

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

Réalisé par : NAIT ALI KHALID

Encadré par : Prof. Azeddine KHIAT

Année universitaire : 2025 – 2026

---

## Table des matières

1. Introduction
  2. Architecture et configuration réseau
    - 2.1 Mise en place du VPC et du sous-réseau
    - 2.2 Configuration de la sécurité (Security Groups)
    - 2.3 Connectivité Internet et routage
    - 2.4 Schéma de l'architecture finale
  3. Architecture des instances EC2
    - 3.1 Tableau récapitulatif des ressources
    - 3.2 État du déploiement dans la console AWS
  4. Déploiement du serveur Zabbix via Docker
    - 4.1 Installation de Docker et Docker Compose
    - 4.2 Configuration de l'orchestration Docker Compose
    - 4.3 Lancement et vérification
  5. Configuration des agents de supervision
    - 5.1 Installation de l'agent sur Linux Client
    - 5.2 Configuration du service
    - 5.3 Initialisation et persistance
    - 5.4 Enregistrement sur l'interface Web
    - 5.5 Installation de l'agent sur Windows
    - 5.6 Validation de la connectivité
    - 5.7 Enregistrement et supervision
  6. Monitoring et validation
    - 6.1 Création du Dashboard global
    - 6.2 Mise en place d'un Trigger (alerte proactive)
    - 6.3 Synthèse visuelle
  7. Conclusion
- 

## 1. Introduction

Dans le cadre de la gestion moderne des infrastructures informatiques, la supervision constitue un élément essentiel afin de garantir la disponibilité, la performance et la sécurité des systèmes. Le recours aux solutions de Cloud Computing permet aujourd'hui de déployer rapidement des architectures robustes et évolutives.

Ce projet a pour objectif la mise en œuvre d’une infrastructure cloud de supervision centralisée hébergée sur Amazon Web Services (AWS), basée sur l’outil open source **Zabbix**. Le serveur Zabbix est déployé sous forme conteneurisée à l’aide de Docker afin de simplifier la gestion des services et d’assurer une meilleure portabilité.

L’infrastructure permet la supervision d’un environnement hybride composé de machines Linux et Windows, avec la collecte de métriques telles que l’utilisation du processeur, de la mémoire et l’état général des systèmes.

Lien du dépôt GitHub : .....

## 2. Architecture et configuration réseau

L’infrastructure réseau est isolée dans un VPC dédié, situé dans la région **us-east-1**, avec une configuration stricte des flux via les groupes de sécurité afin d’assurer la sécurité des communications.

### 2.1 Mise en place du VPC et du sous-réseau

Un VPC nommé **VPC-Projet-Zabbix** a été créé avec un bloc CIDR **10.0.0.0/16**. À l’intérieur de ce VPC, un sous-réseau public **10.0.0.0/24** a été configuré pour héberger les différentes instances EC2.

Figure 1 – Création du VPC Projet Zabbix

<input type="checkbox"/>	Name	ID de VPC	État	ID de contrôl...	Mode de contrôle d...	Bloquer l'ac...	CIDR IPv4	CII
<input type="checkbox"/>	-	<a href="#">vpc-0c4a255da33006a00</a>	Available	-	-	Désactivé	172.31.0.0/16	-
<input type="checkbox"/>	VPC_Projet_Zabbix	<a href="#">vpc-01173f63b155e2cec</a>	Available	-	-	Désactivé	10.0.0.0/16	-

Figure 2 – Configuration du sous-réseau (Subnet)

### 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.

Subnet-VPC-Projet-Zabbix

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.

Aucune préférence

**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**

10.0.0.0/24 256 IPs

## 2.2 Configuration de la sécurité (Security Groups)

Deux groupes de sécurité ont été mis en place :

- **Zabbix-Server-SG :**
  - Port 22 (SSH)
  - Ports 80 et 443 (HTTP/HTTPS)
  - Port 10051 (communication serveur Zabbix)
- **Agents-SG :**
  - Port 10050 (agent Zabbix)
  - Port 22 (administration Linux)
  - Port 3389 (RDP Windows)

Figure 3 – Règles du groupe de sécurité Zabbix Server

Détails de base

Nom du groupe de sécurité Informations

Zabbix-Server-SG

Le nom ne peut pas être modifié après sa création.

Description Informations

Security group for Zabbix Server

VPC Informations

vpc-01173f63b155e2cec (VPC\_Projet\_Zabbix)

Règles entrantes Informations

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

Ajouter une règle

Règles sortantes Informations

Type Informations	Protocole Informations	Plage de ports Informations	Destination Informations	Description - facultatif Informations	
SSH	TCP	22	Mon IP		Supprimer
HTTP	TCP	80	N'import...	41.141.186.247/32	Supprimer
HTTPS	TCP	443	N'import...	0.0.0.0/0	Supprimer
TCP personnalisé	TCP	10051	N'import...	0.0.0.0/0	Supprimer
				0.0.0.0/0	Supprimer

Ajouter une règle

Figure 4 – Règles du groupe de sécurité des Agents

Détails de base

Nom du groupe de sécurité Informations

Agents-SG

Le nom ne peut pas être modifié après sa création.

Description Informations

Security group for Zabbix Server

VPC Informations

vpc-01173f63b155e2cec (VPC\_Projet\_Zabbix)

Règles entrantes Informations

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

Ajouter une règle

Règles sortantes Informations

Type Informations	Protocole Informations	Plage de ports Informations	Destination Informations	Description - facultatif Informations	
SSH	TCP	22	Mon IP		Supprimer
RDP	TCP	3389	Mon IP	41.141.186.247/32	Supprimer
TCP personnalisé	TCP	10050	N'import...	41.141.186.247/32	Supprimer
				0.0.0.0/0	Supprimer

Ajouter une règle

## 2.3 Connectivité Internet et routage

Une **Internet Gateway (IGW)** a été attachée au VPC afin de permettre l'accès à Internet. Une route par défaut **0.0.0.0/0** a été ajoutée dans la table de routage associée au sous-réseau public.

