

Mise à jour des Agents Wazuh

4.13.1 → 4.14.2

Auteur: Andrea hardy

LinkedIn: <https://www.linkedin.com/in/andrea-h-702a97188/>

Date: 19/01/2026



Qu'est-ce que Wazuh et pourquoi est-il essentiel ?

Wazuh est une plateforme de sécurité open source qui protège vos serveurs en temps réel. C'est comme avoir un gardien numérique qui surveille en permanence vos systèmes pour détecter les menaces, les intrusions et les comportements anormaux.

Pourquoi Wazuh est crucial pour votre infrastructure ?

- Détection des intrusions
- Conformité réglementaire
- Analyse des logs
- Alertes intelligentes
- Détection de malwares
- Surveillance de l'intégrité des fichiers

Mise à jour des Agents Wazuh 4.13.1 → 4.14.2

Inventaire des Serveurs

Serveurs à mettre à jour (4.13.1 → 4.14.2)

Hostname	IP	OS	Agent ID	Version actuelle
DNS-Slave-ct	10.3.20.54	Debian 13	002	4.13.1
teleport	10.3.30.7	Debian 12	004	4.13.1
WIN-THP53NH297I	10.3.30.200	Windows Server 2022	005	4.13.1
HAProxy-DB	10.3.30.90	Debian 12	006	4.13.1
Wordpress2	10.3.30.85	Debian 12	007	4.13.1
zabbix	10.3.30.19	Debian 13	008	4.13.1

Serveur déjà à jour

Hostname	IP	OS	Version
DNS	10.3.20.53	Debian 13	4.14.2 

Prérequis

Sur votre serveur Terraform/Ansible

1. **Ansible installé** (version 2.9 ou supérieure)

```
ansible --version
```

2. **Accès SSH configuré** vers tous les serveurs Linux

3. **Accès WinRM** pour le serveur Windows

4. **Connectivité réseau**

5. **Modules Python requis**

```
apt install -y python3-winrm
```

Structure du Projet Ansible

```
wazuh-upgrade/  
├─ inventory/  
│   └─ hosts.yml  
├─ playbooks/  
│   ├── upgrade-linux.yml  
│   ├── upgrade-windows.yml  
│   └─ report.yml  
├─ group_vars/  
│   ├── all.yml  
│   └─ windows.yml  
└─ ansible.cfg
```

Fichiers de Configuration

1. `inventory/hosts.yml`

```
all:
  children:
    wazuh_agents_linux:
      hosts:
        dns-slave-ct:
          ansible_host: 10.3.20.54
          ansible_user: root
          agent_id: "002"

        teleport:
          ansible_host: 10.3.30.7
          ansible_user: root
          agent_id: "004"

        haproxy-db:
          ansible_host: 10.3.30.90
          ansible_user: root
          agent_id: "006"

        wordpress2:
          ansible_host: 10.3.30.85
          ansible_user: root
          agent_id: "007"

        zabbix:
          ansible_host: 10.3.30.19
          ansible_user: root
          agent_id: "008"

      vars:
        ansible_python_interpreter: /usr/bin/python3
        ansible_become: yes
        ansible_become_method: sudo

    wazuh_agents_windows:
      hosts:
        win-server:
          ansible_host: 10.3.30.200
          agent_id: "005"

      vars:
        ansible_user: Administrateur
```



```
ansible_password: "AndrePaul87" # À sécuriser avec ansible-vault
ansible_connection: winrm
ansible_winrm_server_cert_validation: ignore
ansible_winrm_transport: ntlm
ansible_port: 5985
ansible_become: false

vars:
  wazuh_version: "4.14.2"
  wazuh_manager_ip: "10.3.20.92"
```

2. `ansible.cfg`

```
[defaults]
inventory = inventory/hosts.yml
host_key_checking = False
timeout = 30
gathering = smart
fact_caching = jsonfile
fact_caching_connection = /tmp/ansible_facts
fact_caching_timeout = 3600
retry_files_enabled = False

[privilege_escalation]
become = True
become_method = sudo
become_user = root
become_ask_pass = False

[ssh_connection]
ssh_args = -o ControlMaster=auto -o ControlPersist=60s -o StrictHostKeyChecking=no
pipelining = True

[vault]
vault_password_file = ~/.vault_pass
```

