



Université Mundiapolis

Année universitaire : 2025-2026

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)

Étudiant : Aymane El yamani
Encadrant : Prof. Azeddine KHIAT
Filière : 2ANCI – Génie Informatique

Table des matières

1	Introduction	1
1.1	Contexte et outils	1
2	Objectifs	2
2.1	Objectifs opérationnels	2
3	Architecture de la solution	3
3.1	Architecture réseau (VPC et sous-réseaux)	3
3.2	Sécurité (Security Groups)	4
3.3	Instances EC2 (serveur et clients)	6
4	Étapes d'installation et de configuration	9
4.1	Préparation du serveur Zabbix (Docker)	9
4.1.1	Commandes d'installation (exemple)	9
4.2	Déploiement du serveur Zabbix conteneurisé	10
4.3	Configuration du client Linux (agent Zabbix)	12
4.3.1	Installation et paramétrage	12
4.4	Configuration du client Windows (agent Zabbix)	14
4.5	Monitoring et tableaux de bord	16
5	Problèmes rencontrés et solutions	19
5.1	Problèmes réseau et accès	19
5.2	Contraintes du Learner Lab	19
6	Résultats	20
7	Conclusion	21
7.1	Perspectives	21
7.2	Dépôt GitHub	21

Chapitre 1

Introduction

Ce projet porte sur la mise en place d'une infrastructure de supervision centralisée dans le cloud (AWS), afin de superviser un parc hybride composé de machines Linux et Windows. La solution retenue s'appuie sur **Zabbix** déployé en **conteneurs Docker** sur une instance EC2, et sur des **agents Zabbix** installés sur les machines clientes.

1.1 Contexte et outils

Les outils et services mobilisés sont :

- **AWS (EC2, VPC)** pour l'hébergement et le réseau.
- **Docker / Docker Compose** pour déployer Zabbix de manière reproductible.
- **Zabbix** pour la collecte, l'agrégation et la visualisation des métriques.

Chapitre 2

Objectifs

L'objectif principal est de déployer une infrastructure de monitoring sur AWS avec Zabbix (Docker), permettant de superviser un parc hybride (Linux & Windows).

2.1 Objectifs opérationnels

- Concevoir un réseau (VPC, sous-réseau) adapté à l'accès et à la supervision.
- Mettre en place des règles de sécurité (Security Groups) cohérentes avec les flux Zabbix.
- Déployer un serveur Zabbix conteneurisé (serveur, interface web, base de données).
- Installer et configurer les agents Zabbix sur Linux et Windows.
- Valider la supervision : hôtes visibles, statut "ZBX" au vert, données en temps réel, graphiques.

Chapitre 3

Architecture de la solution

3.1 Architecture réseau (VPC et sous-réseaux)

Afin de simplifier l'accès dans le cadre du Lab, l'architecture repose sur un VPC avec un sous-réseau public.

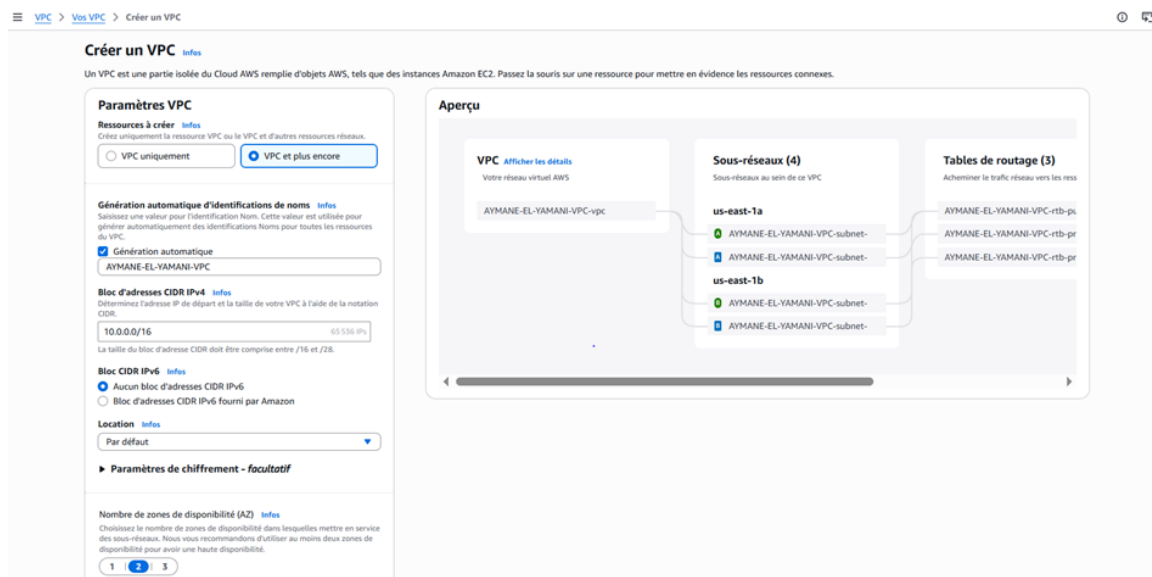


FIGURE 3.1 – Création du VPC

☰ [VPC](#) > [Sous-réseaux](#) > Créer un sous-réseau

Créer un sous-réseau Infos

VPC

ID de VPC
Créez des sous-réseaux dans ce VPC.

vpc-017337ec58b329efb (AYMANE-EL-YAMANI-VPC-vpc) ▼

CIDR de VPC associés

CIDR IPv4
10.0.0.0/16

Paramètres du sous-réseau
Précisez les blocs d'adresse CIDR et la zone de disponibilité pour le sous-réseau.

Sous-réseau 1 sur 1

Nom du sous-réseau (subnet)
Créez une balise avec une clé « Name » et une valeur à spécifier.

AYMANE-EL-YAMANI-Public-Subnet

Le nom peut comporter jusqu'à 256 caractères.

Zone de disponibilité Infos
Choisissez la zone dans laquelle votre sous-réseau résidera ou laissez Amazon en choisir une pour vous.

États-Unis (Virginie du Nord) / use1-az2 (us-east-1a) ▼

Bloc d'adresse CIDR IPv4 VPC Infos
Choisissez le bloc d'adresse CIDR IPv4 du VPC pour le sous-réseau. L'adresse CIDR IPv4 du sous-réseau doit se trouver dans ce bloc.

10.0.0.0/16 ▼

Bloc d'adresse CIDR de sous-réseau IPv4

10.0.0.0/20

FIGURE 3.2 – Création du sous-réseau

☰ [VPC](#) > [Sous-réseaux](#)

Tableau de bord du VPC

AWS Global View

Filtrer par VPC

Cloud privé virtuel

Vos VPC

[Sous-réseaux](#)

Tables de routage

Passerelles Internet

Passerelles Internet de sortie uniquement

Passerelles de l'opérateur

Jeux d'options DHCP

Adresses IP Elastic

Listes de préfixes gérées

Passerelles NAT

Connexions d'appariage

Serveurs de routage

Sécurité

ACL réseau

Groupes de sécurité

PrivateLink et Lattice

Mise en route

Points de terminaison

Vous avez bien supprimé subnet-006a82a164402c399, subnet-070203d8973012aab.

