Rapport de Projet : Automatisation des Sauvegardes et Restaurations avec Ansible et PBS

Objectif du projet :
L'objectif principal de ce projet est d'automatiser les sauvegardes des machines virtuelles (VM) sous Proxmox en utilisant Ansible et Proxmox Backup Server (PBS). Cette approche permet de garantir une haute disponibilité et une reprise après sinistre efficace pour les machines virtuelles.

Technologies utilisées :

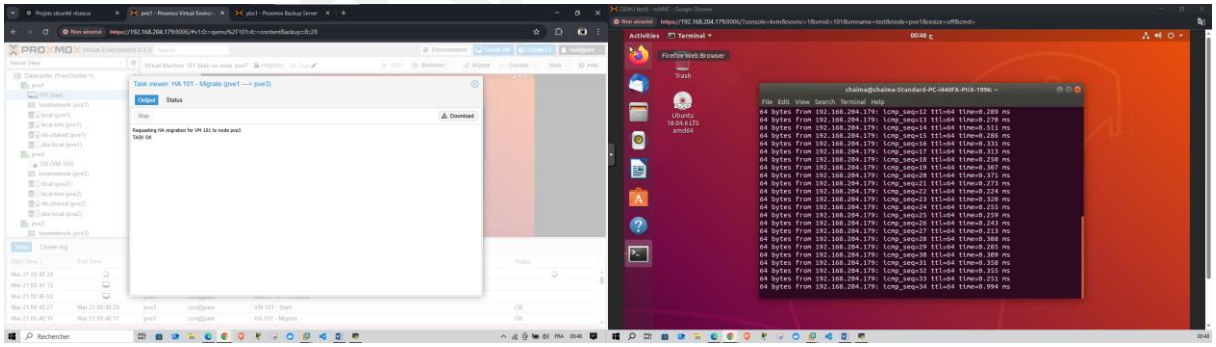
- **Ansible :** Automatisation des tâches de gestion de la configuration.
- **Proxmox Backup Server (PBS) :** Gestion des sauvegardes et restauration des VMs.
- **Shell scripting :** Scripts pour exécuter les tâches de sauvegarde et de restauration.
- **SSH :** Pour l'accès sécurisé entre les nœuds du cluster.
- **Cron :** Pour la planification des sauvegardes.
- **Zabbix :** Pour la supervision et la surveillance de l'état du système.

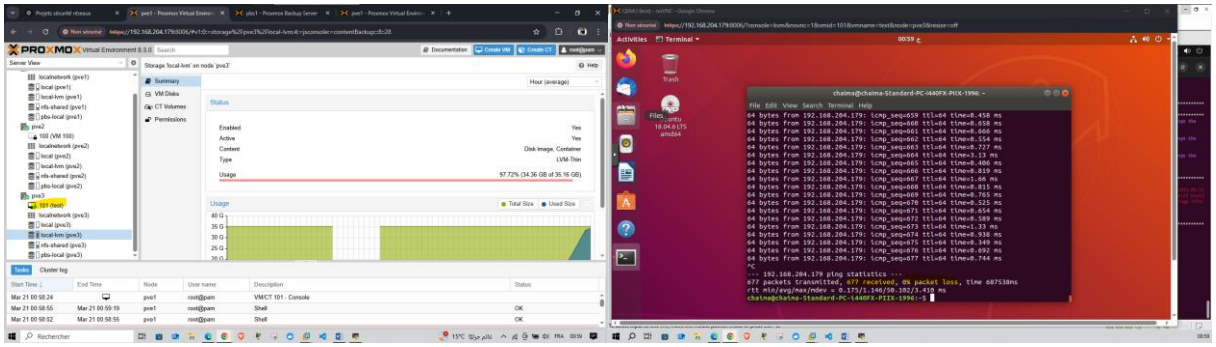
Pourquoi ce projet est important ?
Le projet permet d'assurer une solution robuste pour la sauvegarde et la restauration des machines virtuelles, combinant haute disponibilité et reprise après sinistre. Ce système est crucial pour maintenir la continuité des services, en particulier dans un environnement de production.

Architecture et Infrastructure :

1. **Cluster Proxmox :** Le cluster Proxmox est composé de trois nœuds ayant les adresses IP suivantes :
   - **192.168.204.179**
   - **192.168.204.180**
   - **192.168.204.181**

   Ce cluster permet une gestion centralisée et une migration à chaud des machines virtuelles entre les nœuds.





2. **Stockage NFS :** Un stockage NFS est utilisé pour la migration à chaud des VMs entre les nœuds du cluster. Cela permet de transférer les machines virtuelles sans arrêt de service, ce qui est essentiel pour la haute disponibilité.
3. **Proxmox Backup Server (PBS) :** Le PBS, avec l'adresse IP **192.168.204.182**, est utilisé pour stocker les sauvegardes des VMs et faciliter la récupération des données en cas de panne.
4. **Ansible :** Ansible est installé sur la machine virtuelle NFS pour gérer les sauvegardes à l'aide de playbooks. Ces playbooks automatisent le processus de sauvegarde et de restauration, réduisant ainsi les risques d'erreurs humaines et assurant une cohérence des sauvegardes.

Ansible est un outil d'automatisation simple à utiliser pour gérer des serveurs et des machines à distance. Il repose sur une architecture très simple, sans besoin d'installer des agents sur les machines que tu souhaites gérer.

