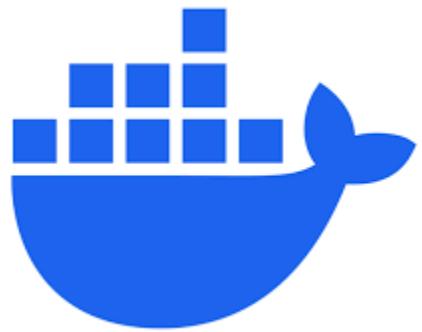




Mise en œuvre d'une infrastructure cloud de supervision centralisée sous AWS

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

Réalisé par:	EL FELLAH Meryem
Encadré par:	KHIAT Azeddine
Filière:	2 ACI INFO groupe A
Module:	Ingénierie des infrastructures CLOUD
Université:	Mundiapolis



ZABBIX

Remerciements

Je tiens à exprimer ma profonde gratitude à mon encadrant, M. Azeddine KHIAT, pour ses directives précieuses et son accompagnement pédagogique tout au long de ce module de **Cloud Computing**. Ce projet a constitué une opportunité majeure pour consolider mes compétences en administration système et en architecture cloud. La mise en œuvre concrète des concepts théoriques abordés en cours, notamment à travers le déploiement de services **AWS** tels que **EC2** et le **VPC**, ainsi que l'usage de la conteneurisation, m'a permis d'acquérir une expertise pratique essentielle.

Avant-propos

La supervision des infrastructures informatiques constitue désormais un pilier stratégique des opérations en entreprise. Dans un contexte de généralisation du Cloud, la surveillance en temps réel de la disponibilité et des performances est devenue un enjeu critique pour assurer la continuité d'activité.

Réalisé dans le cadre du module **Cloud Computing**, ce projet simule un environnement de production réel hébergé sur **Amazon Web Services (AWS)**. L'objectif dépasse le simple déploiement d'instances : il s'agit de concevoir une architecture résiliente et proactive, capable d'alerter les administrateurs dès l'apparition d'un incident.

L'utilisation de la solution **Zabbix**, associée à la conteneurisation via **Docker**, a été choisie pour répondre aux impératifs d'agilité et de standardisation, compétences fondamentales pour un ingénieur. Ce rapport détaille l'intégralité de la démarche technique, de la structuration du réseau VPC à l'analyse fine des métriques de performance du parc hybride.

1. Introduction et Périmètre du Projet

Ce projet, réalisé dans le cadre du module **Cloud Computing**, porte sur la conception et le déploiement d'une infrastructure de supervision centralisée sur le cloud **Amazon Web Services (AWS)**. L'objectif majeur est de mettre en place une solution de monitoring performante capable d'assurer la visibilité et le suivi d'un parc informatique hybride, composé d'instances **Linux** et **Windows Server**.

Cette approche permet de garantir une gestion proactive des ressources cloud tout en s'adaptant à l'hétérogénéité des systèmes d'exploitation rencontrés en milieu professionnel.

1.1 Écosystème Technologique

Pour répondre aux exigences de flexibilité et de scalabilité du projet, les technologies suivantes ont été mobilisées :

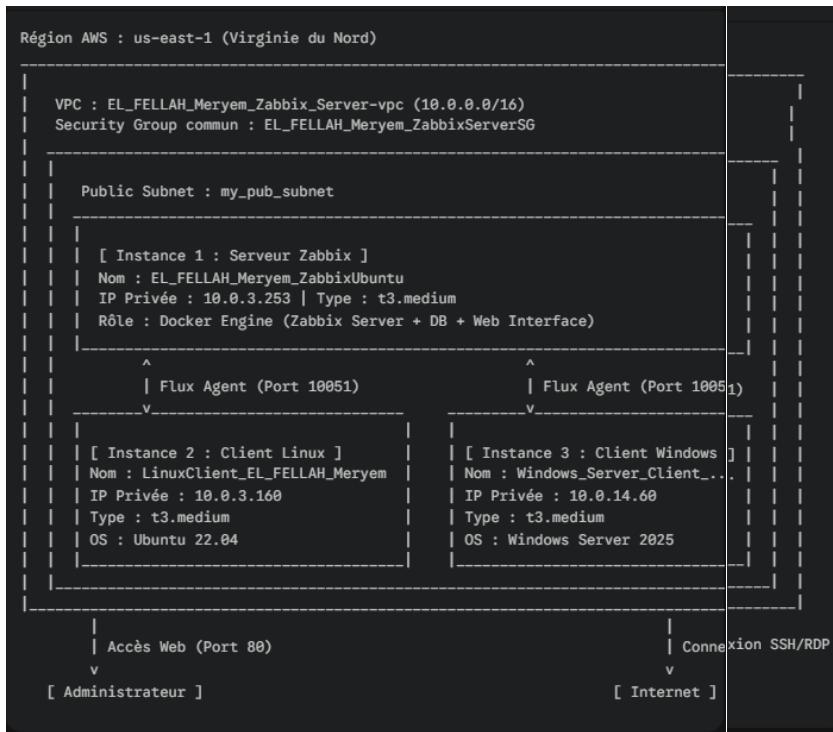
- **Amazon Web Services (AWS)** : Fournisseur de l'infrastructure Cloud. L'architecture repose sur l'utilisation d'instances **EC2** pour le calcul, d'un **VPC** pour l'isolation réseau et de **Security Groups** pour le contrôle granulaire des flux de données.
- **Docker** : Utilisé pour la conteneurisation du serveur Zabbix. Cette technologie permet un déploiement rapide, une portabilité accrue et une gestion simplifiée des dépendances (base de données, serveur web, serveur Zabbix).
- **Zabbix** : Solution de monitoring *open-source* de référence, choisie pour sa capacité à collecter des métriques complexes et à générer des alertes intelligentes via des agents légers installés sur les clients.