Figure 5 – Création de l'Internet Gateway

> [Passerelles Internet](#) > Créer une passerelle Internet

### Créer une passerelle Internet Infos

Une passerelle Internet est un routeur virtuel qui connecte un VPC à Internet. Pour créer une nouvelle passerelle Internet, spécifiez le nom de la passerelle ci-dessous.

#### Paramètres de passerelle Internet

##### Identification de nom

Créez une identification avec une clé du « Nom » et une valeur que vous spécifiez.

IGW\_Zabbix

#### Balises - facultatif

Une balise est une étiquette que vous attribuez à une ressource AWS. Chaque balise est constituée d'une clé et d'une valeur facultative. Vous pouvez utiliser des balises pour rechercher et filtrer vos ressources ou réaliser le suivi vos coûts AWS.

**Clé**

Q, Name X

**Valeur - facultatif**

Q, IGW\_Zabbix X Supprimer

Ajouter une nouvelle balise

Vous pouvez ajouter 49 d'autres balises.

Annuler Créer une passerelle Internet

Figure 6 – Configuration de la table de routage

VPC > [Tables de routage](#) > Créer une table de routage

### Créer une table de routage Infos

Une table de routage spécifie comment les paquets sont transférés entre les sous-réseaux au sein de votre VPC, sur Internet et votre connexion VPN.

#### Paramètres de la table de routage

##### Nom - facultatif

Créez une balise avec une clé « Name » et une valeur à spécifier.

route\_zabbix

##### VPC

VPC à utiliser pour cette table de routage.

vpc-01173f63b155e2cec (VPC\_Projet\_Zabbix)

#### Balises

Une balise est une étiquette que vous attribuez à une ressource AWS. Chaque balise est constituée d'une clé et d'une valeur facultative. Vous pouvez utiliser des balises pour rechercher et filtrer vos ressources ou réaliser le suivi vos coûts AWS.

**Clé**

Q, Name X

**Valeur - facultatif**

Q, route\_zabbix X Supprimer

Ajouter une nouvelle balise

Vous pouvez ajouter 49 d'autres balises.

Annuler Créer une table de routage

## 2.4 Schéma de l'architecture finale

Le schéma ci-dessous illustre l'architecture globale de l'infrastructure déployée sous AWS.

Figure 7 – Visualisation globale de l'architecture réseau AWS



## 3. Architecture des instances EC2

L'infrastructure repose sur trois instances EC2 distinctes, dimensionnées selon leur rôle.

### 3.1 Tableau récapitulatif des ressources

Rôle	Type d'instance	Système d'exploitation	Fonction
Serveur Zabbix	t3.large	Ubuntu 22.04 LTS	Supervision centrale
Client Linux	t3.medium	Ubuntu 22.04 LTS	Agent de monitoring

Rôle	Type d'instance	Système d'exploitation	Fonction
Client Windows	t3.large	Windows Server 2022	Agent de monitoring

Tableau 1 – Caractéristiques techniques des instances déployées

### 3.2 État du déploiement dans la console AWS

Les instances ont été vérifiées depuis la console AWS et sont toutes en état **Running**.

Figure 8 – Instances EC2 en état Running

<input type="checkbox"/>	Zabbix-Server	i-0a3c7d284f40a377b	En cours d'...	t3.large	3/3 vérifications r	<a href="#">Afficher les alarm</a>	us-east-1f	-	3.238.126.110	-	-
<input type="checkbox"/>	Linux-client-zabbix	i-0e88c405dc78ccb7f	En cours d'...	t3.medium	3/3 vérifications r	<a href="#">Afficher les alarm</a>	us-east-1f	-	98.93.0.56	-	-
<input type="checkbox"/>	Windows-client-zabbix	i-0a80c6a3bf79358d6	En cours d'...	t3.large	Initialisation en cc	<a href="#">Afficher les alarm</a>	us-east-1f	-	3.255.224.91	-	-

## 4. Déploiement du serveur Zabbix via Docker

### 4.1 Installation de Docker et Docker Compose

Docker et Docker Compose ont été installés sur l'instance Ubuntu afin d'orchestrer les services Zabbix.

```
ubuntu@ip-10-0-0-130: ~  
ubuntu@ip-10-0-0-130:~$ sudo apt update && sudo apt install docker . io docker - compose -y  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease  
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]  
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]  
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]  
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]  
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]  
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]  
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]  
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]  
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]  
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]  
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]  
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1738 kB]  
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [324 kB]  
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]  
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [16.5 kB]  
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1528 kB]  
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [313 kB]  
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [386 kB]  
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [31.9 kB]  
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [2582 kB]  
Get:22 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1431 kB]  
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [591 kB]  
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]  
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [556 B]  
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [32.1 kB]  
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [6816 B]  
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]  
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [496 B]  
  
No VM guests are running outside hypervisor (qemu) binaries  
ubuntu@ip-10-0-0-130:~$ docker --version  
Docker version 28.2.2, build 28.2.2-0ubuntu1~24.04.1  
ubuntu@ip-10-0-0-130:~$ docker-compose --version  
docker-compose version 1.29.2, build unknown  
ubuntu@ip-10-0-0-130:~$ |
```

## 4.2 Configuration de l'orchestration Docker Compose

Un fichier `docker-compose.yml` a été utilisé pour déployer les services suivants :

- Base de données MySQL
- Zabbix Server
- Interface Web Zabbix (Nginx)