Sous-réseaux (8) Infos

Rechercher des sous-réseaux par attribut ou par balise

	Name	ID de sous-réseau	État	VPC	Bloquer l'ac...	CIDR IPv4	CIDR
<input type="checkbox"/>	-	subnet-02547ef3a5e5c14de	Available	vpc-0307fd91a8bd67b15	Désactivé	172.31.64.0/20	-
<input type="checkbox"/>	-	subnet-06fa33af6efc99af	Available	vpc-0307fd91a8bd67b15	Désactivé	172.31.48.0/20	-
<input type="checkbox"/>	-	subnet-05bc5204e0809960d	Available	vpc-0307fd91a8bd67b15	Désactivé	172.31.32.0/20	-
<input type="checkbox"/>	-	subnet-0132fc92a25493cab	Available	vpc-0307fd91a8bd67b15	Désactivé	172.31.16.0/20	-
<input type="checkbox"/>	-	subnet-09a8db411d814e27a	Available	vpc-0307fd91a8bd67b15	Désactivé	172.31.80.0/20	-
<input type="checkbox"/>	-	subnet-0b926ce715c82fd85	Available	vpc-0307fd91a8bd67b15	Désactivé	172.31.0.0/20	-
<input type="checkbox"/>	AYMANE-EL-YAMANI-VPC-subnet-privat...	subnet-067c01196e95e5676	Available	vpc-017337ec58b329efb AYM...	Désactivé	10.0.128.0/20	-
<input type="checkbox"/>	AYMANE-EL-YAMANI-VPC-subnet-publi...	subnet-0f75bd2162684b43e	Available	vpc-017337ec58b329efb AYM...	Désactivé	10.0.16.0/20	-

Sélectionner un sous-réseau

FIGURE 3.3 – Organisation des sous-réseaux (public/privé)

3.2 Sécurité (Security Groups)

Les groupes de sécurité autorisent les ports nécessaires :

- 80/443 (interface Web Zabbix),
- 10050/10051 (agent/serveur Zabbix),
- 22 (SSH) et 3389 (RDP) pour l'administration.

☰ VPC > Groupes de sécurité > Créer un groupe de sécurité ⓘ

Créer un groupe de sécurité Informations

Un groupe de sécurité agit comme un pare-feu virtuel pour votre instance afin de contrôler le trafic entrant et sortant. Pour créer un groupe de sécurité, complétez les champs ci-dessous.

Détails de base

Nom du groupe de sécurité Informations

Le nom ne peut pas être modifié après sa création.

Description Informations

VPC Informations

Règles entrantes Informations

Ce groupe de sécurité n'a pas de règles entrantes.

[Ajouter une règle](#)

Règles sortantes Informations

Type <small>Informations</small>	Protocole <small>Informations</small>	Plage de ports <small>Informations</small>	Destination <small>Informations</small>	Description - facultatif <small>Informations</small>	
Tout le trafic	Tous	Tous	Personn...	<input type="text" value="Q"/>	Supprimer

[0.0.0.0/0](#) ✕

FIGURE 3.4 – Création d'un groupe de sécurité

Règles entrantes Informations

Type <small>Informations</small>	Protocole <small>Informations</small>	Plage de ports <small>Informations</small>	Source <small>Informations</small>	Description - facultatif <small>Informations</small>	
SSH	TCP	22	N'impor... <input type="text" value="Q"/>	<input type="text"/>	Supprimer
HTTP	TCP	80	N'impor... <input type="text" value="Q"/>	<input type="text"/>	Supprimer
HTTPS	TCP	443	N'impor... <input type="text" value="Q"/>	<input type="text"/>	Supprimer
TCP personnalisé	TCP	10050	N'impor... <input type="text" value="Q"/>	<input type="text"/>	Supprimer
TCP personnalisé	TCP	10051	N'impor... <input type="text" value="Q"/>	<input type="text"/>	Supprimer
RDP	TCP	3389	Personn... <input type="text" value="Q"/>	<input type="text"/>	Supprimer

[Ajouter une règle](#)

FIGURE 3.5 – Règles entrantes (ports requis)

Règles sortantes Informations

Type <small>Informations</small>	Protocole <small>Informations</small>	Plage de ports <small>Informations</small>	Destination <small>Informations</small>	Description - facultatif <small>Informations</small>	
Tout le trafic	Tous	Tous	Personn... <input type="text" value="Q"/>	<input type="text"/>	Supprimer

[0.0.0.0/0](#) ✕

[Ajouter une règle](#)

FIGURE 3.6 – Règles sortantes

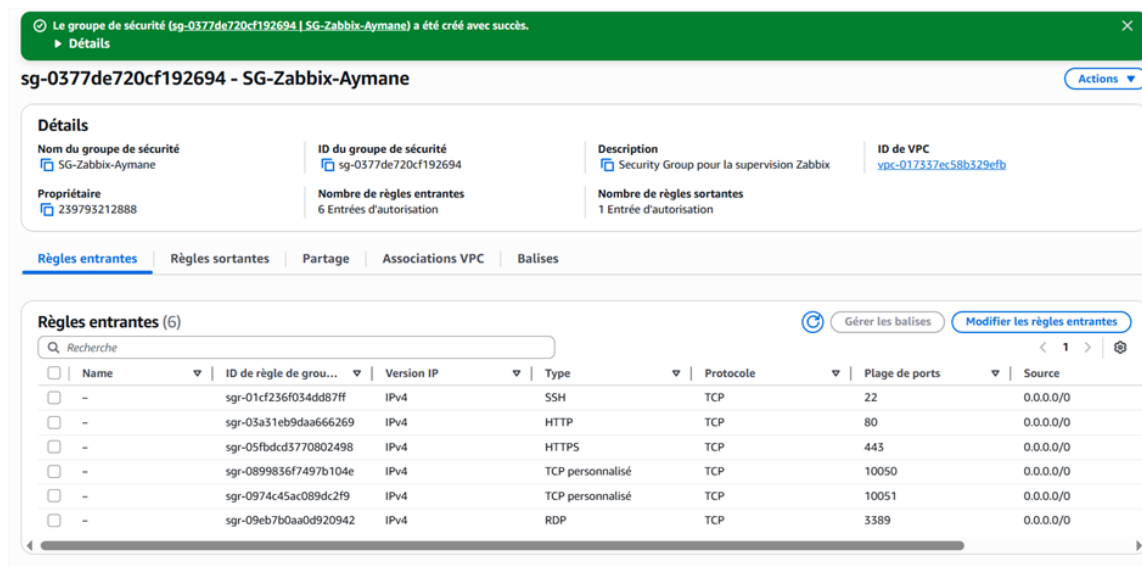


FIGURE 3.7 – Groupe de sécurité créé avec succès

3.3 Instances EC2 (serveur et clients)

L'architecture retient trois instances : un serveur Zabbix, un client Linux et un client Windows. Les tailles d'instances sont choisies pour rester compatibles avec les limitations du Lab.

Rôle	Système	Type (exemple)
Serveur Zabbix	Ubuntu	t3.large
Client Linux	Ubuntu	t3.medium
Client Windows	Windows Server	t3.large

TABLE 3.1 – Instances EC2 utilisées

Lancer une instance [Informations](#)

Amazon EC2 vous permet de créer des machines virtuelles, ou des instances, qui s'exécutent sur le Cloud AWS. Démarrez rapidement en suivant les étapes simples indiquées ci-dessous.

Nom et balises [Informations](#)

Nom

AYMANE-EL-YAMANI-Zabbix-Server

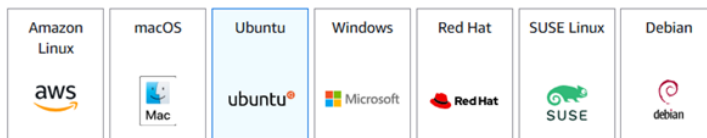
[Ajouter des balises supplémentaires](#)

▼ Images d'applications et de systèmes d'exploitation (Amazon Machine Image) [Informations](#)

Une AMI contient le système d'exploitation, le serveur d'applications et les applications de votre instance. Si aucune AMI appropriée ne s'affiche ci-dessous, utilisez le champ de recherche ou choisissez [Parcourir d'autres AMI](#).