3. `group_vars/all.yml`

```
wazuh_target_version: "4.14.2"
wazuh_current_version: "4.13.1"
wazuh_repo_version: "4.x"

wazuh_manager_ip: "10.3.20.92"
wazuh_manager_port: 1514

backup_enabled: true
backup_dir: "/var/backups/wazuh"
service_timeout: 120
```

4. `group_vars/windows.yml`

```
wazuh_windows_download_url: "https://packages.wazuh.com/4.x/windows/wazuh-agent-{{ wa
wazuh_windows_install_path: "C:\\Program Files (x86)\\ossec-agent"
wazuh_windows_temp_path: "C:\\Temp"
```

Playbooks Ansible

`playbooks/upgrade-linux.yml`

Correction : `backup_dir` existe, on télécharge le `.deb` via `wget`, puis installation via `dpkg` (car `apt` ne trouve pas le paquet)

```
- name: Mise a jour des agents Wazuh Linux vers 4.14.2
hosts: wazuh_agents_linux
become: yes
serial: 1

vars:
  wazuh_target_version: "4.14.2-1"
  backup_dir: "/var/backups/wazuh"
  wazuh_deb_url: "https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-
  wazuh_deb_local: "/tmp/wazuh-agent_{{ wazuh_target_version }}_amd64.deb"

pre_tasks:
  - name: Verification de la version actuelle
    shell: |
      set -o pipefail
      /var/ossec/bin/wazuh-control info | grep WAZUH_VERSION | cut -d'"' -f2
    register: current_version
    changed_when: false
    failed_when: false

  - name: Affichage de la version actuelle
    debug:
      msg: "Version actuelle : {{ current_version.stdout }}"

  - name: Skip si deja a jour
    meta: end_host
    when: current_version.stdout is search(wazuh_target_version.split('-')[0])

tasks:
  - name: Creation du repertoire de backup
    file:
      path: "{{ backup_dir }}"
      state: directory
      mode: '0755'
      check_mode: no

  - name: Sauvegarde de la configuration actuelle
    copy:
      src: /var/ossec/etc/ossec.conf
      dest: "{{ backup_dir }}/ossec.conf.{{ ansible_date_time.epoch }}"
      remote_src: yes

  - name: Installer wget si absent
    apt:
      name: wget
      state: present
```

```

    when: ansible_os_family == "Debian"

- name: Telechargement du paquet Wazuh 4.14.2
  shell: |
    wget -O "{{ wazuh_deb_local }}" "{{ wazuh_deb_url }}"
  args:
    creates: "{{ wazuh_deb_local }}"

- name: Verification que le deb existe
  stat:
    path: "{{ wazuh_deb_local }}"
  register: deb_stat

- name: Installation du paquet Wazuh via dpkg
  apt:
    deb: "{{ wazuh_deb_local }}"
    state: present
  when: deb_stat.stat.exists

- name: Redemarrage du service Wazuh
  systemd:
    name: wazuh-agent
    state: restarted
    enabled: yes

- name: Verification de la nouvelle version
  shell: |
    set -o pipefail
    /var/ossec/bin/wazuh-control info | grep WAZUH_VERSION | cut -d'"' -f2
  register: new_version
  changed_when: false

- name: Verification du statut du service
  systemd:
    name: wazuh-agent
  register: service_status

- name: Resultat de la mise a jour
  debug:
    msg:
      - " Mise a jour reussie !"
      - "Version precedente : {{ current_version.stdout }}"
      - "Version actuelle : {{ new_version.stdout }}"
      - "Service actif : {{ service_status.status.ActiveState }}"

post_tasks:
  - name: Envoi d'une notification de succes