Les variables sensibles ont été stockées dans un fichier `.env`.

```
version: '3.8'

services:
  mysql:
    image: mysql:8.0
    container_name: zabbix-mysql
    restart: unless-stopped
    environment:
      MYSQL_DATABASE: zabbix
      MYSQL_USER: zabbix
      MYSQL_PASSWORD: zabbix_pass
      MYSQL_ROOT_PASSWORD: root_pass
    volumes:
      - mysql_data:/var/lib/mysql
    networks:
      - zabbix-net

  zabbix-server:
    image: zabbix/zabbix-server-mysql:7.4-latest
    container_name: zabbix-server
    restart: unless-stopped
    depends_on:
      - mysql
    environment:
      DB_SERVER_HOST: mysql
      MYSQL_DATABASE: zabbix
      MYSQL_USER: zabbix
      MYSQL_PASSWORD: zabbix_pass
      MYSQL_ROOT_PASSWORD: root_pass
    ports:
      - "10051:10051"
    networks:
      - zabbix-net

  zabbix-web:
    image: zabbix/zabbix-web-nginx-mysql:7.4-latest
    container_name: zabbix-web
    restart: unless-stopped
    depends_on:
      - mysql
      - zabbix-server
    environment:
      DB_SERVER_HOST: mysql
      MYSQL_DATABASE: zabbix
      MYSQL_USER: zabbix
      MYSQL_PASSWORD: zabbix_pass
      ZBX_SERVER_HOST: zabbix-server
      PHP_TZ: Africa/Casablanca
    ports:
      - "80:8080"
    networks:
      - zabbix-net

volumes:
  mysql_data:

networks:
  zabbix-net:
    driver: bridge
```



```

ubuntu@ip-10-0-0-130:~/zabbix$ sudo docker-compose up -d
Creating network "zabbix_default" with the default driver
Creating volume "zabbix_mysql_data" with default driver
Pulling mysql-server (mysql:8.0)...
8.0: Pulling from library/mysql
4f37333d1be6: Pull complete
bde62e757594: Pull complete
f508d7fab5b3: Pull complete
d442b2c1726e: Pull complete
a9a9deeee02a: Pull complete
23fbf4028535: Pull complete
2e2c1f6f8d57: Pull complete
ce98f3559366: Pull complete
bae900376130: Pull complete
e7a04c019bde: Pull complete
e05db5310ebc: Pull complete
Digest: sha256:99d774bf02a48a1bb1c599920d2571946d31e5940b854b02737d5e95c184358f
Status: Downloaded newer image for mysql:8.0
Pulling zabbix-server (zabbix/zabbix-server-mysql:latest)...
latest: Pulling from zabbix/zabbix-server-mysql
589002ba0eae: Pull complete
1e121fca2202: Pull complete
b3be40b1c527: Pull complete
17cc83c10761: Pull complete
ca8469c34092: Pull complete
27f081603218: Pull complete
4f4fb700ef54: Pull complete
eade5553925f: Pull complete
Digest: sha256:7f511490e6e479d2a7ff1a078dccc94b7a1a51f8eb164c174f2678137cee36cf
Status: Downloaded newer image for zabbix/zabbix-server-mysql:latest

```

```

ubuntu@ip-10-0-0-130:~$ sudo docker ps

```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
d585f17bf40c	zabbix/zabbix-web-nginx-mysql:latest	"docker-entrypoint.sh"	2 minutes ago	Up 2 minutes (healthy)
078cedf7927c	zabbix/zabbix-server-mysql:latest	"/usr/bin/docker-ent..."	2 minutes ago	Restarting (1) 13 seconds ago
954b758e0981	mysql:8.0	"docker-entrypoint.s..."	2 minutes ago	Up 2 minutes

```

ubuntu@ip-10-0-0-130:~$

