REPUBLIQUE DU SENEGAL

Un peuple-un but-une foi

Ministère de l'Enseignement Supérieur, de la Recherche et de l'Innovation

Direction de l'Enseignement Supérieur Privé

INSTITUT SUPERIEURE D'INFORMATIQUE

GROUPE ISI

Grade : Licence 3

Filière : Réseaux et Systèmes Informatiques

Installation et Configuration de Wazuh sur Debian 12

Présenté par :

M. Bakary DIAKHITE

Sous la direction de :

M. Massamba LÔ

Année Académique : 2025 -2026

wazuh.

Table des matières

wazuh.

Installation et configuration de Wazuh

Introduction

Wazuh est une solution open-source de détection d'intrusion (HIDS), de surveillance de l'intégrité des fichiers, de détection de malwares et de réponse aux incidents. Il permet aussi l'analyse des logs et la gestion centralisée des agents sur plusieurs systèmes. Ce rapport présente l'installation complète de Wazuh sur un système Debian, depuis la préparation de l'environnement jusqu'à la configuration du tableau de bord et l'ajout d'agents.

1 Paramétrage de base

Sur cette partie nous allons fixer les paramètres ip de notre interfaces (ens34) à savoir l'addresse ip, la passerelle par défaut, le serveur Dns, etc

- **Editer le fichier de configuration**

```
root@etudiant:~# vim /etc/network/interfaces
```

- **Fixer les paramètres ip du serveur**

```
# The primary network interface
allow-hotplug ens33
iface ens33 inet dhcp

allow-hotplug ens34
iface ens34 inet static
address 192.168.10.1
netmask 255.255.255.0
gateway 192.168.10.1
dns-nameservers 192.168.10.1
search badiakhite.sn
~
```

Service déjà installer et configurer sur le serveur

- **ssh** (pour les connections à distance)
- **dhcp** (pour attribuer dynamiquement les paramètres ip aux clients)

2 Installation des prérequis

Avant de commencer l'installation de Wazuh, il est indispensable de s'assurer que le système dispose des paquets de base nécessaires.

apt -y install debconf adduser procps curl gnupg apt-transport-https filebeat debhelper libcap2-bin curl gpg

```
root@etudiant:~#
root@etudiant:~# apt -y install debconf adduser procps curl gnupg apt-transport-https filebeat debhelper libcap2-bin curl gpg
```

- **Importation de la clé GPG wazuh**

Cette commande sécurise le dépôt Wazuh via une **clé GPG**, utilisée pour vérifier que les paquets téléchargés proviennent bien du dépôt officiel Wazuh et n'ont pas été modifiés.

curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg --import && chmod 644 /usr/share/keyrings/wazuh.gpg

```
root@etudiant:~# curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --no-default-keyring --keyr
ing gnupg-ring:/usr/share/keyrings/wazuh.gpg --import && chmod 644 /usr/share/keyrings/wazuh.gpg
```

- **Ajout du dépôt wazuh**

Cette commande ajoute le dépôt officiel de Wazuh dans la liste des sources apt, afin que le système puisse ensuite installer les paquets **wazuh-manager, wazuh-dashboard, etc.**

echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main" | tee -a /etc/apt/sources.list.d/wazuh.list

```
root@etudiant:~# echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt
/ stable main" | tee -a /etc/apt/sources.list.d/wazuh.list
deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main
root@etudiant:~#
```

- **Mettre à jour les paquets**

cette étape est recommandée avant toute installation de nouveaux logiciels afin d'éviter les conflits ou les dépendances obsolètes.

```
root@etudiant:~# apt -y update && apt -y  upgrade
```

- **Téléchargement du script en wazuh-certs-tool.sh et le fichier de figuration config.yml**

  - **wazuh-certs-tool.sh** : un script fourni par Wazuh permettant de générer les certificats TLS nécessaires à la communication sécurisée entre les différents composants (indexer, manager, dashboard).

- **config.yml** : un fichier de configuration utilisé par le script pour générer les certificats selon les paramètres définis (noms d'hôtes, IP, etc.).

curl -sO https://packages.wazuh.com/4.12/wazuh-certs-tool.sh && curl -sO https://packages.wazuh.com/4.12/config.yml

```
root@etudiant:~#
root@etudiant:~# curl -sO https://packages.wazuh.com/4.12/wazuh-certs-tool.sh && curl -sO https://packages.wazuh.com/4.12/config.yml
```

➢ **Modification du fichier de configuration ./config.yml**

```
root@etudiant:~#
root@etudiant:~# vim ./config.yml
```

- Définir l'addresse ip de l'indexeur
- Définir l'addresse ip du server Wazuh
- Définir l'addresse du tableau de bord

```
nodes:
  # Wazuh indexer nodes
  indexer:
    - name: node-1
      ip: "192.168.10.1"
    #- name: node-2
    #  ip: "<indexer-node-ip>"
    #- name: node-3
    #  ip: "<indexer-node-ip>"

  # Wazuh server nodes
  # If there is more than one Wazuh server
  # node, each one must have a node_type
  server:
    - name: wazuh-1
      ip: "192.168.10.1"
    #  node_type: master
    #- name: wazuh-2
    #  ip: "<wazuh-manager-ip>"
    #  node_type: worker
    #- name: wazuh-3
    #  ip: "<wazuh-manager-ip>"
    #  node_type: worker

  # Wazuh dashboard nodes
  dashboard:
    - name: dashboard
      ip: "192.168.10.1"
~
```

➢ **Génération et archivage des certificats avec wazuh-certs-tool.sh**

Une fois le fichier **config.yml** correctement configuré, on exécute le script suivant pour générer automatiquement les certificats nécessaires à la sécurisation des communications entre les composants Wazuh.

```
root@etudiant:~# bash ./wazuh-certs-tool.sh -A
23/06/2025 21:14:37 INFO: Verbose logging redirected to /root/wazuh-certificates-tool.log
23/06/2025 21:14:37 INFO: Generating the root certificate.
23/06/2025 21:14:37 INFO: Generating Admin certificates.
23/06/2025 21:14:37 INFO: Admin certificates created.
23/06/2025 21:14:38 INFO: Generating Wazuh indexer certificates.
23/06/2025 21:14:38 INFO: Wazuh indexer certificates created.
23/06/2025 21:14:38 INFO: Generating Filebeat certificates.
23/06/2025 21:14:38 INFO: Wazuh Filebeat certificates created.
23/06/2025 21:14:38 INFO: Generating Wazuh dashboard certificates.
23/06/2025 21:14:38 INFO: Wazuh dashboard certificates created.
root@etudiant:~#
```

Afin de faciliter la sauvegarde de ces certificats, il est recommandé de les compresser dans une archive tar. La commande suivante permet de créer une archive nommée **wazuh-certificates.tar** à partir du dossier **wazuh-certificates**

```
root@etudiant:~# tar -cvf ./wazuh-certificates.tar -C ./wazuh-certificates/ .
./
./root-ca.key
./wazuh-1-key.pem
./dashboard-key.pem
./admin-key.pem
./root-ca.pem
./node-1.pem
./admin.pem
./wazuh-1.pem
./dashboard.pem
./node-1-key.pem
root@etudiant:~#
```

