UNIVERSITÉ MUNDIAPOLIS
École d'Ingénieurs
Casablanca

2ème Année CI
Génie Informatique
Filière : Informatique

Mise en œuvre d'une infrastructure cloud de supervision centralisée sous AWS

Déploiement de Zabbix conteneurisé pour le monitoring d'un parc hybride (Linux & Windows)

Réalisé par :
FELLAH Youssef

Encadré par :
Prof. KHIAT Azeddine

Année universitaire : 2025/2026

Remerciements

Je tiens tout d'abord à remercier mon encadrant, M. **Azeddine KHIAT**, pour ses directives claires et son accompagnement pédagogique tout au long de ce module de Cloud Computing.

Ce projet m'a permis de consolider mes compétences en administration système et en architecture cloud, en mettant en pratique les concepts théoriques abordés en cours, notamment sur les services AWS (EC2, VPC) et la conteneurisation.

Avant-propos

La supervision des infrastructures informatiques est devenue une composante indispensable de la stratégie opérationnelle des entreprises modernes. Avec la migration croissante vers le Cloud, la capacité à surveiller en temps réel la disponibilité et la performance des services est un enjeu critique pour garantir la continuité d'activité.

Ce projet, réalisé dans le cadre du module de **Cloud Computing**, a pour vocation de simuler un environnement de production réel hébergé sur Amazon Web Services (AWS). Il ne s'agit pas uniquement de déployer des instances, mais de construire une architecture résiliente et surveillée, capable d'alerter les administrateurs au moindre incident.

Le choix de la solution **Zabbix**, couplée à la technologie de conteneurisation **Docker**, répond à un besoin d'agilité et de standardisation, compétences clés pour un futur ingénieur en informatique. Ce rapport retrace ainsi la démarche technique adoptée, depuis la conception de l'architecture réseau jusqu'à l'analyse des métriques de performance.

Table des matières

Table des figures

Liste des codes sources

1 Introduction

Ce projet s'inscrit dans le cadre du module de **Cloud Computing** et vise la mise en œuvre d'une infrastructure de supervision centralisée hébergée sur le cloud Amazon Web Services (AWS). L'objectif est de déployer une solution de monitoring pour un parc hybride (Linux et Windows).

1.1 Outils et Technologies

- **AWS :** Infrastructure (EC2, VPC, Security Groups).
- **Docker :** Conteneurisation du serveur Zabbix.
- **Zabbix :** Solution de monitoring open-source.

1.2 Ressources du Projet (GitHub)

Dépôt GitHub

Le code source et la documentation sont disponibles ici :
`https://github.com/yss-ef/Infrastructure-Cloud-Supervision-AWS`

2 Architecture Réseau

L'infrastructure réseau repose sur un VPC personnalisé avec un sous-réseau public.

2.1 Création du VPC

Le VPC `Fellah-Youssef-VPC-Projet-Zabbix` a été créé avec le bloc CIDR `10.0.0.0/16`.