Voici les principaux éléments de l'architecture d'Ansible expliqués simplement :

1. **Serveur de Contrôle** : C'est la machine où Ansible est installé. Tu y exécutes des commandes ou des **playbooks** (des scripts d'automatisation). Le serveur de contrôle gère l'ensemble du processus.
2. **Hôtes Distants (Nœuds)** : Ce sont les machines que tu veux gérer (par exemple, des serveurs ou des ordinateurs). Ansible se connecte à ces machines à distance via **SSH** (pour Linux) ou **WinRM** (pour Windows). Aucune installation d'agent n'est nécessaire sur ces machines.
3. **Modules** : Ce sont de petites unités de code utilisées pour effectuer des actions spécifiques sur les machines distantes, comme installer un logiciel, démarrer un service, ou créer un fichier. Ansible utilise ces modules pour appliquer des configurations ou exécuter des commandes sur les hôtes distants.
4. **Playbooks** : Ce sont des fichiers écrits en YAML qui définissent les tâches à effectuer sur les hôtes distants. Un playbook contient une série d'instructions (telles que "installer Apache", "copier un fichier", etc.) que tu souhaites automatiser.
5. **Inventaire** : C'est un fichier où tu définis les machines distantes sur lesquelles tu veux exécuter des actions. Cela permet à Ansible de savoir sur quels hôtes il doit se connecter pour appliquer les configurations.

Fonctionnement d'Ansible :

1. **Tu écris un playbook** : Ce fichier contient des instructions pour automatiser des tâches.
2. **Tu lances le playbook depuis le serveur de contrôle**.
3. **Ansible se connecte aux machines distantes via SSH (ou WinRM)** et exécute les actions définies dans le playbook.
4. **Les résultats sont renvoyés sur le serveur de contrôle** pour que tu puisses savoir si tout s'est bien passé.

Ansible est donc très simple à utiliser, car il n'y a pas besoin d'installer des logiciels complexes ou des agents sur les machines distantes. Il se sert de **SSH** pour communiquer avec les hôtes, ce qui rend la configuration et l'exécution des tâches rapides et efficaces.

5. **Zabbix :** Zabbix est utilisé pour la surveillance de l'infrastructure, y compris l'état des nœuds Proxmox, l'utilisation du stockage et le statut des sauvegardes. Cela permet de recevoir des alertes en cas de problème avec les sauvegardes ou les services critiques.

Configuration des Nœuds Proxmox avec Stockage PBS et NFS

Dans l'environnement de haute disponibilité et de sauvegarde, chaque nœud Proxmox (PVE) est configuré pour utiliser un stockage de sauvegarde afin de garantir une gestion fiable des machines virtuelles. Pour cela, deux types de stockage sont utilisés : **PBS (Proxmox Backup Server)** et **NFS (Network File System)**.

1. **Stockage PBS (Proxmox Backup Server)** : Le stockage PBS est dédié aux sauvegardes des VMs sur le serveur Proxmox Backup. Chaque nœud PVE est configuré pour se connecter au serveur PBS afin de stocker et restaurer les sauvegardes des machines virtuelles. Le serveur PBS est installé sur une machine dédiée, souvent sur un nœud spécifique du cluster, et chaque nœud PVE doit être configuré pour pointer vers ce serveur via l'interface web de Proxmox, en spécifiant l'adresse IP du serveur PBS (ici, `192.168.204.182`), le type de stockage PBS, et les options de compression des sauvegardes. Le stockage PBS permet des sauvegardes incrémentielles, réduisant ainsi la consommation de bande passante et d'espace de stockage.
2. **Stockage NFS (Network File System)** : En parallèle, le stockage NFS permet de partager un répertoire de sauvegarde entre plusieurs nœuds du cluster Proxmox. Ce stockage est souvent utilisé pour stocker des sauvegardes ou des ISO partagés entre les nœuds PVE. Le stockage NFS est monté sur chaque nœud du cluster en utilisant le protocole NFS pour accéder au partage réseau. Ce stockage est souvent utilisé pour la migration à chaud des VMs, où les fichiers de la machine virtuelle sont déplacés entre les nœuds sans interruption du service. Le partage NFS est monté dans l'interface web de Proxmox pour garantir que tous les nœuds PVE puissent accéder à ce stockage commun.

En combinant ces deux types de stockage, tu bénéficies de la flexibilité du stockage réseau (NFS) et de la sécurité et efficacité des sauvegardes Proxmox Backup Server (PBS) pour les besoins de sauvegarde et de restauration, tout en maximisant la performance et la sécurité dans ton environnement virtualisé.

```
root@pve3:~# pvesm status
Name             Type     Status           Total            Used       Availabl
e        %
local             dir     active        31682392         2929108        2711836
8    9.25%
local-lvm     lvmthin     active        34336768        33553889          78287
8   97.72%
nfs-shared        nfs     active        40453376        18474752        1989145
6   45.67%
pbs-local         pbs     active       205313000         3238204       19157270
4    1.58%
root@pve3:~#
```

Installation d'Ansible

L'installation d'Ansible sur une machine virtuelle dédiée pour la gestion des sauvegardes et de la configuration des nœuds Proxmox est une étape clé pour automatiser les processus. Pour installer Ansible, voici les étapes principales :

1. **Mise à jour des packages** : Commence par mettre à jour la liste des paquets de ton système avec la commande suivante :

   ```
   sudo apt-get update
   ```

2. **Installation d'Ansible** : Ensuite, tu peux installer Ansible en utilisant les dépôts officiels d'Ubuntu (ou de ta distribution) :