Après avoir créé et archivé les certificats, on peut supprimer le dossier source pour libérer de l'espace ou éviter une duplication inutile des fichiers

```
root@etudiant:~# rm -rf ./wazuh-certificates
root@etudiant:~#
```

3 Installation des paquets Wazuh

L'étape suivante consiste à installer les trois composants principaux de la solution Wazuh à l'aide de la commande **apt**

- **wazuh-indexer** : moteur d'indexation basé sur OpenSearch (anciennement Elasticsearch).
- **wazuh-manager** : cœur du système Wazuh, chargé de collecter et traiter les données des agents.
- **wazuh-dashboard** : interface web permettant de visualiser les alertes, événements et rapports de sécurité.

  apt -y install wazuh-indexer wazuh-manager wazuh-dashboard

```
root@etudiant:~# apt -y install wazuh-indexer wazuh-manager wazuh-dashboard
```

Une fois les paquets installés, il est nécessaire de configurer le service **wazuh-indexer** en modifiant son fichier de configuration.

```
root@etudiant:~# vim /etc/wazuh-indexer/opensearch.yml
```

Dans ce fichier, il faut :

Renseigner **l'adresse IP** du serveur sur lequel l'indexer fonctionne.

Définir un **nom unique pour l'indexer**.

Vérifier **les chemins vers les certificats TLS**.

S'assurer que la configuration reflète bien les paramètres définis dans le fichier **config.yml** utilisé précédemment.

```
network.host: "192.168.10.1"
node.name: "node-1"
cluster.initial_master_nodes:
- "node-1"
#- "node-2"
#- "node-3"
cluster.name: "wazuh-cluster"
#discovery.seed_hosts:
#  - "node-1-ip"
#  - "node-2-ip"
#  - "node-3-ip"
node.max_local_storage_nodes: "3"
path.data: /var/lib/wazuh-indexer
path.logs: /var/log/wazuh-indexer

plugins.security.ssl.http.pemcert_filepath: /etc/wazuh-indexer/certs/indexer.pem
plugins.security.ssl.http.pemkey_filepath: /etc/wazuh-indexer/certs/indexer-key.pem
plugins.security.ssl.http.pemtrustedcas_filepath: /etc/wazuh-indexer/certs/root-ca.pem
plugins.security.ssl.transport.pemcert_filepath: /etc/wazuh-indexer/certs/indexer.pem
plugins.security.ssl.transport.pemkey_filepath: /etc/wazuh-indexer/certs/indexer-key.pem
plugins.security.ssl.transport.pemtrustedcas_filepath: /etc/wazuh-indexer/certs/root-ca.pem
plugins.security.ssl.http.enabled: true
```

- **Déploiement des certificats de l'indexer**

Cette série de commandes permet de déployer et configurer les certificats TLS pour sécuriser la communication du Wazuh Indexer.

NODE_NAME=node-1

mkdir /etc/wazuh-indexer/certs

tar -xf ./wazuh-certificates.tar -C /etc/wazuh-indexer/certs/ ./$NODE_NAME.pem ./$NODE_NAME-key.pem ./admin.pem ./admin-key.pem ./root-ca.pem

mv -n /etc/wazuh-indexer/certs/$NODE_NAME.pem /etc/wazuh-indexer/certs/indexer.pem

mv -n /etc/wazuh-indexer/certs/$NODE_NAME-key.pem /etc/wazuh-indexer/certs/indexer-key.pem

chmod 500 /etc/wazuh-indexer/certs

*chmod 400 /etc/wazuh-indexer/certs/**

chown -R wazuh-indexer:wazuh-indexer /etc/wazuh-indexer/certs

```
root@etudiant:~# NODE_NAME=node-1
root@etudiant:~# mkdir /etc/wazuh-indexer/certs
root@etudiant:~# tar -xf ./wazuh-certificates.tar -C /etc/wazuh-indexer/certs/ ./$NODE_NAME.pem ./$NODE
_NAME-key.pem ./admin.pem ./admin-key.pem ./root-ca.pem
root@etudiant:~#
root@etudiant:~# mv -n /etc/wazuh-indexer/certs/$NODE_NAME.pem /etc/wazuh-indexer/certs/indexer.pem
root@etudiant:~#
root@etudiant:~# mv -n /etc/wazuh-indexer/certs/$NODE_NAME-key.pem /etc/wazuh-indexer/certs/indexer-key
.pem
root@etudiant:~#
root@etudiant:~# chmod 500 /etc/wazuh-indexer/certs
root@etudiant:~#
root@etudiant:~# chmod 400 /etc/wazuh-indexer/certs/*
root@etudiant:~#
root@etudiant:~# chown -R wazuh-indexer:wazuh-indexer /etc/wazuh-indexer/certs
root@etudiant:~#
root@etudiant:~#
```

Recharger le daemon

```
root@etudiant:~# systemctl daemon-reload
root@etudiant:~#
```

Démarrer le service

```
root@etudiant:~# systemctl start wazuh-indexer
root@etudiant:~#
```

Activer le service au démarrage du serveur

```
root@etudiant:~# systemctl enable wazuh-indexer
Synchronizing state of wazuh-indexer.service with SysV service script with /lib/systemd/systemd-sysv-in
stall.
Executing: /lib/systemd/systemd-sysv-install enable wazuh-indexer
Created symlink /etc/systemd/system/multi-user.target.wants/wazuh-indexer.service → /lib/systemd/system
/wazuh-indexer.service.
root@etudiant:~#
```