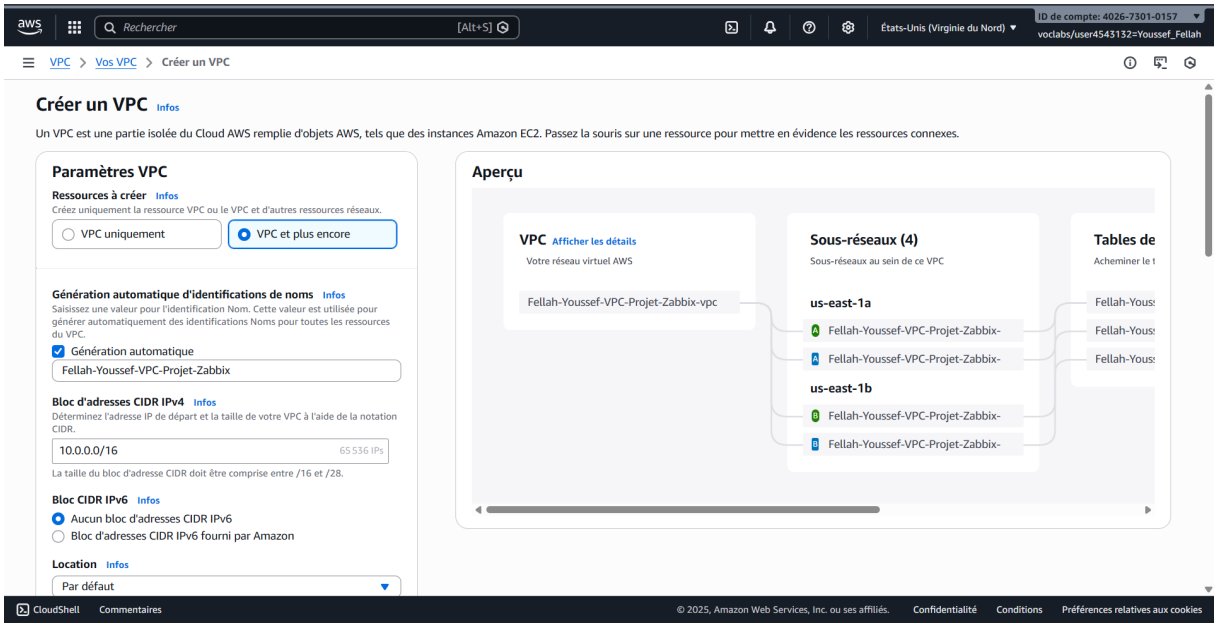

FIGURE 1 – *Aperçu de la topologie du VPC*

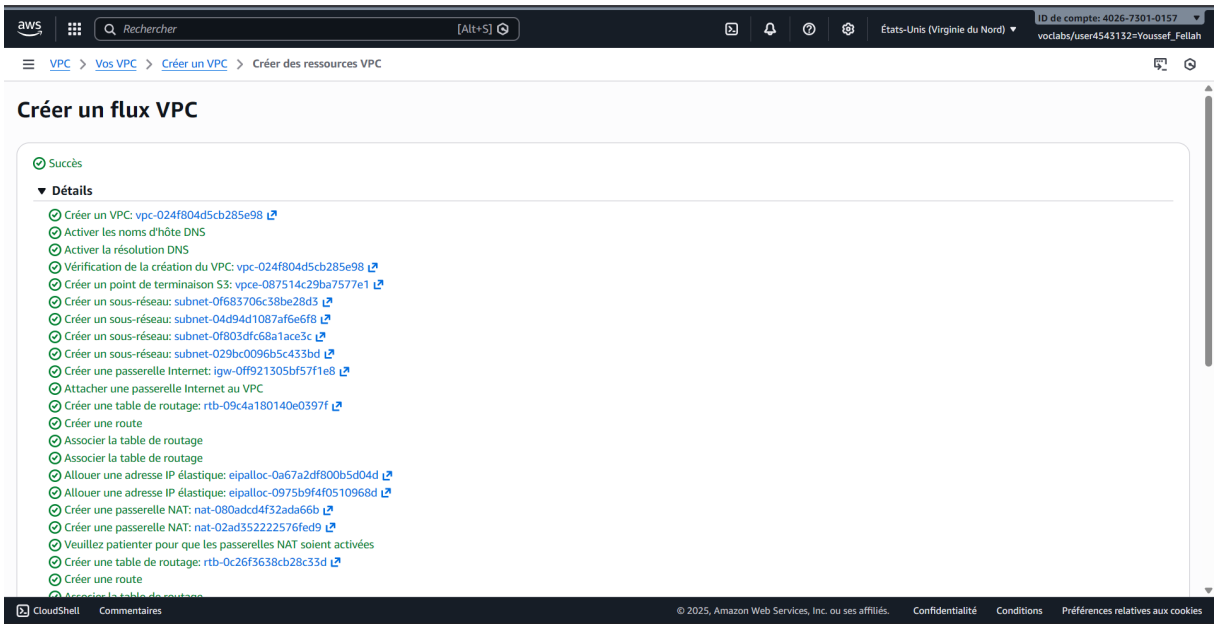

FIGURE 2 – *Validation de la création des ressources VPC*

2.2 Groupes de Sécurité

Le Security Group autorise les flux nécessaires au monitoring (Ports 10050, 10051) et à l'administration (22, 3389).

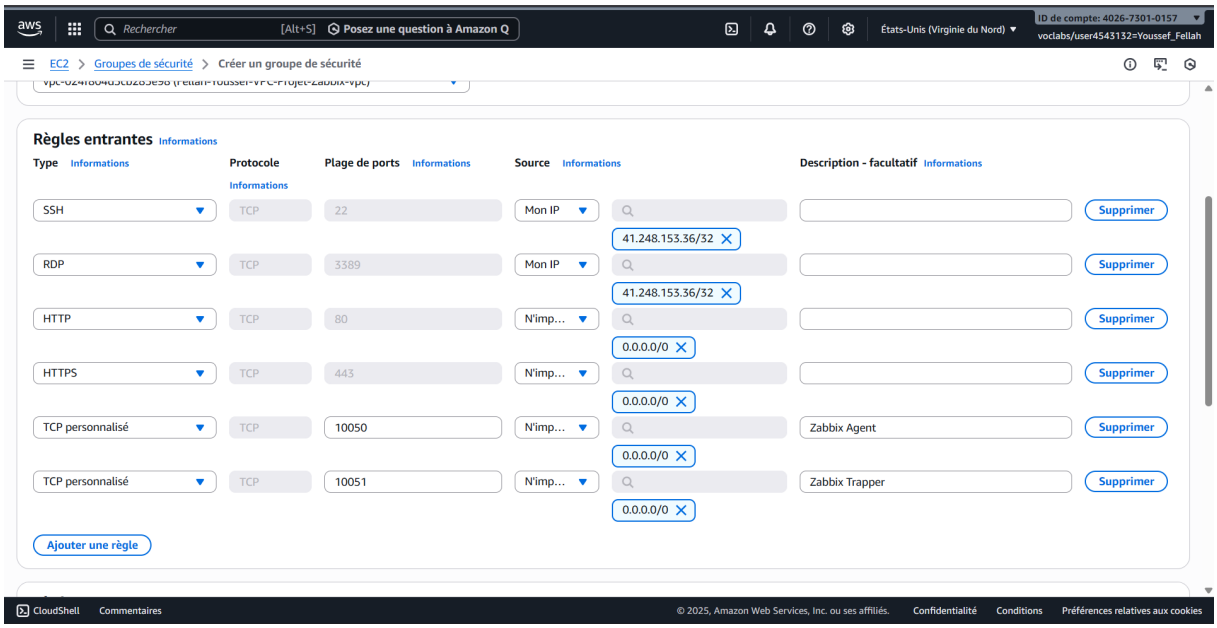

FIGURE 3 – *Configuration des règles entrantes (Inbound Rules)*

3 Architecture des Instances EC2

Trois instances ont été déployées pour simuler l'environnement de production.

- **Serveur Zabbix :** Ubuntu 22.04, t2.medium (4Go RAM).

- **Client Linux :** Ubuntu 22.04, t3.micro.
- **Client Windows :** Windows Server 2022, t3.medium.