Effectuer une recherche dans notre catalogue complet, qui comprend des milliers d'images d'applications et de systèmes d'exploitation

Démarrage rapide



[Explorer plus d'AMI](#)
Y compris les AMI d'AWS, de Marketplace et de la communauté

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
ami-0ecb62995f68bb549 (64 bits (x86)) / ami-01b9f1e7dc427266e (64 bits (Arm))
Virtualisation: hvm ENA activé: true Type de périphérique racine: ebs

Éligible à l'offre gratuite

FIGURE 3.8 – Lancement de l'instance EC2 (serveur Zabbix)

▼ Type d'instance [Informations](#) | [Obtenez des conseils](#)

Type d'instance

t3.large
Famille: t3 2 vCPU 8 Gio Mémoire Génération actuelle: true
À la demande Linux base tarification: 0.0832 USD par heure
À la demande Windows base tarification: 0.1108 USD par heure À la demande RHEL base tarification: 0.112 USD par heure
À la demande SUSE base tarification: 0.1395 USD par heure À la demande Ubuntu Pro base tarification: 0.0867 USD par heure

☐ Toutes les générations

[Comparer les types d'instance](#)

Des frais supplémentaires s'appliquent pour les AMI avec un logiciel préinstallé

▼ Paire de clés (connexion) [Informations](#)

Vous pouvez utiliser une paire de clés pour vous connecter en toute sécurité à votre instance. Assurez-vous d'avoir accès à la paire de clés sélectionnée avant de lancer l'instance.

Nom de la paire de clés - obligatoire

CLE-AYMANE-EL-YAMANI-Zabbix-Server

[Créer une paire de clés](#)

▼ Paramètres réseau [Informations](#)

VPC - obligatoire [Informations](#)

vpc-017337ec58b329efb (AYMANE-EL-YAMANI-VPC-vpc)
10.0.0.0/16

Sous-réseau [Informations](#)

subnet-0f75bd2162684b43e AYMANE-EL-YAMANI-VPC-subnet-public2-us-east-1b
VPC: vpc-017337ec58b329efb Propriétaire: 239793212888 Zone de disponibilité: us-east-1b (use1-az4)
Type de zone: Zone de disponibilité Adresses IP disponibles: 4091 CIDR: 10.0.16.0/20

[Créer un nouveau sous-réseau](#)

Attribuer automatiquement l'adresse IP publique [Informations](#)

Activer

FIGURE 3.9 – Création de la clé de connexion (Key Pair)

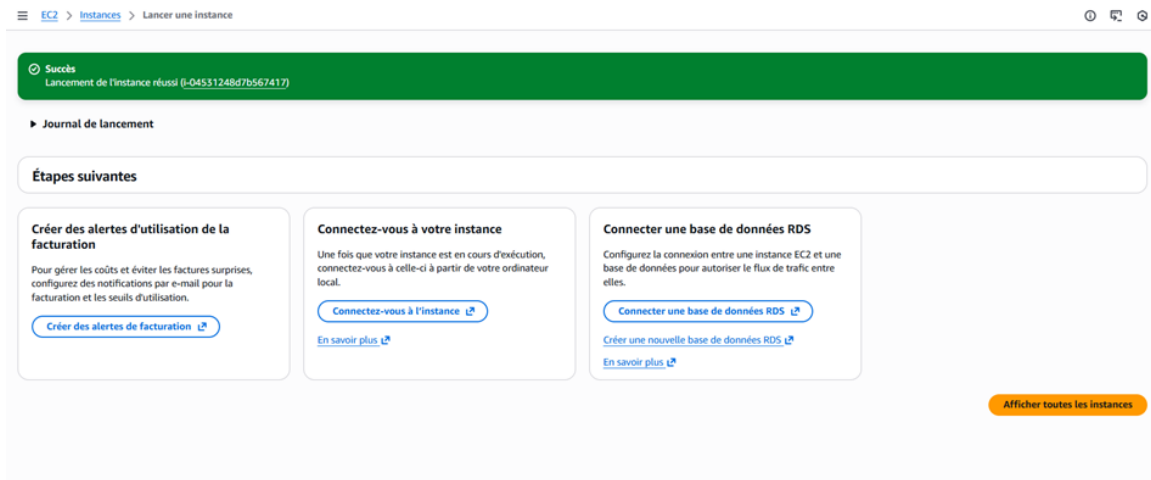


FIGURE 3.10 – Instance créée avec succès

```
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-107-22-27-112.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Thu Jan 1 20:14:04 UTC 2026

System load:  0.02           Temperature:   -273.1 C
Usage of /:   25.8% of 6.71GB Processes:      116
Memory usage: 22%           Users logged in: 0
Swap usage:   0%            IPv4 address for ens5: 10.0.29.2

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-29-2:~$
ubuntu@ip-10-0-29-2:~$
```

FIGURE 3.11 – Connexion SSH réussie au serveur

Chapitre 4

Étapes d'installation et de configuration

4.1 Préparation du serveur Zabbix (Docker)

Cette section décrit l'installation de Docker et Docker Compose sur Ubuntu, puis la vérification du bon fonctionnement des conteneurs.

```
ubuntu@ip-10-0-29-2:~$ docker --version
Docker version 28.2.2, build 28.2.2-0ubuntu1~24.04.1
ubuntu@ip-10-0-29-2:~$ docker-compose --version
docker-compose version 1.29.2, build unknown
ubuntu@ip-10-0-29-2:~$
```

FIGURE 4.1 – Installation de Docker et Docker Compose

4.1.1 Commandes d'installation (exemple)

Listing 4.1 – Installation Docker (exemple Ubuntu)

```
sudo apt update
sudo apt install -y ca-certificates curl gnupg

sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg

echo \
    "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/
    docker.gpg] https://download.docker.com/linux/ubuntu \
    $(. /etc/os-release && echo $VERSION_CODENAME) stable" \
    | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin
```

```
sudo systemctl enable --now docker
docker --version
docker compose version
```

```
ubuntu@ip-10-0-29-2:~/zabbix$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
NAMES
85d251fec9b   zabbix/zabbix-web-nginx-pgsql:alpine-7.0-latest  "docker-entrypoint.sh"  9 seconds ago  Up 8 seconds (health: starting)  8443/tcp, 0.0.0.0:80->8080/tcp, [::]:80->8080/tcp
649b7b7a2ff5   zabbix/zabbix-server-pgsql:alpine-7.0-latest    "/usr/bin/docker-ent..."  9 seconds ago  Up 9 seconds                    0.0.0.0:10051->10051/tcp, [::]:10051->10051/tcp
4d439cfe4071   postgres:16-alpine                      "docker-entrypoint.s..."  9 seconds ago  Up 9 seconds                    5432/tcp
ubuntu@ip-10-0-29-2:~/zabbix$
```

FIGURE 4.2 – Vérification des conteneurs Docker

4.2 Déploiement du serveur Zabbix conteneurisé

Le serveur Zabbix est déployé via Docker Compose (serveur, interface web et base de données).