4 Initiation du cluster

Ce script **(indexer-security-init.sh**) configure la sécurité du cluster **Wazuh Indexer** (basé sur OpenSearch) en appliquant des règles d'authentification, des rôles et des permissions.

```
root@etudiant:~# /usr/share/wazuh-indexer/bin/indexer-security-init.sh
Security Admin v7
Will connect to 192.168.10.1:9200 ... done
Connected as "CN=admin,OU=Wazuh,O=Wazuh,L=California,C=US"
OpenSearch Version: 2.19.1
Contacting opensearch cluster 'opensearch' and wait for YELLOW clusterstate ...
Clustername: wazuh-cluster
Clusterstate: GREEN
Number of nodes: 1
Number of data nodes: 1
.opendistro_security index does not exists, attempt to create it ... done (0-all replicas)
Populate config from /etc/wazuh-indexer/opensearch-security/
Will update '/config' with /etc/wazuh-indexer/opensearch-security/config.yml
   SUCC: Configuration for 'config' created or updated
Will update '/roles' with /etc/wazuh-indexer/opensearch-security/roles.yml
   SUCC: Configuration for 'roles' created or updated
Will update '/rolesmapping' with /etc/wazuh-indexer/opensearch-security/roles_mapping.yml
   SUCC: Configuration for 'rolesmapping' created or updated
Will update '/internalusers' with /etc/wazuh-indexer/opensearch-security/internal_users.yml
   SUCC: Configuration for 'internalusers' created or updated
Will update '/actiongroups' with /etc/wazuh-indexer/opensearch-security/action_groups.yml
   SUCC: Configuration for 'actiongroups' created or updated
Will update '/tenants' with /etc/wazuh-indexer/opensearch-security/tenants.yml
   SUCC: Configuration for 'tenants' created or updated
Will update '/nodesdn' with /etc/wazuh-indexer/opensearch-security/nodes_dn.yml
   SUCC: Configuration for 'nodesdn' created or updated
Will update '/whitelist' with /etc/wazuh-indexer/opensearch-security/whitelist.yml
   SUCC: Configuration for 'whitelist' created or updated
Will update '/audit' with /etc/wazuh-indexer/opensearch-security/audit.yml
   SUCC: Configuration for 'audit' created or updated
Will update '/allowlist' with /etc/wazuh-indexer/opensearch-security/allowlist.yml
   SUCC: Configuration for 'allowlist' created or updated
SUCC: Expected 10 config types for node {"updated_config_types":["allowlist","tenants","rolesmapping","
nodesdn","audit","roles","whitelist","actiongroups","config","internalusers"],"updated_config_size":10,
"message":null} is 10 (["allowlist","tenants","rolesmapping","nodesdn","audit","roles","whitelist","act
iongroups","config","internalusers"]) due to: null
Done with success
root@etudiant:~#
```

Test du cluster

Cette commande vérifie que le cluster répond correctement via l'API REST

```
root@etudiant:~# curl -k -u admin:admin https://192.168.10.1:9200
{
  "name" : "node-1",
  "cluster_name" : "wazuh-cluster",
  "cluster_uuid" : "S0Sf3kCQRkidSnvXjAbPOw",
  "version" : {
    "number" : "7.10.2",
    "build_type" : "deb",
    "build_hash" : "dae2bfc93896178873b43cdf4781f183c72b238f",
    "build_date" : "2025-04-30T10:51:28.815931460Z",
    "build_snapshot" : false,
    "lucene_version" : "9.12.1",
    "minimum_wire_compatibility_version" : "7.10.0",
    "minimum_index_compatibility_version" : "7.0.0"
  },
  "tagline" : "The OpenSearch Project: https://opensearch.org/"
}
root@etudiant:~#
```

```
root@etudiant:~# curl -k -u admin:admin https://192.168.10.1:9200/_cat/nodes?v
ip           heap.percent ram.percent cpu load_1m load_5m load_15m node.role node.roles
                          cluster_manager name
192.168.10.1           60          91   4    0.08    0.16     0.14 dimr      cluster_manager,data,inges
t,remote_cluster_client *              node-1
root@etudiant:~#
```

5 Configuration du (wazuh-manager).

Le **Wazuh Manager** est le composant central qui gère les agents, analyse les alertes et stocke les données.

Démarrer le service

```
root@etudiant:~# systemctl start wazuh-manager
root@etudiant:~#
```

Activation du service au démarrage du serveur

Crée un lien symbolique pour démarrer automatiquement le service au boot

```
root@etudiant:~# systemctl enable wazuh-manager
Created symlink /etc/systemd/system/multi-user.target.wants/wazuh-manager.service → /lib/systemd/system
/wazuh-manager.service.
root@etudiant:~#
```

Vérifier le status du service

```
root@etudiant:~# systemctl status wazuh-manager
● wazuh-manager.service - Wazuh manager
     Loaded: loaded (/lib/systemd/system/wazuh-manager.service; enabled; preset: enabled)
     Active: active (running) since Mon 2025-06-23 21:41:14 CEST; 1min 24s ago
      Tasks: 154 (limit: 2273)
     Memory: 592.5M
        CPU: 1min 46.430s
     CGroup: /system.slice/wazuh-manager.service
             ├─89599 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
             ├─89639 /var/ossec/bin/wazuh-authd
             ├─89655 /var/ossec/bin/wazuh-db
             ├─89670 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
             ├─89671 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
             ├─89674 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
             ├─89677 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh_apid.py
             ├─89690 /var/ossec/bin/wazuh-execd
             ├─89704 /var/ossec/bin/wazuh-analysisd
             ├─89746 /var/ossec/bin/wazuh-syscheckd
             ├─89761 /var/ossec/bin/wazuh-remoted
             ├─89795 /var/ossec/bin/wazuh-logcollector
             ├─89814 /var/ossec/bin/wazuh-monitord
             └─89837 /var/ossec/bin/wazuh-modulesd
```

6 Configuration du ficher filebeat.

Filebeat est utilisé pour envoyer les logs du Wazuh Manager vers Wazuh Indexer

Installer le paquet filebeat si ce n'est pas encore fait

apt -y install filebeat

- **Téléchargement du fichier de configuration filebeat**

Récupère un modèle préconfiguré pour Wazuh.

curl -so /etc/filebeat/filebeat.yml

https://packages.wazuh.com/4.7/tpl/wazuh/filebeat/filebeat.yml

```
root@etudiant:~# curl -so /etc/filebeat/filebeat.yml https://packages.wazuh.com/4.7/tpl/wazuh/filebeat/
filebeat.yml
root@etudiant:~#
```

Edition du fichier de configuration

```
root@etudiant:~# vim /etc/filebeat/filebeat.yml
```

Sur **hosts** : addresse ip plus le port

ssl : Active le chiffrement TLS avec les certificats.

username/password : Variables stockées dans le keystore.

```
# Wazuh - Filebeat configuration file
output.elasticsearch:
  hosts: ["192.168.10.1:9200"]
  protocol: https
  username: ${username}
  password: ${password}
  ssl.certificate_authorities:
    - /etc/filebeat/certs/root-ca.pem
  ssl.certificate: "/etc/filebeat/certs/filebeat.pem"
  ssl.key: "/etc/filebeat/certs/filebeat-key.pem"
setup.template.json.enabled: true
setup.template.json.path: '/etc/filebeat/wazuh-template.json'
setup.template.json.name: 'wazuh'
setup.ilm.overwrite: true
setup.ilm.enabled: false
```

- **Création du fichier Keystore**

Stocke les identifiants de manière sécurisée (au lieu de les écrire en clair dans le fichier YAML).

```
root@etudiant:~# filebeat keystore create
Created filebeat keystore
root@etudiant:~#
```

Ajout de l'utilisateur et du mot de passe par défaut

```
root@etudiant:~# echo admin | filebeat keystore add username --stdin --force
Successfully updated the keystore
root@etudiant:~#
root@etudiant:~# echo admin | filebeat keystore add password --stdin --force
Successfully updated the keystore
root@etudiant:~#
root@etudiant:~#
```

7 Téléchargement d'un modèle d'alerte

Le fichier wazuh-template.json structure les données Wazuh dans Elasticsearch/OpenSearch pour optimiser les performances et la lisibilité.

