

Département Mathématique et Informatique

**Filière :**  
**«Ingénierie Informatique, Big Data et Cloud Computing»**

**Projet fin Module: Virtualisation et Cloud computing**

**Cloud\_Sécurité des endpoints et supervision  
SIEM : étude de cas multi-OS (Linux &  
Windows)**

Réalisé par :

M. WAHYUDI Muhammad Irfan

Encadré par :

Pr. Azeddine KHIAT

Année universitaire : 2025/2026

## Contents

Introduction .....	5
Objectifs .....	5
Architecture .....	6
Mise en place de l'infrastructure AWS .....	8
1. Configuration Réseau (VPC) .....	8
2. Configuration Sécurité (Security Groups).....	8
3. Création des instances EC2 .....	10
4. Installation Wazuh All-in-One sur Ubuntu (serveur).....	11
Enrôle agents .....	16
5. Enrôler le client Linux (Ubuntu) .....	16
6. Enrôler le client Windows (agent + événements) .....	18
Démo SIEM + EDR : scénarios d'événements à générer .....	20
7. Démo SIEM côté Linux (rapide, visible tout de suite).....	20
8. Démo EDR côté Windows (événements sécurité + option Sysmon).....	23
EDR Avancé avec Sysmon .....	26
9. Installation de sysmon .....	26
10. Integrer Sysmon sur Wazuh .....	27
SIEM, EDR et Gestion des Accès (IAM / PAM).....	30
1. SIEM vs EDR .....	30
2. IAM et PAM.....	30
3. Threat Hunting (3 requêtes simples) .....	31
Conclusion.....	32

Figure 1: Architecture de projet .....	6
Figure 2: Configuration Réseau (VPC) .....	8
Figure 3: Configuration Sécurité (Security Groups) .....	9
Figure 4: EC2-1 Wazuh Server .....	10
Figure 5: EC2-2 Linux Client.....	10
Figure 6: EC2-3 Windows Client.....	11
Figure 7: toutes les instances.....	11
Figure 8: Connecter to server .....	11
Figure 9: Telecharger Wazuh part1 .....	11
Figure 10: Telecharger Wazuh part2.....	12
Figure 11: Verifier installation Wazuh Manager .....	13
Figure 12: Verifier installation Wazuh Indexer .....	13
Figure 13: Verifier installation Wazuh Dashboard .....	14
Figure 14: Wazuh login.....	14
Figure 15: Wazuh dashboard.....	15
Figure 16: Configurer nouveau agent (linux client).....	16
Figure 17: installer et lancer agent (linux client) .....	17
Figure 18: Connecter to windows client depuis RDP .....	18
Figure 19: Configurer nouveau agent (windows client).....	19
Figure 20: installer et lancer agent (windows client) .....	19
Figure 21: Verifier agents .....	20
Figure 22: Test connecter depuis fake user .....	20
Figure 23: alertes authentication failed .....	21
Figure 24: test élévation de privilèges.....	21
Figure 25: alertes événements sudo .....	22
Figure 26: test modification fichier sensible .....	22
Figure 27: alerte Surveillance de l'intégrité des fichiers .....	23
Figure 28: dashboard des alertes (linux client) .....	23
Figure 29: test to connecter depuis fake user .....	24
Figure 30: alertes événements Windows Security (Failed logon).....	24
Figure 31: test Création d'un utilisateur local .....	25
Figure 32: alertes événements “user created / group changed” .....	25
Figure 33: Installation de sysmon part1 .....	26
Figure 34: Installation de sysmon part2 .....	26
Figure 35: Sysmon logs.....	27
Figure 36: Intégrer Sysmon sur Wazuh part1 .....	28
Figure 37: Intégrer Sysmon sur Wazuh part2 .....	28
Figure 38: Wazuh dashboard (windows client).....	29

Figure 39: Threat Hunting requête 1 .....	31
Figure 40: Threat Hunting requête 2 .....	31
Figure 41: Threat Hunting requête 3 .....	32

## Introduction

Ce projet présente la mise en œuvre d'une plateforme de sécurité basée sur **Wazuh**, combinant les fonctionnalités **SIEM** et **EDR**, déployée dans un environnement **Cloud AWS**. L'architecture repose sur un **serveur Wazuh**, des **systèmes Linux et Windows supervisés**, et un **VPC AWS sécurisé**, permettant la collecte, la centralisation et l'analyse des événements de sécurité.

À travers des scénarios pratiques, le projet met en évidence la **sécurité des endpoints**, la **gestion des identités et des accès (IAM)**, ainsi que la **surveillance et la détection des menaces**, offrant une vision concrète du fonctionnement d'un **SOC moderne** dans un environnement Cloud.

## Objectifs

Les objectifs principaux de ce projet sont :

- Mettre en place une architecture SIEM/EDR fonctionnelle dans le Cloud
- Superviser des endpoints Linux et Windows
- Centraliser et analyser les événements de sécurité
- Illustrer les concepts de Endpoint Security, IAM/PAM et Threat Hunting
- Produire des alertes exploitables à des fins SOC

# Architecture

L'architecture de ce projet repose sur une **infrastructure AWS EC2 simple, sécurisée et représentative d'un SOC moderne**, intégrant les fonctionnalités **SIEM** et **EDR** via la solution **Wazuh All-in-One**.

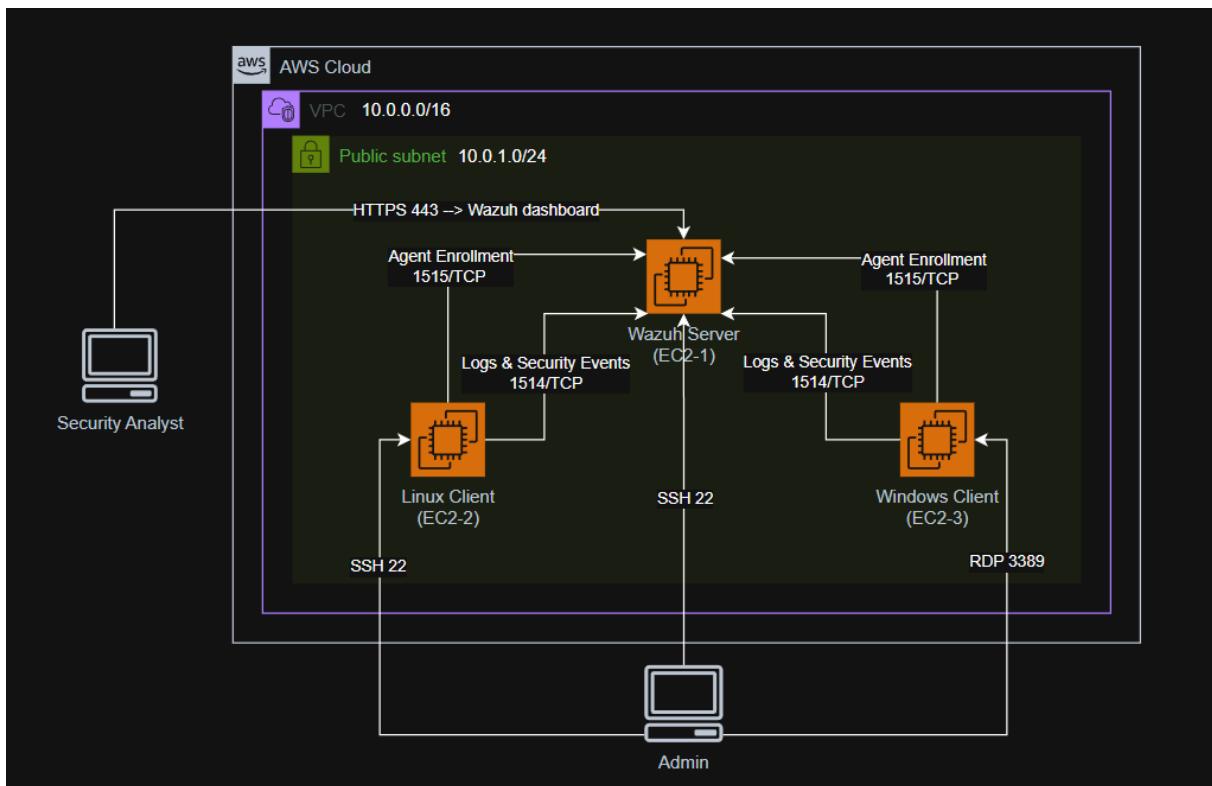


Figure 1: Architecture de projet

## 1. EC2-1 : Serveur Wazuh All-in-One (Ubuntu)

Cette instance constitue le cœur de la plateforme de sécurité. Elle regroupe l'ensemble des composants Wazuh :

- **Wazuh Manager** : collecte, analyse, corrélation et détection des événements de sécurité
- **Wazuh Indexer** : indexation et stockage des logs
- **Wazuh Dashboard** : visualisation des alertes, supervision en temps réel et analyse SIEM via une interface web sécurisée

## 2. EC2-2 : Client Linux (Ubuntu)

Cette instance représente un endpoint Linux supervisé par Wazuh.

- Installation de l'agent Wazuh
- Surveillance des logs système, de l'intégrité des fichiers (FIM), des activités suspectes et des tentatives d'élévation de privilèges
- Transmission des événements vers le serveur Wazuh

Elle permet de démontrer les mécanismes de **sécurité et de hardening des systèmes Linux**.

### 3. EC2-3 : Client Windows Server

Cette instance simule un **environnement Windows d'entreprise**.

- Installation de l'agent Wazuh pour Windows
- Surveillance des journaux Windows, des activités utilisateurs et des tentatives de connexion suspectes
- Option avancée : **Sysmon**, pour enrichir les événements EDR et améliorer la détection des processus, connexions réseau et comportements anormaux

Cette instance met en évidence les **capacités EDR de Wazuh en environnement Windows**.

### 4. Flux de communication

Les communications sont **strictement contrôlées** via le VPC AWS et les Security Groups :

- **Agents → Wazuh Server** : port 1514 (TCP/UDP) – transmission des événements
- **Enrôlement des agents** : port 1515 (TCP)
- **Accès au Dashboard** : HTTPS (443) – accès sécurisé à l'interface SIEM

# Mise en place de l'infrastructure AWS

## 1. Configuration Réseau (VPC)

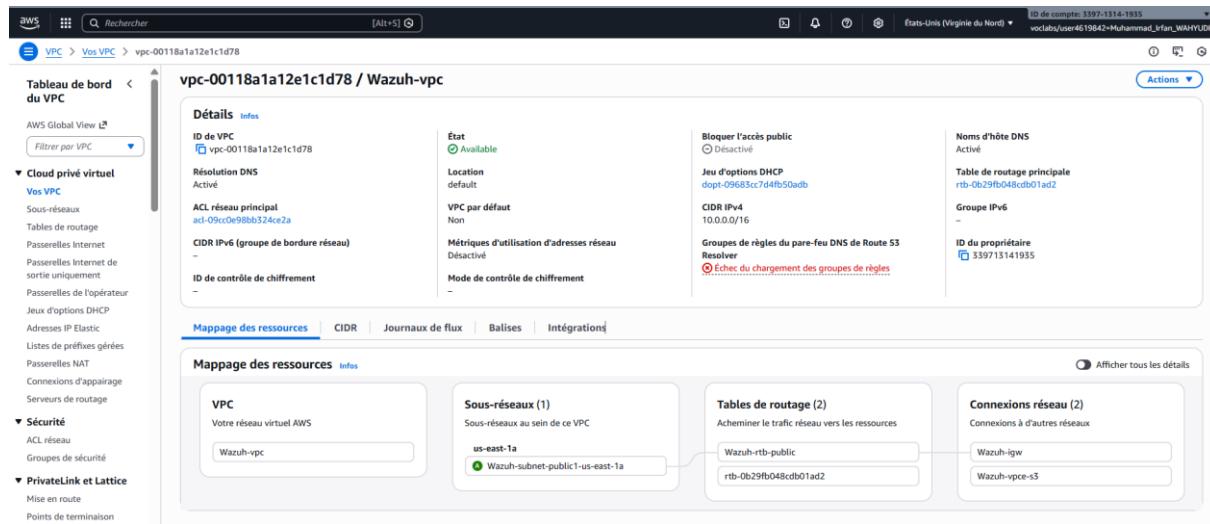


Figure 2: Configuration Réseau (VPC)

Le projet repose sur un **VPC AWS sécurisé et isolé**, structuré pour représenter un SOC moderne. Ce VPC permet de contrôler finement les flux entre les instances EC2 et de séparer les composants critiques, tout en assurant la communication nécessaire pour la collecte et l'analyse des événements de sécurité. Les sous-réseaux, les Security Groups et les règles de routage sont configurés de manière à respecter le principe du **moindre privilège**, garantir la **sécurité interne**, et permettre la **scalabilité** et la gestion centralisée des agents Wazuh sans exposer le trafic sensible à Internet.

**Remarque:** Aucune passerelle NAT n'a été déployée, les instances étant placées dans un subnet public avec un accès Internet direct via une Internet Gateway.

## 2. Configuration Sécurité (Security Groups)

Dans ce projet, les règles de sécurité réseau ont été **centralisées au sein d'un seul Security Group**, attaché aux différentes instances EC2. Ce choix vise à **simplifier la gestion**, tout en conservant un **niveau de sécurité adapté** au périmètre du projet.

Le Security Group unique autorise uniquement les flux strictement nécessaires au fonctionnement de la plateforme Wazuh et à l'administration des instances :

- **Accès sécurisé au Wazuh Dashboard** via HTTPS (443) depuis l'adresse IP de l'administrateur
- **Communication des agents Wazuh vers le serveur** sur les ports 1514 et 1515

**Remarque :** Les ports 1514 et 1515 permettent aux agents Wazuh d'envoyer respectivement les logs et événements, et de s'enregistrer auprès du serveur, avec une connexion toujours initiée par l'agent. Pour la sécurité et la flexibilité, la source des flux est un Security Group plutôt qu'une IP, garantissant que seules les instances autorisées peuvent communiquer, indépendamment des changements d'IP ou de l'Auto Scaling. Les ports ne sont jamais exposés à Internet, l'accès HTTPS (443) est limité à l'administrateur, et cette configuration respecte le principe du moindre privilège, assurant une architecture SOC Cloud professionnelle, sécurisée et scalable.