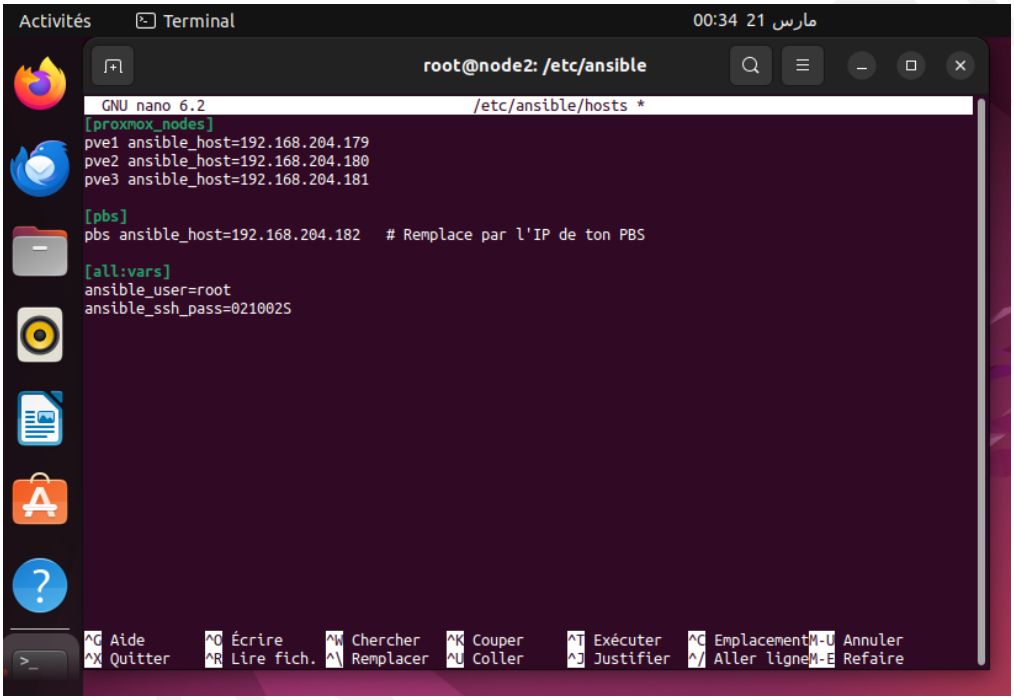
   ```
   sudo apt-get install ansible
   ```

3. **Vérification de l'installation** : Une fois l'installation terminée, tu peux vérifier qu'Ansible est bien installé en exécutant :

   ```
   ansible --version
   ```

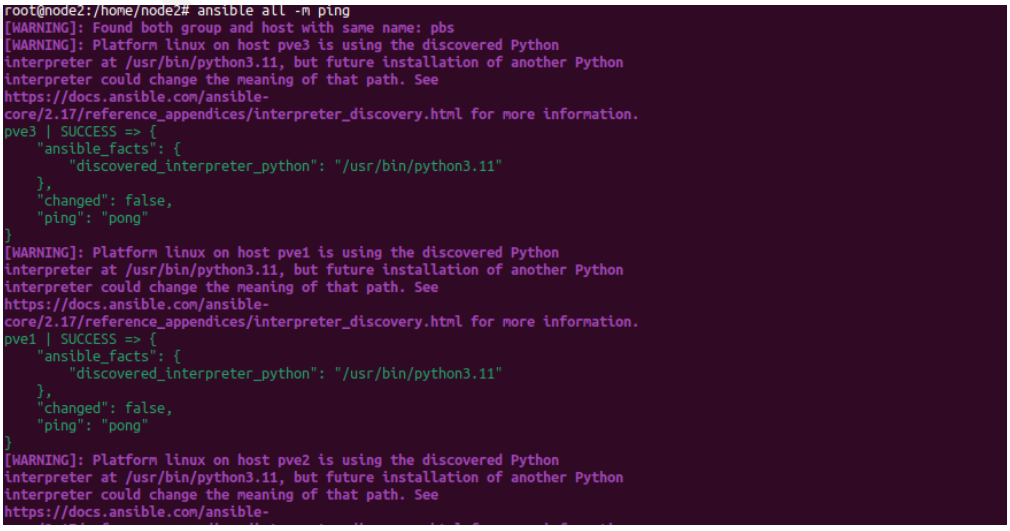
   Cela te retournera la version d'Ansible installée, confirmant ainsi que l'installation a réussi.

4. **Configuration des hôtes** : Après l'installation, tu dois configurer ton fichier d'inventaire Ansible (souvent situé dans `/etc/ansible/hosts` ou un fichier spécifique que tu définis) pour inclure les hôtes à gérer, comme les nœuds Proxmox et le serveur PBS.



5. **Test de la connexion** : Utilise une commande `ping` pour tester la connexion SSH entre la machine Ansible et les nœuds cibles :

   ```
   ansible all -m ping -u root
   ```

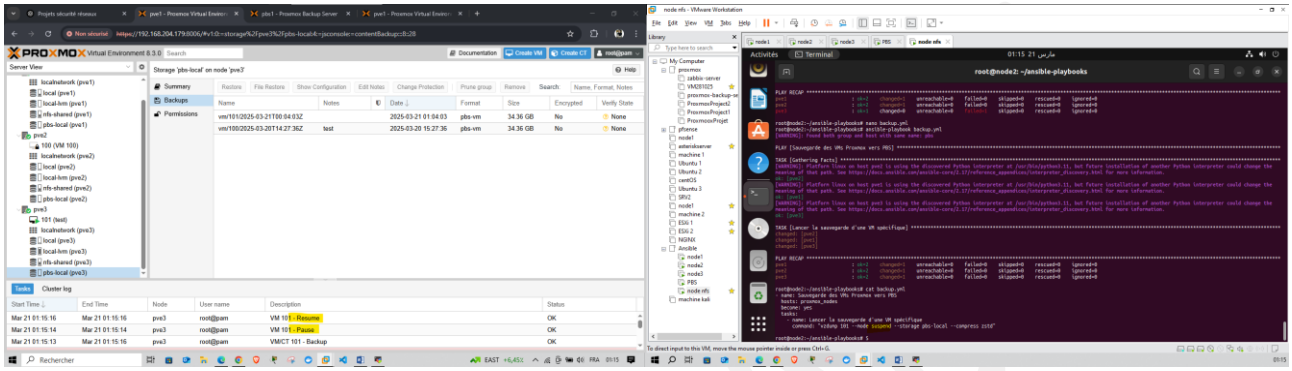


Cela configure Ansible pour commencer à gérer les machines cibles, automatiser les sauvegardes, et exécuter des playbooks pour différentes tâches d'administration système.