curl -so /etc/filebeat/wazuh-template.json https://raw.githubusercontent.com/wazuh/wazuh/v4.7.2/extensions/elasticsearch/7.x/wazuh-template.json

```
root@etudiant:~# curl -so /etc/filebeat/wazuh-template.json https://raw.githubusercontent.com/wazuh/waz
uh/v4.7.2/extensions/elasticsearch/7.x/wazuh-template.json
root@etudiant:~#
```

Donner les permissions de lecture à Filebeat

```
root@etudiant:~# chmod go+r /etc/filebeat/wazuh-template.json
root@etudiant:~#
```

8 Installation des modules supplémentaire pour filebeat

Ces modules améliorent le traitement des logs Wazuh (parsing, pipelines) avant leur envoi à Elasticsearch.

curl -s https://packages.wazuh.com/4.x/filebeat/wazuh-filebeat-0.3.tar.gz | tar -xvz -C /usr/share/filebeat/module

```
root@etudiant:~# curl -s https://packages.wazuh.com/4.x/filebeat/wazuh-filebeat-0.3.tar.gz | tar -xvz -
C /usr/share/filebeat/module
wazuh/
wazuh/archives/
wazuh/archives/ingest/
wazuh/archives/ingest/pipeline.json
wazuh/archives/config/
wazuh/archives/config/archives.yml
wazuh/archives/manifest.yml
wazuh/_meta/
wazuh/_meta/config.yml
wazuh/_meta/docs.asciidoc
wazuh/_meta/fields.yml
wazuh/alerts/
wazuh/alerts/ingest/
wazuh/alerts/ingest/pipeline.json
wazuh/alerts/config/
wazuh/alerts/config/alerts.yml
wazuh/alerts/manifest.yml
wazuh/module.yml
root@etudiant:~#
```

- **Déploiement des certificats**

Cette étape est cruciale pour sécuriser la communication entre Filebeat (qui collecte les logs) et Wazuh Indexer (qui les stocke).

NODE_NAME=node-1

mkdir /etc/filebeat/certs

tar -xf ./wazuh-certificates.tar -C /etc/filebeat/certs/ ./$NODE_NAME.pem ./$NODE_NAME-key.pem ./root-ca.pem

mv -n /etc/filebeat/certs/$NODE_NAME.pem /etc/filebeat/certs/filebeat.pem

mv -n /etc/filebeat/certs/$NODE_NAME-key.pem /etc/filebeat/certs/filebeat-key.pem

chmod 500 /etc/filebeat/certs

*chmod 400 /etc/filebeat/certs/**

chown -R root:root /etc/filebeat/certs

```
root@etudiant:~# NODE_NAME=node-1
root@etudiant:~# mkdir /etc/filebeat/certs
root@etudiant:~#
root@etudiant:~# tar -xf ./wazuh-certificates.tar -C /etc/filebeat/certs/ ./$NODE_NAME.pem ./$NODE_NAME
-key.pem ./root-ca.pem
root@etudiant:~#
root@etudiant:~# mv -n /etc/filebeat/certs/$NODE_NAME.pem /etc/filebeat/certs/filebeat.pem
root@etudiant:~#
root@etudiant:~# mv -n /etc/filebeat/certs/$NODE_NAME-key.pem /etc/filebeat/certs/filebeat-key.pem
root@etudiant:~#
root@etudiant:~# chmod 500 /etc/filebeat/certs
root@etudiant:~#
root@etudiant:~# chmod 400 /etc/filebeat/certs/*
root@etudiant:~#
root@etudiant:~# chown -R root:root /etc/filebeat/certs
root@etudiant:~#
root@etudiant:~#
```

Démarrer le service

```
root@etudiant:~# systemctl start filebeat
root@etudiant:~#
```

Activation du service au démarrage du serveur

Crée un lien symbolique pour démarrer Filebeat au boot.

```
root@etudiant:~# systemctl enable filebeat
Synchronizing state of filebeat.service with SysV service script with /lib/systemd/systemd-sysv-install
.
Executing: /lib/systemd/systemd-sysv-install enable filebeat
Created symlink /etc/systemd/system/multi-user.target.wants/filebeat.service → /lib/systemd/system/file
beat.service.
root@etudiant:~#
```

Test de filebeat

Cette commande vérifie que Filebeat peut communiquer avec le Wazuh Indexer.

```
root@etudiant:~# filebeat test output
elasticsearch: https://192.168.10.1:9200...
  parse url... OK
  connection...
    parse host... OK
    dns lookup... OK
    addresses: 192.168.10.1
    dial up... OK
  TLS...
    security: server's certificate chain verification is enabled
    handshake... OK
    TLS version: TLSv1.3
    dial up... OK
  talk to server... OK
  version: 7.10.2
root@etudiant:~#
```

9 Configuration du tableau de bord (dashboard)

Le fichier configure l'accès au tableau de bord et la connexion à Wazuh Indexer.

```
root@etudiant:~# vim /etc/wazuh-dashboard/opensearch_dashboards.yml
```

```
server.host: 192.168.10.1
server.port: 443
opensearch.hosts: https://192.168.10.1:9200
opensearch.ssl.verificationMode: certificate
#opensearch.username:
#opensearch.password:
opensearch.requestHeadersAllowlist: ["securitytenant","Authorization"]
opensearch_security.multitenancy.enabled: false
opensearch_security.readonly_mode.roles: ["kibana_read_only"]
server.ssl.enabled: true
server.ssl.key: "/etc/wazuh-dashboard/certs/dashboard-key.pem"
server.ssl.certificate: "/etc/wazuh-dashboard/certs/dashboard.pem"
opensearch.ssl.certificateAuthorities: ["/etc/wazuh-dashboard/certs/root-ca.pem"]
uiSettings.overrides.defaultRoute: /app/wz-home
~
```

Déploiement des certificats du tableau de bord wazuh

Cette étape sécurise les communications entre le Wazuh Dashboard (interface web) et les autres composants.

NODE_NAME=node-1

mkdir /etc/wazuh-dashboard/certs

tar -xf ./wazuh-certificates.tar -C /etc/wazuh-dashboard/certs/ ./$NODE_NAME.pem ./$NODE_NAME-key.pem ./root-ca.pem

mv -n /etc/wazuh-dashboard/certs/$NODE_NAME.pem /etc/wazuh-dashboard/certs/dashboard.pem

mv -n /etc/wazuh-dashboard/certs/$NODE_NAME-key.pem /etc/wazuh-dashboard/certs/dashboard-key.pem

chmod 500 /etc/wazuh-dashboard/certs

*chmod 400 /etc/wazuh-dashboard/certs/**

chown -R wazuh-dashboard:wazuh-dashboard /etc/wazuh-dashboard/certs

```
root@etudiant:~# NODE_NAME=node-1
root@etudiant:~# mkdir /etc/wazuh-dashboard/certs
root@etudiant:~# tar -xf ./wazuh-certificates.tar -C /etc/wazuh-dashboard/certs/ ./$NODE_NAME.pem ./$NO
DE_NAME-key.pem ./root-ca.pem
root@etudiant:~#
root@etudiant:~# mv -n /etc/wazuh-dashboard/certs/$NODE_NAME.pem /etc/wazuh-dashboard/certs/dashboard.p
em
root@etudiant:~#
root@etudiant:~# mv -n /etc/wazuh-dashboard/certs/$NODE_NAME-key.pem /etc/wazuh-dashboard/certs/dashboa
rd-key.pem
root@etudiant:~#
root@etudiant:~# chmod 500 /etc/wazuh-dashboard/certs
root@etudiant:~#
root@etudiant:~# chmod 400 /etc/wazuh-dashboard/certs/*
root@etudiant:~#
root@etudiant:~# chown -R wazuh-dashboard:wazuh-dashboard /etc/wazuh-dashboard/certs
root@etudiant:~#
root@etudiant:~#
```

Démarrer le service

```
root@etudiant:~# systemctl start wazuh-dashboard
root@etudiant:~#
```

Activer le service au démarrage

Active le démarrage automatique au boot.