```

### 4.3 Lancement et vérification

Les services ont été lancés via la commande `docker-compose up -d`. L'interface Web Zabbix est accessible via l'adresse IP publique du serveur.

```
ubuntu@ip-10-0-0-130:~/zabbix$ sudo docker-compose up -d
Creating network "zabbix_default" with the default driver
Creating volume "zabbix_zabbix-db-data" with default driver
Pulling zabbix-db (mysql:8.0)...
8.0: Pulling from library/mysql
4f37333d1be6: Pull complete
bde62e757594: Pull complete
f508d7fab5b3: Pull complete
d442b2c1726e: Pull complete
a9a9deeee02a: Pull complete
23fbf4028535: Pull complete
2e2c1f6f8d57: Pull complete
ce98f3559366: Pull complete
bae900376130: Pull complete
e7a04c019bde: Pull complete
e05db5310ebc: Pull complete
Digest: sha256:99d774bf02a48a1bb1c599920d2571946d31e5940b854b02737d5e95c184358f
Status: Downloaded newer image for mysql:8.0
Pulling zabbix-server (zabbix/zabbix-server-mysql:ubuntu-6.4-latest)...
ubuntu-6.4-latest: Pulling from zabbix/zabbix-server-mysql
de44b265507a: Pull complete
c15ab064eea3: Pull complete
a2ab853cdf6c: Pull complete
5f5562f77cda: Pull complete
66367a2ddd2d: Pull complete
210f7f100111: Pull complete

ubuntu@ip-10-0-0-130:~/zabbix$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND NAMES	CREATED	STATUS
189bd43fe3c0	zabbix/zabbix-web-nginx-mysql:ubuntu-6.4-latest	"docker-entrypoint.sh"	About a minute ago	Up 59 sec
onds (healthy)	8443/tcp, 0.0.0.0:80->8080/tcp, [::]:80->8080/tcp	zabbix-web		
4e7c0fc6d142	zabbix/zabbix-server-mysql:ubuntu-6.4-latest	"/usr/bin/tini -- /u..."	About a minute ago	Up About
a minute	0.0.0.0:10051->10051/tcp, [::]:10051->10051/tcp	zabbix-server		
8cf4eab36863	mysql:8.0	"docker-entrypoint.s..."	About a minute ago	Up About
a minute	3306/tcp, 33060/tcp	zabbix-db		

Figure 9 – Lancement des conteneurs



Username

Admin

Password

.....

☒ Remember me for 30 days

Sign in

[Help](#) • [Support](#)

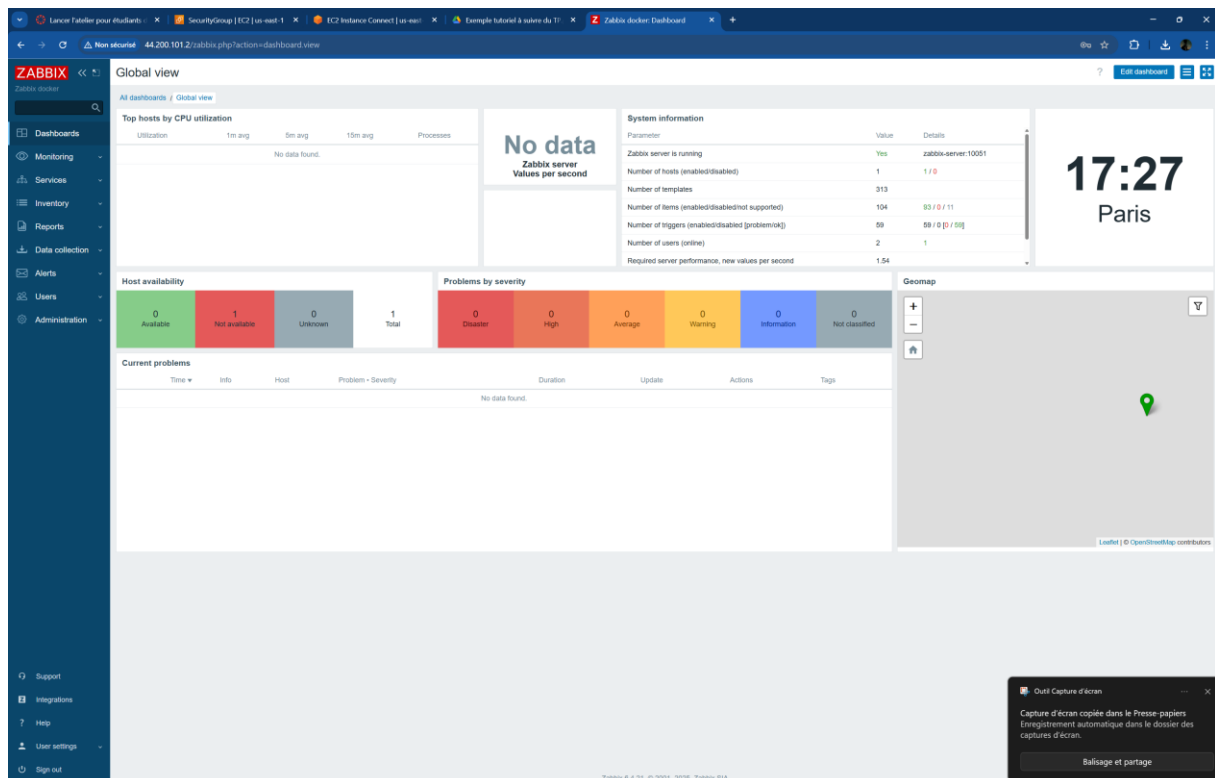


Figure 10 – Interface de connexion Zabbix

## 5. Configuration des agents de supervision

### 5.1 Installation de l'agent sur Linux Client

L'agent Zabbix a été installé depuis le dépôt officiel et configuré pour communiquer avec le serveur central.

```
ubuntu@ip-10-0-0-233:~$ wget https://repo.zabbix.com/zabbix/6.4/ubuntu/pool/main/z/zabbix-release/zabbix-release_6.4-1+ubuntu22.04_all.deb
--2026-02-06 18:09:31-- https://repo.zabbix.com/zabbix/6.4/ubuntu/pool/main/z/zabbix-release/zabbix-release_6.4-1+ubuntu22.04_all.deb
Resolving repo.zabbix.com (repo.zabbix.com)... 178.128.6.101, 2604:a880:2:d0::2062:d001
Connecting to repo.zabbix.com (repo.zabbix.com)|178.128.6.101|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3744 (3.7K) [application/octet-stream]
Saving to: 'zabbix-release_6.4-1+ubuntu22.04_all.deb'

zabbix-release_6.4-1+ubuntu22.04_all.deb 100%[=====] 3.66K --.-KB/s in 0s

2026-02-06 18:09:31 (812 MB/s) - 'zabbix-release_6.4-1+ubuntu22.04_all.deb' saved [3744/3744]

ubuntu@ip-10-0-0-233:~$ sudo dpkg -i zabbix-release_6.4-1+ubuntu22.04_all.deb
Selecting previously unselected package zabbix-release.
(Reading database ... 71752 files and directories currently installed.)
Preparing to unpack zabbix-release_6.4-1+ubuntu22.04_all.deb ...
Unpacking zabbix-release (1:6.4-1+ubuntu22.04) ...
Setting up zabbix-release (1:6.4-1+ubuntu22.04) ...
ubuntu@ip-10-0-0-233:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 https://repo.zabbix.com/zabbix/6.4/ubuntu jammy InRelease [2883 B]
Get:7 https://repo.zabbix.com/zabbix/6.4/ubuntu jammy/main Sources [23.8 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1431 kB]
Get:9 https://repo.zabbix.com/zabbix/6.4/ubuntu jammy/main amd64 Packages [63.5 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:11 https://repo.zabbix.com/zabbix/6.4/ubuntu jammy/main all Packages [12.7 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [232 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.6 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [9888 B]
Get:15 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [929 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:17 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [212 kB]
Get:18 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [74.2 kB]
```