- **Accès d'administration :**
  - SSH (22) pour les instances Linux
  - RDP (3389) pour l'instance Windows
- **Trafic sortant autorisé** afin de permettre aux agents d'initier les connexions et aux instances d'accéder aux mises à jour nécessaires

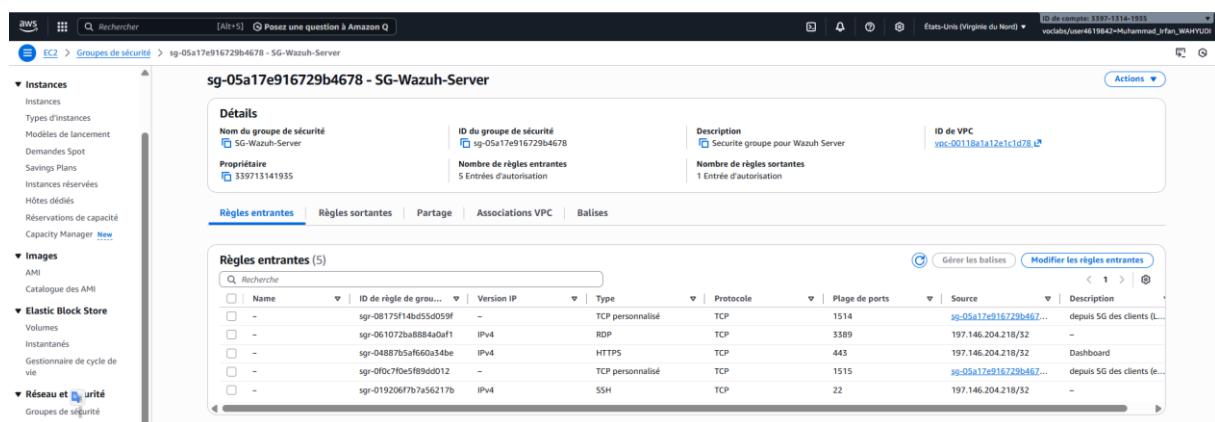
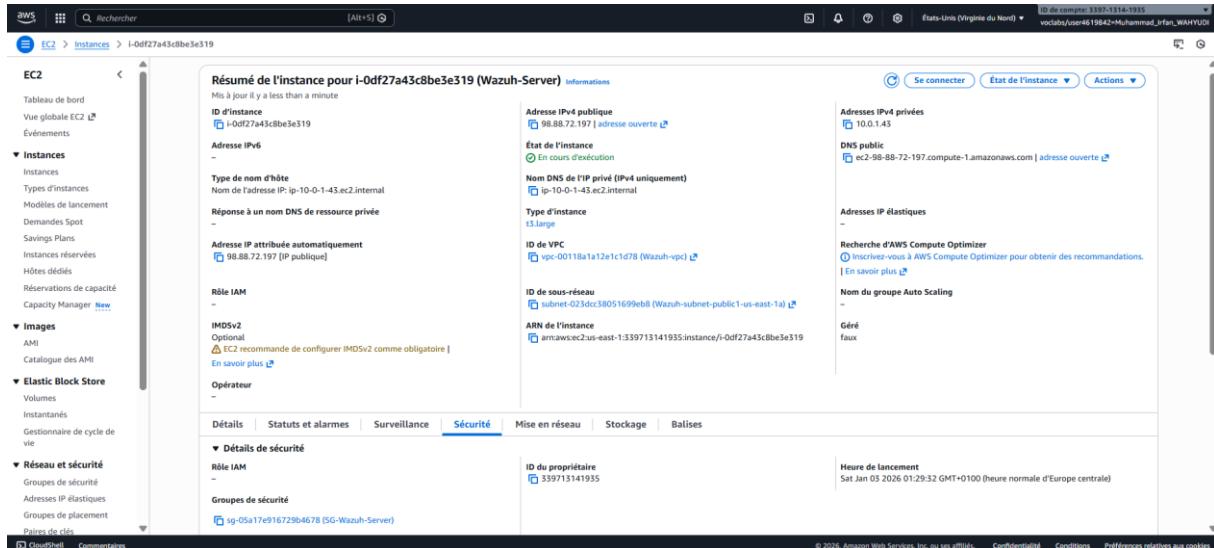


Figure 3: Configuration Sécurité (Security Groups)

Cette approche permet de maintenir une **architecture lisible et fonctionnelle**, représentative d'un SOC Cloud simplifié, tout en respectant le principe du **moindre privilège** pour les communications réseau essentielles.

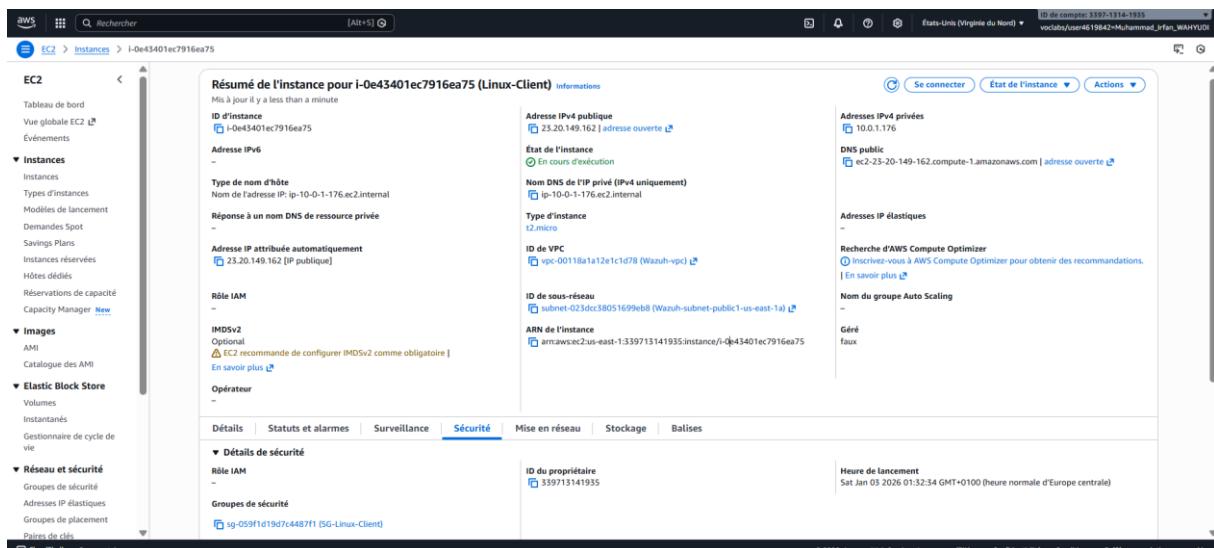
### 3. Création des instances EC2



The screenshot shows the AWS Management Console with the EC2 service selected. The left sidebar shows various navigation options like EC2, Instances, Images, and Network & security. The main content area displays the details for an instance named 'Wazuh-Server' with ID i-0df27a43c8be3e319. Key information includes:

- Address IPv4 publique:** 98.88.72.197 | adresse ouverte
- État de l'instance:** En cours d'exécution
- Nom DNS de l'IP privé (IPv4 uniquement):** ip-10-0-1-43.ec2.internal
- Type d'instance:** t2.large
- ID de VPC:** vpc-0011ba1a12e1cd78 (Wazuh-vpc)
- Rôle IAM:** –
- ID de sous-réseau:** subnet-023dc38051699eb8 (Wazuh-subnet-public1-us-east-1-a)
- ARN de l'instance:** arnaws:ec2:us-east-1:339713141935:instance/i-0df27a43c8be3e319
- Heure de lancement:** Sat Jan 05 2026 01:29:32 GMT+0100 (heure normale d'Europe centrale)

Figure 4: EC2-1 Wazuh Server



The screenshot shows the AWS Management Console with the EC2 service selected. The left sidebar shows various navigation options like EC2, Instances, Images, and Network & security. The main content area displays the details for an instance named 'Linux-Client' with ID i-0e43401ec7916ea75. Key information includes:

- Address IPv4 publique:** 23.20.149.162 | adresse ouverte
- État de l'instance:** En cours d'exécution
- Nom DNS de l'IP privé (IPv4 uniquement):** ip-10-0-1-176.ec2.internal
- Type d'instance:** t2.micro
- ID de VPC:** vpc-0011ba1a12e1cd78 (Wazuh-vpc)
- Rôle IAM:** –
- ID de sous-réseau:** subnet-023dc38051699eb8 (Wazuh-subnet-public1-us-east-1-a)
- ARN de l'instance:** arnaws:ec2:us-east-1:339713141935:instance/i-0e43401ec7916ea75
- Heure de lancement:** Sat Jan 05 2026 01:32:34 GMT+0100 (heure normale d'Europe centrale)

Figure 5: EC2-2 Linux Client

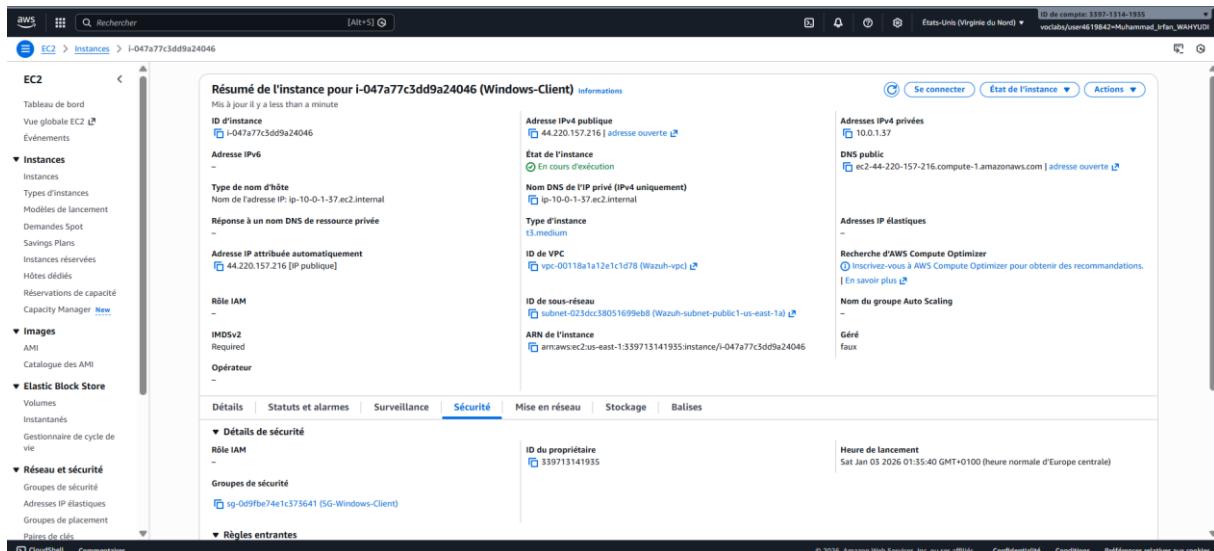


Figure 6: EC2-3 Windows Client

Instances (3) Informations										
Date de la dernière mise à jour Il y a less than a minute										
En cours d'exécution										
<input type="checkbox"/>	Name	i-De43401ec7916ea75	État de l'insta...	En cours d'...	Type d'insta...	t2.micro	Contrôle des statu...	Statut d'alarm...	Zone de dispon...	DNS IPv4 public
<input type="checkbox"/>	Linux-Client	i-De43401ec7916ea75	En cours d'...	En cours d'...	En cours d'...	t2.micro	2/2 vérifications r	Afficher les alarm...	us-east-1a	ec2-23-20-149-162.co...
<input type="checkbox"/>	Wazuh-Server	i-0df27a43c8be3e319	En cours d'...	En cours d'...	En cours d'...	t3.large	3/3 vérifications r	Afficher les alarm...	us-east-1a	ec2-98-88-72-197.com...
<input type="checkbox"/>	Windows-Client	i-047a77c3dd9a24046	En cours d'...	En cours d'...	En cours d'...	t3.medium	3/3 vérifications r	Afficher les alarm...	us-east-1a	ec2-44-220-157-216.co...

Figure 7: toutes les instances

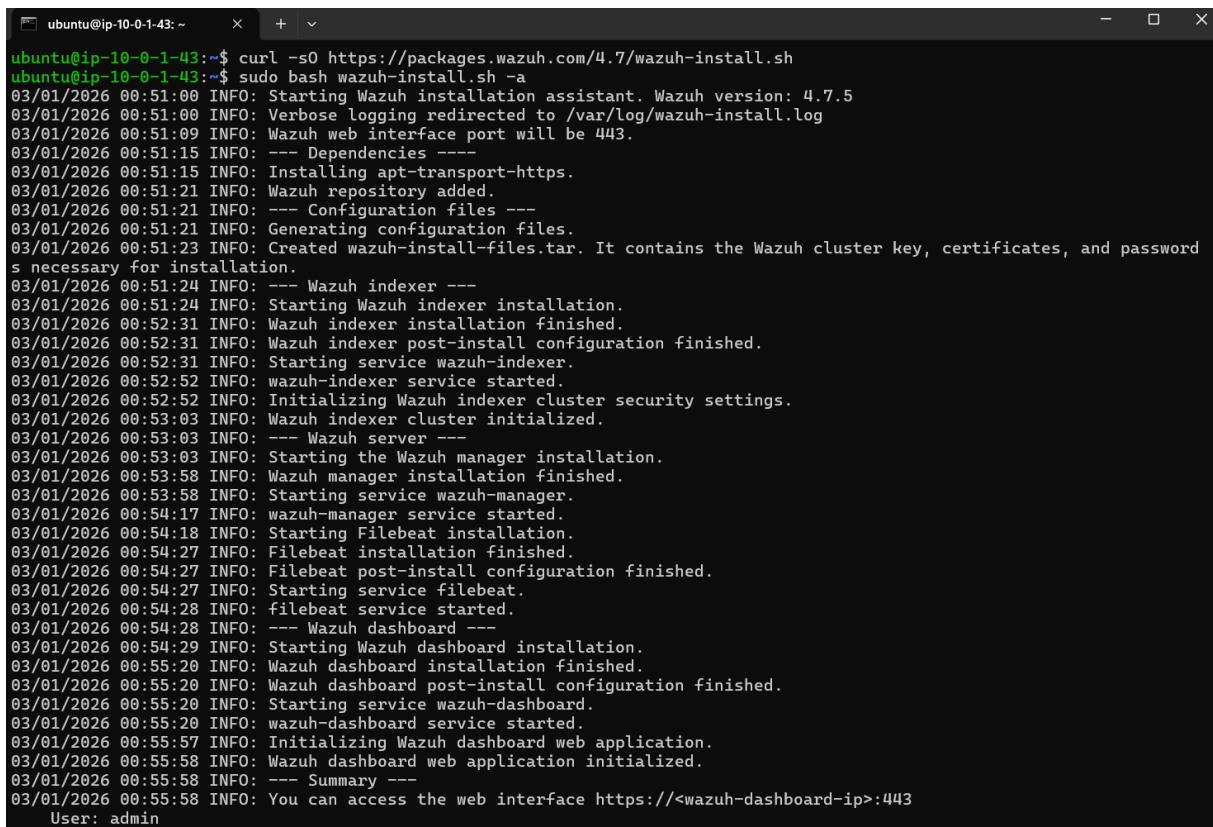
## 4. Installation Wazuh All-in-One sur Ubuntu (serveur)

```
C:\Users\NITRO 5\Desktop\Cloud Project>ssh -i Wazuh-Server-Cle.pem ubuntu@98.88.72.197
```