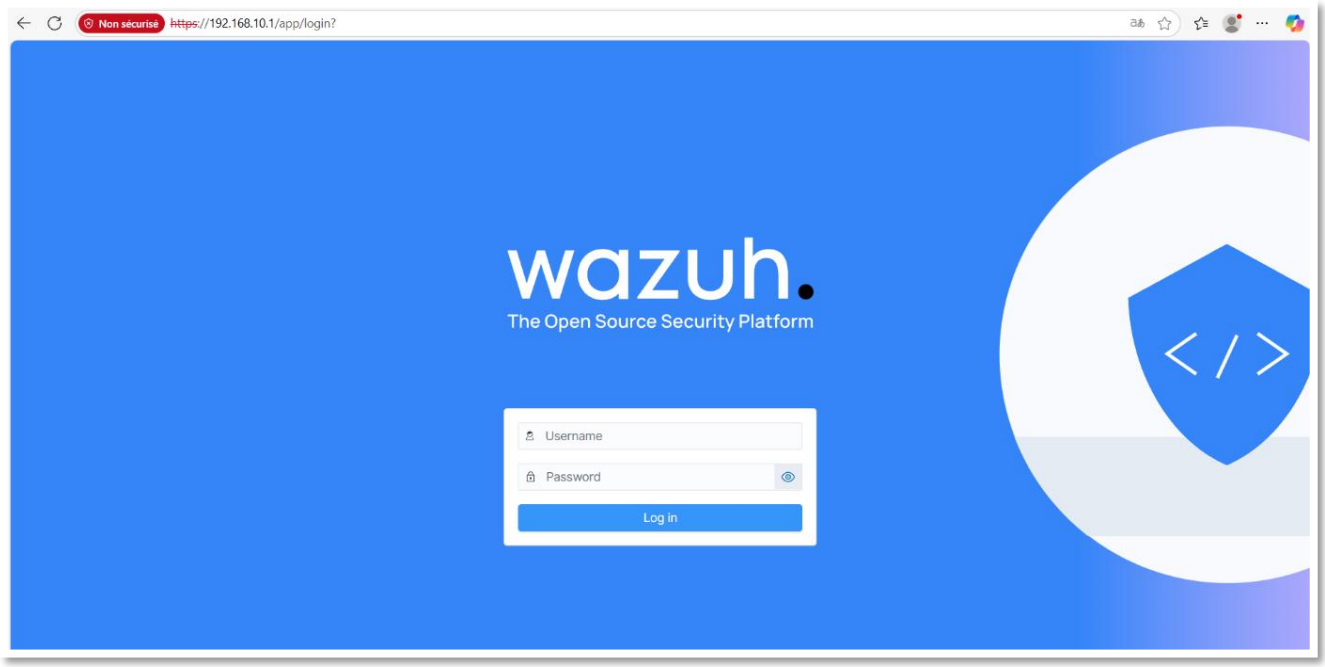
```
root@etudiant:~# systemctl enable wazuh-dashboard
Created symlink /etc/systemd/system/multi-user.target.wants/wazuh-dashboard.service → /etc/systemd/syst
em/wazuh-dashboard.service.
root@etudiant:~#
```

Accès au tableau de bord de Wazuh

Sur navigateur web du machine cliente on mets https://192.168.10.1

Utilisateur par défaut : **admin**

Mot de passe : **admin**



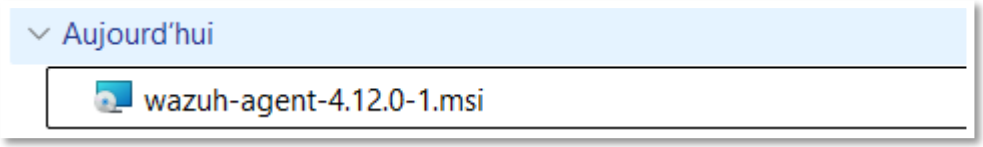
10 Installation et configuration des agent Wazuh

Cette section décrit la procédure d'installation et de configuration de l'agent Wazuh sur un système Windows. L'agent permet de collecter et d'envoyer des logs et événements de sécurité au serveur Wazuh Manager pour analyse centralisée.

