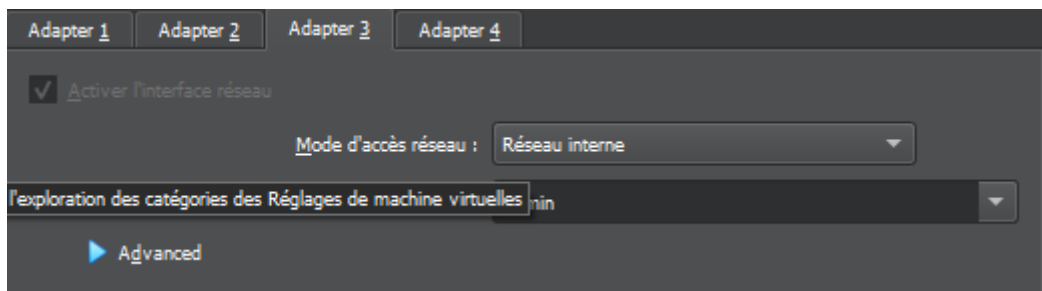
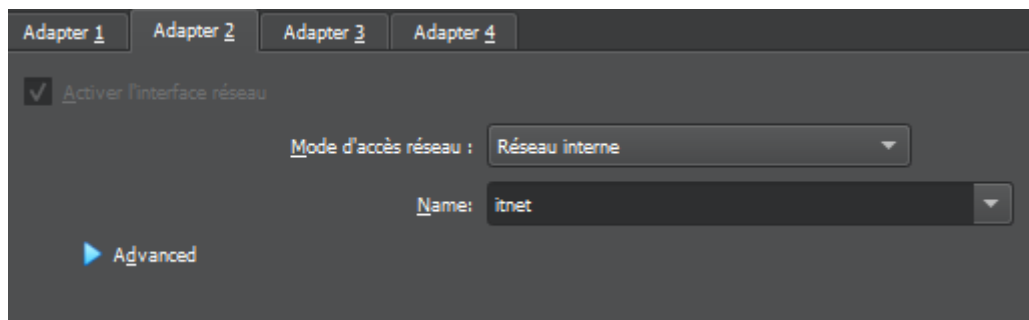
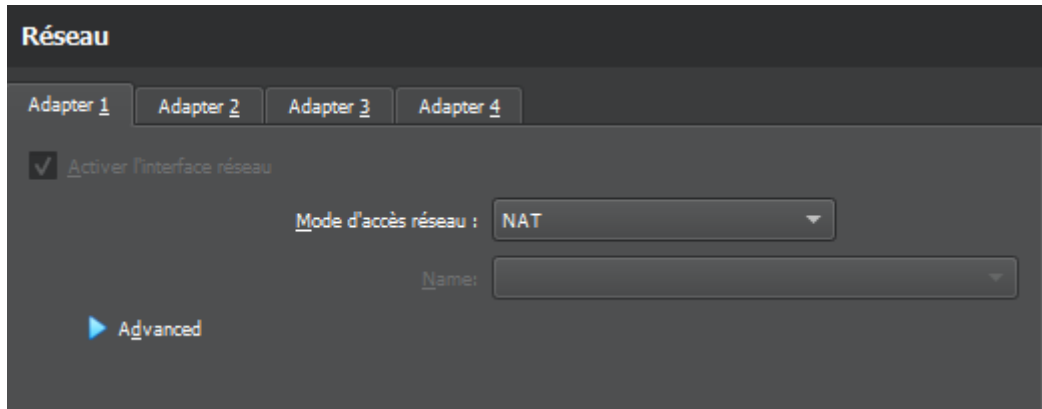
## Prérequis Configuration Machines

### OPNSense

Commencez par télécharger puis installer l'iso OPNSense :

<https://opnsense.org/download/>

Modifiez ensuite les interfaces réseau sur virtual box de la machine OPNSense.



Une fois cela fait, connectez-vous puis assignez les différentes interfaces réseaux.

WAN:	em0
LAN:	em1
Admin:	em2
Client:	em3

Attribuer les adresses IP correspondant à chaque réseau sur OPNSense puis à chaque machine selon le plan réseau défini plus haut dans l'infrastructure

Lorsque chaque machine sera installée, connectez-vous ensuite à l'interface OPNSense depuis une des machines en questions pour autoriser les réseaux à communiquer via la modification des règles OPNSense.