```

ubuntu@ip-10-0-0-233:~$ sudo apt install zabbix-agent -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libdbus5
The following NEW packages will be installed:
  libdbus5 zabbix-agent
0 upgraded, 2 newly installed, 0 to remove and 78 not upgraded.
Need to get 299 kB of archives.
After this operation, 823 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libdbus5 amd64 3.1.10-1ubuntu1 [34.4 kB]
Get:2 https://repo.zabbix.com/zabbix/6.4/ubuntu noble/main amd64 zabbix-agent amd64 1:6.4.21-1+ubuntu24.04 [264 kB]
Fetched 299 kB in 1s (470 kB/s)
Selecting previously unselected package libdbus5:amd64.
(Reading database ... 71759 files and directories currently installed.)
Preparing to unpack .../libdbus5_3.1.10-1ubuntu1_amd64.deb ...
Unpacking libdbus5:amd64 (3.1.10-1ubuntu1) ...
Selecting previously unselected package zabbix-agent.
Preparing to unpack .../zabbix-agent_1%3a6.4.21-1+ubuntu24.04_amd64.deb ...
Unpacking zabbix-agent (1:6.4.21-1+ubuntu24.04) ...
Setting up libdbus5:amd64 (3.1.10-1ubuntu1) ...
Setting up zabbix-agent (1:6.4.21-1+ubuntu24.04) ...
Created symlink /etc/systemd/system/multi-user.target.wants/zabbix-agent.service → /usr/lib/systemd/system/zabbix-agent.service.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-10-0-0-233:~$

```

## 5.2 Configuration du service

Les paramètres **Server**, **ServerActive** et **Hostname** ont été configurés dans le fichier `zabbix_agentd.conf`.

Fichier	Modifier	Affichage
<code>Server=10.0.0.130</code>		
<code>ServerActive=10.0.0.130</code>		
<code>Hostname=Linux-client-zabbix</code>		

## 5.3 Initialisation et persistance

Le service Zabbix Agent a été redémarré et activé au démarrage du système.

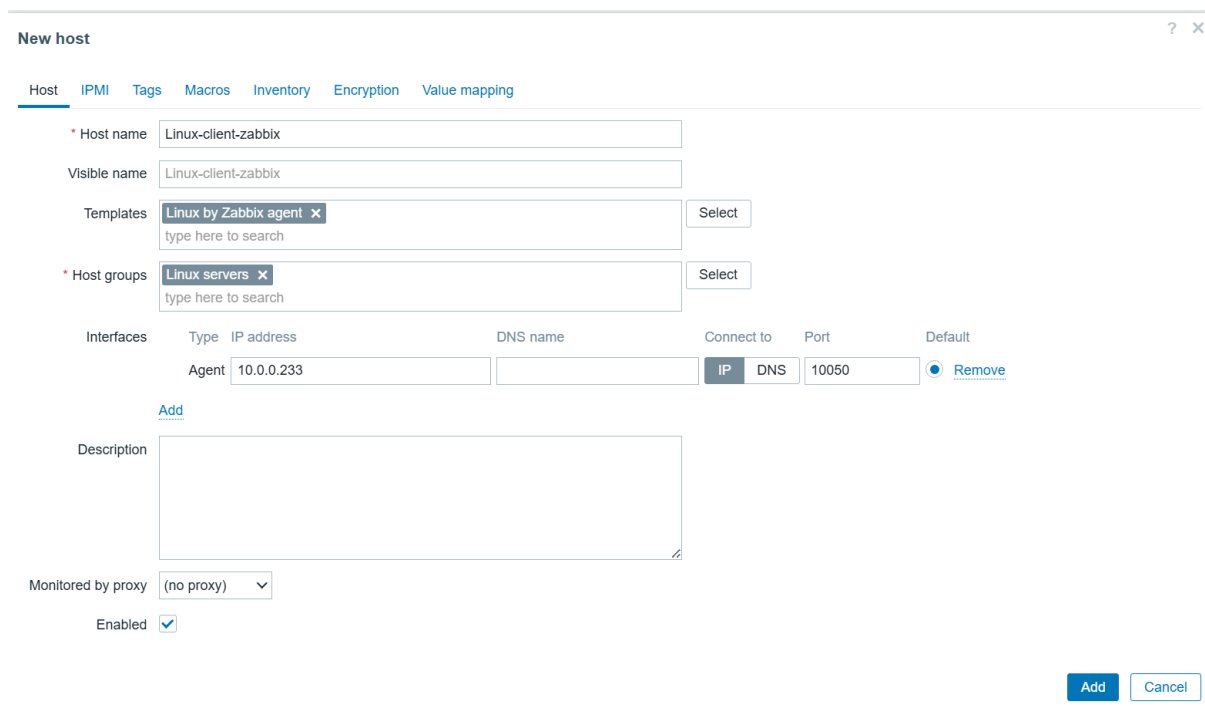
```