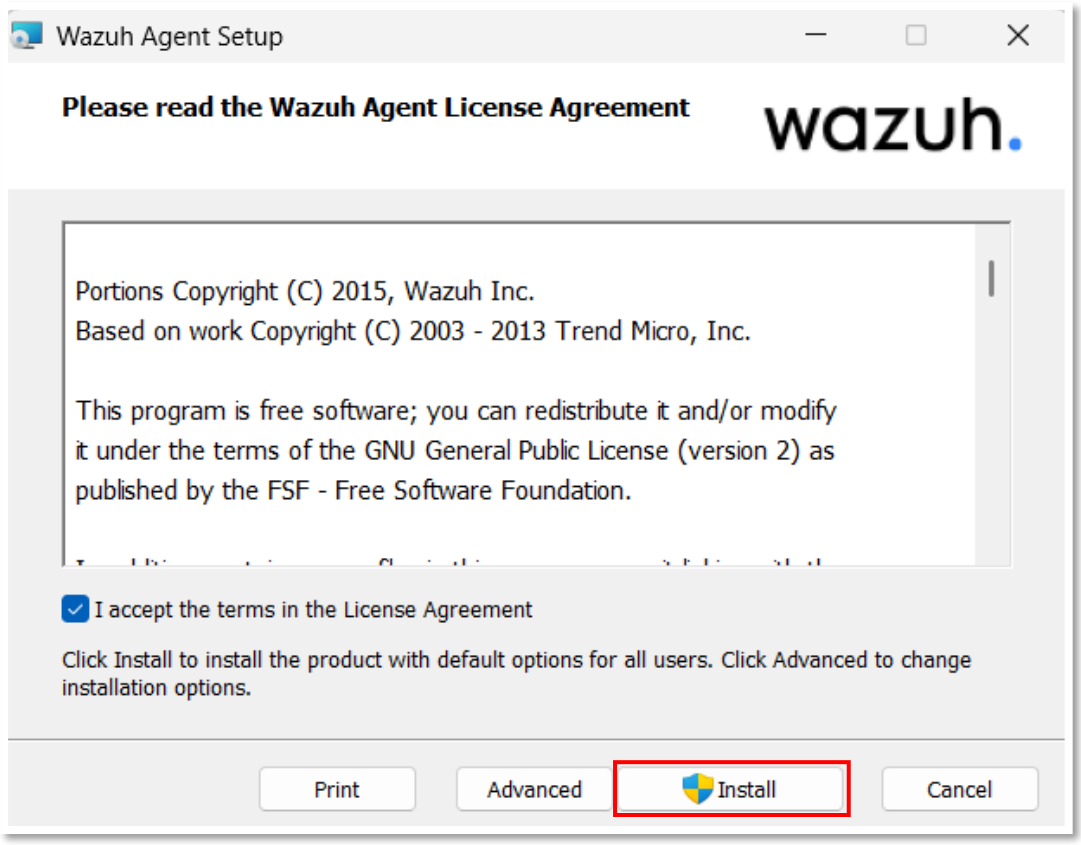
10.1 Agent Windows

Téléchargement de l'agent

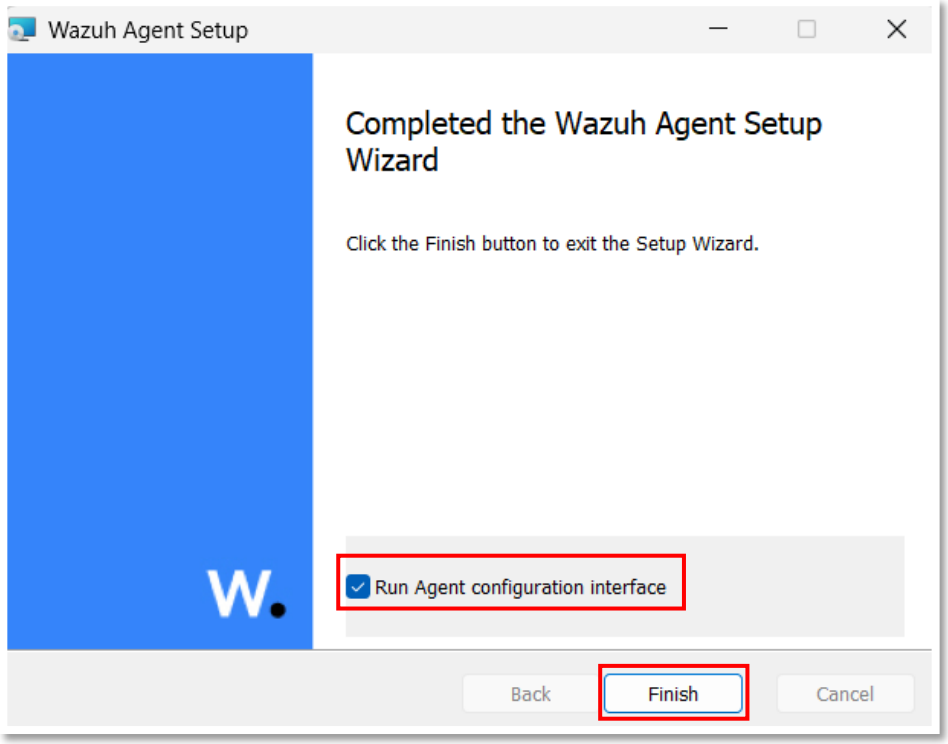
Puis double clique sur le setup



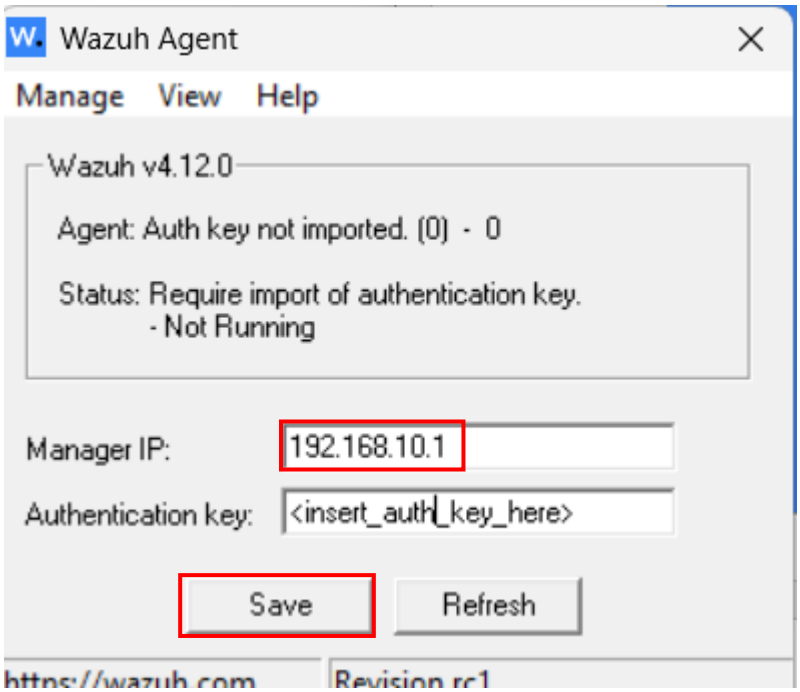
Accepter les conditions d'utilisations puis **install**



Cliquer sur finish



Renseigner l'addresse ip de management puis **Save**



Démarrer le service

Sous PowerShell

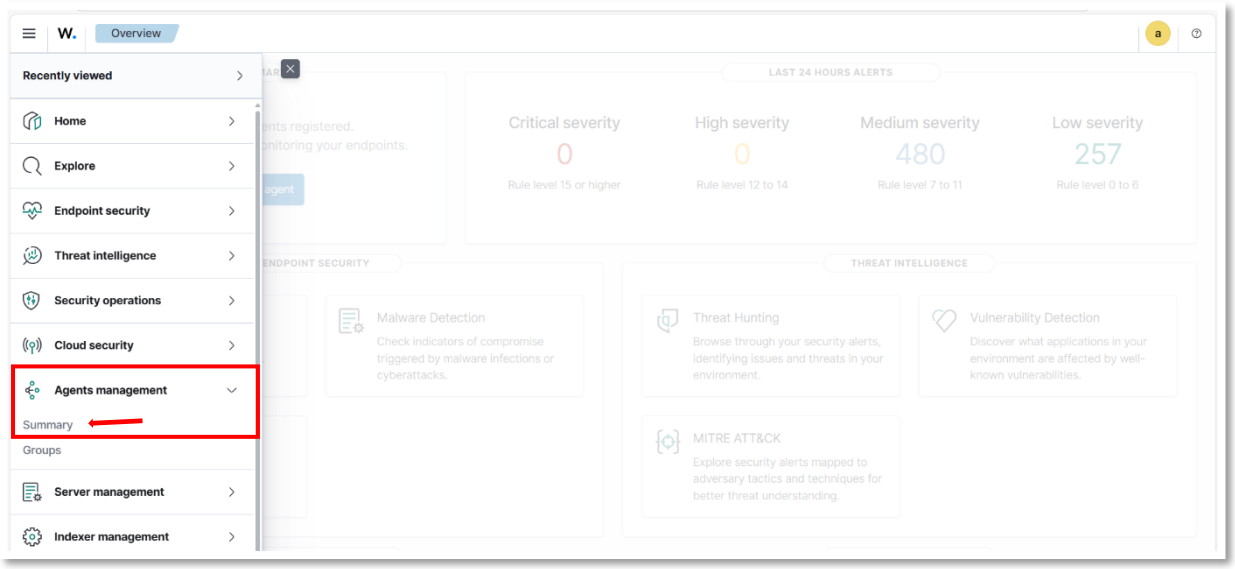
```
PS C:\WINDOWS\system32> Start-Service WazuhSvc
PS C:\WINDOWS\system32>
```