```

```
debug:
  msg: " {{ inventory_hostname }} mis a jour avec succes vers Wazuh {{ new_vers
```

playbooks/upgrade-windows.yml

Correction : le dossier `C:\Windows\Temp` doit exister, puis téléchargement MSI, puis installation

```
- name: Mise a jour de l'agent Wazuh Windows vers 4.14.2
hosts: wazuh_agents_windows
gather_facts: yes

vars:
  wazuh_version: "4.14.2"
  temp_path: "C:\\Windows\\Temp"
  wazuh_msi_url: "https://packages.wazuh.com/4.x/windows/wazuh-agent-{{ wazuh_version }}-1.msi"
  wazuh_msi_path: "{{ temp_path }}\\wazuh-agent-{{ wazuh_version }}-1.msi"

tasks:
  - name: Creation du dossier temporaire
    win_file:
      path: "{{ temp_path }}"
      state: directory

  - name: Telechargement du MSI Wazuh 4.14.2
    win_get_url:
      url: "{{ wazuh_msi_url }}"
      dest: "{{ wazuh_msi_path }}"
      force: yes
    # IMPORTANT : on ne veut pas télécharger en check mode
    when: not ansible_check_mode

  - name: Verifier que le MSI existe
    win_stat:
      path: "{{ wazuh_msi_path }}"
    register: msi_stat

  - name: Afficher le resultat du telechargement
    debug:
      msg:
        - "MSI existe : {{ msi_stat.stat.exists }}"
        - "Taille : {{ msi_stat.stat.size | default('N/A') }}"

  - name: Installation de Wazuh 4.14.2
    win_shell: |
      msixexec.exe /i "{{ wazuh_msi_path }}" /qn /norestart
    args:
      creates: "{{ wazuh_msi_path }}"
    when: not ansible_check_mode
```

playbooks/report.yml

```
- name: Generation du rapport de la mise a jour
  hosts: all
  gather_facts: yes

  tasks:
    - name: Version Wazuh (Linux)
      shell: /var/ossec/bin/wazuh-control info | grep WAZUH_VERSION | cut -d'"' -f2
      register: version_linux
      when: ansible_os_family == "Debian"
      changed_when: false

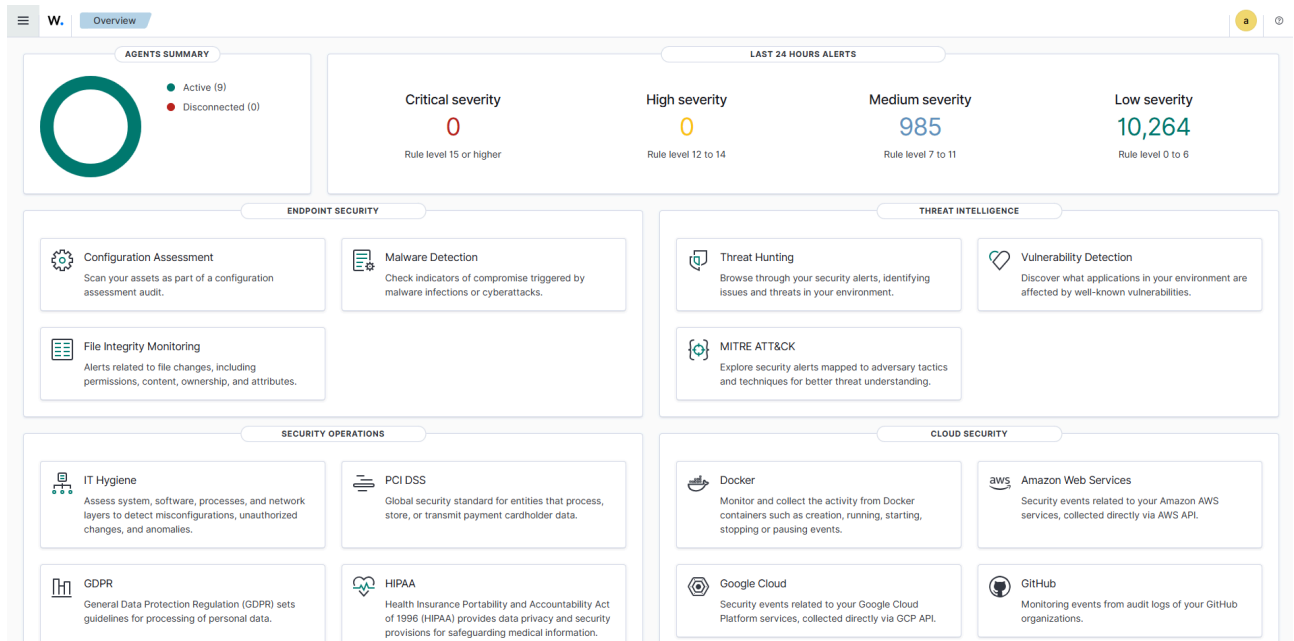
    - name: Version Wazuh (Windows)
      win_shell: Get-Content "C:\Program Files (x86)\ossec-agent\VERSION"
      register: version_windows
      when: ansible_os_family == "Windows"
      changed_when: false

    - name: Affichage du rapport
      debug:
        msg:
          - "Serveur : {{ inventory_hostname }}"
          - "IP : {{ ansible_host }}"
          - "OS : {{ ansible_distribution }} {{ ansible_distribution_version | default }}"
          - "Version Wazuh : {{ version_linux.stdout | default(version_windows.stdout) }}"
          - "Statut : ACTIF"
```