ubuntu@ip-10-0-0-233:~$ sudo systemctl restart zabbix-agent
sudo systemctl enable zabbix-agent
Synchronizing state of zabbix-agent.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable zabbix-agent
ubuntu@ip-10-0-0-233:~$ sudo systemctl status zabbix-agent

```

## 5.4 Enregistrement sur l'interface Web

L'hôte Linux a été ajouté dans l'interface Zabbix avec le template approprié.

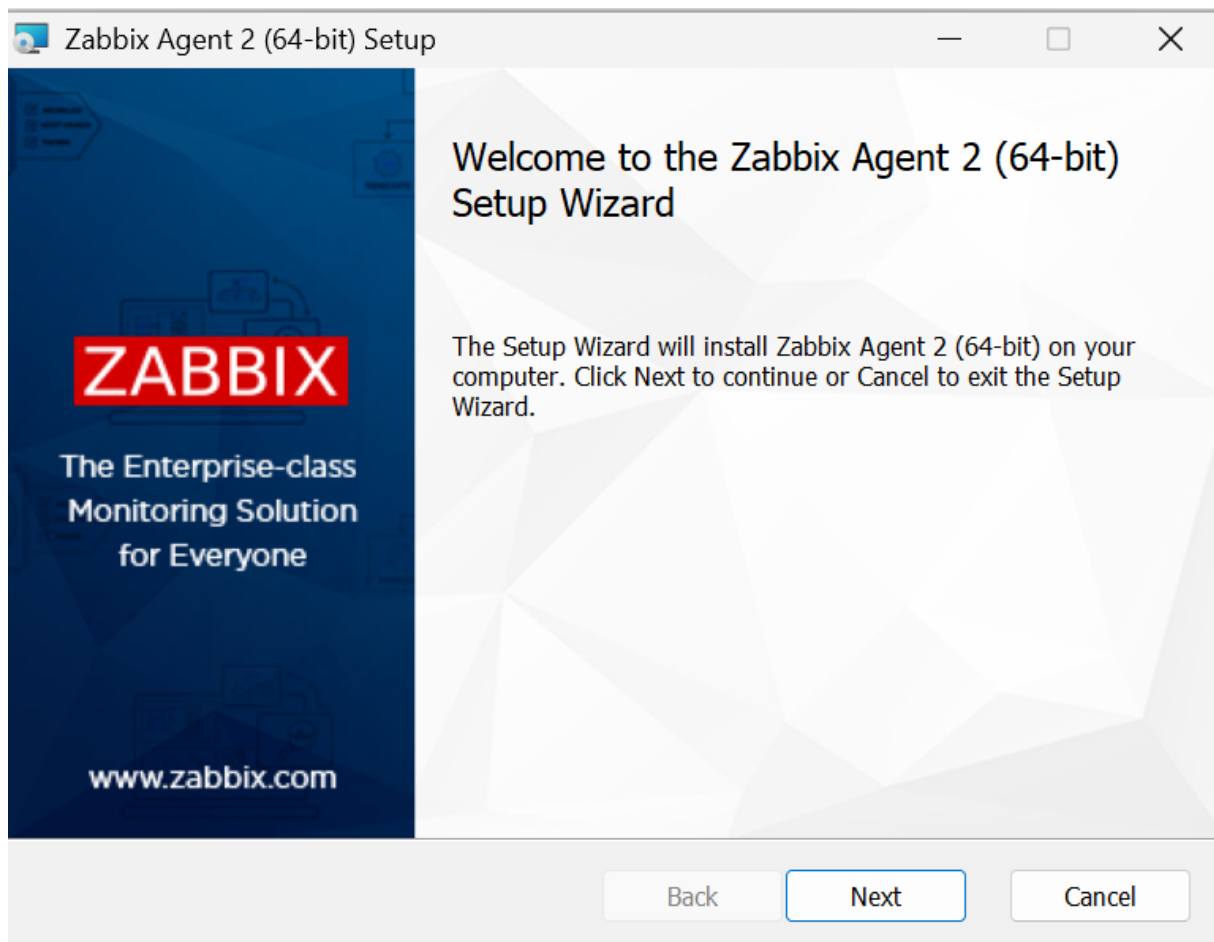


The screenshot shows the 'New host' configuration window in Zabbix. The 'Host' tab is selected, showing fields for 'Host name' (Linux-client-zabbix), 'Visible name' (Linux-client-zabbix), 'Templates' (Linux by Zabbix agent), and 'Host groups' (Linux servers). Below these is a table for 'Interfaces' with one entry: 'Agent' at IP '10.0.0.233' connected to 'IP' on port '10050'. The 'Description' field is empty. At the bottom, 'Monitored by proxy' is set to '(no proxy)' and 'Enabled' is checked. 'Add' and 'Cancel' buttons are at the bottom right.

Interfaces	Type	IP address	DNS name	Connect to	Port	Default
Agent		10.0.0.233		IP	DNS	10050

## 5.5 Installation de l'agent sur Windows

L'agent Zabbix a été installé sur Windows Server via le package MSI officiel.



## 5.6 Validation de la connectivité

Un test de connectivité réseau a confirmé le bon fonctionnement du port 10051.