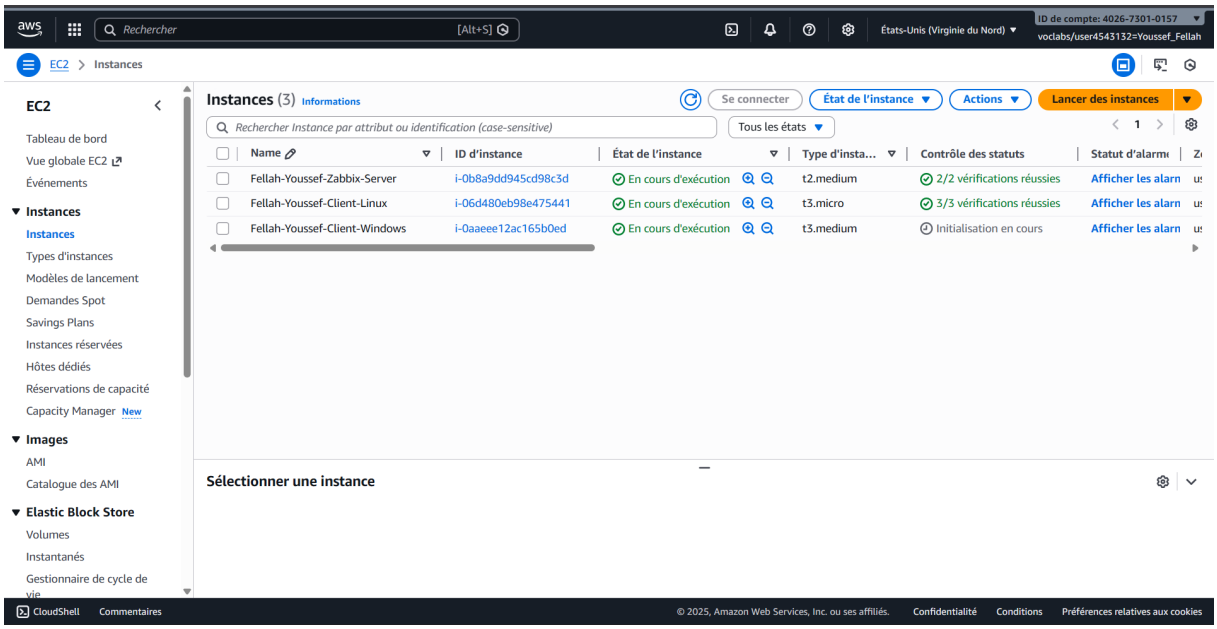

FIGURE 4 – *Instances EC2 en cours d'exécution (Running)*

4 Déploiement du Serveur Zabbix

Le déploiement du serveur de supervision a été réalisé de manière conteneurisée sur l'instance Ubuntu dédiée.

4.1 Installation de Docker

Nous avons d'abord préparé l'environnement en installant les paquets nécessaires et en configurant les droits utilisateurs.

```
# 1. Mise    jour du syst me
sudo apt update && sudo apt upgrade -y

# 2. Installation de Docker et Docker Compose
sudo apt install docker.io -y
sudo apt install docker-compose -y

# 3. D marrage et activation du service Docker
sudo systemctl start docker
sudo systemctl enable docker

# 4. Ajout de l'utilisateur actuel au groupe docker
sudo usermod -aG docker $USER
```

Listing 1 – *Installation des dépendances Docker*

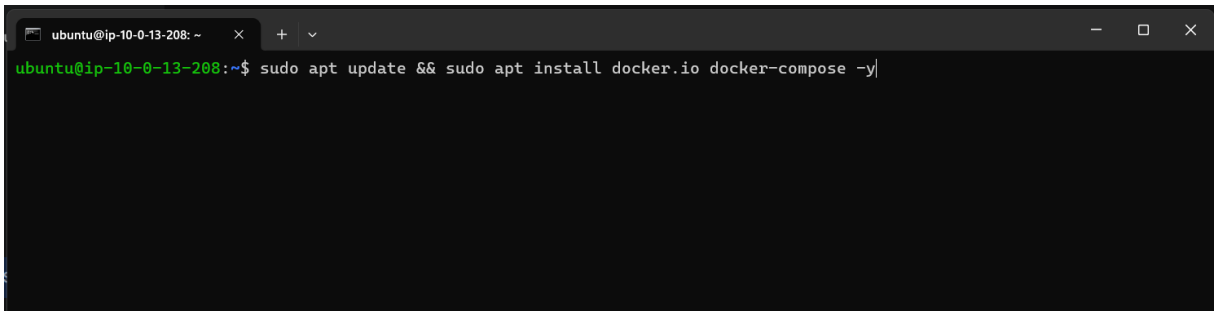

FIGURE 5 – *Installation des paquets Docker sur le terminal*

4.2 Configuration de l'Orchestrateur

Nous avons créé l'arborescence du projet dans le dossier `zabbix-docker`.

```
mkdir zabbix-docker && cd zabbix-docker
nano docker-compose.yml
```

Listing 2 – *Création du répertoire de travail*

Le fichier `docker-compose.yml` a été configuré pour utiliser des fichiers d'environnement externes (`.env_db_mysql`, etc.) afin de sécuriser les identifiants.

```
version: '3.5'
services:
  zabbix-db:
    image: mysql:8.0
    command: --character-set-server=utf8 --collation-server=utf8_bin --
    default-authentication-plugin=mysql_native_password
    volumes:
      - ./zbx_db_data:/var/lib/mysql
    env_file:
      - .env_db_mysql

  zabbix-server:
    image: zabbix/zabbix-server-mysql:ubuntu-6.4-latest
    ports:
      - "10051:10051"
    env_file:
      - .env_db_mysql
      - .env_srv
    depends_on:
      - zabbix-db

  zabbix-web:
    image: zabbix/zabbix-web-apache-mysql:ubuntu-6.4-latest
    ports:
      - "80:8080"
      - "443:8443"
    env_file:
      - .env_db_mysql
      - .env_web
    depends_on:
      - zabbix-server
```