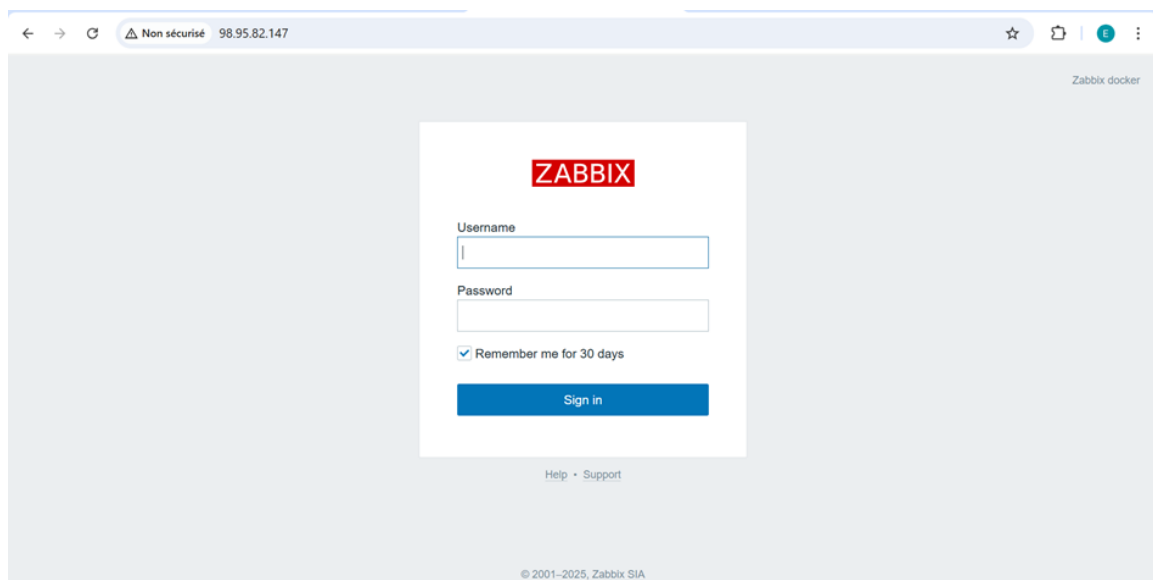


FIGURE 4.3 – Lancement des services Zabbix via Docker

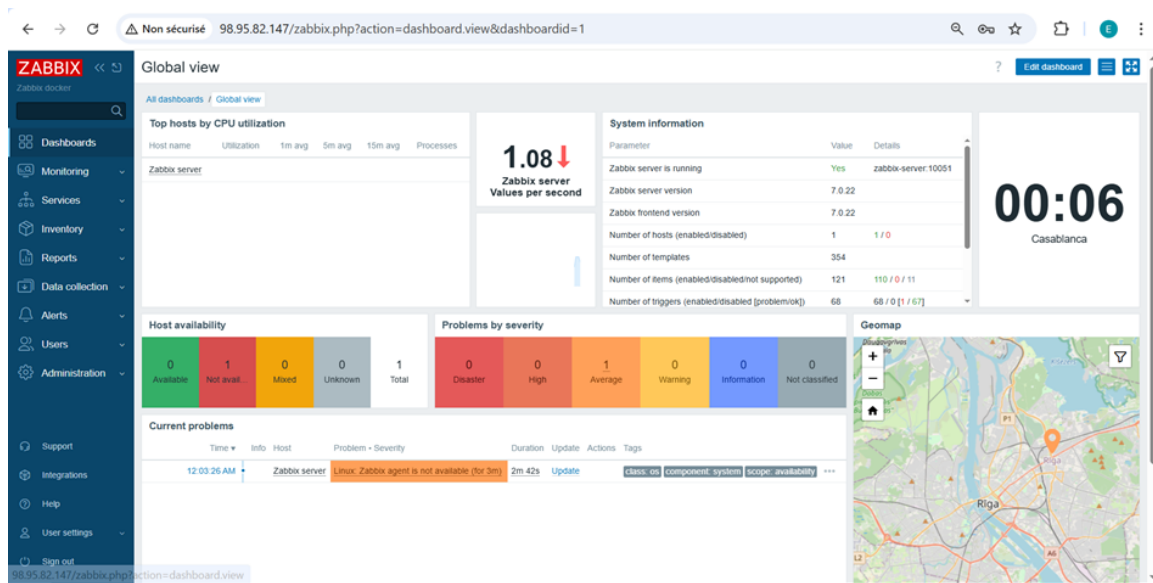


FIGURE 4.4 – Accès au tableau de bord Zabbix

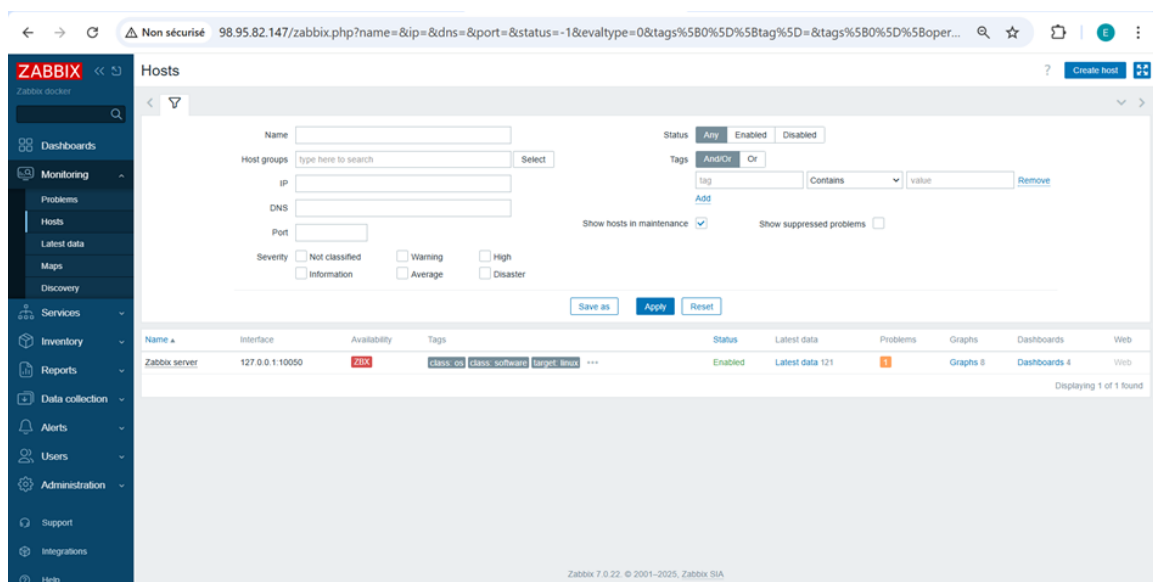


FIGURE 4.5 – Vue “Hosts” (supervision des hôtes)