Figure 8: Connecter to server

```
ubuntu@ip-10-0-1-43:~$ curl -s0 https://packages.wazuh.com/4.7/wazuh-install.sh
```

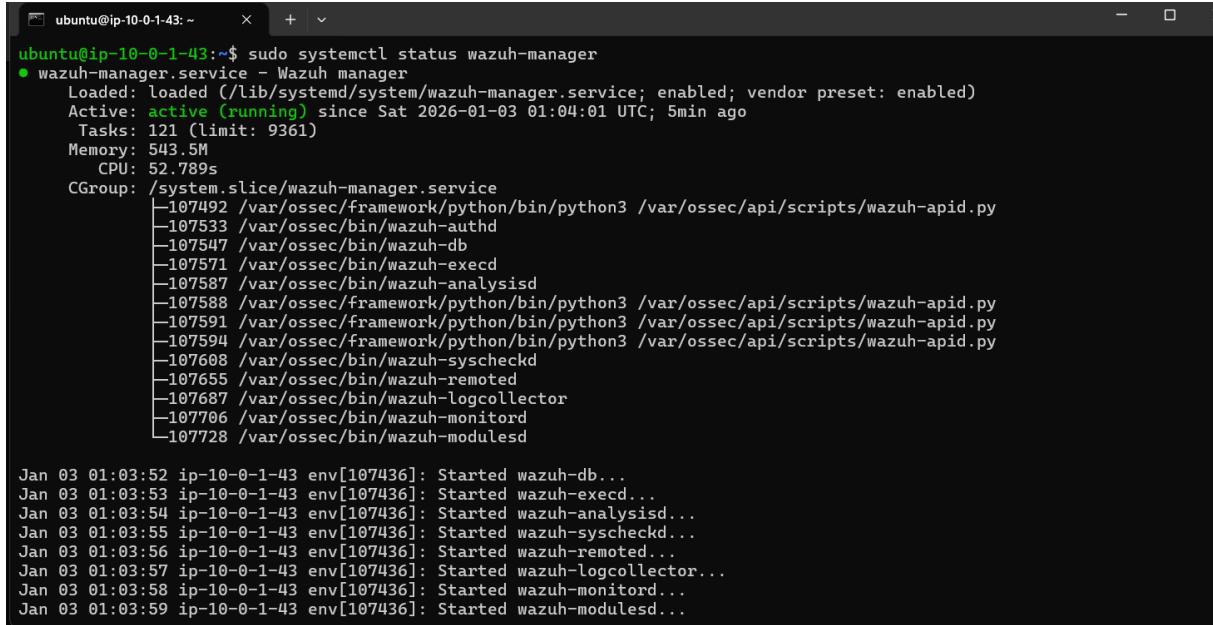
Figure 9: Telecharger Wazuh part1



A terminal window titled "ubuntu@ip-10-0-1-43:~" showing the output of a Wazuh installation script. The logs detail the process from curling the install.sh script to the final summary and access instructions.

```
ubuntu@ip-10-0-1-43:~$ curl -s0 https://packages.wazuh.com/4.7/wazuh-install.sh
ubuntu@ip-10-0-1-43:~$ sudo bash wazuh-install.sh -
03/01/2026 00:51:00 INFO: Starting Wazuh installation assistant. Wazuh version: 4.7.5
03/01/2026 00:51:00 INFO: Verbose logging redirected to /var/log/wazuh-install.log
03/01/2026 00:51:09 INFO: Wazuh web interface port will be 443.
03/01/2026 00:51:15 INFO: --- Dependencies ---
03/01/2026 00:51:15 INFO: Installing apt-transport-https.
03/01/2026 00:51:21 INFO: Wazuh repository added.
03/01/2026 00:51:21 INFO: --- Configuration files ---
03/01/2026 00:51:21 INFO: Generating configuration files.
03/01/2026 00:51:23 INFO: Created wazuh-install-files.tar. It contains the Wazuh cluster key, certificates, and password
s necessary for installation.
03/01/2026 00:51:24 INFO: --- Wazuh indexer ---
03/01/2026 00:51:24 INFO: Starting Wazuh indexer installation.
03/01/2026 00:52:31 INFO: Wazuh indexer installation finished.
03/01/2026 00:52:31 INFO: Wazuh indexer post-install configuration finished.
03/01/2026 00:52:31 INFO: Starting service wazuh-indexer.
03/01/2026 00:52:52 INFO: wazuh-indexer service started.
03/01/2026 00:52:52 INFO: Initializing Wazuh indexer cluster security settings.
03/01/2026 00:53:03 INFO: Wazuh indexer cluster initialized.
03/01/2026 00:53:03 INFO: --- Wazuh server ---
03/01/2026 00:53:03 INFO: Starting the Wazuh manager installation.
03/01/2026 00:53:58 INFO: Wazuh manager installation finished.
03/01/2026 00:53:58 INFO: Starting service wazuh-manager.
03/01/2026 00:54:17 INFO: wazuh-manager service started.
03/01/2026 00:54:18 INFO: Starting Filebeat installation.
03/01/2026 00:54:27 INFO: Filebeat installation finished.
03/01/2026 00:54:27 INFO: Filebeat post-install configuration finished.
03/01/2026 00:54:27 INFO: Starting service filebeat.
03/01/2026 00:54:28 INFO: filebeat service started.
03/01/2026 00:54:28 INFO: --- Wazuh dashboard ---
03/01/2026 00:54:29 INFO: Starting Wazuh dashboard installation.
03/01/2026 00:55:20 INFO: Wazuh dashboard installation finished.
03/01/2026 00:55:20 INFO: Wazuh dashboard post-install configuration finished.
03/01/2026 00:55:20 INFO: Starting service wazuh-dashboard.
03/01/2026 00:55:20 INFO: wazuh-dashboard service started.
03/01/2026 00:55:57 INFO: Initializing Wazuh dashboard web application.
03/01/2026 00:55:58 INFO: Wazuh dashboard web application initialized.
03/01/2026 00:55:58 INFO: --- Summary ---
03/01/2026 00:55:58 INFO: You can access the web interface https://<wazuh-dashboard-ip>:443
User: admin
```

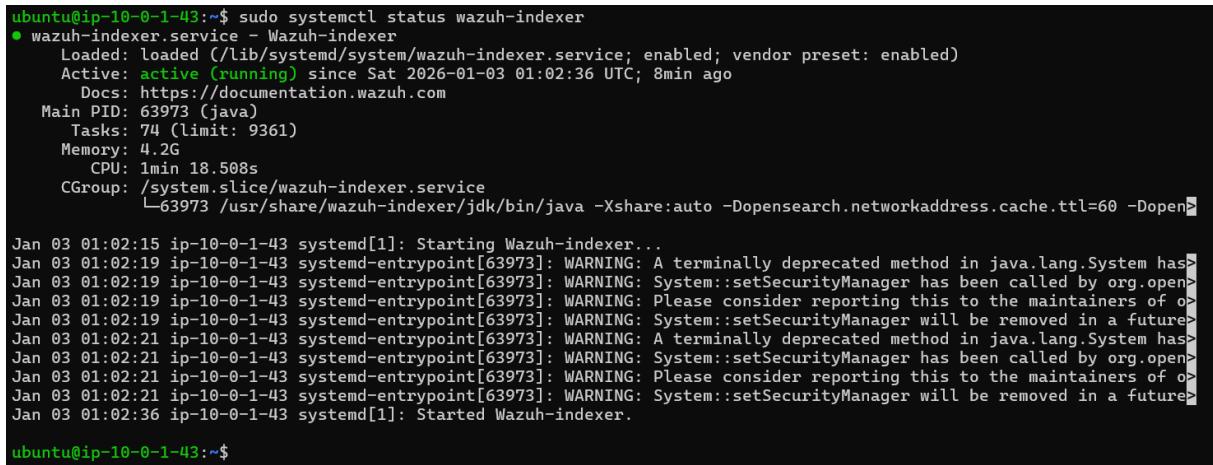
Figure 10: Telecharger Wazuh part2



```
ubuntu@ip-10-0-1-43:~$ sudo systemctl status wazuh-manager
● wazuh-manager.service - Wazuh manager
   Loaded: loaded (/lib/systemd/system/wazuh-manager.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2026-01-03 01:04:01 UTC; 5min ago
     Tasks: 121 (limit: 9361)
    Memory: 543.5M
      CPU: 52.789s
     CGroup: /system.slice/wazuh-manager.service
             ├─107492 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
             ├─107533 /var/ossec/bin/wazuh-authd
             ├─107547 /var/ossec/bin/wazuh-db
             ├─107571 /var/ossec/bin/wazuh-execd
             ├─107587 /var/ossec/bin/wazuh-analysisd
             ├─107588 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
             ├─107591 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
             ├─107594 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
             ├─107608 /var/ossec/bin/wazuh-syscheckd
             ├─107655 /var/ossec/bin/wazuh-remoted
             ├─107687 /var/ossec/bin/wazuh-logcollector
             ├─107706 /var/ossec/bin/wazuh-monitord
             └─107728 /var/ossec/bin/wazuh-modulesd

Jan 03 01:03:52 ip-10-0-1-43 env[107436]: Started wazuh-db...
Jan 03 01:03:53 ip-10-0-1-43 env[107436]: Started wazuh-execd...
Jan 03 01:03:54 ip-10-0-1-43 env[107436]: Started wazuh-analysisd...
Jan 03 01:03:55 ip-10-0-1-43 env[107436]: Started wazuh-syscheckd...
Jan 03 01:03:56 ip-10-0-1-43 env[107436]: Started wazuh-remoted...
Jan 03 01:03:57 ip-10-0-1-43 env[107436]: Started wazuh-logcollector...
Jan 03 01:03:58 ip-10-0-1-43 env[107436]: Started wazuh-monitord...
Jan 03 01:03:59 ip-10-0-1-43 env[107436]: Started wazuh-modulesd...
```

Figure 11: Vérifier l'installation Wazuh Manager



```
ubuntu@ip-10-0-1-43:~$ sudo systemctl status wazuh-indexer
● wazuh-indexer.service - Wazuh-indexer
   Loaded: loaded (/lib/systemd/system/wazuh-indexer.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2026-01-03 01:02:36 UTC; 8min ago
     Docs: https://documentation.wazuh.com
   Main PID: 63973 (java)
      Tasks: 74 (limit: 9361)
     Memory: 4.2G
       CPU: 1min 18.508s
      CGroup: /system.slice/wazuh-indexer.service
              └─63973 /usr/share/wazuh-indexer/jdk/bin/java -Xshare:auto -Dopensearch.networkaddress.cache.ttl=60 -Dopen...
```

Jan 03 01:02:15 ip-10-0-1-43 systemd[1]: Starting Wazuh-indexer...
Jan 03 01:02:19 ip-10-0-1-43 systemd-entrapoint[63973]: WARNING: A terminally deprecated method in java.lang.System has...
Jan 03 01:02:19 ip-10-0-1-43 systemd-entrapoint[63973]: WARNING: System:::setSecurityManager has been called by org.open...

Jan 03 01:02:19 ip-10-0-1-43 systemd-entrapoint[63973]: WARNING: Please consider reporting this to the maintainers of o...

Jan 03 01:02:19 ip-10-0-1-43 systemd-entrapoint[63973]: WARNING: System:::setSecurityManager will be removed in a future...

Jan 03 01:02:21 ip-10-0-1-43 systemd-entrapoint[63973]: WARNING: A terminally deprecated method in java.lang.System has...

Jan 03 01:02:21 ip-10-0-1-43 systemd-entrapoint[63973]: WARNING: System:::setSecurityManager has been called by org.open...

Jan 03 01:02:21 ip-10-0-1-43 systemd-entrapoint[63973]: WARNING: Please consider reporting this to the maintainers of o...

Jan 03 01:02:21 ip-10-0-1-43 systemd-entrapoint[63973]: WARNING: System:::setSecurityManager will be removed in a future...

Jan 03 01:02:36 ip-10-0-1-43 systemd[1]: Started Wazuh-indexer.