## Mise en place Ansible

### Fonctionnement

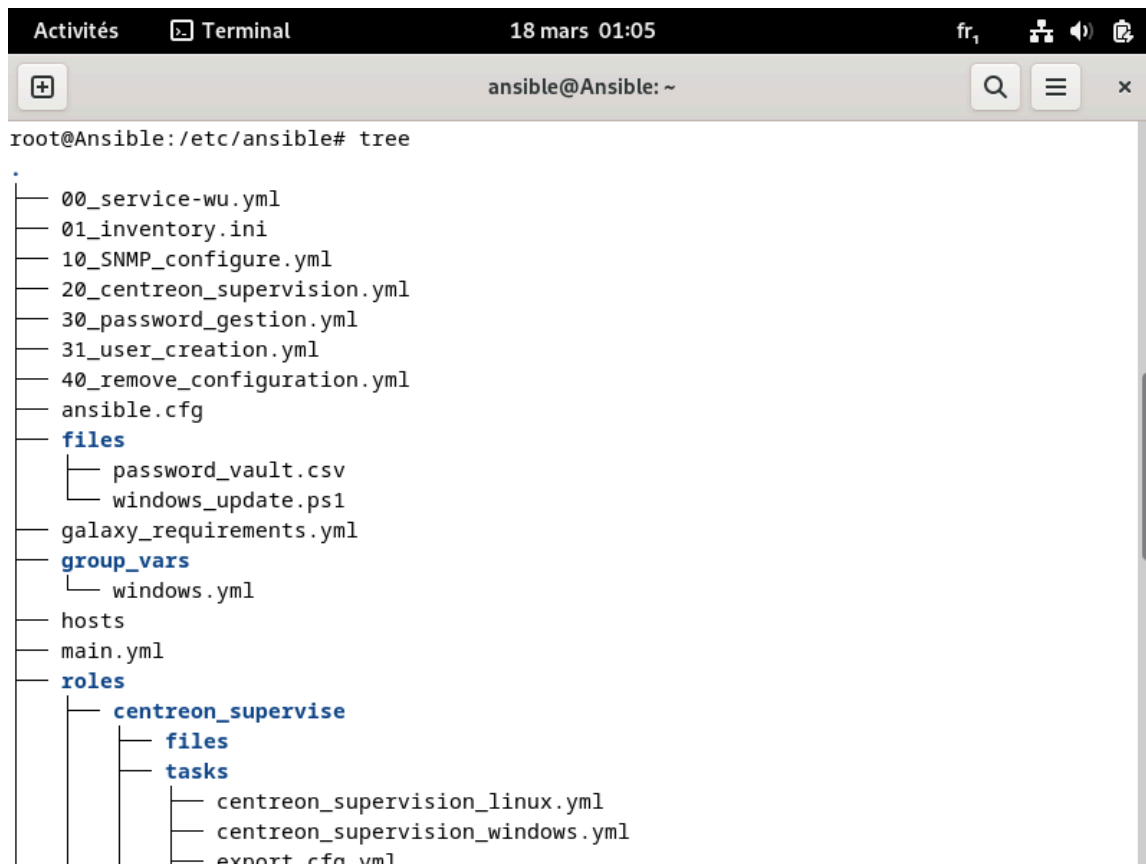
Ansible va fonctionner sur notre machine Ansible debian et possèdera toutes l'arborescence des dossiers/fichiers lié à ansible.

Cela fonctionnera avec un fichier appelé Inventory qui se verra contenir un groupe d'hosts (adresse ip de machine présente dans le réseau) afin de définir des variables lui permettant de savoir avec qui ansible devra communiquer via une connexion ssh lors de l'exécution de tâches spécifiques sur diverses machines.