Vérifier l'état du service

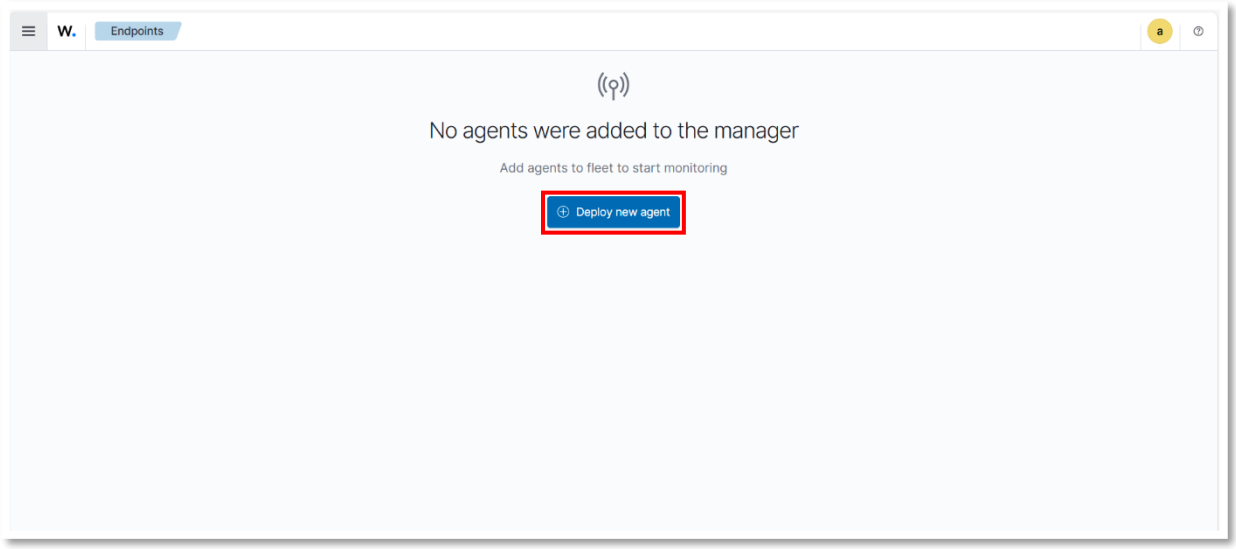
```
PS C:\WINDOWS\system32> Get-Service WazuhSvc

Status   Name               DisplayName
------   ----               -----------
Running  WazuhSvc           Wazuh


PS C:\WINDOWS\system32>
```