Listing 3 – *Extrait du fichier docker-compose.yml*

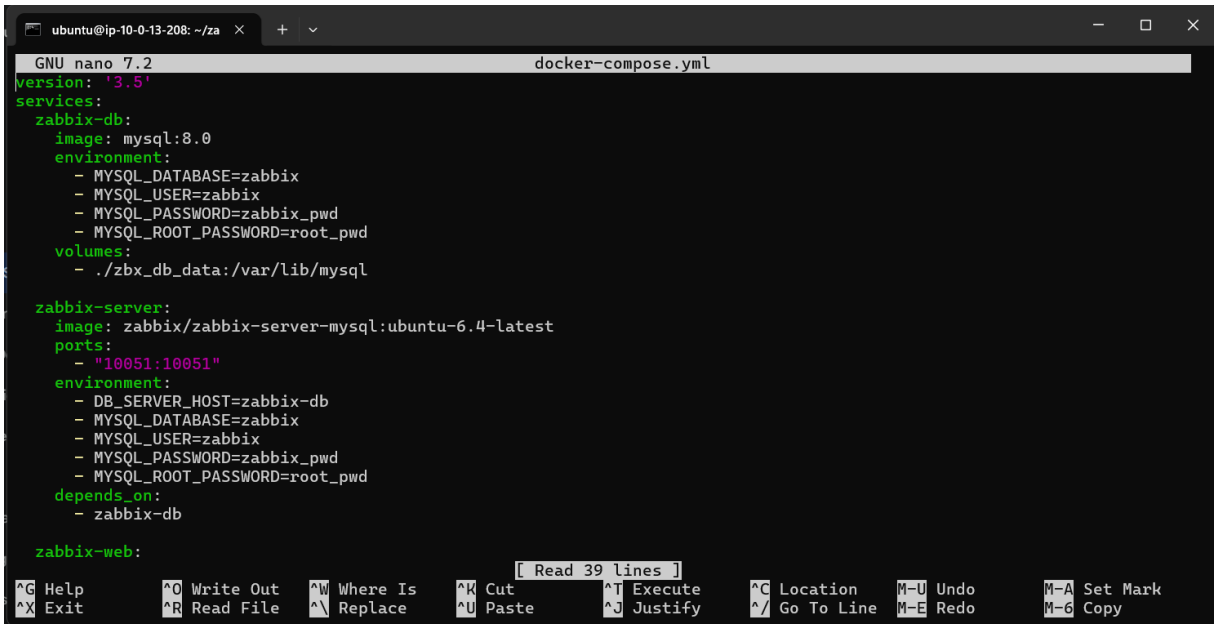

FIGURE 6 – *Configuration du fichier docker-compose.yml*

4.3 Lancement et Validation

Le service est démarré en mode détaché.

```
1 docker-compose up -d
```

Listing 4 – *Lancement de la stack Zabbix*

La vérification de l'état des conteneurs confirme que le déploiement est un succès :

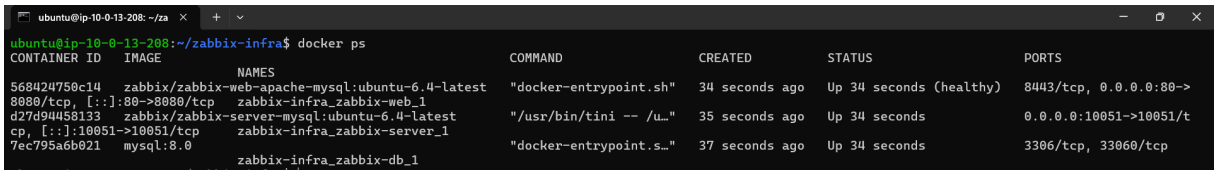

FIGURE 7 – *Validation des conteneurs actifs via docker ps*

4.4 Accès à l'Interface Web

L'interface est accessible via l'adresse IP publique de l'instance.

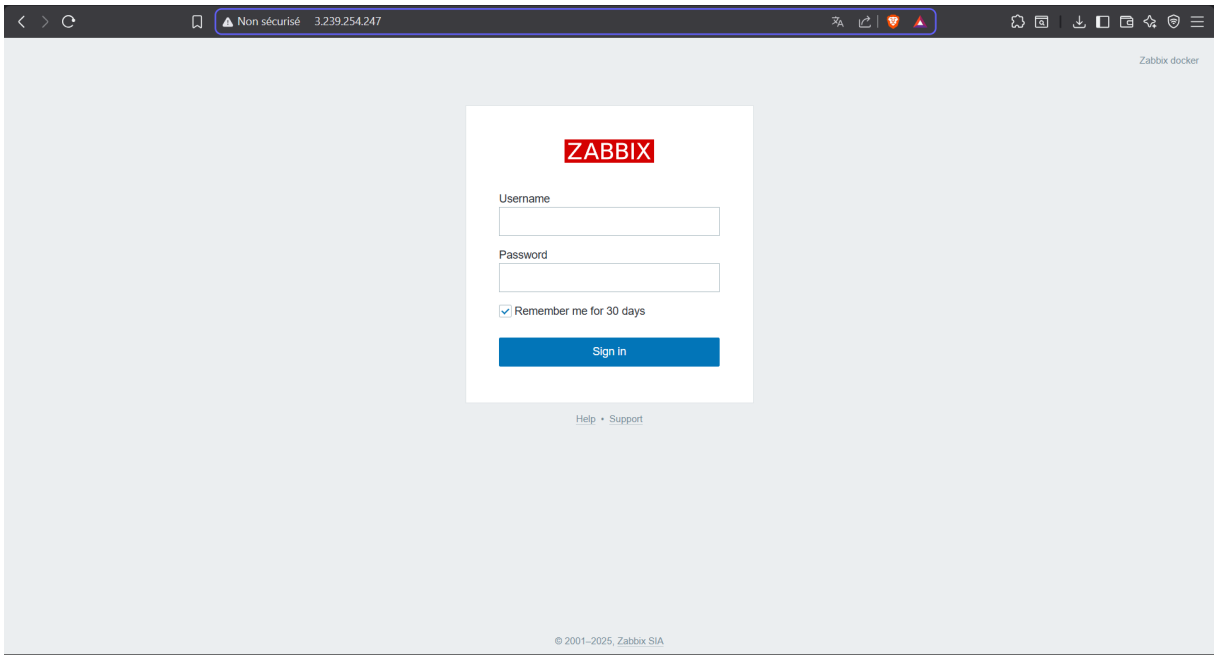

FIGURE 8 – *Page de connexion Zabbix*

5 Configuration des Clients (Agents)

Pour permettre la remontée des métriques vers le serveur central, l'agent Zabbix doit être installé et configuré sur chaque machine cible.

5.1 Client Linux (Ubuntu)

Sur l'instance `Client-Linux`, nous avons procédé à l'installation manuelle de l'agent en utilisant le dépôt officiel Zabbix pour garantir la compatibilité avec la version 6.4 du serveur.