4.3 Configuration du client Linux (agent Zabbix)

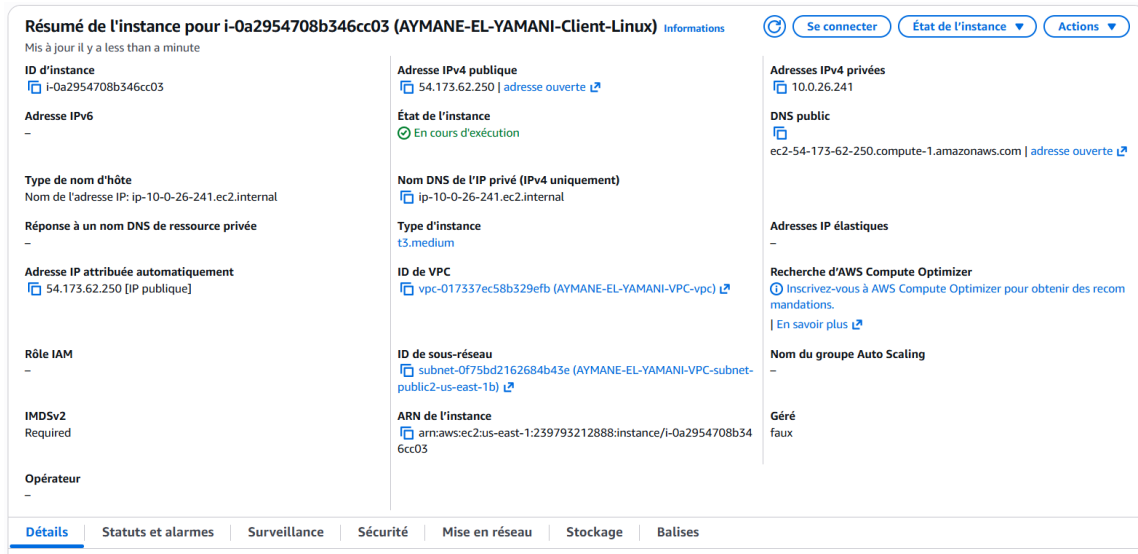


FIGURE 4.6 – Lancement de l'instance EC2 (client Linux)

4.3.1 Installation et paramétrage

Listing 4.2 – Installation de l'agent Zabbix sur Ubuntu (exemple)

```
sudo apt update
sudo apt install -y zabbix-agent

# Adapter l'IP du serveur Zabbix
sudo sed -i 's/^Server=.* /Server=<IP_SERVEUR_ZABBIX>/' /etc/zabbix/zabbix_agentd.conf
sudo sed -i 's/^ServerActive=.* /ServerActive=<IP_SERVEUR_ZABBIX>/' /etc/zabbix/zabbix_agentd.conf
sudo sed -i 's/^Hostname=.* /Hostname=client-linux/' /etc/zabbix/zabbix_agentd.conf

sudo systemctl enable --now zabbix-agent
sudo systemctl status zabbix-agent
```

```

# IP addresses must be enclosed in square brackets if port for that host is specified.
# If port is not specified, square brackets for IPv6 addresses are optional.
# If this parameter is not specified, active checks are disabled.
# Example for Zabbix proxy:
#   ServerActive=127.0.0.1:10051
# Example for multiple servers:
#   ServerActive=127.0.0.1:20051;zabbix.domain,[::1]:30051,[12fc::1]
# Example for high availability:
#   ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
# Example for high availability with two clusters and one server:
#   ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster2.node1;zabbix.cluster2.node2;zabbix.domain
#
# Mandatory: no
# Default:
# ServerActive=

ServerActive=10.0.29.2

## Option: Hostname
#   List of comma delimited unique, case sensitive hostnames.
#   Required for active checks and must match hostnames as configured on the server.
#   Value is acquired from Hostnameitem if undefined.
#
# Mandatory: no
# Default:
# Hostname=

Hostname=AYMANE-EL-YAMANI-Client-Linux

## Option: Hostnameitem
#   Item used for generating Hostname if it is undefined. Ignored if Hostname is defined.
#   Does not support UserParameters or aliases.
#
# Mandatory: no
# Default:
# Hostnameitem=system.hostname

```

FIGURE 4.7 – Configuration “server / client / agent”

```

ubuntu@ip-10-0-26-241:~$ sudo nano /etc/zabbix/zabbix_agent2.conf
ubuntu@ip-10-0-26-241:~$ sudo systemctl enable zabbix-agent2
Synchronizing state of zabbix-agent2.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable zabbix-agent2
ubuntu@ip-10-0-26-241:~$ sudo systemctl start zabbix-agent2
ubuntu@ip-10-0-26-241:~$ systemctl status zabbix-agent2
● zabbix-agent2.service - Zabbix Agent 2
   Loaded: loaded (/usr/lib/systemd/system/zabbix-agent2.service; enabled; preset: enabled)
   Active: active (running) since Sat 2026-01-03 00:13:36 UTC; 7s ago
   Main PID: 3370 (zabbix_agent2)
   Tasks: 9 (limit: 4515)
   Memory: 7.9M (peak: 10.7M)
   CPU: 80ms
   CGroup: /system.slice/zabbix-agent2.service
           └─3370 /usr/sbin/zabbix_agent2 -c /etc/zabbix/zabbix_agent2.conf

Jan 03 00:13:36 ip-10-0-26-241 systemd[1]: Started zabbix-agent2.service - Zabbix Agent 2.
Jan 03 00:13:36 ip-10-0-26-241 zabbix_agent2[3370]: Starting Zabbix Agent 2 (7.0.22)
Jan 03 00:13:36 ip-10-0-26-241 zabbix_agent2[3370]: Zabbix Agent2 hostname: [AYMANE-EL-YAMANI-Client-Linux]
Jan 03 00:13:36 ip-10-0-26-241 zabbix_agent2[3370]: Press Ctrl+C to exit.
ubuntu@ip-10-0-26-241:~$