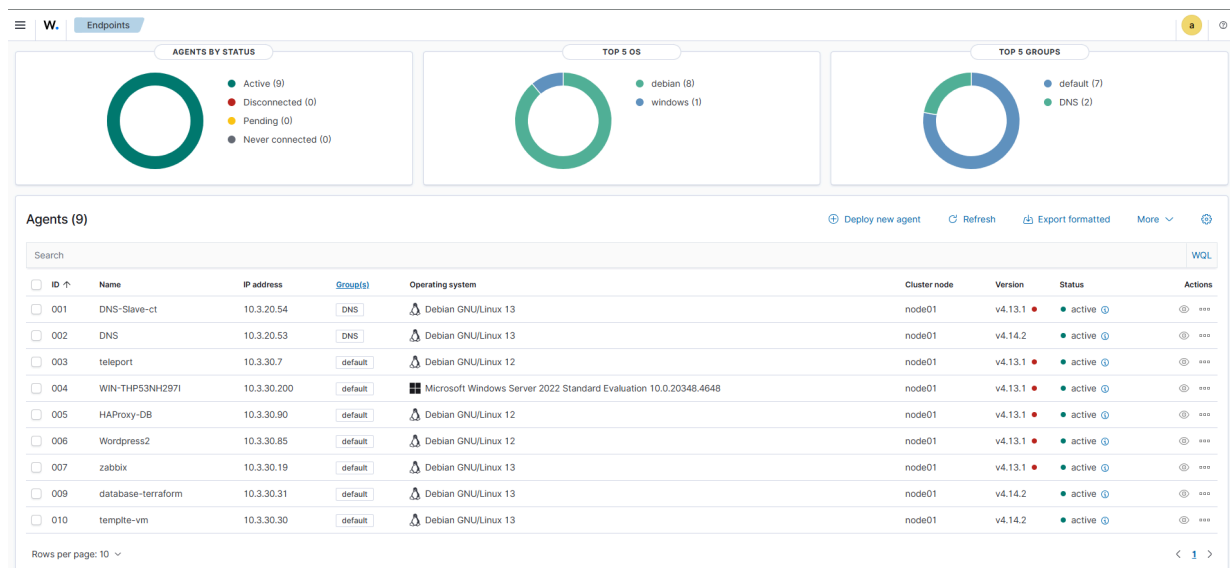
Procédure d'Exécution

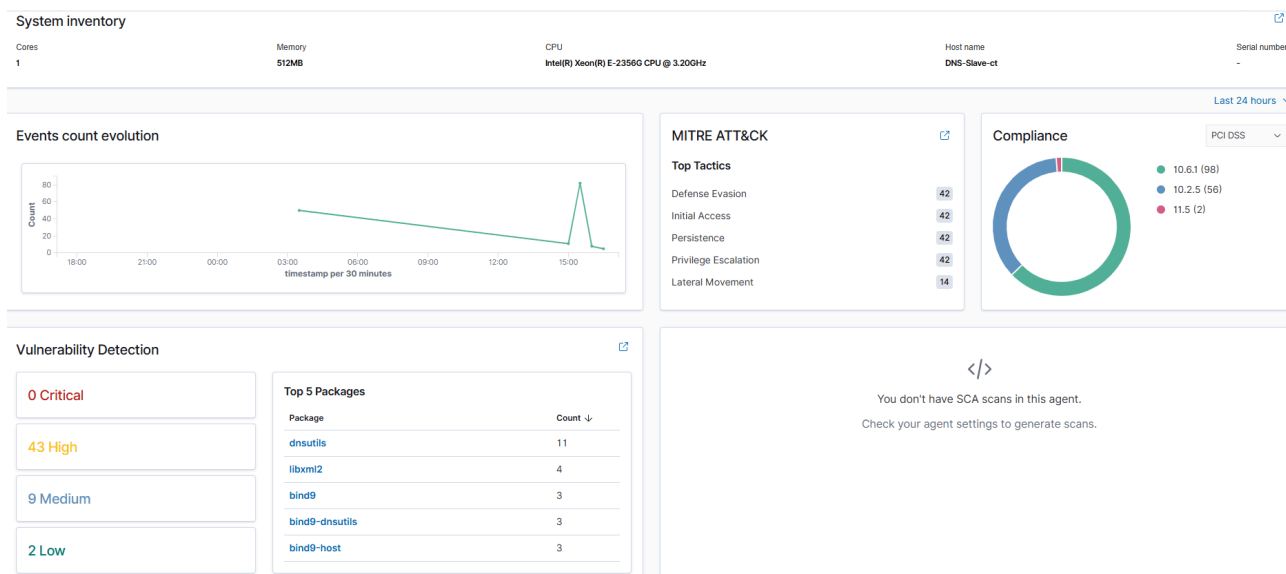
1) Dashboard Wazuh

- Dashboard Wazuh



- Agent détail





2) Vérifications avant exécution

Linux

1. Statut Wazuh

```
systemctl status wazuh-agent
```

```
root@teleport:~# systemctl status wazuh-agent
* wazuh-agent.service - Wazuh agent
   Loaded: loaded (/lib/systemd/system/wazuh-agent.service; enabled; preset: enabled)
   Active: active (running) since Mon 2026-01-19 14:05:12 UTC; 2h 7min ago
   Process: 43922 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited, status=0/SUCCESS)
   Tasks: 31 (limit: 76562)
   Memory: 55.0M
   CPU: 5.939s
   CGroup: /system.slice/wazuh-agent.service
           └─43944 /var/ossec/bin/wazuh-execd
             └─43952 /var/ossec/bin/wazuh-agentd
               └─43966 /var/ossec/bin/wazuh-syscheckd
                 └─43978 /var/ossec/bin/wazuh-logcollector
                   └─43996 /var/ossec/bin/wazuh-modulesd

Jan 19 14:05:08 teleport systemd[1]: Starting wazuh-agent.service - Wazuh agent...
Jan 19 14:05:08 teleport env[43922]: Starting Wazuh v4.13.1...
Jan 19 14:05:08 teleport env[43922]: Started wazuh-execd...
Jan 19 14:05:09 teleport env[43922]: Started wazuh-agentd...
Jan 19 14:05:09 teleport env[43922]: Started wazuh-syscheckd...