5.1.1 Installation des paquets

Les commandes suivantes ont été exécutées pour ajouter le dépôt et installer l'agent :

```
# 1. R cup ration et installation du d p t officiel
wget https://repo.zabbix.com/zabbix/6.4/ubuntu/pool/main/z/zabbix-release/
    zabbix-release_latest+ubuntu24.04_all.deb
sudo dpkg -i zabbix-release_latest+ubuntu24.04_all.deb
sudo apt update

# 2. Installation du paquet de l'agent
sudo apt install zabbix-agent -y
```

Listing 5 – *Script d'installation de l'agent Zabbix sur Ubuntu*

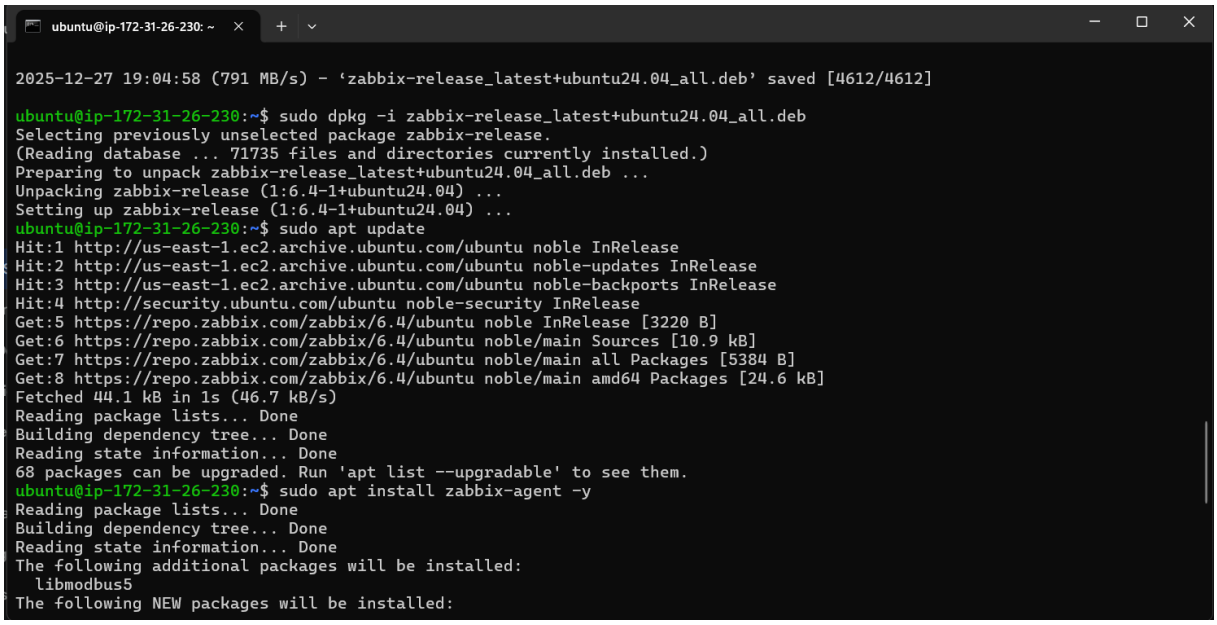

FIGURE 9 – *Installation réussie du paquet zabbix-agent via apt*

5.1.2 Configuration de l'Agent

L'étape critique consiste à modifier le fichier `/etc/zabbix/zabbix_agentd.conf` pour autoriser le serveur de supervision à communiquer avec cet agent.

Nous avons édité les paramètres suivants :

- **Server=10.0.13.208** : Adresse IP privée de notre serveur Zabbix (autorisé à faire du polling).
- **ServerActive=10.0.13.208** : Adresse IP pour les vérifications actives.
- **Hostname=Client-Linux** : Nom unique identifiant la machine.

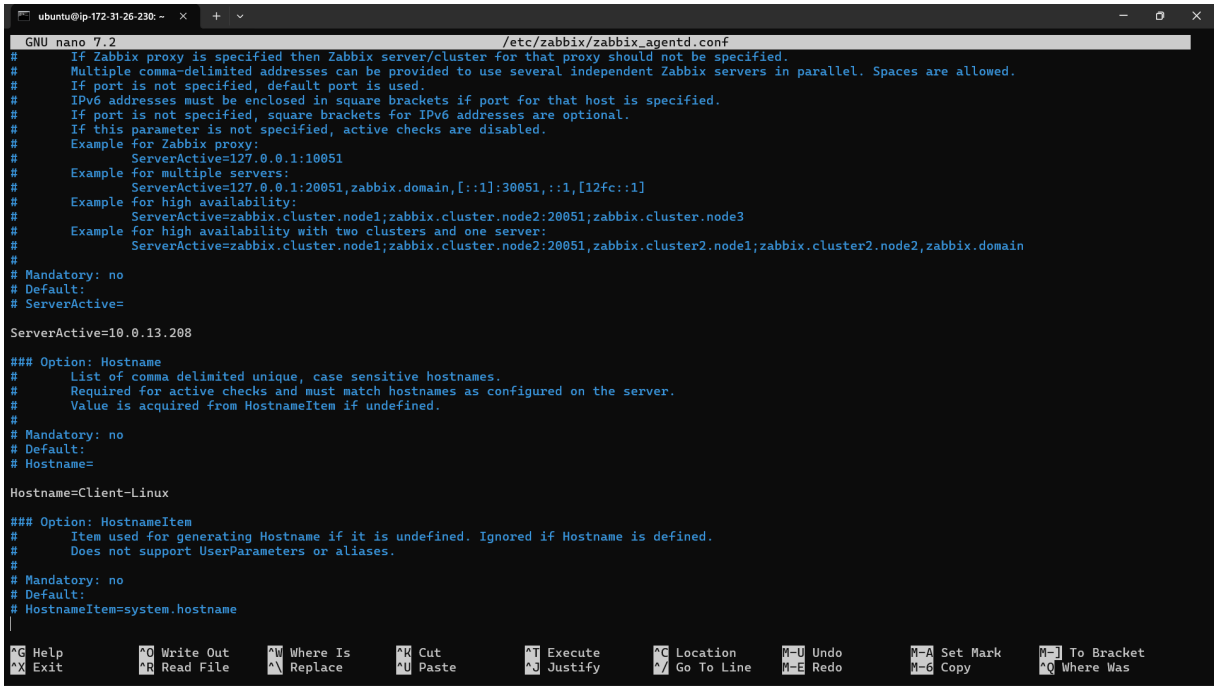

FIGURE 10 – *Configuration du Hostname et de l'IP du serveur dans zabbix_agentd.conf*