Procédure de Sauvegarde :

Le processus de sauvegarde utilise la commande `vzdump` pour effectuer des sauvegardes des machines virtuelles. Voici les options principales utilisées :

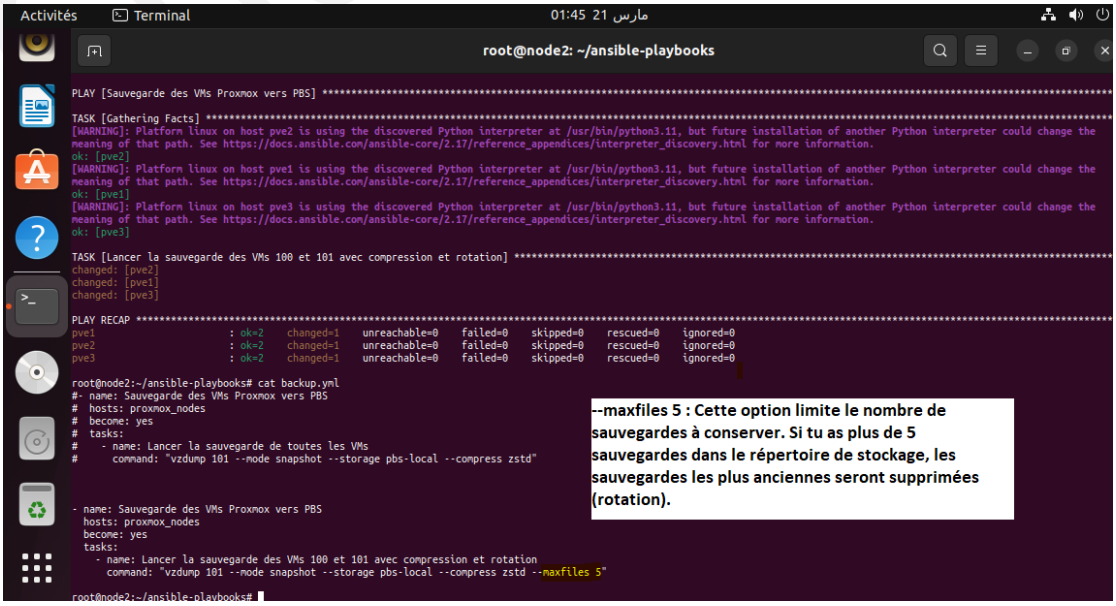
- **Mode de Sauvegarde :**
  - `snapshot` : Prend un instantané de la VM sans l'arrêter.
  - `suspend` : Suspend la VM avant de faire la sauvegarde.



  - `stop` : Arrête la VM avant de la sauvegarder.
- **Compression :**
  - `zstd` est utilisé pour la compression rapide des sauvegardes.
- **Conservation des Sauvegardes :**
  - L'option `--maxfiles` est utilisée pour conserver un nombre limité de fichiers de sauvegarde.
- **Planification avec Cron :**
  Le playbook est planifié via Cron pour effectuer des sauvegardes automatiques à intervalles réguliers.

Exemple de commande de sauvegarde avec Ansible :

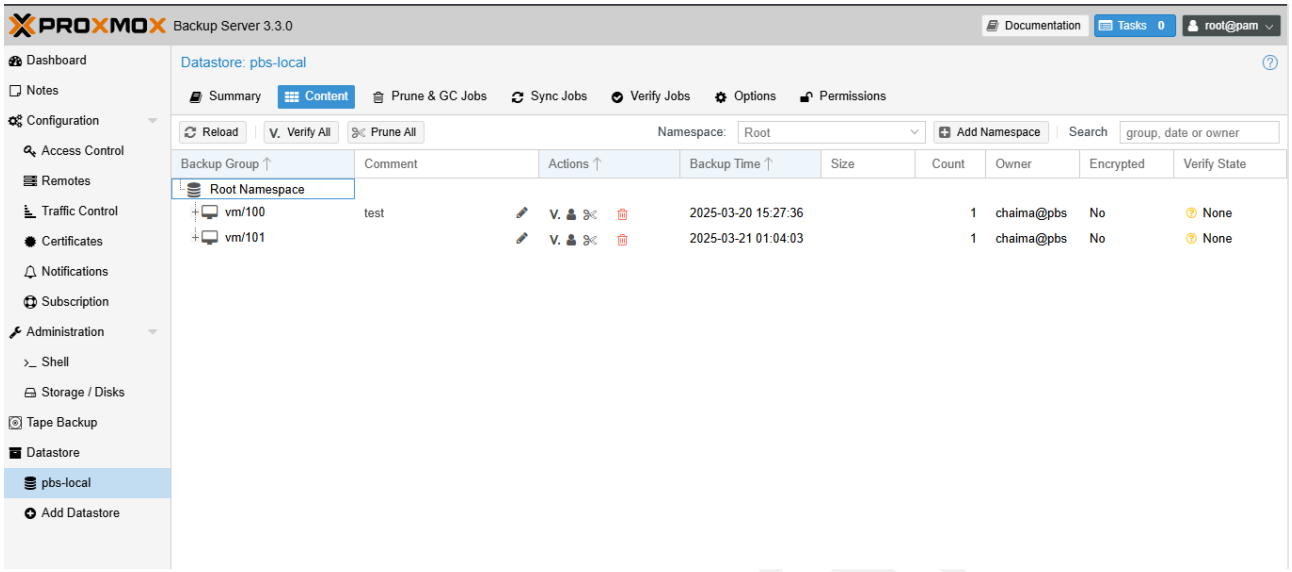
- `vzdump --all 1 --mode snapshot --storage pbs-local --compress zstd --maxfiles 5 --exclude /var/log`