```
ubuntu@ip-10-0-0-130:~$ nc -zv 10.0.0.130 10051
Connection to 10.0.0.130 10051 port [tcp/zabbix-trapper] succeeded!
ubuntu@ip-10-0-0-130:~$ |
```

## 5.7 Enregistrement et supervision

L'hôte Windows a été ajouté à l'interface Zabbix et les métriques ont été correctement reçues.

New host

HostIPMI

Tags

Macros

Inventory

Encryption

Value mapping

\* Host name

Windows-server

Visible name

Windows-server

Templates

Windows by Zabbix agent x

type here to search

Select

\* Host groups

Windows servers (new) x

type here to search

Select

Interfaces

Type

IP address

DNS name

Connect to

Port

Default

Agent

98.92.102.152

IPDNS

10050

Remove

Add

Description

Monitored by proxy

(no proxy)

Enabled

Add

Cancel

Subfilter affects only filtered data

Save as

Apply

Reset

HOSTS

zabbix-client-zabbix +57

Windows-server

Zabbix server +104

TAGS

component 34

TAG VALUES

component: application 1cpu 0memory 12os 4raw 2storage 4system 7

JAVA

With dataWithout data

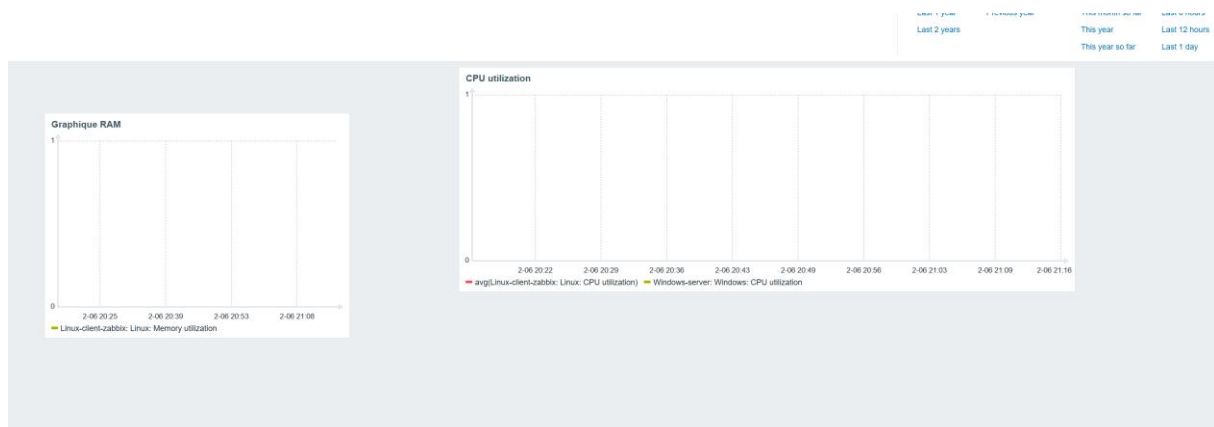
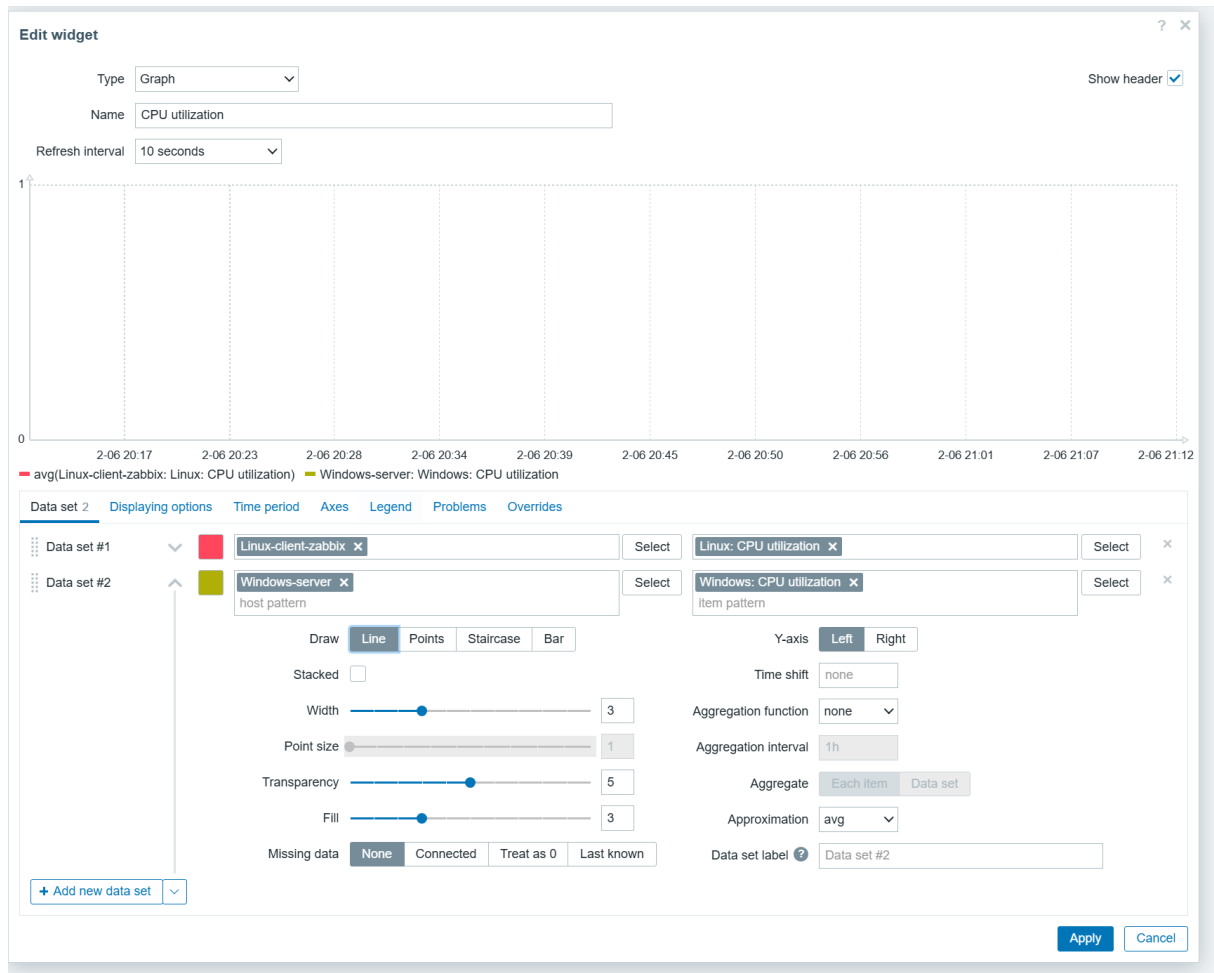
Host	Name	Last check	Last value	Change	Tags	Info
<input type="checkbox"/>	Windows-server				component:memory	Graph
<input type="checkbox"/>	Windows-server				component:cpu	Graph
<input type="checkbox"/>	Windows-server				component:cpu	Graph
<input type="checkbox"/>	Windows-server				component:cpu	Graph
<input type="checkbox"/>	Windows-server				component:cpu	Graph
<input type="checkbox"/>	Windows-server				component:cpu	Graph
<input type="checkbox"/>	Windows-server				component:cpu	Graph
<input type="checkbox"/>	Windows-server				component:cpu	Graph
<input type="checkbox"/>	Windows-server				component:cpu	Graph
<input type="checkbox"/>	Windows-server				component:cpu	Graph
<input type="checkbox"/>	Windows-server				component:memorycomponent:storage	Graph
<input type="checkbox"/>	Windows-server				component:memorycomponent:storage	Graph
<input type="checkbox"/>	Windows-server				component:memory	Graph
<input type="checkbox"/>	Windows-server				component:raw	Graph
<input type="checkbox"/>	Windows-server				component:system	History
<input type="checkbox"/>	Windows-server				component:memory	Graph