```

FIGURE 4.8 – Activation du service agent côté client Linux

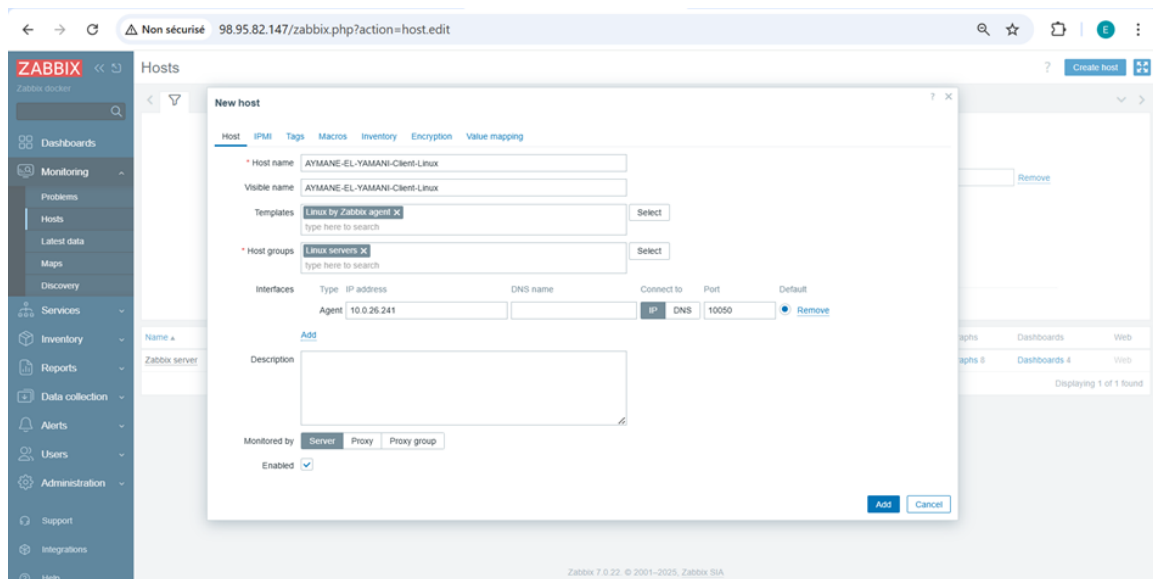


FIGURE 4.9 – Modification/validation des paramètres de l'hôte Linux

4.4 Configuration du client Windows (agent Zabbix)

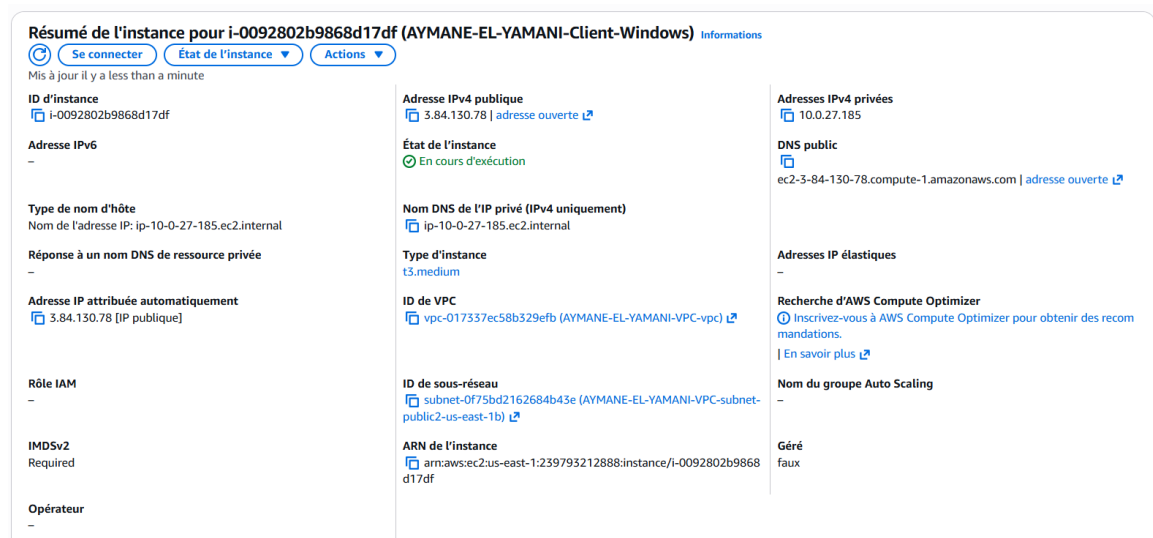


FIGURE 4.10 – Lancement de l'instance EC2 (client Windows)