1 Sauvegarde de toutes les VMs (option `--all`)

Si tu veux sauvegarder toutes les VMs sans spécifier leurs ID, tu peux utiliser l'option `--all` :

- `command: "vzdump --all 1 --mode snapshot --storage pbs-local --compress zstd"`
- `--all 1` : Sauvegarde toutes les VMs actives.



---

2 Mode de Sauvegarde (Options `--mode`)

Tu as plusieurs modes de sauvegarde :

- **`snapshot`** : Crée un snapshot de la VM sans l'arrêter (idéal pour la production).
- **`suspend`** : Suspends la VM avant de faire la sauvegarde.
- **`stop`** : Arrête la VM avant de la sauvegarder.

Exemple avec **`suspend`** :

```
"vzdump 101 --mode suspend --storage pbs-local --compress zstd"
```

---

3 Compression des Sauvegardes

Tu peux choisir différents formats de compression :

- **`zstd`** : Compression rapide et moderne.
- **`gzip`** : Compression standard.
- **`lzo`** : Compression plus rapide mais moins efficace.

Exemple avec **`gzip`** :

- `command: "vzdump 101 --mode snapshot --storage pbs-local --compress gzip"`

---

4 Limiter la Taille des Sauvegardes (Option `--maxfiles`)

Tu peux spécifier un nombre maximal de fichiers de sauvegarde à conserver pour éviter d'en accumuler trop :

- `command: "vzdump 101 --mode snapshot --storage pbs-local --compress zstd --maxfiles 5"`

Cela garde **5** fichiers de sauvegarde maximum et supprime les plus anciens.

---

5 Sauvegarde Incrémentielle (Option `--dumpdir`)

Tu peux également définir un répertoire spécifique pour chaque sauvegarde :

- `command: "vzdump 101 --mode snapshot --storage pbs-local --compress zstd --dumpdir /mnt/backupdir"`

---

6 Planification de la Sauvegarde avec Cron

Tu peux automatiser la sauvegarde via **Cron** sur Ansible pour planifier les sauvegardes chaque nuit, par exemple.

Ajoute cette tâche dans ton playbook :

```
- name: Planifier la sauvegarde quotidienne
  cron:
    name: "Sauvegarde quotidienne des VMs"
    minute: "0"
    hour: "2"
    job: "/usr/sbin/vzdump --all 1 --mode snapshot --storage pbs-local --compress zstd"
```

Cela exécutera la commande **tous les jours à 2h00 du matin**.

---

7 Exclure des Disques de la Sauvegarde (Option `--exclude` et `--exclude-path`)

Tu peux exclure certains disques ou chemins de la sauvegarde pour éviter de sauvegarder des fichiers non essentiels :

- `command: "vzdump 101 --mode snapshot --storage pbs-local --compress zstd --exclude /path/to/folder"`

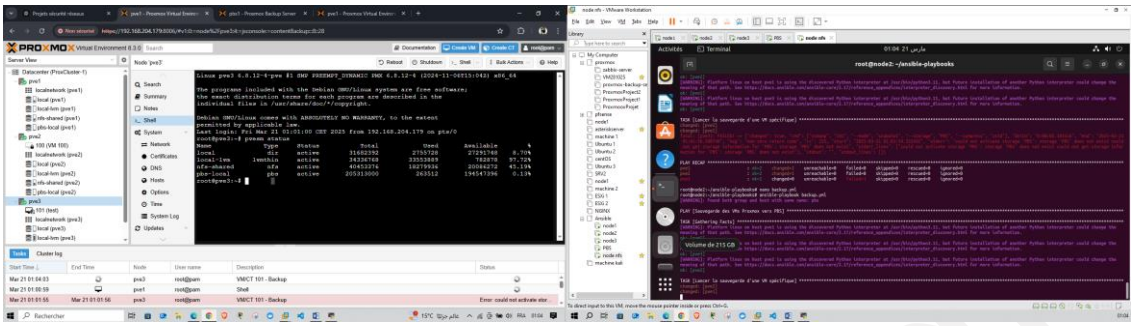
Exemple complet avec plusieurs options

Voici un exemple de playbook amélioré avec plusieurs options :

```
- name: Sauvegarde des VMs Proxmox vers PBS
  hosts: proxmox_nodes
  become: yes
  tasks:
    - name: Lancer la sauvegarde de toutes les VMs avec compression et conservation des fichiers
```

- `"vzdump --all 1 --mode snapshot --storage pbs-local --compress zstd --maxfiles 5 --exclude /var/log"`