Jan 19 14:05:10 teleport env[43922]: Started wazuh-logcollector...
Jan 19 14:05:10 teleport env[43922]: Started wazuh-modulesd...
Jan 19 14:05:12 teleport env[43922]: Completed.
Jan 19 14:05:12 teleport systemd[1]: Started wazuh-agent.service - Wazuh agent.
root@teleport:~#
```

1. Vérification réseau / SSH

```
ssh root@10.3.20.54
```

Windows

1. WinRM

```
winrm quickconfig
```

```
beaupeyrat\administrateur@WIN-THP53NH297I C:\Users\Administrateur\.ssh>winrm quickconfig
Le service WinRM est déjà en cours d'exécution sur cet ordinateur.
WinRM est déjà configuré pour la gestion à distance sur cet ordinateur.
```

3) Tests Ansible (ping)

Linux

```
ansible wazuh_agents_linux -m ping
```

Windows

```
ansible wazuh_agents_windows -m win_ping
```

```
root@terraform:~/wazuh-upgrade# ansible wazuh_agents_linux -m ping
teleport | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
haproxy-db | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
wordpress2 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
zabbix | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
dns-slave-ct | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
root@terraform:~/wazuh-upgrade# ansible wazuh_agents_windows -m win_ping
win-server | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
root@terraform:~/wazuh-upgrade# |
```