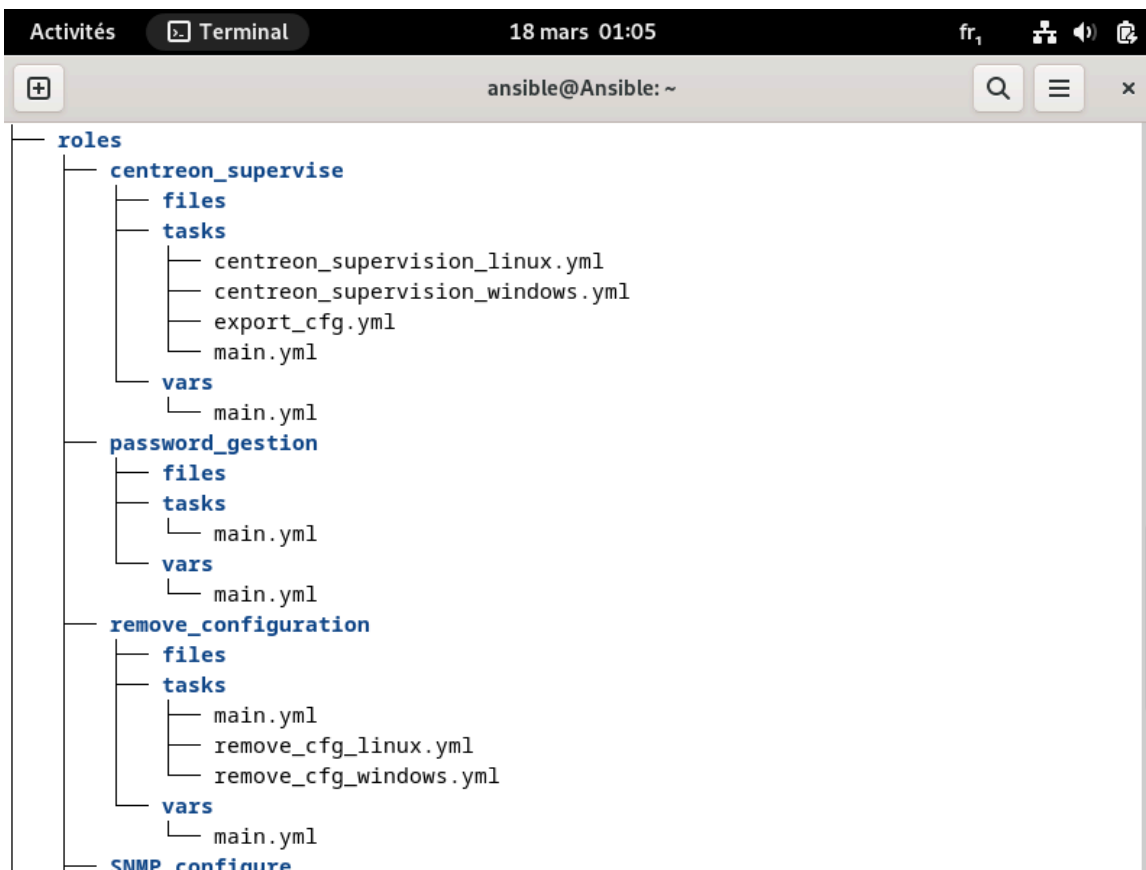
Il existera aussi ce que l'on appelle des Playbook qui permettront d'effectuer une ou plusieurs suite(s) de tâches spécifiques.

On pourra aussi sauvegarder des données Vars et les chiffrer via ansible-vault afin de permettre d'augmenter la sécurité et de globaliser des variables au sein des divers playbook existant.