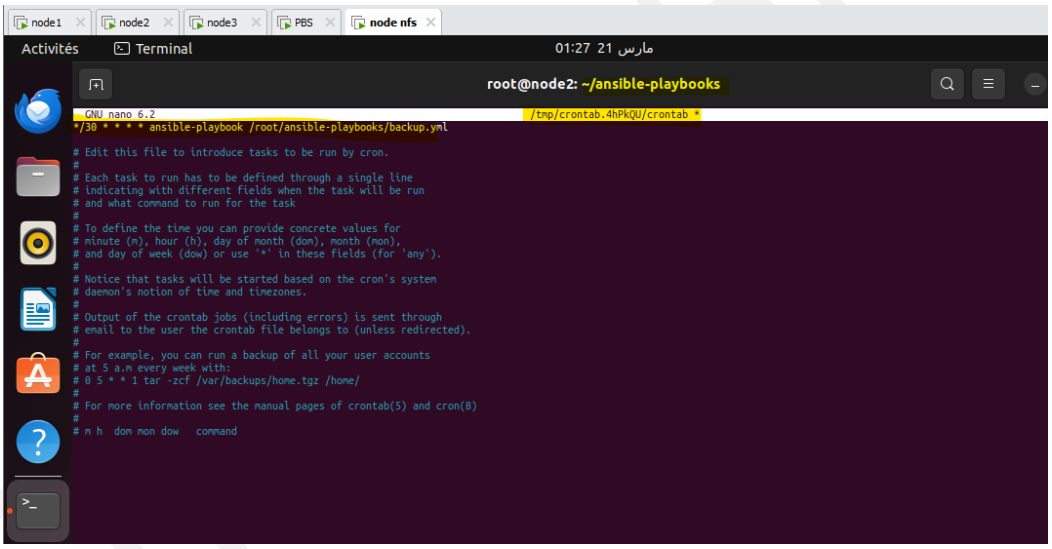
Cela devrait te donner une plus grande flexibilité pour automatiser et personnaliser tes sauvegardes ! Si tu veux tester d'autres options ou ajuster quelque chose, fais-le moi savoir !



Planification des Sauvegardes avec Cron :

Les sauvegardes sont planifiées pour s'exécuter automatiquement tous les jours à 30min, et peuvent être ajustées selon les besoins. Exemple de configuration Cron :

- `*/30 * * * ansible-playbook /home/chayma/ansible/backup.yml`



Cela permet de garantir que les sauvegardes sont effectuées régulièrement sans intervention manuelle.

Tests et Validation :

Des tests ont été effectués pour valider le bon fonctionnement du système de sauvegarde et de restauration. Le stockage NFS a été utilisé pour la migration à chaud des VMs, et la planification des sauvegardes via Cron a été testée avec succès.

Supervision avec Zabbix :

- Zabbix est intégré pour superviser les performances et l'état des sauvegardes ainsi que des machines virtuelles.
- La gestion et la configuration de Zabbix (installation, monitoring des VMs, alertes) sont automatisées et gérées via Ansible. Cela permet une supervision centralisée et efficace.

Un guide détaillé pour installer **Zabbix Server** sur **Proxmox Backup Server (PBS)** et **Zabbix Agent** sur les nœuds du cluster Proxmox, le tout automatisé avec **Ansible**.

Étape 1 : Préparation de l'Infrastructure

Avant de commencer, assurez-vous que les nœuds sont accessibles via SSH sans mot de passe (en utilisant des clés SSH) et que le serveur Ansible a accès à tous les nœuds et à PBS.

Étape 2 : Installation de Zabbix Server sur PBS via Ansible

1. Mise à jour des paquets et installation des dépendances

Nous allons commencer par configurer un playbook Ansible pour installer Zabbix Server sur PBS.

Playbook Ansible pour installer Zabbix Server :

```
---
- name: Installer Zabbix Server sur PBS
  hosts: pbs-server  # Remplacer par l'adresse de votre PBS dans l'inventaire
Ansible
  become: yes
  tasks:
    - name: Mise à jour des paquets
      apt:
        update_cache: yes
        upgrade: dist

    - name: Installer les dépendances nécessaires
      apt:
        name:
          - wget
          - gnupg
        state: present

    - name: Ajouter le dépôt Zabbix
      apt_repository:
        repo: 'deb http://repo.zabbix.com/zabbix/6.0/debian/stable main'
        state: present
        filename: zabbix

    - name: Ajouter la clé GPG pour le dépôt Zabbix
      apt_key:
        url: 'https://repo.zabbix.com/zabbix-official-repo.gpg'
        state: present

    - name: Installer Zabbix Server et le frontend
      apt:
        name:
          - zabbix-server-pgsql
          - zabbix-frontend-php
          - zabbix-apache-conf
        state: present

    - name: Installer le serveur de base de données PostgreSQL
      apt:
        name: postgresql
        state: present

    - name: Démarrer et activer le service Zabbix Server
      systemd:
        name: zabbix-server
        state: started
```

```
        enabled: yes
```