4) Lancement des playbooks

Linux

```
ansible-playbook playbooks/upgrade-linux.yml #pour exécuter les taches
```

```

TASK [Verification de la nouvelle version] *****
skipping: [zabbix]

TASK [Verification du statut du service] *****
ok: [zabbix]

TASK [Resultat de la mise a jour] *****
ok: [zabbix] => {
  "msg": [
    "✅ Mise a jour reussie !",
    "Version precedente : ",
    "Version actuelle : ",
    "Service actif : active"
  ]
}

TASK [Envoi d'une notification de succes] *****
ok: [zabbix] => {
  "msg": "🚀 zabbix mis a jour avec succes vers Wazuh "
}

PLAY RECAP *****
dns-slave-ct      : ok=11  changed=3  unreachable=0  failed=0  skipped=3  rescued=0  ignored=0
haproxy-db        : ok=11  changed=3  unreachable=0  failed=0  skipped=3  rescued=0  ignored=0
teleport          : ok=11  changed=3  unreachable=0  failed=0  skipped=3  rescued=0  ignored=0
wordpress2        : ok=11  changed=3  unreachable=0  failed=0  skipped=3  rescued=0  ignored=0
zabbix            : ok=11  changed=3  unreachable=0  failed=0  skipped=3  rescued=0  ignored=0

```

Windows

```
ansible-playbook playbooks/upgrade-windows.yml
```

```

root@terraform:~/wazuh-upgrade# ansible-playbook playbooks/upgrade-windows.yml

PLAY [Mise a jour de l'agent Wazuh Windows vers 4.14.2] *****

TASK [Gathering Facts] *****
ok: [win-server]

TASK [Creation du dossier temporaire] *****
ok: [win-server]

TASK [Telechargement du MSI Wazuh 4.14.2] *****
changed: [win-server]

TASK [Verifier que le MSI existe] *****
ok: [win-server]

TASK [Afficher le resultat du telechargement] *****
ok: [win-server] => {
  "msg": [
    "MSI existe : True",
    "Taille : 5939200"
  ]
}

TASK [Installation de Wazuh 4.14.2] *****
skipping: [win-server]

PLAY RECAP *****
win-server        : ok=5    changed=1  unreachable=0  failed=0  skipped=1  rescued=0  ignored=0

root@terraform:~/wazuh-upgrade#

```

Vérifications Post-Mise à Jour

Sur chaque serveur Linux

```
/var/ossec/bin/wazuh-control info  
systemctl status wazuh-agent  
tail -f /var/ossec/logs/ossec.log
```

```
root@teleport:~# /var/ossec/bin/wazuh-control info  
WAZUH_VERSION="v4.13.1"  
WAZUH_REVISION="rc1"  
WAZUH_TYPE="agent"  
root@teleport:~# /var/ossec/bin/wazuh-control info  
WAZUH_VERSION="v4.14.2"  
WAZUH_REVISION="rc4"  
WAZUH_TYPE="agent"  
root@teleport:~# |
```

Sur le serveur Windows

```
Get-Service WazuhSvc  
Get-Content "C:\Program Files (x86)\ossec-agent\VERSION"  
Get-Content "C:\Program Files (x86)\ossec-agent\ossec.log" -Tail 50
```

Sur le Manager Wazuh

```
/var/ossec/bin/agent_control -l
```

```
root@Lynis-Wazuh:~# /var/ossec/bin/agent_control -l

Wazuh agent_control. List of available agents:
  ID: 000, Name: Lynis-Wazuh (server), IP: 127.0.0.1, Active/Local
  ID: 001, Name: DNS-Slave-ct, IP: any, Active
  ID: 002, Name: DNS, IP: any, Active
  ID: 003, Name: teleport, IP: any, Active
  ID: 004, Name: WIN-THP53NH297I, IP: any, Active
  ID: 005, Name: HAProxy-DB, IP: 10.3.30.90, Active
  ID: 006, Name: Wordpress2, IP: 10.3.30.85, Active
  ID: 007, Name: zabbix, IP: 10.3.30.19, Active
  ID: 009, Name: database-terraform, IP: 10.3.30.31, Active
  ID: 010, Name: templte-vm, IP: any, Active

List of agentless devices:

root@Lynis-Wazuh:~# |
```