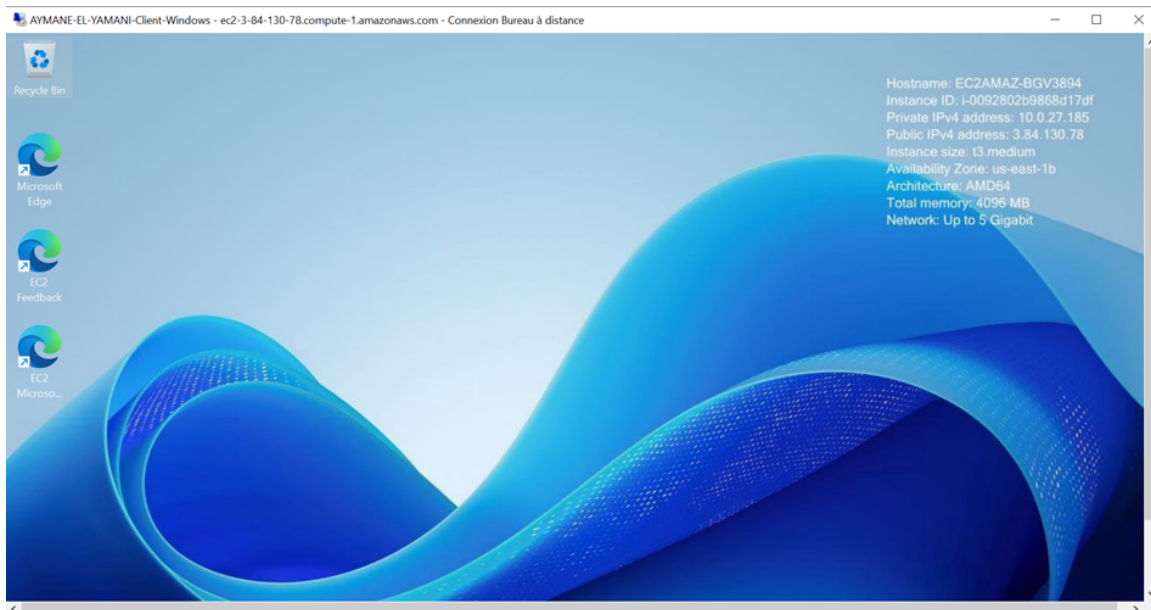


FIGURE 4.11 – Affichage et accès au client Windows

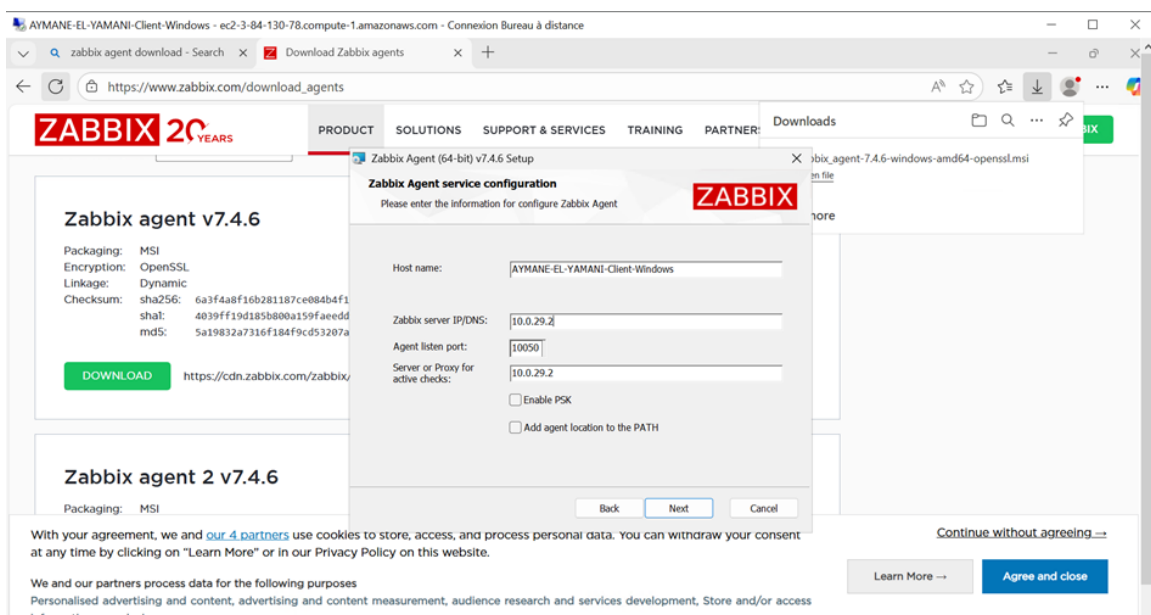


FIGURE 4.12 – Installation de l'agent Zabbix sur Windows

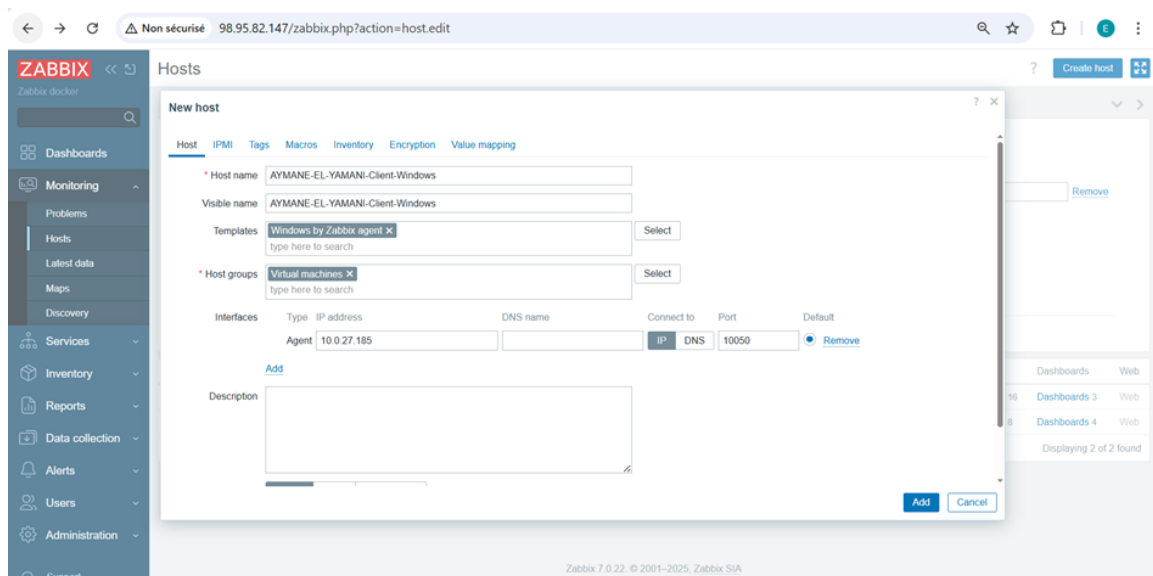


FIGURE 4.13 – Paramétrage de l’hôte Windows dans Zabbix

4.5 Monitoring et tableaux de bord

Après ajout des hôtes, la réception des données est vérifiée via le statut et les écrans “Latest data” et “Graphs”.

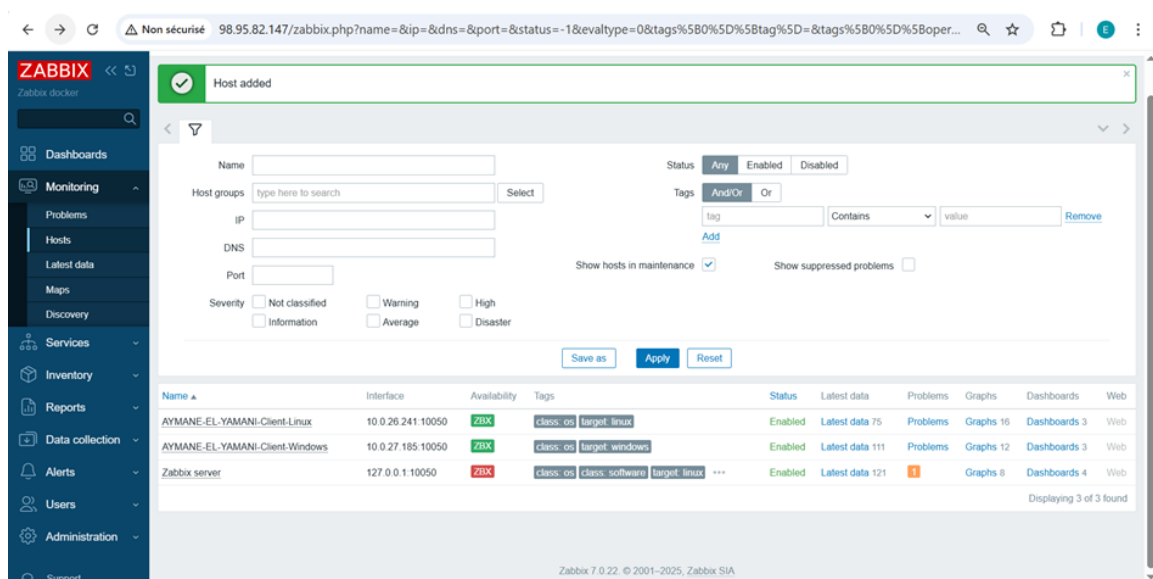


FIGURE 4.14 – Ajout des hôtes réussi

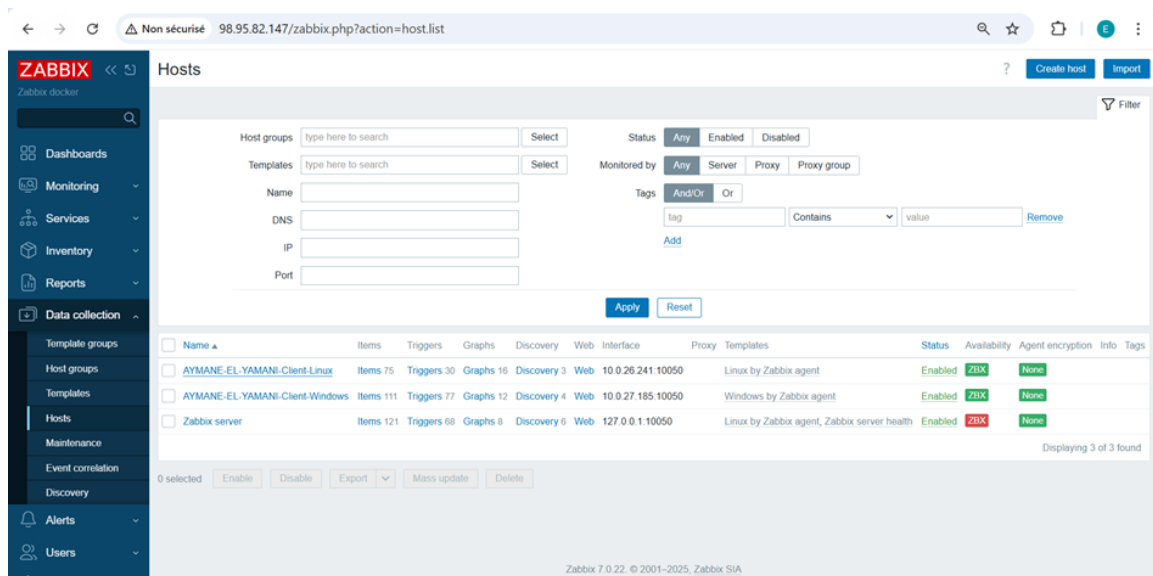


FIGURE 4.15 – Liste des hôtes (statut de connexion)