➢ **Configurer l'agent Windows sur wazuh**

Dans le Wazuh Dashboard, allez dans : **Agents management ➔ Summary**



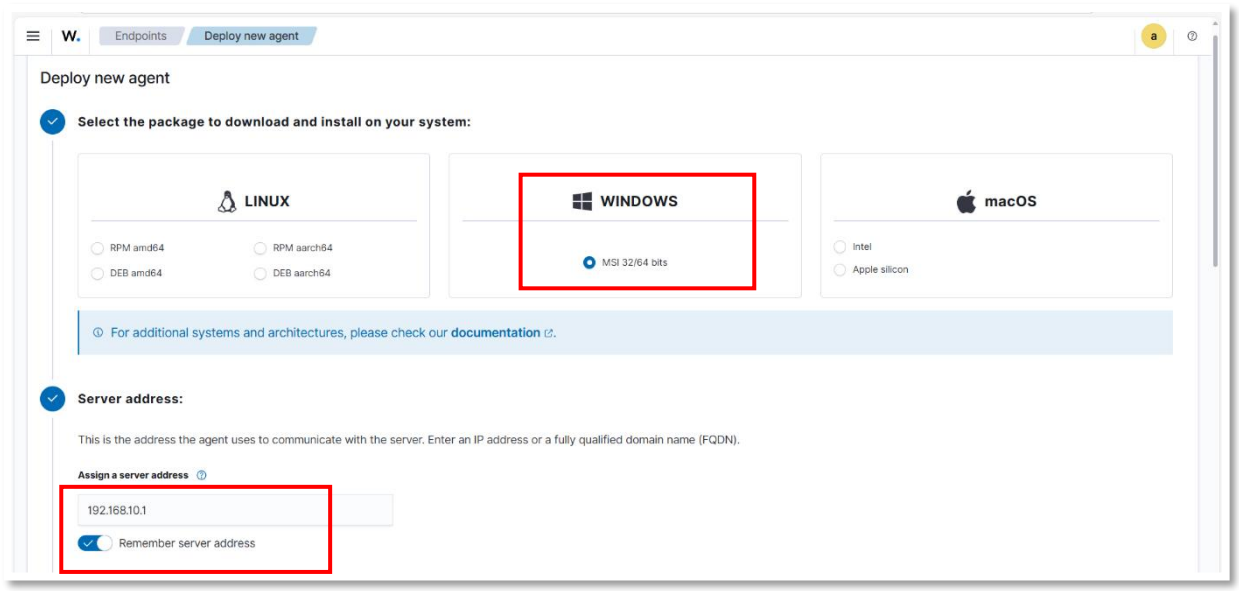
Sélectionner **Deploy new agent**



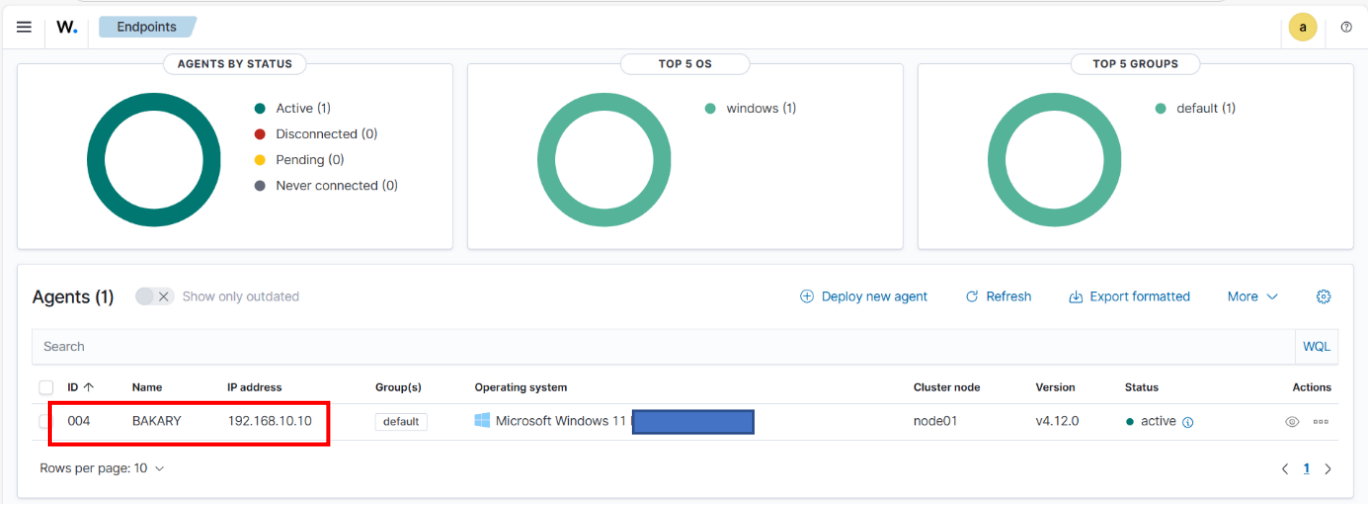
Choisir le package correspondant à notre système : **Windows**

Entrez **l'adresse IP** du serveur Wazuh Manager

Cochez "**Remember server address**" pour sauvegarder cette information.



Voici l'agent



10.2 **Agent linux**

Ajoutez le dépôt Wazuh pour télécharger les packages officiels.

Installez la clé GPG

Authentifie les paquets Wazuh via une signature numérique

```
root@bakary:~# curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --no-default-keyring --key
ring gnupg-ring:/usr/share/keyrings/wazuh.gpg --import && chmod 644 /usr/share/keyrings/wazuh.gpg
gpg: keyring '/usr/share/keyrings/wazuh.gpg' created
gpg: directory '/root/.gnupg' created
gpg: /root/.gnupg/trustdb.gpg: trustdb created
gpg: key 96B3EE5F29111145: public key "Wazuh.com (Wazuh Signing Key) <support@wazuh.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1
root@bakary:~# _
```

Active le téléchargement des paquets officiels

```
root@bakary:~#
root@bakary:~# echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/ap
t/ stable main" | tee -a /etc/apt/sources.list.d/wazuh.list
deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main
root@bakary:~# _
```

Mettre à jour les packages

```
root@bakary:~# apt -y update
Réception de :1 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Atteint :2 http://sn.archive.ubuntu.com/ubuntu jammy InRelease
Réception de :3 http://sn.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Réception de :4 https://packages.wazuh.com/4.x/apt stable InRelease [17,3 kB]
Réception de :5 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [2 423 kB]
```

- **Installation de l'agent**

Spécifiez l'IP du serveur Wazuh Manager (**192.168.10.1)** :

```
root@bakary:~# WAZUH_MANAGER="192.168.10.1" apt -y install wazuh-agent
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les NOUVEAUX paquets suivants seront installés :
  wazuh-agent
0 mis à jour, 1 nouvellement installés, 0 à enlever et 105 non mis à jour.
```

Recharger le daemon

```
root@bakary:~# systemctl daemon-reload
root@bakary:~#
```

Démarrer le service

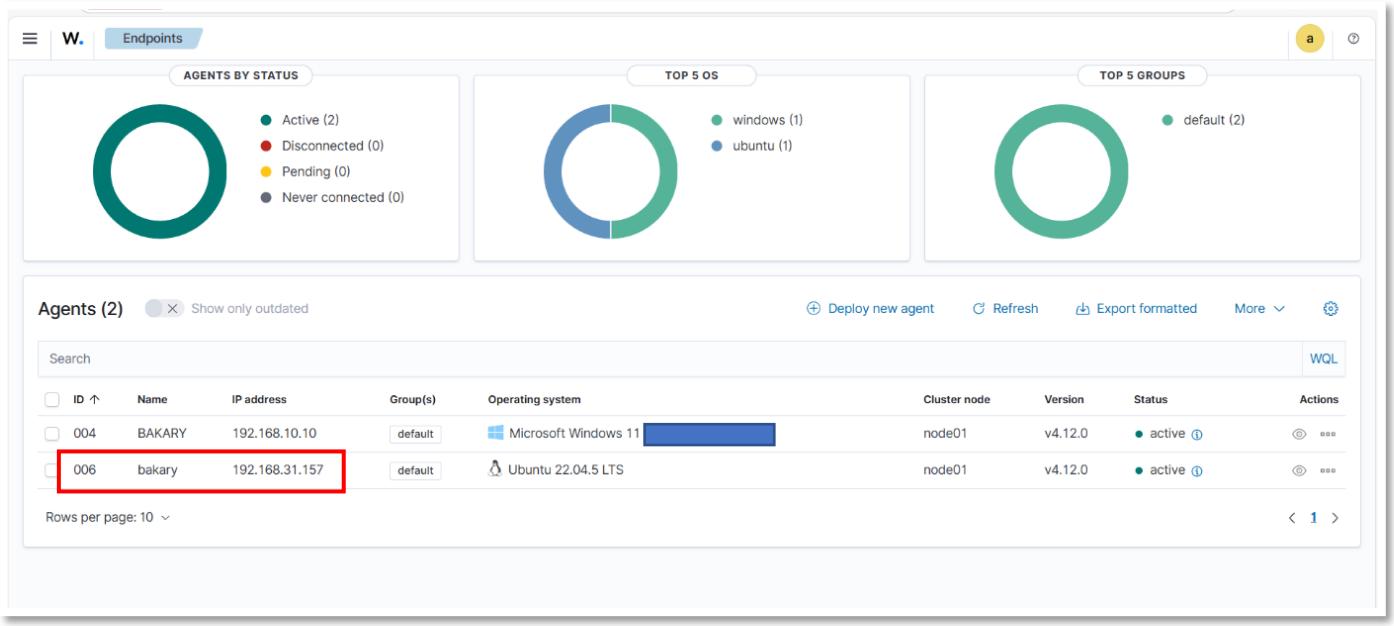
```
root@bakary:~# systemctl start wazuh-agent
root@bakary:~# _
```

Activer le service au démarrage du serveur

```
root@bakary:~# systemctl enable wazuh-agent
Created symlink /etc/systemd/system/multi-user.target.wants/wazuh-agent.service → /lib/systemd/syste
m/wazuh-agent.service.
root@bakary:~# _
```

Test sur le serveur

Une fois sur le tableau de bord de wazuh nous allons actualiser la page



Conclusion :

Wazuh offre une solution open-source puissante pour la sécurité des systèmes, combinant simplicité de déploiement et fonctionnalités professionnelles. Avec cette implémentation, votre infrastructure est désormais équipée pour détecter, analyser et répondre aux menaces de manière proactive