Dépannage (problèmes courants)

Agent ne se reconnecte pas

Linux :

```
systemctl restart wazuh-agent
journalctl -u wazuh-agent -n 50
```

Windows :

```
Restart-Service WazuhSvc
Get-Service WazuhSvc
```

Sécurité des Credentials

Ne jamais commiter les mots de passe en clair !

```
ansible-vault encrypt group_vars/windows.yml
```

Planning de Mise à Jour

PLANNING DE MISE À JOUR

Planification des mises à jour des serveurs sur 5 jours

 Jour	 Serveurs	 Horaire
Jour 1	HAProxy-DB (test)	17h00
Jour 2	DNS-Slave-ct, DNS	17h00
Jour 3	Wordpress2, Zabbix	17h00
Jour 4	Teleport	17h00
Jour 5	WIN-THP53NH297I	17h00

Conclusion

Cette documentation permet une mise à jour automatisée, sécurisée et reproductible.
Ansible garantit la cohérence et la répétabilité.

Risques & Mitigation

RISQUES & MITIGATION

Gestion des risques potentiels lors des mises à jour des serveurs

Risque 	Impact 	Mitigation 
Agent ne redémarre pas	Perte de monitoring	Rollback + Restart
WinRM non configuré	Blocage Windows	Vérification WinRM + Configuration
Paquet non disponible	Échec installation	Téléchargement direct .deb

Suivi du projet – Outil de gestion des tâches (Noteia)

Vue d'ensemble du projet

Cette capture d'écran présente le **tableau de suivi du projet "Mise à jour Agents Wazuh 4.13.1 → 4.14.2"** réalisé à l'aide de l'outil **Noteia**.

Elle permet de visualiser l'état d'avancement global du projet ainsi que le statut de chaque phase.

Le projet est actuellement avancé à **75 %**, avec **3 phases terminées sur 4**.

Mise à jour Agents Wazuh 4.13.1 → 4.14.2

75% (3/4)

Tâche	Responsable	Priorité	Statut
Préparation 2/2	Andréa	Haute	Terminé
srv-dns Mise à jour Linux 2/2	-	Haute	Terminé
win-10.3.30.2... Mise à jour Windows* 2/2	-	Haute	Terminé
Vérification 0/2	-	Haute	À faire

Détail des tâches

Le projet est découpé en **4 grandes phases**, chacune correspondant à une étape clé de la mise à jour.

1. Préparation

- **Responsable** : Andréa
- **Priorité** : Haute
- **Statut** : Terminé
- **Progression** : 2/2 tâches complétées

Cette phase comprend :

- La configuration de l'environnement Ansible
- La vérification des accès SSH (Linux) et WinRM (Windows)
- Les tests de connectivité avant déploiement

2. Mise à jour Linux

- **Cible** : srv-dns
- **Priorité** : Haute
- **Statut** : Terminé
- **Progression** : 2/2 tâches complétées

Cette étape correspond à :

- La mise à jour des agents Wazuh sur les serveurs Linux
- L'exécution des playbooks Ansible
- La vérification du bon fonctionnement des services après mise à jour

3. Mise à jour Windows

- **Cible** : win-10.3.30.2...
- **Priorité** : Haute
- **Statut** : Terminé
- **Progression** : 2/2 tâches complétées

Cette phase inclut :

- La préparation de l'environnement WinRM
- Le déploiement de l'agent Wazuh via MSI
- La validation de l'installation sur les machines Windows

4. Vérification

- **Priorité** : Haute
- **Statut** : À faire
- **Progression** : 0/2 tâches complétées

Cette dernière phase concerne :

- La vérification des agents depuis le **tableau de bord Wazuh**
- La confirmation que tous les agents sont actifs et à jour
- La collecte des captures d'écran finales pour la documentation

Conclusion

Ce tableau de suivi montre une **gestion de projet structurée**, avec :

- Une priorisation claire des tâches
- Un suivi précis de l'avancement

- Une séparation logique des phases (préparation, déploiement, vérification)

Il permet de garantir une **mise à jour maîtrisée, progressive et traçable** des agents Wazuh.