Figure 12: Vérifier l'installation Wazuh Indexer

```
ubuntu@ip-10-0-1-43:~$ sudo systemctl status wazuh-dashboard
● wazuh-dashboard.service - wazuh-dashboard
   Loaded: loaded (/etc/systemd/system/wazuh-dashboard.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2026-01-03 01:05:20 UTC; 7min ago
     Main PID: 109641 (node)
        Tasks: 11 (limit: 9361)
       Memory: 173.9M
          CPU: 14.489s
        CGroup: /system.slice/wazuh-dashboard.service
                  └─109641 /usr/share/wazuh-dashboard/node/bin/node --no-warnings --max-http-header-size=65536 --unhandled-r>

Jan 03 01:07:49 ip-10-0-1-43 opensearch-dashboards[109641]: {"type":"response","@timestamp":"2026-01-03T01:07:49Z","tag>
Jan 03 01:07:50 ip-10-0-1-43 opensearch-dashboards[109641]: {"type":"response","@timestamp":"2026-01-03T01:07:50Z","tag>
Jan 03 01:07:50 ip-10-0-1-43 opensearch-dashboards[109641]: {"type":"response","@timestamp":"2026-01-03T01:07:50Z","tag>
Jan 03 01:10:00 ip-10-0-1-43 opensearch-dashboards[109641]: {"type":"log","@timestamp":"2026-01-03T01:10:00Z","tags":[">
ubuntu@ip-10-0-1-43:~$
```

Figure 13: Vérifier l'installation Wazuh Dashboard

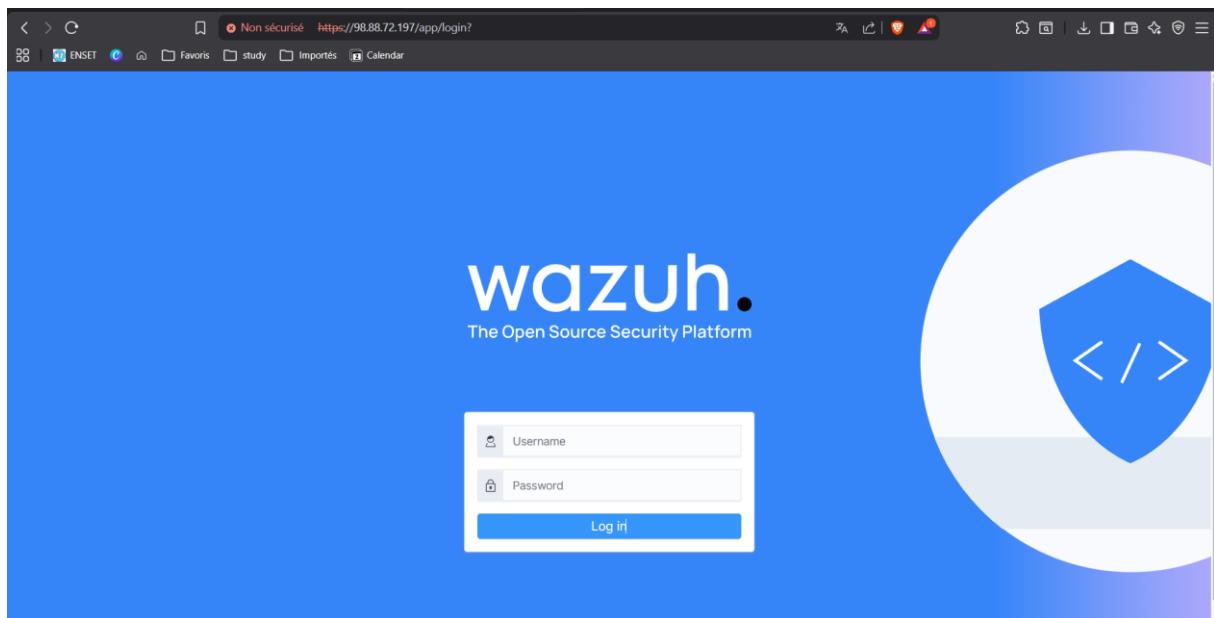


Figure 14: Wazuh login

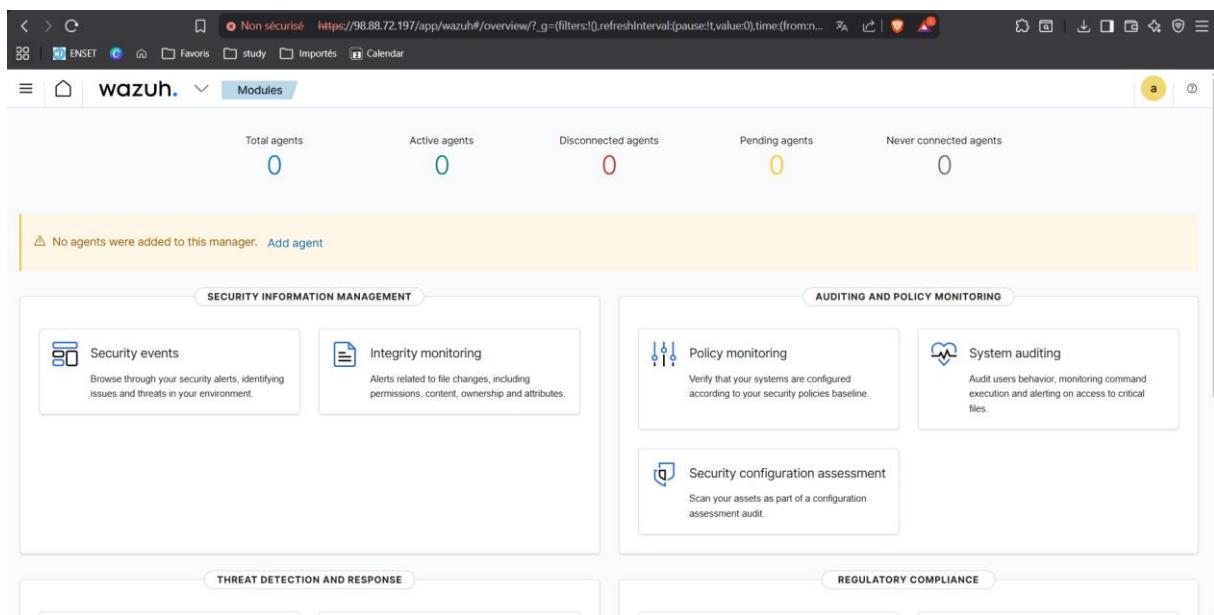


Figure 15: Wazuh dashboard

## Enrôle agents

### 5. Enrôler le client Linux (Ubuntu)

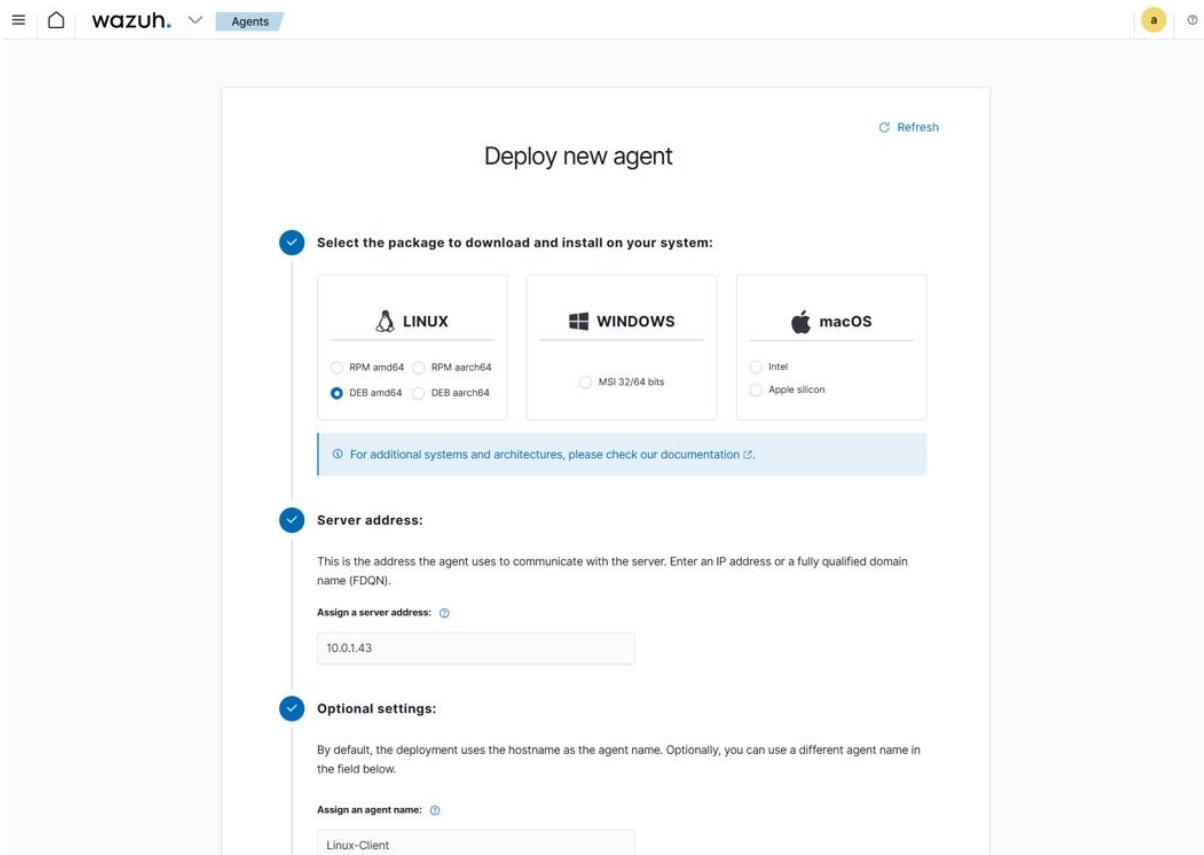
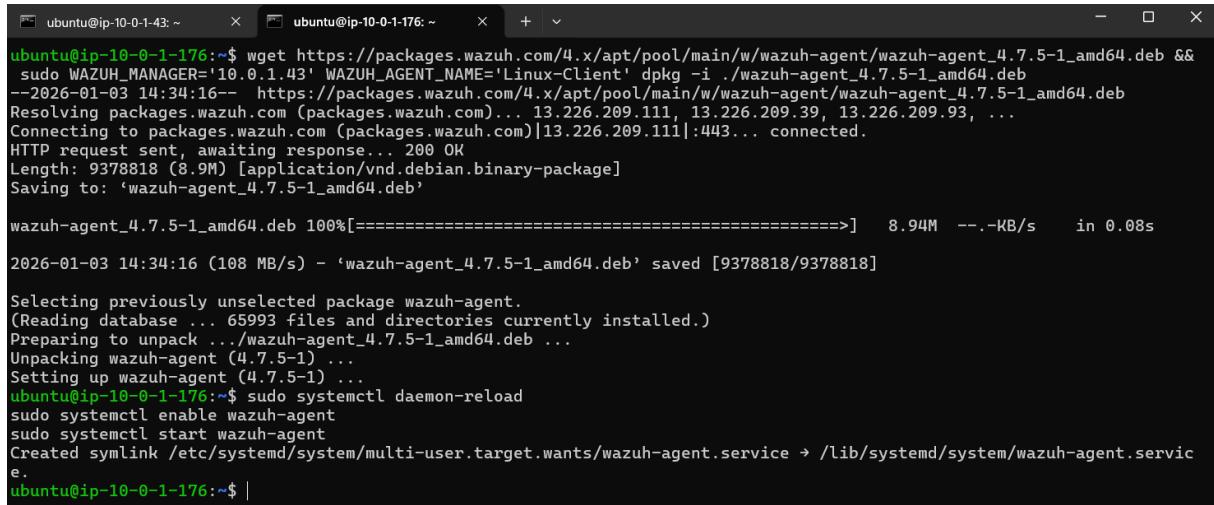


Figure 16: Configurer nouveau agent (linux client)



```
ubuntu@ip-10-0-1-43: ~      x  ubuntu@ip-10-0-1-176: ~      x  +  ~
ubuntu@ip-10-0-1-176:~$ wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.7.5-1_amd64.deb &&
  sudo WAZUH_MANAGER='10.0.1.43' WAZUH_AGENT_NAME='Linux-Client' dpkg -i ./wazuh-agent_4.7.5-1_amd64.deb
--2026-01-03 14:34:16-- https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.7.5-1_amd64.deb
Resolving packages.wazuh.com (packages.wazuh.com)... 13.226.209.111, 13.226.209.39, 13.226.209.93, ...
Connecting to packages.wazuh.com (packages.wazuh.com)|13.226.209.111|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9378818 (8.9M) [application/vnd.debian.binary-package]
Saving to: 'wazuh-agent_4.7.5-1_amd64.deb'

wazuh-agent_4.7.5-1_amd64.deb 100%[=====] 8.94M --.-KB/s   in 0.08s
2026-01-03 14:34:16 (108 MB/s) - 'wazuh-agent_4.7.5-1_amd64.deb' saved [9378818/9378818]

Selecting previously unselected package wazuh-agent.
(Reading database ... 65993 files and directories currently installed.)
Preparing to unpack .../wazuh-agent_4.7.5-1_amd64.deb ...
Unpacking wazuh-agent (4.7.5-1) ...
Setting up wazuh-agent (4.7.5-1) ...
ubuntu@ip-10-0-1-176:~$ sudo systemctl daemon-reload
sudo systemctl enable wazuh-agent
sudo systemctl start wazuh-agent
Created symlink /etc/systemd/system/multi-user.target.wants/wazuh-agent.service → /lib/systemd/system/wazuh-agent.service.
ubuntu@ip-10-0-1-176:~$ |
```

Figure 17: installer et lancer agent (linux client)