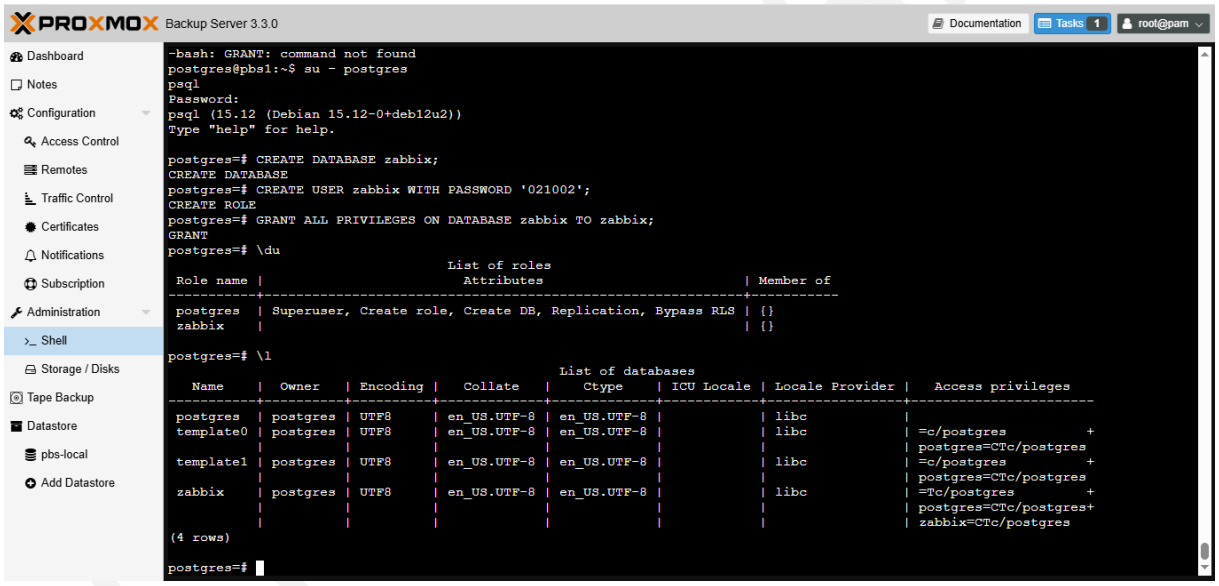
2. Configurer la base de données PostgreSQL

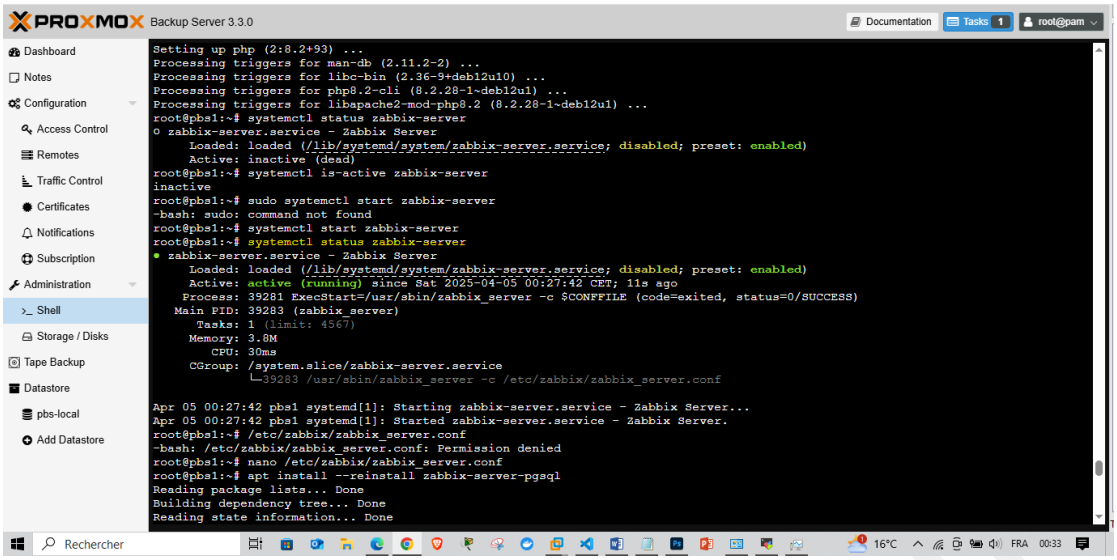
- Vous devez configurer la base de données PostgreSQL pour Zabbix. Voici un exemple de tâche Ansible pour créer la base de données Zabbix.

Playbook pour configurer PostgreSQL :

```
- name: Configurer PostgreSQL pour Zabbix
  hosts: pbs-server
  become: yes
  tasks:
    - name: Créer la base de données Zabbix
      postgresql_db:
        name: zabbix
        state: present
        owner: zabbix

    - name: Créer l'utilisateur Zabbix
      postgresql_user:
        name: zabbix
        password: 'your_password_here'  # Remplacer par un mot de passe sécurisé
        db: zabbix
        priv: 'ALL'
        state: present
```





3. Configurer Zabbix Server

Vous devez maintenant configurer le fichier `zabbix_server.conf`.

Exemple de tâche Ansible pour modifier la configuration :

```
- name: Configurer zabbix_server.conf
  hosts: pbs-server
  become: yes
  tasks:
    - name: Modifier la configuration de Zabbix pour se connecter à PostgreSQL
      lineinfile:
        path: /etc/zabbix/zabbix_server.conf
        regexp: '^# DBPassword='
        line: 'DBPassword=your_password_here'  # Utiliser le mot de passe de
l'utilisateur PostgreSQL
        state: present

    - name: Redémarrer le service Zabbix Server
      systemd:
        name: zabbix-server
        state: restarted
```

Étape 3 : Installation du Zabbix Agent sur les Nœuds Proxmox via Ansible

1. Installer le Zabbix Agent sur les Nœuds

Nous allons installer Zabbix Agent sur chaque nœud Proxmox pour permettre à Zabbix de surveiller les VMs.