Une fois la configuration sauvegardée, le service a été redémarré et activé au démarrage :

```
sudo systemctl restart zabbix-agent
sudo systemctl enable zabbix-agent
```

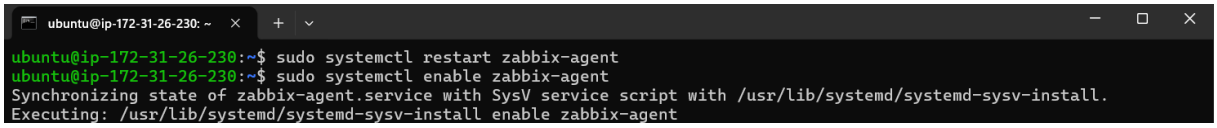

FIGURE 11 – *Redémarrage du service pour appliquer la configuration*

5.1.3 Déclaration de l'hôte dans Zabbix

Enfin, pour finaliser l'appairage, nous avons ajouté l'hôte dans l'interface Web de Zabbix (*Configuration* > *Hosts* > *Create host*) en veillant à ce que le "Host name" corresponde exactement à celui défini dans le fichier de configuration (`Client-Linux`).

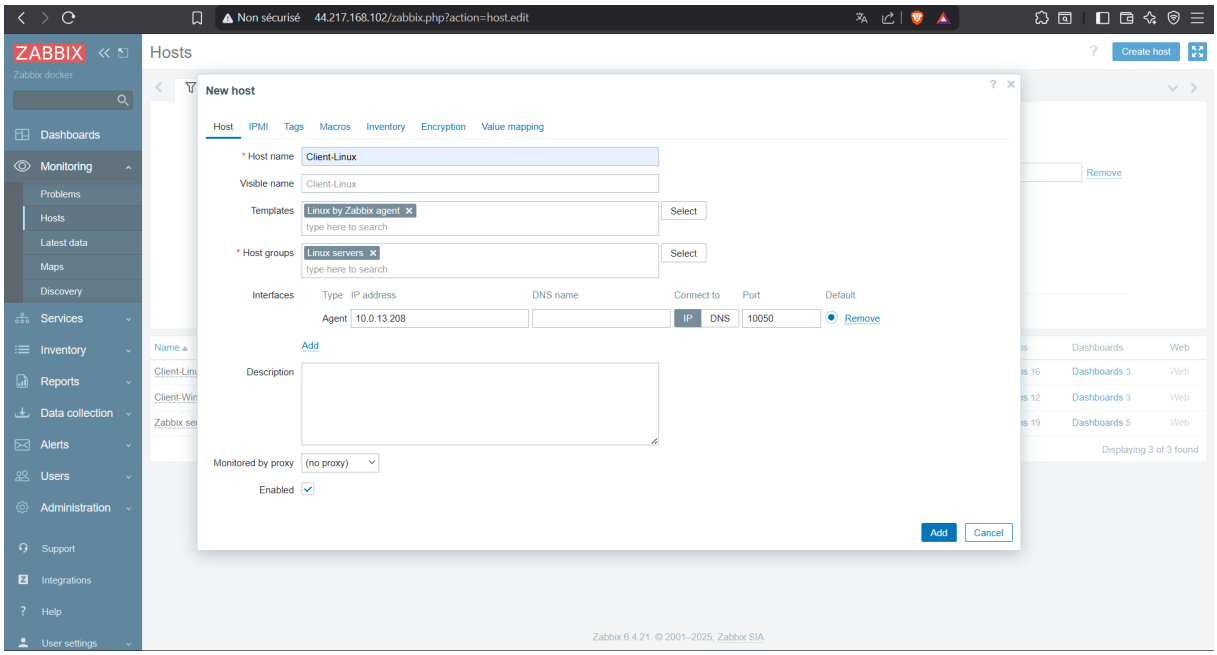

FIGURE 12 – *Ajout de l'hôte Linux dans l'interface de gestion Zabbix*

5.2 Client Windows (Windows Server 2022)

Pour le client Windows, la procédure diffère légèrement, s'appuyant sur une interface graphique via une connexion Bureau à Distance (RDP).

5.2.1 Accès à l'instance

Nous nous sommes connectés à l'instance `Fellah-Youssef-Client-Windows` en utilisant le client RDP et les identifiants administrateur déchiffrés via la clé privée AWS.