## 6. Enrôler le client Windows (agent + événements)

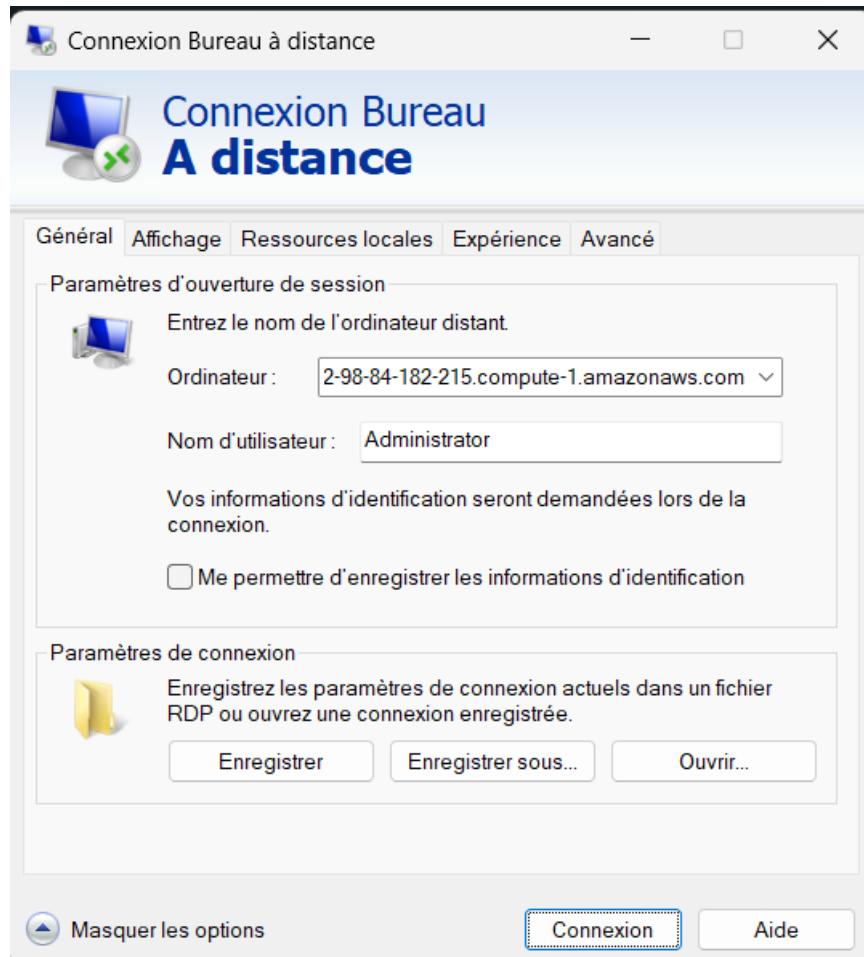


Figure 18: Connecter to windows client depuis RDP

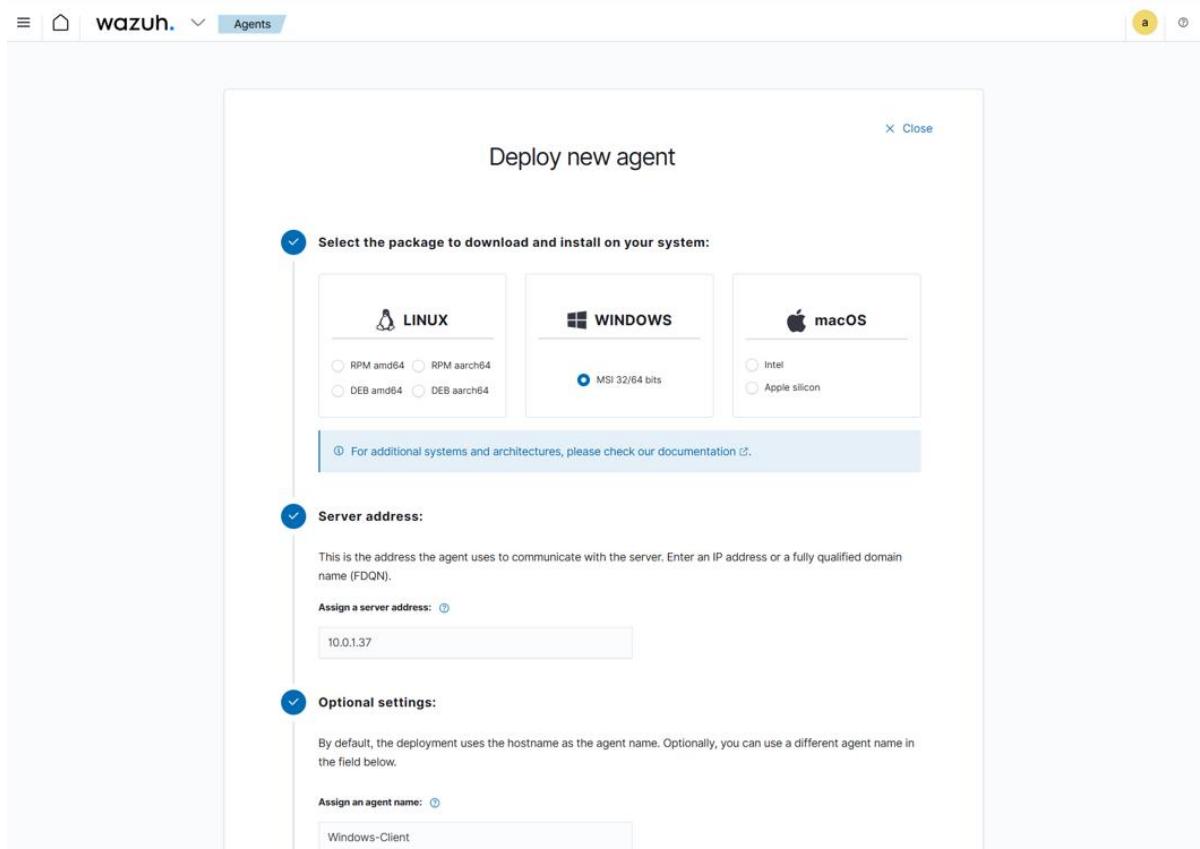


Figure 19: Configurer nouveau agent (windows client)

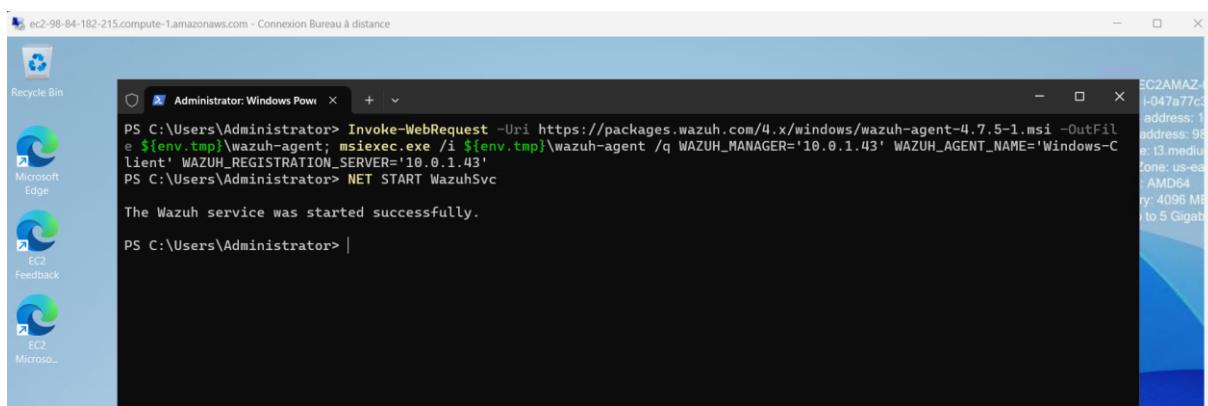


Figure 20: installer et lancer agent (windows client)

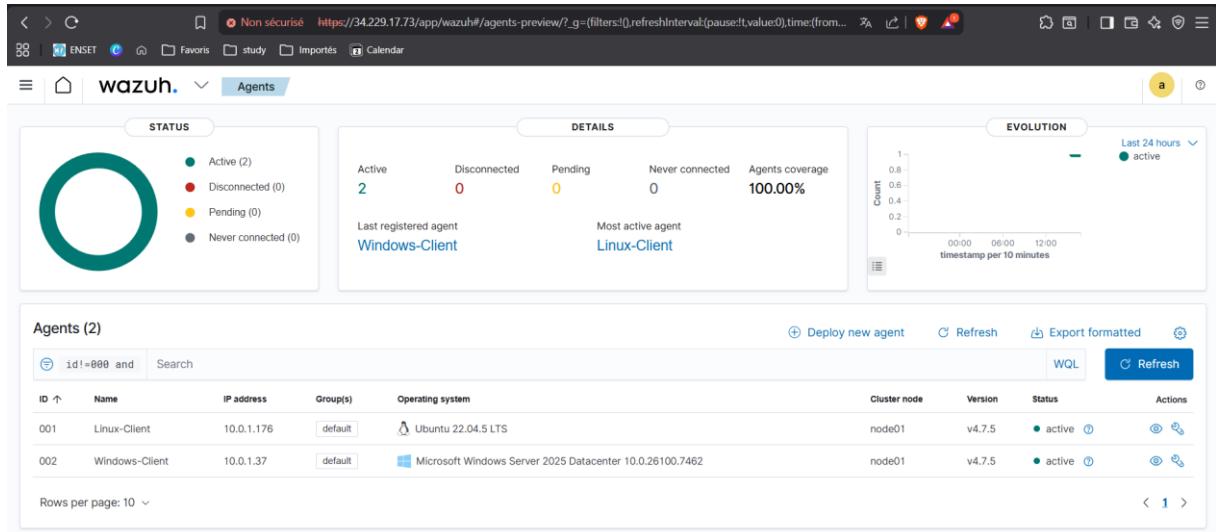
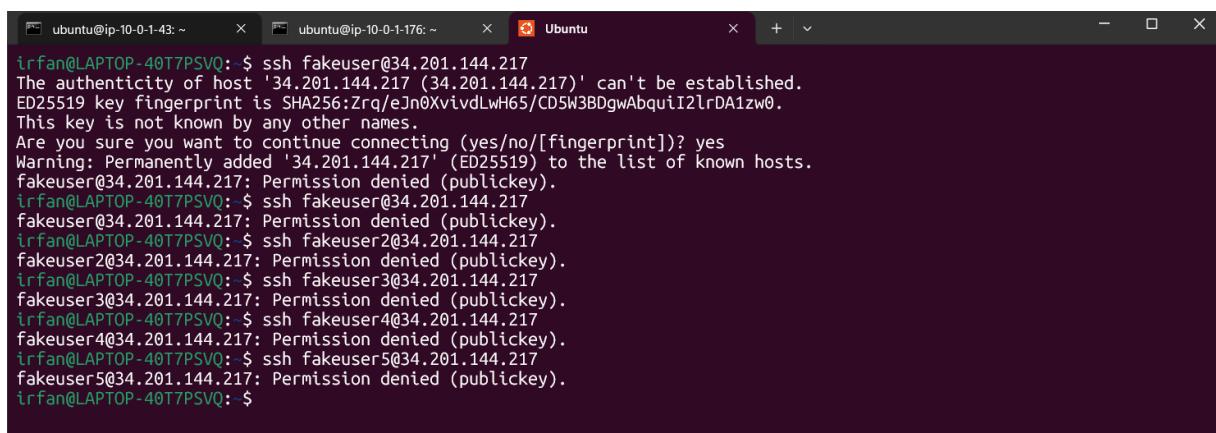


Figure 21: Vérifier agents

## Démo SIEM + EDR : scénarios d'événements à générer

### 7. Démo SIEM côté Linux (rapide, visible tout de suite)

#### Scénario 1 — Tentatives SSH échouées (bruteforce simulé)



```
trfan@LAPTOP-40T7PSVQ: $ ssh fakeuser@34.201.144.217
The authenticity of host '34.201.144.217 (34.201.144.217)' can't be established.
ED25519 key fingerprint is SHA256:Zrq/eJn0XvivdLwH65/CD5W3BDgwAbquiI2lrDA1zw0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '34.201.144.217' (ED25519) to the list of known hosts.
fakeuser@34.201.144.217: Permission denied (publickey).
trfan@LAPTOP-40T7PSVQ: $ ssh fakeuser@34.201.144.217
fakeuser@34.201.144.217: Permission denied (publickey).
trfan@LAPTOP-40T7PSVQ: $ ssh fakeuser2@34.201.144.217
fakeuser2@34.201.144.217: Permission denied (publickey).
trfan@LAPTOP-40T7PSVQ: $ ssh fakeuser3@34.201.144.217
fakeuser3@34.201.144.217: Permission denied (publickey).
trfan@LAPTOP-40T7PSVQ: $ ssh fakeuser4@34.201.144.217
fakeuser4@34.201.144.217: Permission denied (publickey).
trfan@LAPTOP-40T7PSVQ: $ ssh fakeuser5@34.201.144.217
fakeuser5@34.201.144.217: Permission denied (publickey).
trfan@LAPTOP-40T7PSVQ: $
```

Figure 22: Test connecter depuis fake user

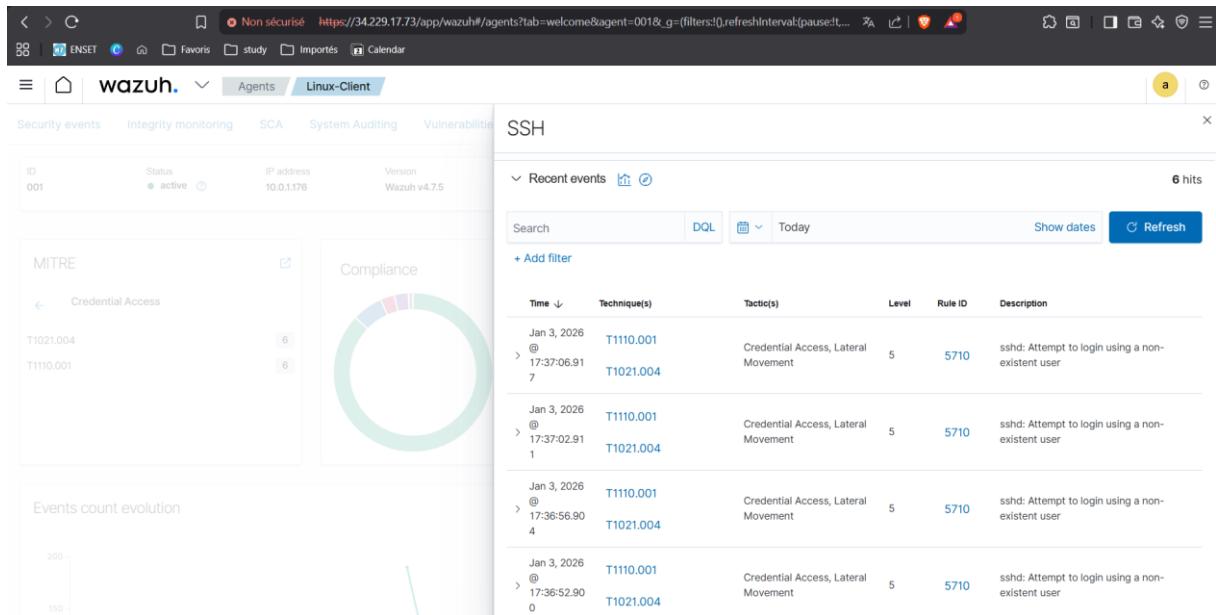
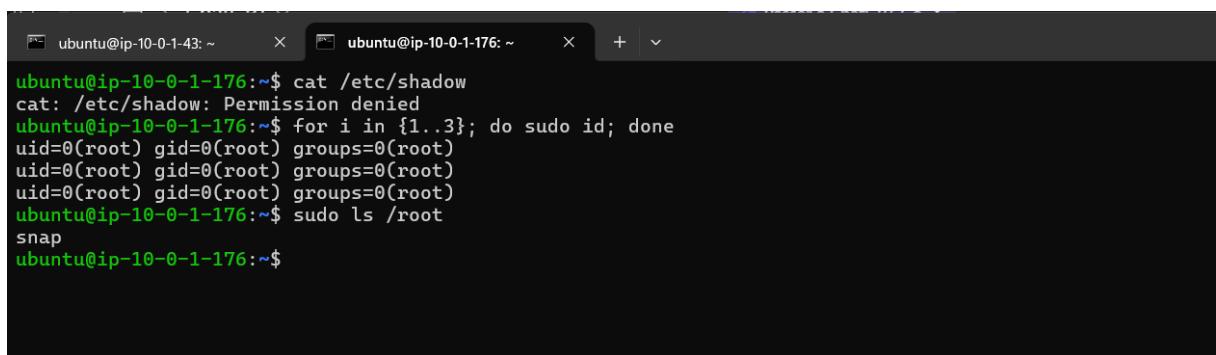