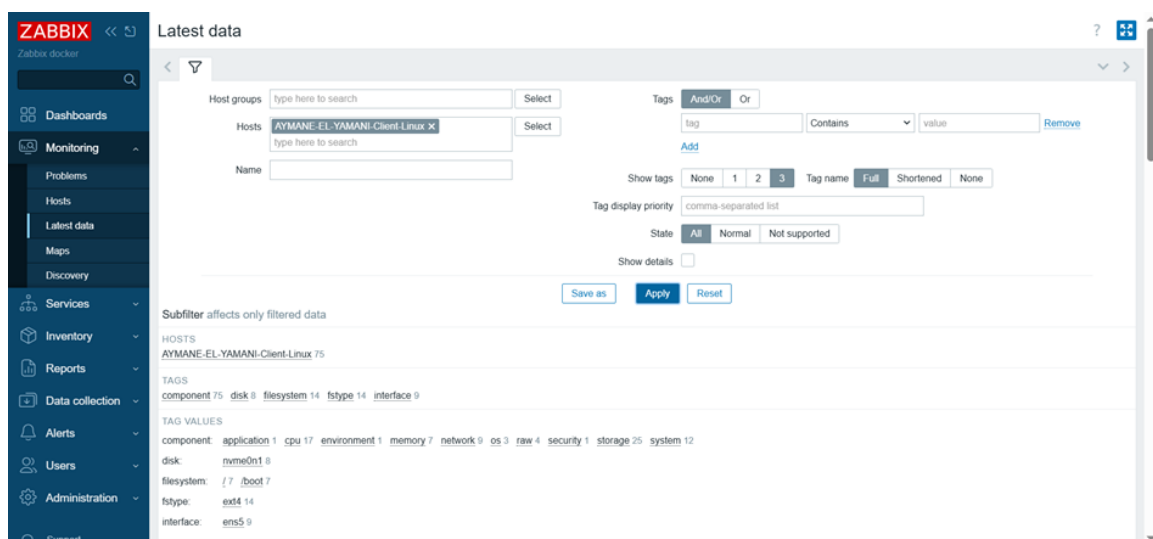


FIGURE 4.16 – “Latest data” pour le client Linux

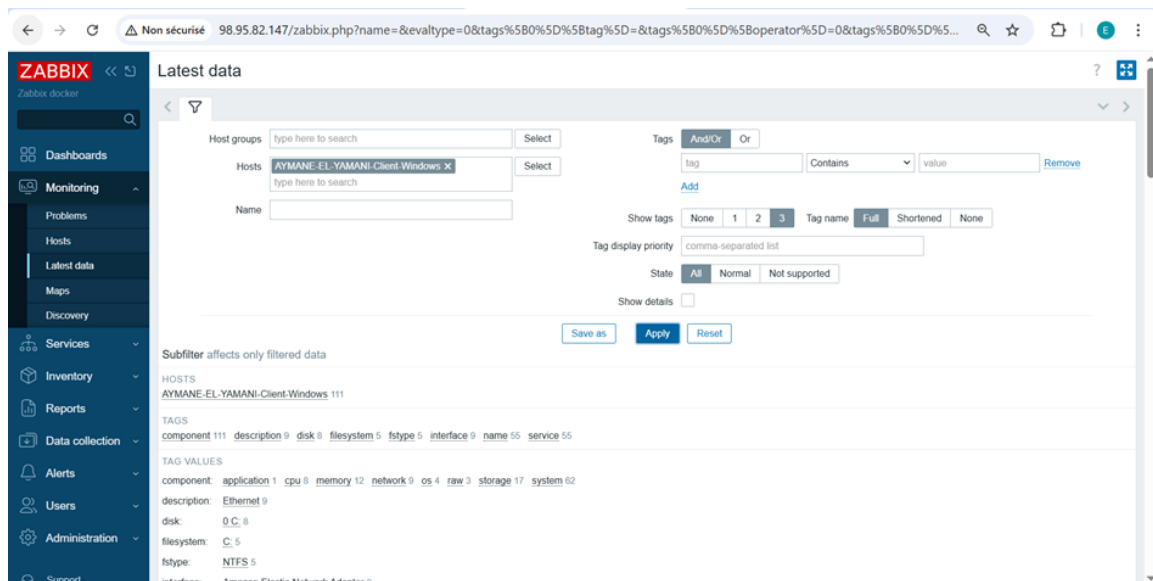


FIGURE 4.17 – “Latest data” pour le client Windows

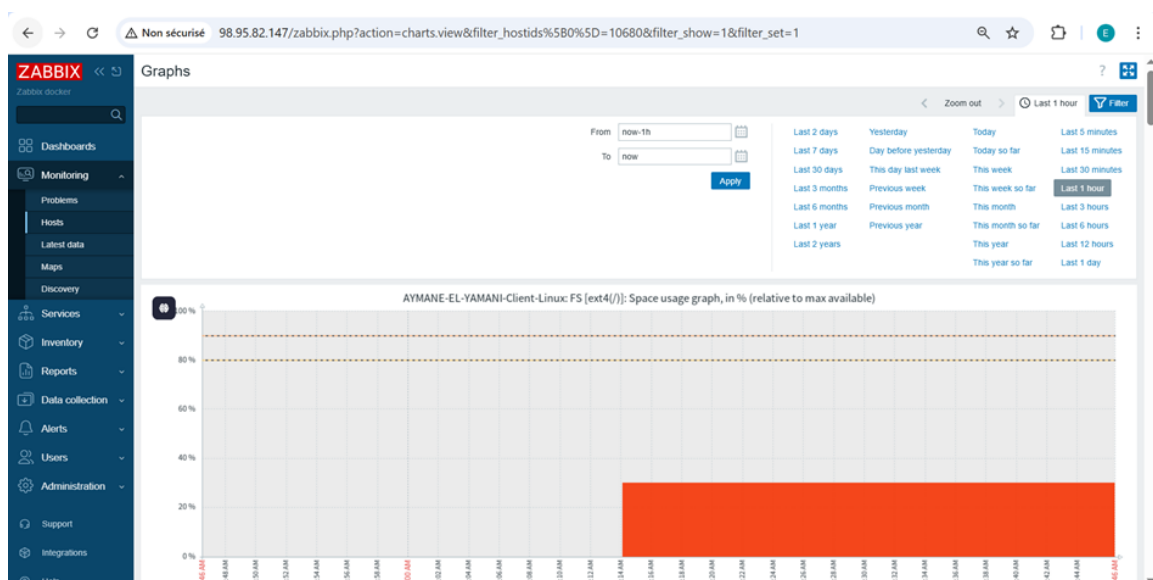


FIGURE 4.18 – Graphiques de supervision (CPU/RAM, etc.)

Chapitre 5

Problèmes rencontrés et solutions

5.1 Problèmes réseau et accès

- **Ports bloqués (10050/10051)** : vérification des règles du Security Group et des pare-feux locaux.
- **Accès web Zabbix** : s'assurer que le port 80/443 est autorisé et que l'IP publique de l'instance est correcte.

5.2 Contraintes du Learner Lab

Dans le cadre du Lab, certaines limitations sont à respecter :

- Types d'instances limités (ex. t3.medium, t3.large).
- Région recommandée : **us-east-1** (N. Virginia).
- Arrêt automatique du Lab : redémarrage nécessaire des services.

Listing 5.1 – Relance des services Docker (exemple)

```
docker compose up -d
```

- Suivi du budget : arrêt des instances hors usage.

Chapitre 6

Résultats

Les résultats attendus sont atteints lorsque :

- les hôtes Linux et Windows apparaissent dans Zabbix avec un statut sain (“ZBX” actif),
- les métriques remontent en temps réel (“Latest data”),
- des graphiques sont disponibles pour les ressources (CPU, RAM, réseau),
- une alerte peut être déclenchée et observée en supervision.

Chapitre 7

Conclusion

Ce projet démontre la mise en place d'une supervision centralisée sur AWS, basée sur Zabbix déployé via Docker. L'approche conteneurisée facilite le déploiement et la maintenance, tandis que l'installation des agents sur Linux et Windows permet d'obtenir une visibilité opérationnelle sur un parc hybride.

7.1 Perspectives

- Sécurisation renforcée (accès privé via VPN/bastion, moindre exposition publique).
- Automatisation (Infrastructure as Code : Terraform/CloudFormation).
- Haute disponibilité (multi-AZ, base de données managée, sauvegardes).

7.2 Dépôt GitHub

<https://github.com/aymaneElyamani/zabbix-docker-aws-monitoring>