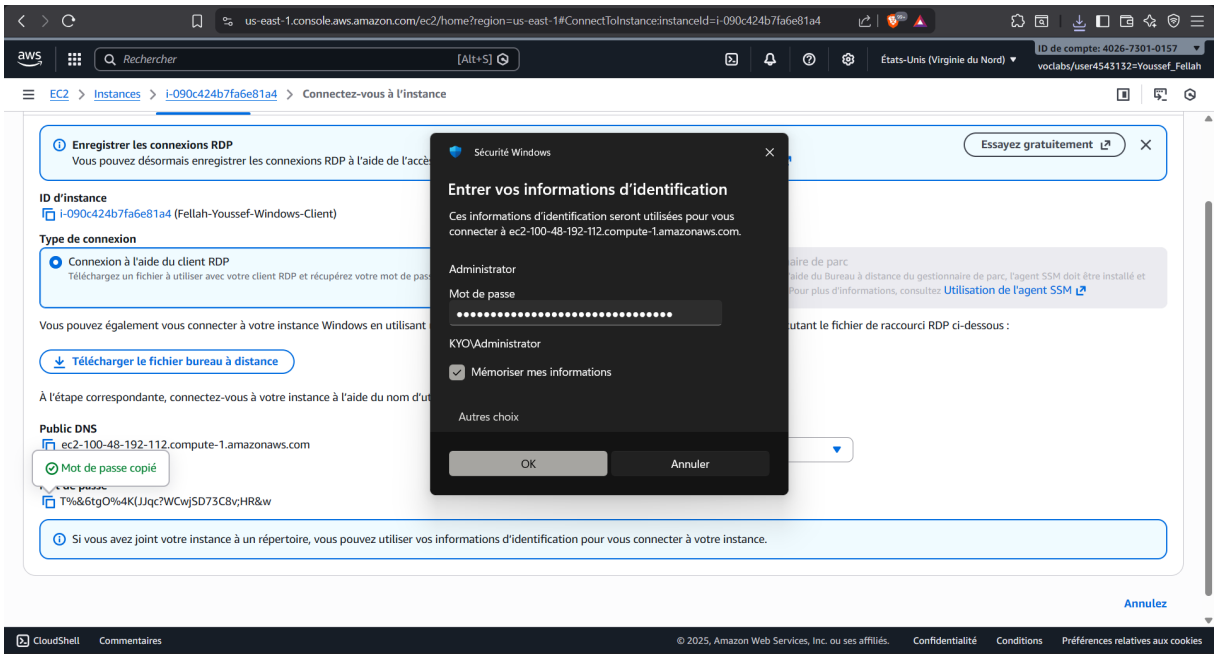

FIGURE 13 – *Connexion RDP à l'instance Windows Server*

5.2.2 Installation et Configuration de l'Agent

Depuis le navigateur de la VM, nous avons téléchargé l'installeur MSI officiel de l'agent Zabbix (version 7.4 LTS, compatible avec notre serveur).

Durant l'installation, nous avons configuré les paramètres de connexion pour lier l'agent à notre serveur de supervision :

- **Host name :** `Client-Windows` (Doit être identique au nom déclaré dans l'interface Web Zabbix).
- **Zabbix server IP :** `10.0.13.208` (L'IP privée de notre serveur Zabbix).
- **Agent listen port :** `10050` (Port par défaut).
- **Add agent location to the PATH** : Coché pour faciliter le débogage en ligne de commande.

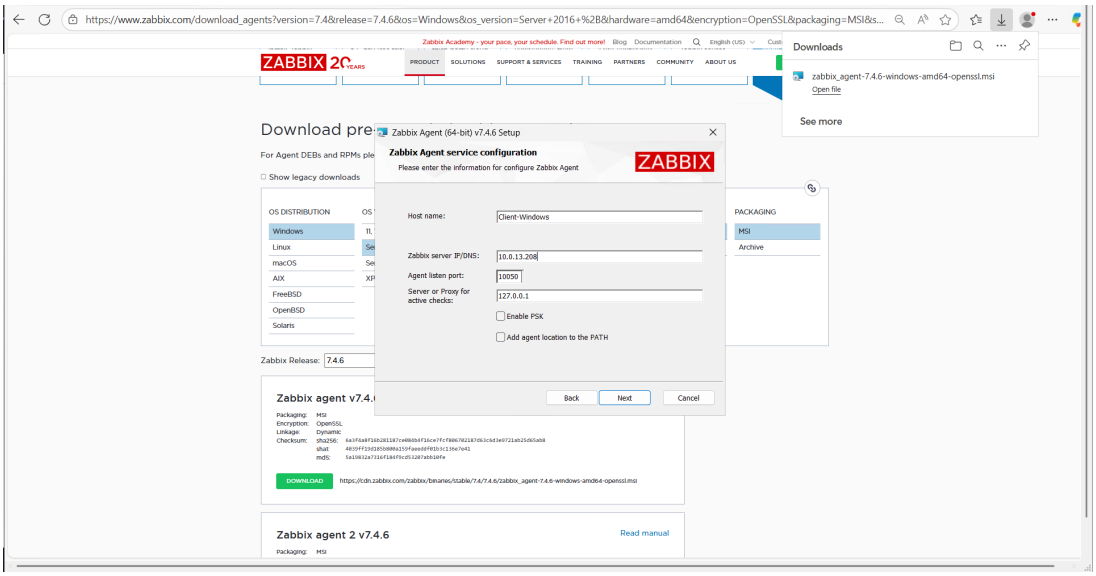

FIGURE 14 – *Configuration de l'agent Zabbix via l'installeur MSI*

L'installation MSI configure automatiquement le Pare-feu Windows (Windows Defender Firewall) pour autoriser le trafic entrant sur le port TCP 10050, rendant l'agent immédiatement opérationnel.

6 Monitoring et Tableaux de Bord

Une fois les agents installés et configurés, la dernière étape consistait à valider la remontée d'informations dans l'interface Zabbix et à visualiser les métriques.

6.1 Validation de la Connectivité (Disponibilité)

Nous avons ajouté nos deux clients ("Client-Linux" et "Client-Windows") dans la section *Configuration* > *Hosts*.

Le tableau ci-dessous présente l'état final du parc. La colonne **Availability** affiche l'icône **ZBX en vert** pour l'ensemble des hôtes, confirmant que le serveur reçoit correctement les données via le port 10050.