Figure 23: alertes authentication failed

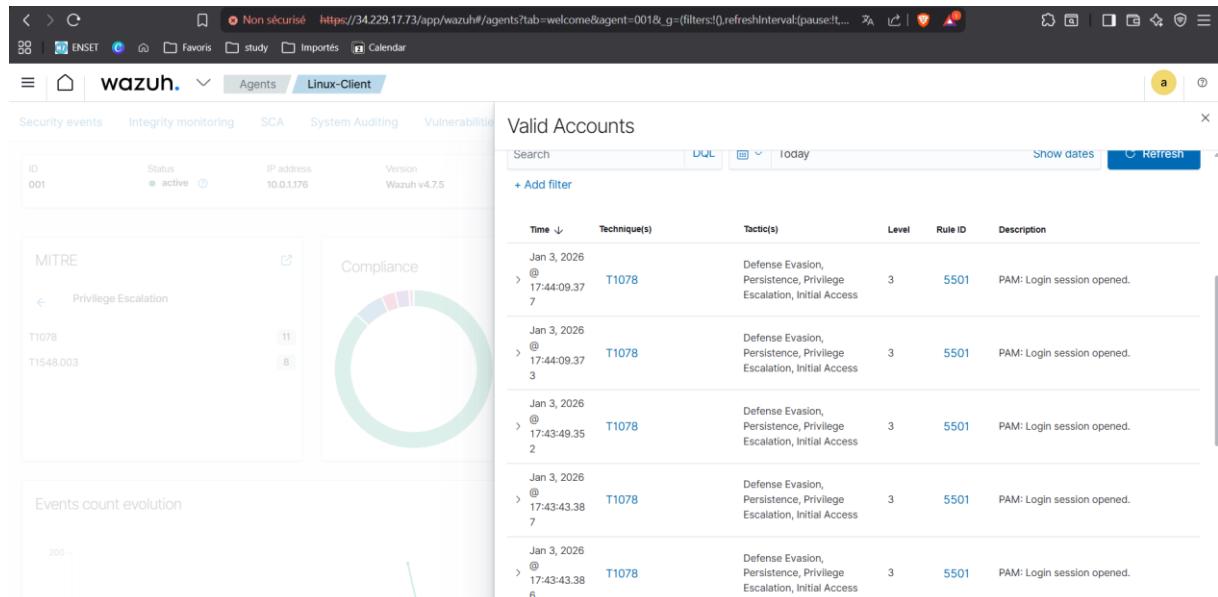
## Scénario 2 — Élévation de privilège



The terminal session shows a user attempting to escalate privileges. The user runs 'cat /etc/shadow' which is denied ('Permission denied'). Then, the user runs a loop command to repeatedly run 'sudo id' to find a root shell. Finally, the user runs 'sudo ls /root' and lists files in the root directory, including 'snap'.

```
ubuntu@ip-10-0-1-43:~$ cat /etc/shadow
cat: /etc/shadow: Permission denied
ubuntu@ip-10-0-1-176:~$ for i in {1..3}; do sudo id; done
uid=0(root) gid=0(root) groups=0(root)
uid=0(root) gid=0(root) groups=0(root)
uid=0(root) gid=0(root) groups=0(root)
ubuntu@ip-10-0-1-176:~$ sudo ls /root
snap
ubuntu@ip-10-0-1-176:~$
```

Figure 24: test élévation de privilèges



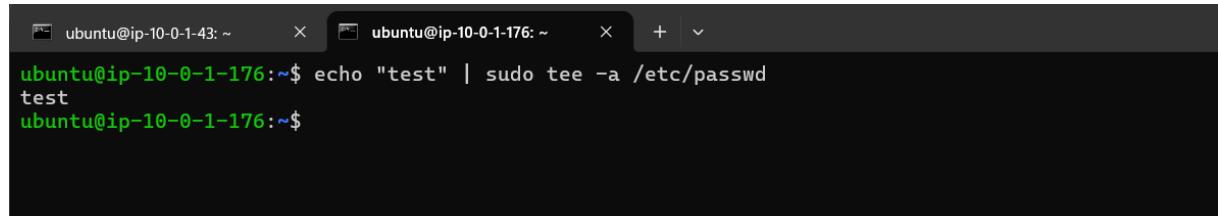
The screenshot shows the Wazuh web interface with the following details:

- Valid Accounts:**
  - Search: DQL, today
  - Show dates: Refresh
  - + Add filter

Time	Technique(s)	Tactic(s)	Level	Rule ID	Description
Jan 3, 2026 17:44:09.37 7	T1078	Defense Evasion, Persistence, Privilege Escalation, Initial Access	3	5501	PAM: Login session opened.
Jan 3, 2026 17:44:09.37 3	T1078	Defense Evasion, Persistence, Privilege Escalation, Initial Access	3	5501	PAM: Login session opened.
Jan 3, 2026 17:43:49.35 2	T1078	Defense Evasion, Persistence, Privilege Escalation, Initial Access	3	5501	PAM: Login session opened.
Jan 3, 2026 17:43:43.38 7	T1078	Defense Evasion, Persistence, Privilege Escalation, Initial Access	3	5501	PAM: Login session opened.
Jan 3, 2026 17:43:43.38 6	T1078	Defense Evasion, Persistence, Privilege Escalation, Initial Access	3	5501	PAM: Login session opened.
- MITRE:**
  - Privilege Escalation
  - T1078
  - T1548.003
- Compliance:** A donut chart showing compliance levels.
- Events count evolution:** A line chart showing event counts over time.

Figure 25: alertes événements sudo

### Scénario 3 — Modification fichier sensible (FIM)



```
ubuntu@ip-10-0-1-43: ~      x  ubuntu@ip-10-0-1-176: ~      x  +  v
ubuntu@ip-10-0-1-176:~$ echo "test" | sudo tee -a /etc/passwd
test
ubuntu@ip-10-0-1-176:~$
```

Figure 26: test modification fichier sensible

The screenshot shows the Wazuh web interface under the 'Agents / Linux-Client' tab. A prominent alert titled 'Sudo and Sudo Caching' is displayed, indicating a 'Privilege Escalation' tactic using 'Defense Evasion'. The alert details a successful sudo command to root at 17:48:09.62 on Jan 3, 2026. The interface includes a sidebar with MITRE ATT&CK data, a compliance donut chart, and an events count evolution graph.

Figure 27: alerte Surveillance de l'intégrité des fichiers

The screenshot shows the Wazuh dashboard for a Linux client (Ubuntu 22.04 LTS). It displays a summary of agent status, system information, and security metrics. The 'MITRE' section shows top tactics like Defense Evasion (30 occurrences) and Privilege Escalation (29). The 'FIM: Recent events' section shows no recent events. The 'SCA: Lastest scans' section shows a CIS Ubuntu Linux 22.04 LTS Benchmark v1.0.0 scan with a score of 74%. The interface also includes a sidebar with MITRE ATT&CK data and an events count evolution graph.

Figure 28: dashboard des alertes (linux client)

## 8. Démo EDR côté Windows (événements sécurité + option Sysmon)

### Scénario 1 — Échecs de login (4625)

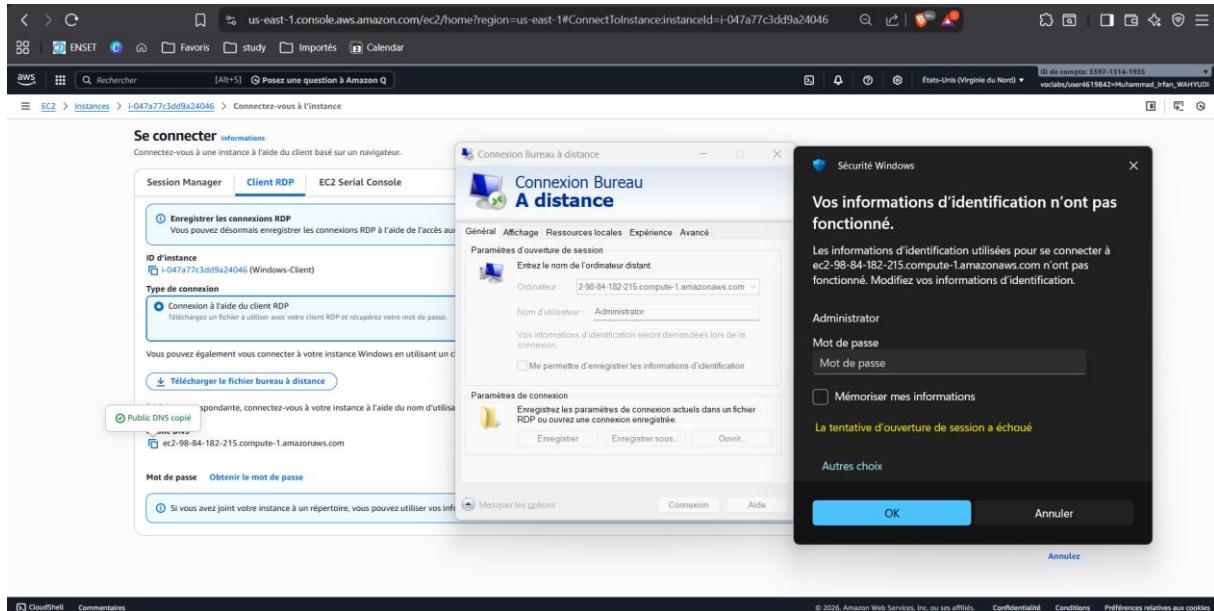
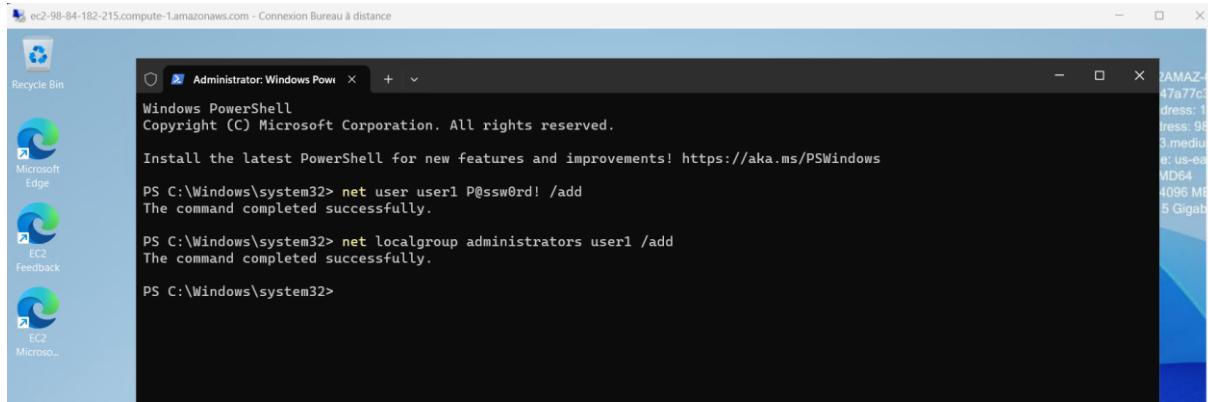


Figure 29: test to connecter depuis fake user

Time	Technique(s)	Tactic(s)	Level	Rule ID	Description
Jan 3, 2026 17:56:56.92 7	@ T1078 T1531	Defense Evasion, Persistence, Privilege Escalation, Initial Access, Impact	5	60122	Logon failure - Unknown user or bad password.
Jan 3, 2026 17:56:53.60 8	@ T1078 T1531	Defense Evasion, Persistence, Privilege Escalation, Initial Access, Impact	5	60122	Logon failure - Unknown user or bad password.
Jan 3, 2026 17:56:44.65 3	@ T1078 T1531	Defense Evasion, Persistence, Privilege Escalation, Initial Access, Impact	5	60122	Logon failure - Unknown user or bad password.

Figure 30: alertes événements Windows Security (Failed logon)

## Scénario 2 — Crédation d'un utilisateur local



```

ec2-98-84-182-215.compute-1.amazonaws.com - Connexion Bureau à distance

Administrator: Windows PowerShell

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

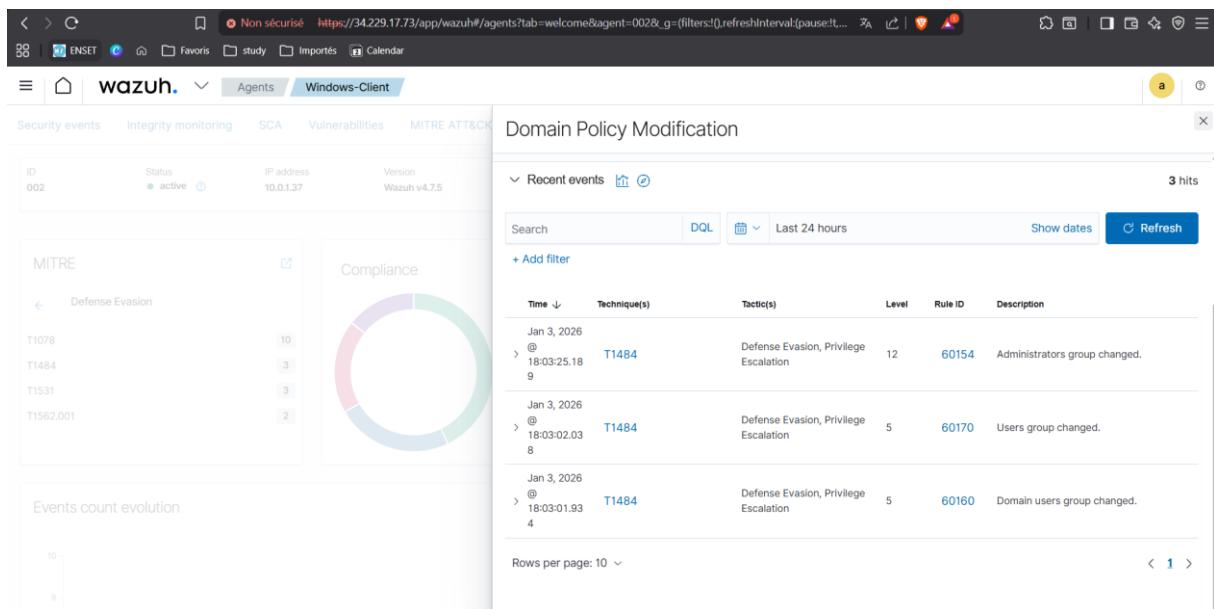
PS C:\Windows\system32> net user user1 P@ssw0rd! /add
The command completed successfully.

PS C:\Windows\system32> net localgroup administrators user1 /add
The command completed successfully.

PS C:\Windows\system32>

```

Figure 31: test Création d'un utilisateur local



Non sécurisé https://34.229.17.73/app/wazuh#/agents?tab=welcome&agent=002&.g=filters!().refreshInterval(pause:1000)

Wazuh Agents Windows-Client

Domain Policy Modification

Recent events 3 hits

Time	Technique(s)	Tactic(s)	Level	Rule ID	Description
Jan 3, 2026 @ 18:03:25.189	T1484	Defense Evasion, Privilege Escalation	12	60154	Administrators group changed.
Jan 3, 2026 @ 18:03:02.038	T1484	Defense Evasion, Privilege Escalation	5	60170	Users group changed.
Jan 3, 2026 @ 18:03:01.934	T1484	Defense Evasion, Privilege Escalation	5	60160	Domain users group changed.