Playbook Ansible pour installer Zabbix Agent sur les nœuds Proxmox :

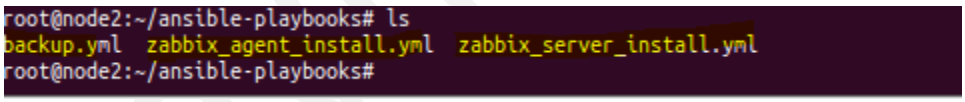
```
---
- name: Installer Zabbix Agent sur les nœuds Proxmox
  hosts: proxmox_nodes  # Remplacer par les adresses de vos nœuds dans
l'inventaire Ansible
  become: yes
  tasks:
    - name: Ajouter le dépôt Zabbix
      apt_repository:
        repo: 'deb http://repo.zabbix.com/zabbix/6.0/debian/stable main'
        state: present
        filename: zabbix
```

```
- name: Ajouter la clé GPG pour le dépôt Zabbix
  apt_key:
    url: 'https://repo.zabbix.com/zabbix-official-repo.gpg'
    state: present

- name: Installer Zabbix Agent
  apt:
    name: zabbix-agent
    state: present

- name: Configurer Zabbix Agent pour pointer vers Zabbix Server
  lineinfile:
    path: /etc/zabbix/zabbix_agentd.conf
    regexp: '^# Server='
    line: 'Server=pbs-server_ip'  # Remplacer par l'adresse IP de votre PBS
    state: present

- name: Redémarrer le service Zabbix Agent
  systemd:
    name: zabbix-agent
    state: restarted
    enabled: yes
```



2. Démarrer et Activer le Zabbix Agent

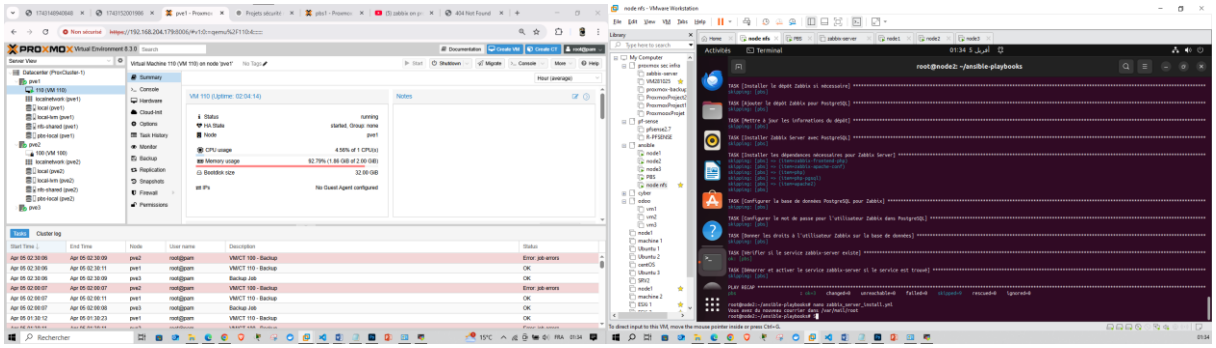
Le service Zabbix Agent sera démarré et configuré pour démarrer automatiquement au démarrage du système.

Étape 4 : Vérification et Tests

- **Vérification de l'installation de Zabbix Server et Agent :**
  - Après avoir exécuté les playbooks Ansible, vous pouvez vérifier l'état de Zabbix Server et des agents avec les commandes suivantes :
    - Vérifier le statut du serveur Zabbix :
    - `systemctl status zabbix-server`

Vérifier le statut de l'agent Zabbix sur chaque nœud :

- `systemctl status zabbix-agent`



- **Vérification via l'interface Web de Zabbix :**
  - Accédez à l'interface Web de Zabbix à l'adresse suivante :

`http://pbs-server_ip/zabbix`

et connectez-vous avec les identifiants administratifs.

  - Ajoutez les hôtes dans Zabbix pour chaque nœud Proxmox et vérifiez que les agents Zabbix sont bien enregistrés et communiquent avec le serveur.

Étape 5 : Configuration de la Supervision

- Une fois que Zabbix Server et les agents sont installés et que les nœuds sont ajoutés, vous pouvez configurer les **templates** et **éléments de surveillance** pour superviser les **machines virtuelles** et les **sauvegardes**.
  - Vous pouvez utiliser des templates prédéfinis pour Proxmox ou personnaliser les éléments de surveillance en fonction des besoins spécifiques de votre infrastructure.

Conclusion :

Ce projet a permis de mettre en place une solution robuste et efficace pour automatiser la gestion des sauvegardes des machines virtuelles sous Proxmox, en utilisant Ansible et Proxmox Backup Server (PBS). L'intégration de ces outils a permis de centraliser et d'automatiser les sauvegardes tout en garantissant une haute disponibilité et une reprise rapide après sinistre.

L'utilisation de NFS pour la migration à chaud des VMs a été un choix stratégique, permettant d'effectuer des migrations sans interruption de service, ce qui assure une continuité optimale des opérations. L'automatisation des sauvegardes via des playbooks Ansible a non seulement facilité la gestion des tâches, mais a également permis d'établir une planification flexible et adaptée aux besoins spécifiques de l'infrastructure.

De plus, l'intégration de Zabbix pour la supervision de l'ensemble du système a renforcé la surveillance en temps réel des sauvegardes et des performances des VMs. Grâce à Zabbix, les alertes et notifications permettent une réaction rapide en cas de problème, minimisant ainsi les risques de pertes de données.

Enfin, l'ensemble de cette infrastructure, géré via Ansible, offre une grande flexibilité et un contrôle total sur les tâches d'automatisation, tout en simplifiant la gestion des sauvegardes et la supervision. Ce projet a ainsi permis d'atteindre ses objectifs de haute disponibilité, de sécurité, et de gestion automatisée des sauvegardes, tout en réduisant les interventions manuelles et les erreurs humaines.

En conclusion, l'automatisation des sauvegardes et la supervision via Ansible et Zabbix constituent une avancée significative dans l'optimisation des processus de gestion des infrastructures virtuelles, contribuant à une gestion plus efficace, sécurisée et fiable.