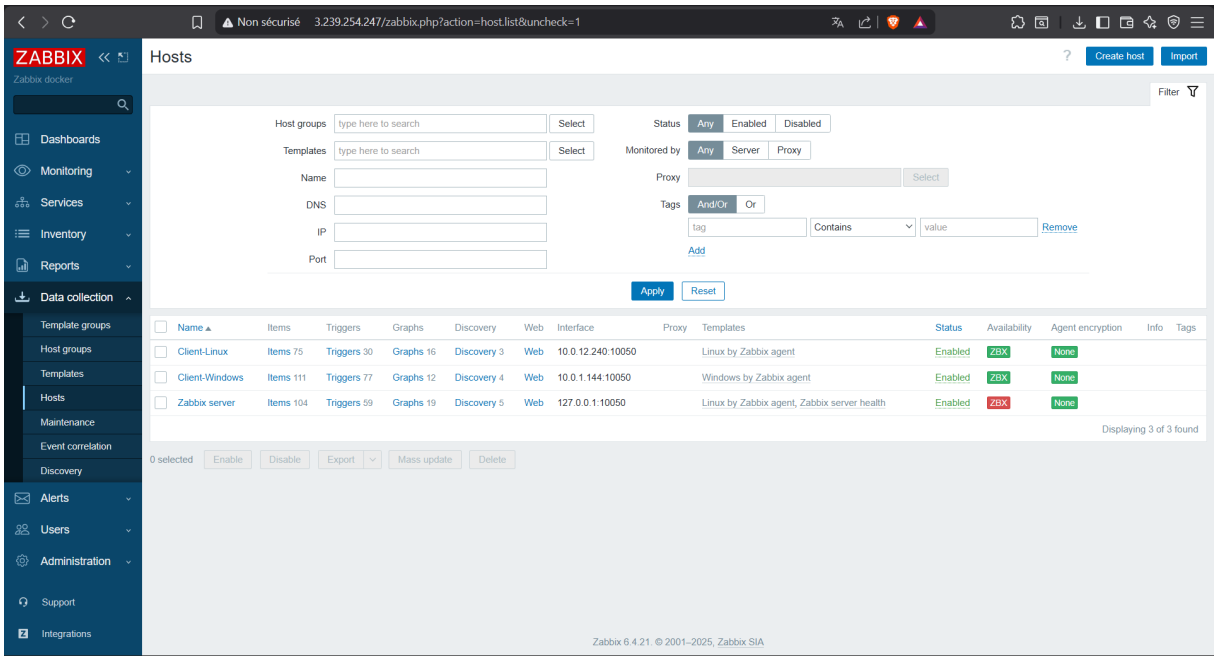

FIGURE 15 – *Vue globale des hôtes : Le statut "Vert" (ZBX) valide la communication*

6.2 Visualisation des Données (Graphiques)

Zabbix collecte désormais des métriques en temps réel (CPU, RAM, Disque, Réseau). Nous avons configuré un tableau de bord pour visualiser l'occupation des systèmes de fichiers du client Linux.

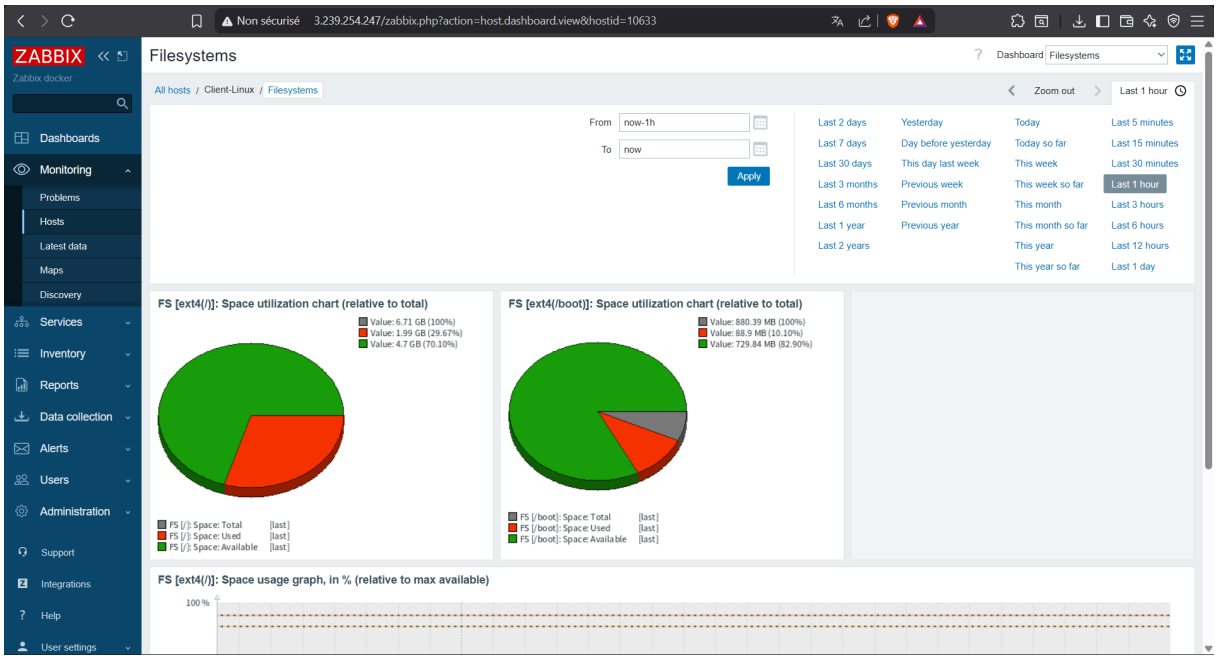

FIGURE 16 – *Graphiques d'utilisation des disques sur le Client Linux*

7 Conclusion

Ce projet nous a permis de déployer avec succès une infrastructure de supervision centralisée sur le cloud AWS, répondant aux exigences d'un environnement hybride moderne.

7.1 Bilan Technique

Nous avons réussi à :

- Mettre en œuvre une architecture réseau sécurisée (VPC, Security Groups) isolant les flux d'administration et de monitoring.
- Conteneuriser une application complexe (Zabbix) via Docker pour en faciliter le déploiement.
- Fédérer des systèmes hétérogènes (Ubuntu et Windows Server) au sein d'une même console de gestion.

7.2 Difficultés Rencontrées et Solutions

Durant la réalisation, nous avons fait face à quelques obstacles techniques :

1. **Configuration des Agents :** Une erreur de saisie lors de la déclaration de l'hôte Linux (doublon dans l'adresse IP, voir capture d'erreur lors des essais) a initialement empêché la connexion. Cela nous a appris l'importance de la validation rigoureuse des fichiers de configuration.
2. **Règles de Sécurité AWS :** La communication sur le port 10050 était initialement bloquée. Il a fallu modifier les *Security Groups* pour autoriser explicitement le trafic TCP sur ce port spécifique entre les instances du VPC.

En conclusion, cette architecture offre une base solide et évolutive pour la supervision d'un parc informatique d'entreprise.

8 Webographie

Pour la réalisation de ce projet, les documentations techniques suivantes ont été consultées :

- **Documentation AWS officielle** (VPC, Security Groups, EC2) : `https://docs.aws.amazon.com/`
- **Documentation Zabbix 6.4** (Installation Agents, Docker) : `https://www.zabbix.com/documentation/6.4/en/manual`
- **Docker Hub** (Images officielles Zabbix) : `https://hub.docker.com/u/zabbix`