MITRE

Defense Evasion

T1078 (10), T1484 (3), T1531 (3), T1562.001 (2)

Events count evolution

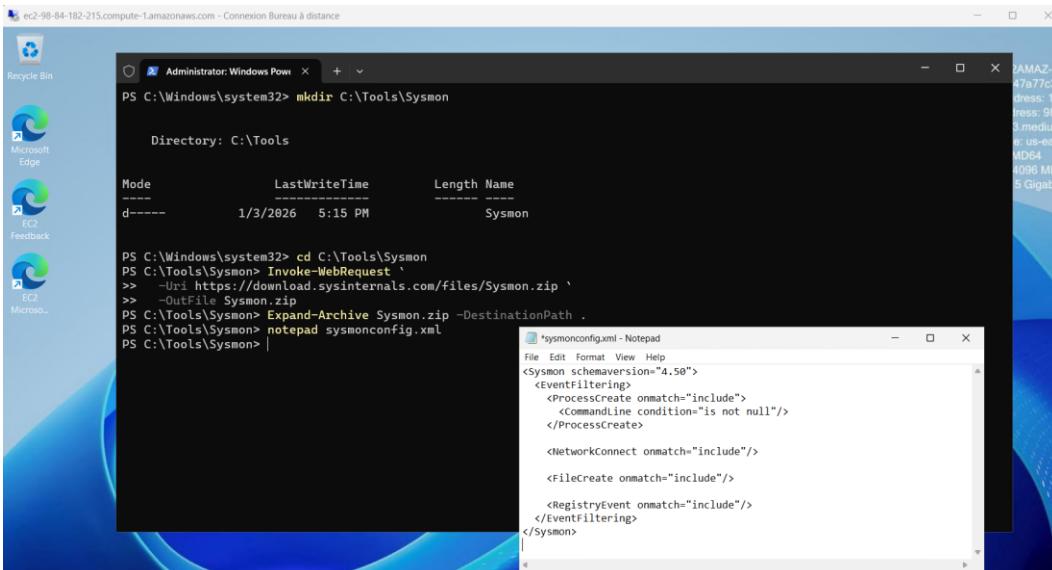
Rows per page: 10 < 1 >

Figure 32: alertes événements “user created / group changed”

## EDR Avancé avec Sysmon

L'option “EDR” consiste à installer **Sysmon**, un outil Microsoft Sysinternals fournit une télémétrie détaillée sur l'activité système (création de processus, connexions réseau, modifications de fichiers, chargement de DLL et activité du registre). Ces événements sont ensuite collectés par **Wazuh**, permettant une démonstration complète des capacités **EDR**.

### 9. Installation de sysmon



```

ec2-98-84-182-215.compute-1.amazonaws.com - Connexion Bureau à distance

Administrator: Windows Pow
PS C:\Windows\system32> mkdir C:\Tools\Sysmon

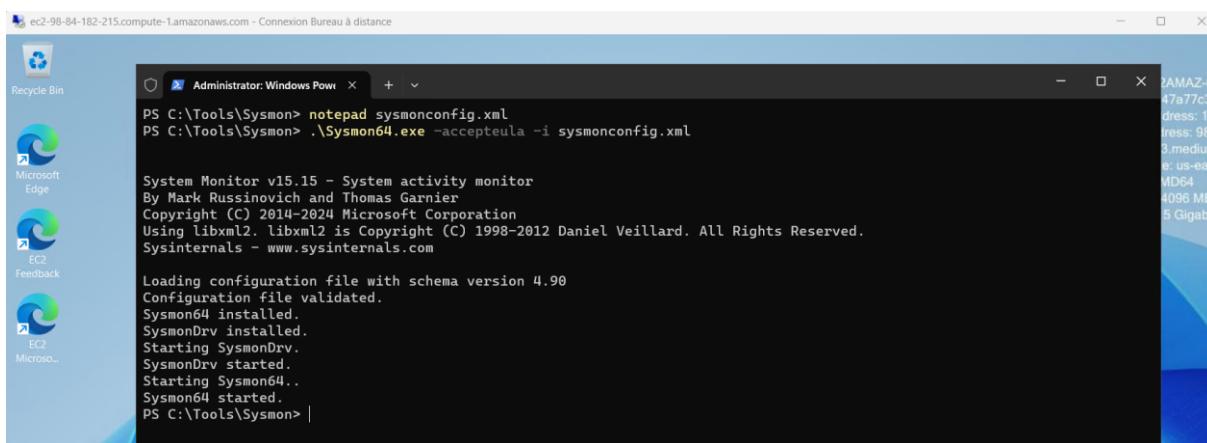
Directory: C:\Tools

Mode LastWriteTime Length Name
d---- 1/3/2026 5:15 PM Sysmon

PS C:\Windows\system32> cd C:\Tools\Sysmon
PS C:\Tools\Sysmon> Invoke-WebRequest `
>> -Uri https://download.sysinternals.com/files/Sysmon.zip `
>> -OutFile Sysmon.zip
PS C:\Tools\Sysmon> Expand-Archive Sysmon.zip -DestinationPath .
PS C:\Tools\Sysmon> notepad sysmonconfig.xml
PS C:\Tools\Sysmon>

```

Figure 33: Installation de sysmon part1



```

ec2-98-84-182-215.compute-1.amazonaws.com - Connexion Bureau à distance

Administrator: Windows Pow
PS C:\Tools\Sysmon> notepad sysmonconfig.xml
PS C:\Tools\Sysmon> .\Sysmon64.exe -accepteula -i sysmonconfig.xml

System Monitor v15.15 - System activity monitor
By Mark Russinovich and Thomas Garnier
Copyright (C) 2014-2024 Microsoft Corporation
Using libxml2. libxml2 is Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.
Sysinternals - www.sysinternals.com

Loading configuration file with schema version 4.90
Configuration file validated.
Sysmon64 installed.
SysmonDrv installed.
Starting SysmonDrv.
SysmonDrv started.
Starting Sysmon64..
Sysmon64 started.
PS C:\Tools\Sysmon>