<input type="checkbox"/>	Windows-server				component:memory	Graph
<input type="checkbox"/>	Windows-server				component:memory	Graph
<input type="checkbox"/>	Windows-server				component:memory	Graph
<input type="checkbox"/>	Windows-server				component:memory	Graph
<input type="checkbox"/>	Windows-server				component:memory	Graph
<input type="checkbox"/>	Windows-server				component:raw	Graph
<input type="checkbox"/>	Windows-server				component:cpu	Graph
<input type="checkbox"/>	Windows-server				component:os	Graph



## 6. Monitoring et validation

### 6.1 Création du Dashboard global

Un tableau de bord global a été créé afin de centraliser la supervision des différents hôtes.



## 6.2 Mise en place d'un Trigger (alerte proactive)

Un trigger simulant une surcharge CPU a été configuré afin de tester le système d'alerte.

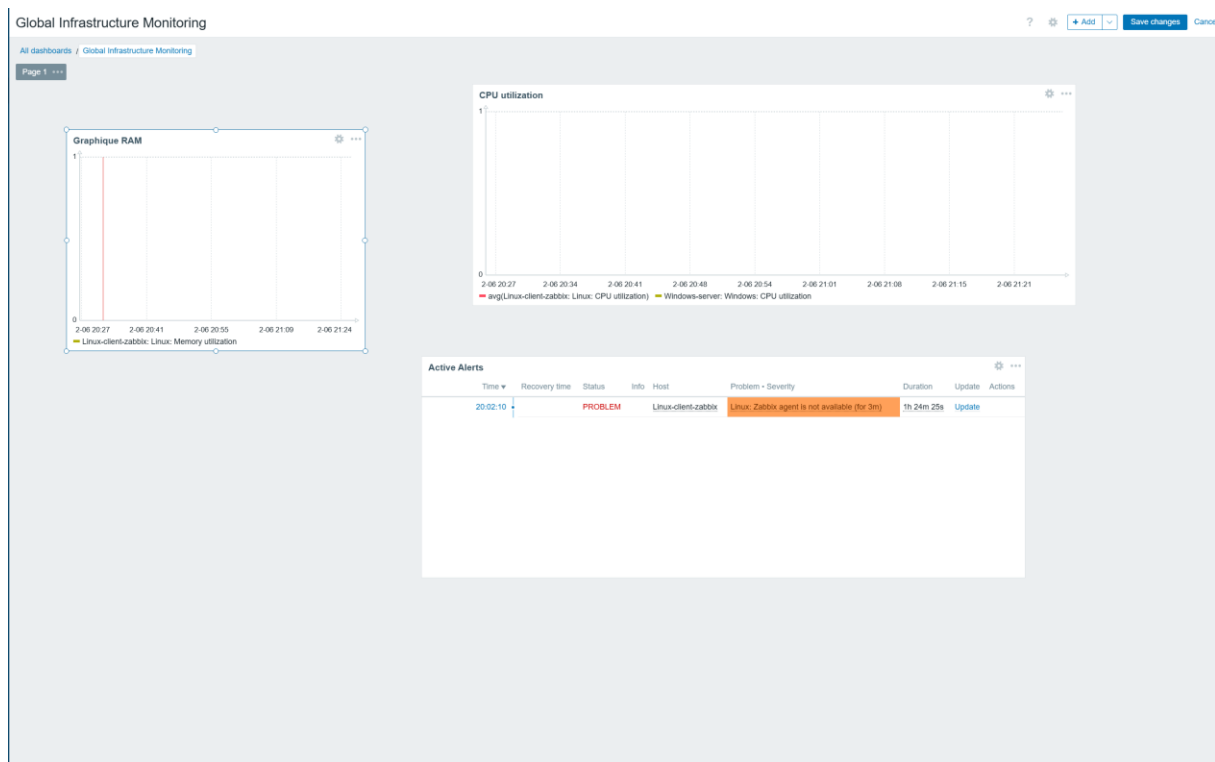
The screenshot shows the Nagios XI interface for configuring a trigger. The main page is titled "Triggers" and shows a list of triggers for the host "Linux-client-zabbix". A dialog box titled "Condition" is open, allowing the user to define the trigger's condition. The dialog box contains the following fields:

- Item:** Linux-client-zabbix: Linux: CPU utilization (selected from a dropdown)
- Function:** last() - Last (most recent) T value (selected from a dropdown)
- Last of (T):** Count (selected from a dropdown)
- Time shift:** now-h (selected from a dropdown)
- Result:** > (selected from a dropdown) 10 (entered in the text field)

The dialog box has "Insert" and "Cancel" buttons. The main page also has "Add" and "Cancel" buttons at the bottom.

## 6.3 Synthèse visuelle

Le dashboard final permet une visualisation complète et centralisée des métriques Linux et Windows.



## 7. Conclusion

Ce projet a permis de mettre en œuvre une infrastructure complète de supervision centralisée sous AWS. L'utilisation de Zabbix conteneurisé via Docker a facilité le déploiement et la maintenance de la solution.

Les objectifs initiaux ont été atteints et l'architecture mise en place constitue une base solide pouvant être enrichie par l'intégration d'outils supplémentaires tels que Grafana ou des systèmes de notification avancés.