## Architecture Ansible



```
root@Ansible:/etc/ansible# tree
.
├── 00_service-wu.yml
├── 01_inventory.ini
├── 10_SNMP_configure.yml
├── 20_centreon_supervision.yml
├── 30_password_gestion.yml
├── 31_user_creation.yml
├── 40_remove_configuration.yml
├── ansible.cfg
├── files
│   ├── password_vault.csv
│   └── windows_update.ps1
├── galaxy_requirements.yml
├── group_vars
│   └── windows.yml
├── hosts
├── main.yml
├── roles
│   ├── centreon_supervise
│   │   ├── files
│   │   └── tasks
│   │       ├── centreon_supervision_linux.yml
│   │       ├── centreon_supervision_windows.yml
│   │       └── export_cfg.yml
│   └── password_gestion
│       ├── files
│       ├── tasks
│       │   └── main.yml
│       └── vars
│           └── main.yml
└── remove_configuration
    ├── files
    ├── tasks
    │   ├── main.yml
    │   ├── remove_cfg_linux.yml
    │   └── remove_cfg_windows.yml
    └── vars
        └── main.yml
```



```
roles
├── centreon_supervise
│   ├── files
│   ├── tasks
│   │   ├── centreon_supervision_linux.yml
│   │   ├── centreon_supervision_windows.yml
│   │   ├── export_cfg.yml
│   │   └── main.yml
│   ├── vars
│   │   └── main.yml
│   └── password_gestion
│       ├── files
│       ├── tasks
│       │   └── main.yml
│       └── vars
│           └── main.yml
├── remove_configuration
│   ├── files
│   ├── tasks
│   │   ├── main.yml
│   │   ├── remove_cfg_linux.yml
│   │   └── remove_cfg_windows.yml
│   └── vars
│       └── main.yml
└── SNMP_configure
```



```
Activités Terminal 18 mars 01:06 fr_ [network] [sound] [refresh]
ansible@Ansible: ~
+
├── main.yml
├── remove_cfg_linux.yml
├── remove_cfg_windows.yml
├── vars
│   └── main.yml
├── SNMP_configure
│   ├── files
│   ├── tasks
│   │   ├── main.yml
│   │   ├── snmp_cfg_linux.yml
│   │   └── snmp_cfg_windows.yml
│   └── vars
│       └── main.yml
├── user_creation
│   ├── files
│   ├── tasks
│   │   └── main.yml
│   └── vars
│       └── main.yml
└── vars
    └── main.yml

25 directories, 32 files
root@Ansible:/etc/ansible#
```

## Inventaire

Dans le fichier d'inventaire, il nous faut un groupe pour chacun des hôtes (Windows Server, Linux Debian, Centreon) et dont on va spécifier l'adresse IP de chacun. Comme cela, nous pourrons facilement choisir sur quel(s) hôte(s) se dérouleront les playbooks.

## Ansible Playbook

### Contexte

Dans le cadre de ce projet, nous avons créé un ensemble de playbook permettant de mettre en place sur une machine Linux Debian et Windows Server la configuration SNMP (*10\_SNMP\_configure.yml*), l'ajout de ces machines sur Centreon (*20\_centreon\_supervision.yml*) ainsi que proposer une façon de gérer des mots de passe (*30\_password\_gestion.yml*).

Chacun de ces playbooks peut être joué indépendamment ou alors successivement via le *main.yml*.

Un playbook permettant de revenir sur un état avant le lancement des playbooks a également été fait (*40\_remove\_configuration.yml*)

### Configuration SNMP

Ce playbook permet d'installer et configurer SNMP sur nos deux machines distantes. Il va appeler le rôle *SNMP\_configure* qui va jouer deux ensembles de tâches, l'un pour l'hôte Windows, l'autre pour Linux.

### Ajout d'host sur Centreon

Ce playbook permet d'ajouter nos deux machines distantes à la machine Centreon. Il va appeler le rôle *centreon\_supervise* qui va jouer deux ensembles de tâches, l'un pour l'hôte Windows, l'autre pour Linux.

### Gestion des mots de passe

Ce playbook propose une façon de gérer les mots de passe des utilisateurs sur les deux hôtes distants. Il va appeler le rôle *password\_gestion* qui va jouer deux ensembles de tâches, l'un pour l'hôte Windows, l'autre pour Linux.

### Suppression de la configuration

Ce playbook a pour but de 'revenir en arrière'. Il va nous permettre de supprimer les différentes installations et modifications de configuration qui ont été faites. Il va appeler le rôle *remove\_configuration* qui va jouer deux ensembles de tâches, l'un pour l'hôte Windows, l'autre pour Linux.

## Ressources

- Repository GitHub : [https://github.com/Amael7/Gu2\\_Ansible\\_Automation](https://github.com/Amael7/Gu2_Ansible_Automation)
- Centreon AlmaLinux : <https://download.centreon.com/>
- Debian Linux : <https://www.debian.org/download.fr.html>
- Windows Server:  
<https://www.microsoft.com/fr-fr/evalcenter/evaluate-windows-server-2019>
- Ansible debian: <https://www.debian.org/download.fr.html>