```

Figure 34: Installation de sysmon part2

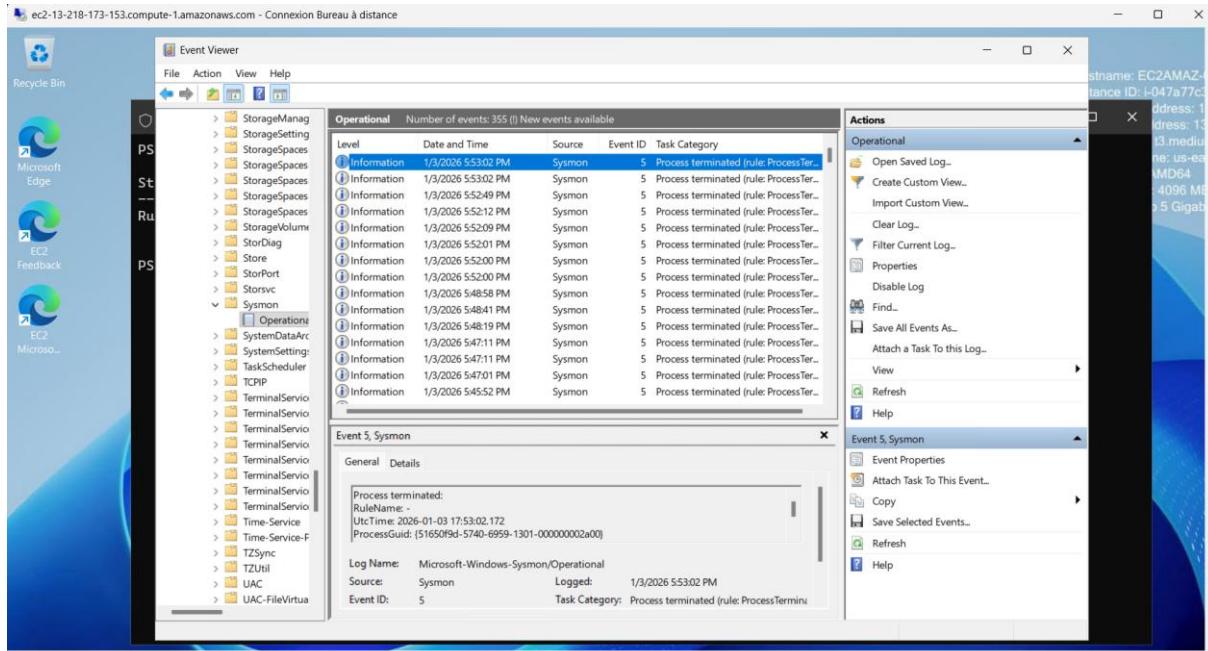


Figure 35: Sysmon logs

## 10. Intégrer Sysmon sur Wazuh

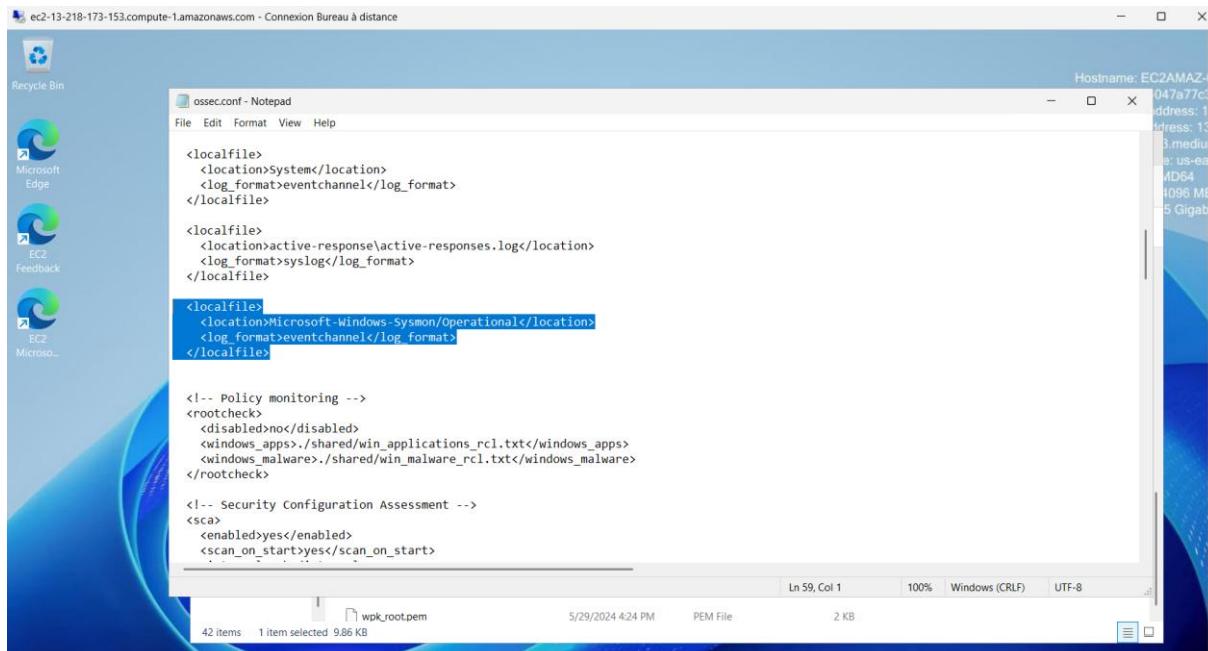


Figure 36: Intégrer Sysmon sur Wazuh part1

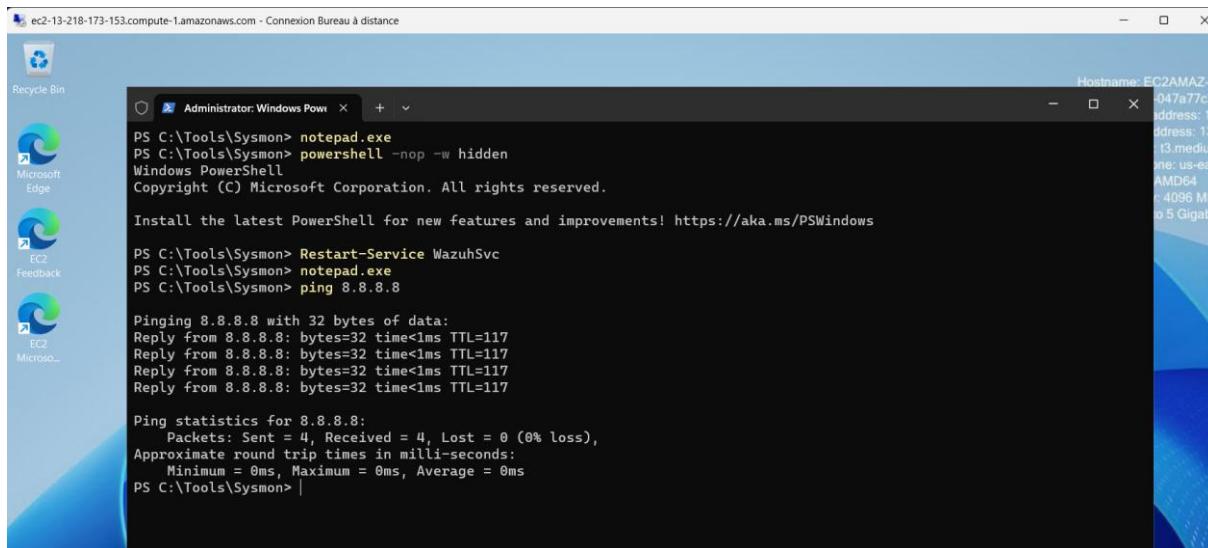


Figure 37: Intégrer Sysmon sur Wazuh part2

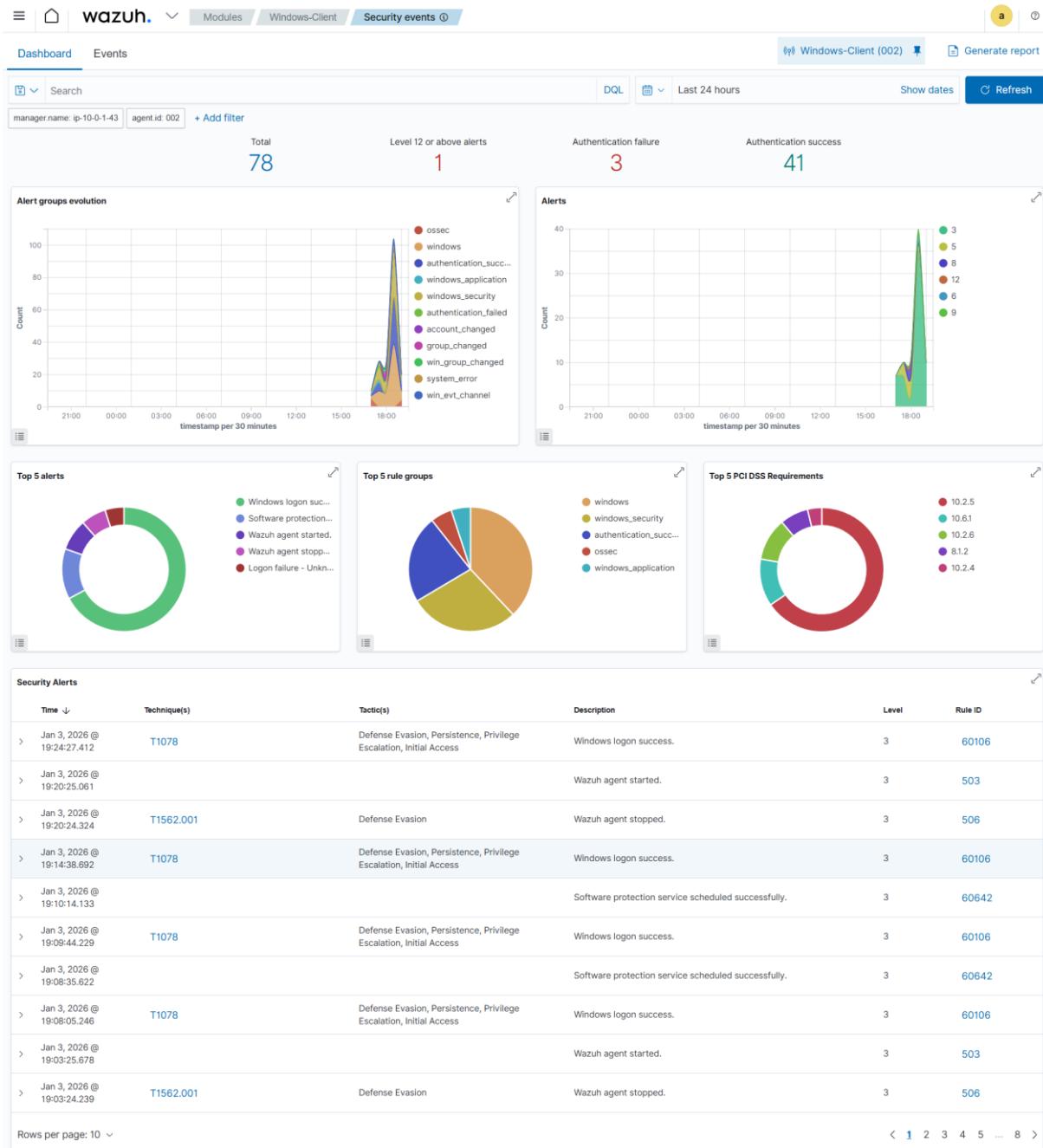


Figure 38: Wazuh dashboard (windows client)

# SIEM, EDR et Gestion des Accès (IAM / PAM)

## 1. SIEM vs EDR

Un **SIEM** (Security Information and Event Management) collecte, centralise et corrèle les logs provenant de serveurs, applications, firewalls et services cloud pour générer des alertes et rapports. Exemple avec **Wazuh** : centralisation des logs, détection d'attaques et alertes de conformité.

Un **EDR** (Endpoint Detection and Response) surveille directement les postes (Windows, Linux), analyse le comportement des processus, détecte les attaques avancées et permet la réponse aux incidents. Exemple avec **Sysmon + Wazuh** : collecte des événements système, analyse et alertes sur les endpoints.

**Déférence clé :** le SIEM offre une vue globale et corrélée, tandis que l'EDR protège les machines individuelles. Les deux sont **complémentaires**.

## 2. IAM et PAM

**IAM** (Identity and Access Management) gère les identités, l'authentification et les autorisations d'accès pour tous les utilisateurs. Exemple : utilisateurs AWS IAM, rôles, politiques et MFA.

**PAM** (Privileged Access Management) sécurise les comptes à privilèges élevés, en limitant l'abus, en assurant la traçabilité et en fournissant un accès temporaire. Exemple : sudo contrôlé, sessions admin surveillées, rotation des mots de passe.

**Déférence clé :** IAM concerne tous les utilisateurs et contrôle l'accès général, PAM protège spécifiquement les comptes privilégiés et prévient l'escalade de privilèges.

### 3. Threat Hunting (3 requêtes simples)

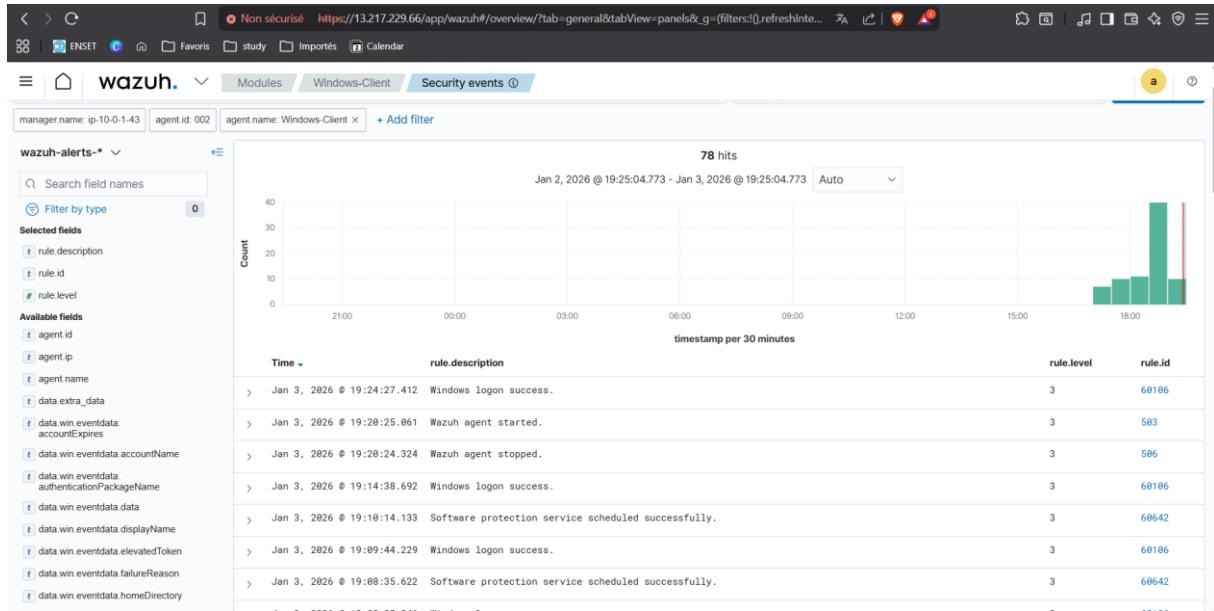


Figure 39: Threat Hunting requête 1

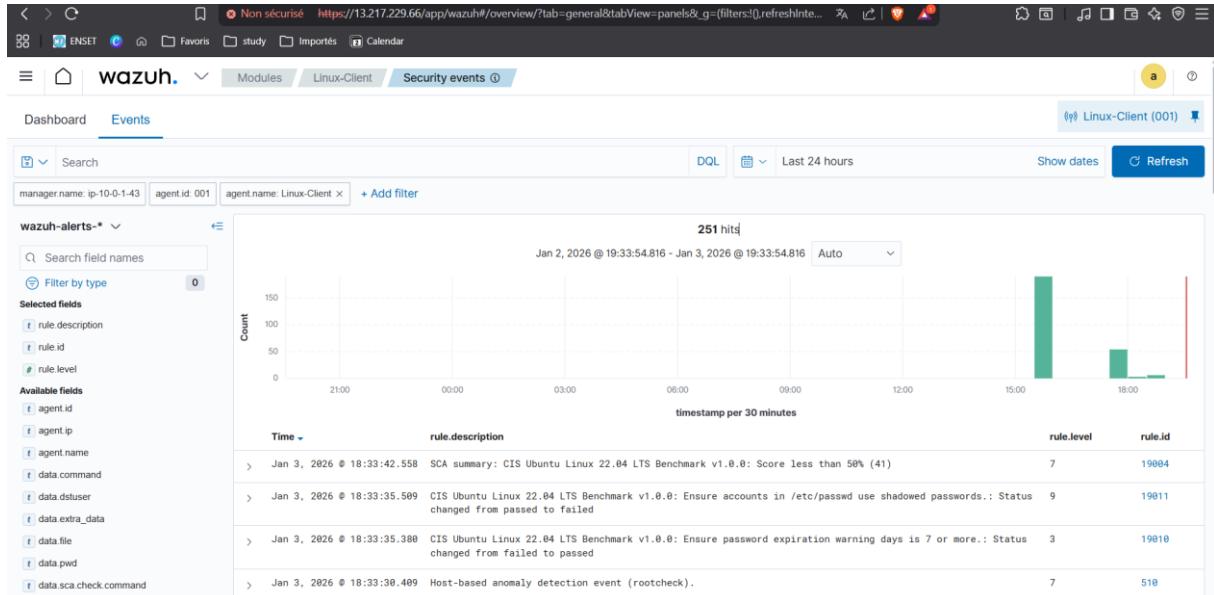


Figure 40: Threat Hunting requête 2

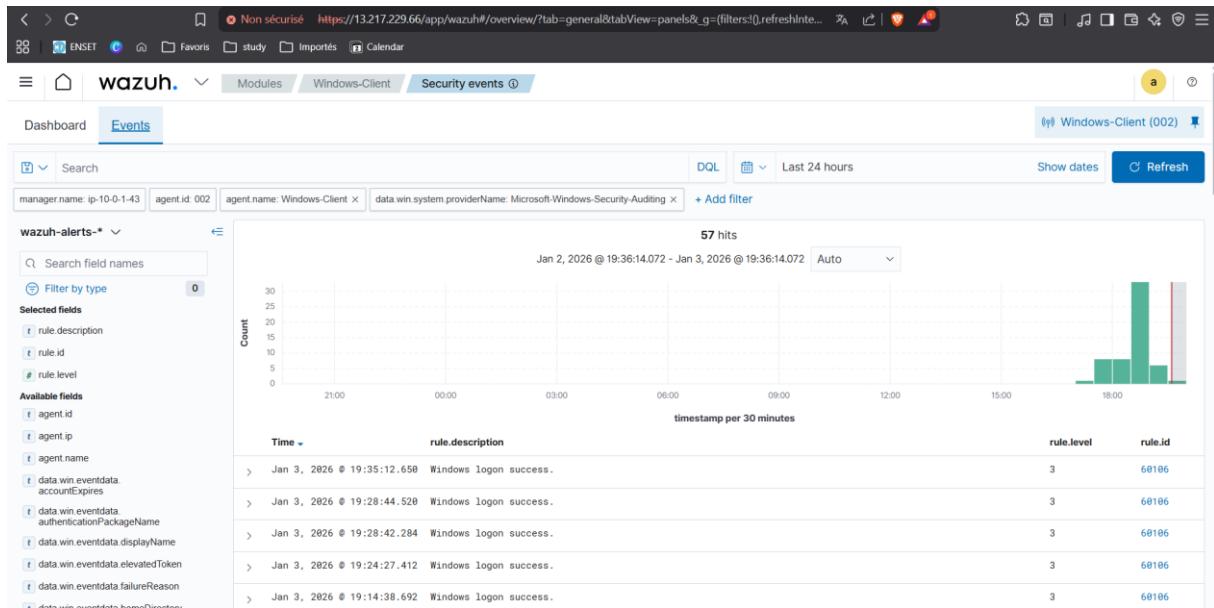


Figure 41: Threat Hunting requête 3

## Conclusion

Ce projet illustre la mise en place d'un **SOC Cloud complet** basé sur **Wazuh**, intégrant les fonctionnalités **SIEM** et **EDR** pour la collecte, l'analyse et la corrélation des événements de sécurité. Les composants déployés sur **AWS** (serveur Wazuh, endpoints Linux et Windows) permettent d'observer en temps réel la sécurité des systèmes, tandis que les bonnes pratiques réseau et la gestion des accès via **IAM/PAM** garantissent une architecture sécurisée, flexible et évolutive. Grâce aux démonstrations concrètes (Sysmon, FIM, surveillance réseau, alertes SIEM), ce projet fournit une **vision opérationnelle d'un SOC moderne**, combinant prévention, détection et réponse aux menaces dans un environnement Cloud.

### Lien Repo Github :

<https://github.com/irfanWh/Cloud-Security-SIEM-EDR-Multi-OS>