1.2 Ressources du projet:

Ce projet a été publié dans un dépôt sur mon compte GitHub:

https://github.com/merra3012/infrastructure_cloud_supervision_centralise_AWS

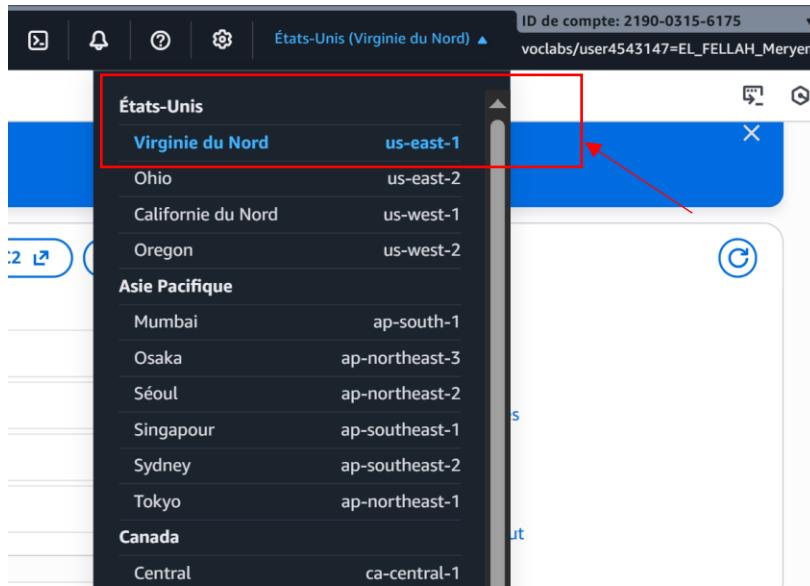
1.3 Architecture globale du projet:



- **Cohérence du Parc :** L'utilisation du type d'instance **t3.medium** pour l'ensemble du parc garantit des performances homogènes pour les tests de charge.
- **Modernité de l'Infrastructure :** Le déploiement du client Windows repose sur l'AMI **Windows Server 2025**, démontrant la capacité de la solution Zabbix à superviser les versions les plus récentes des systèmes d'exploitation Microsoft.
- **Segmentation Réseau :** Bien que les instances soient situées dans des plages d'adresses différentes (10.0.3.x et 10.0.14.x), elles appartiennent au même VPC (**EL_FELLAH_Meryem_Zabbix_Server-vpc**) et au même groupe de sécurité (**EL_FELLAH_Meryem_ZabbixServerSG**), permettant ainsi un routage interne fluide des flux de monitoring.

2. Configuration de l'Infrastructure Cloud (AWS)

Le déploiement de la solution de monitoring repose sur une infrastructure robuste et isolée au sein du cloud **Amazon Web Services (AWS)**. Conformément aux exigences du projet, l'ensemble des ressources a été déployé dans la région **us-east-1 (Virginie du Nord)**, choisie pour sa stabilité et sa compatibilité avec l'environnement *AWS Academy Learner Lab*.



2.1. Architecture Réseau et Sécurité:

Au sein de cette région, un réseau privé virtuel (**VPC**) nommé **EL_FELLAH_Meryem_Zabbix_Server-vpc** a été créé pour assurer l'isolation logique des serveurs. Cette segmentation réseau est fondamentale pour garantir que les flux de métriques entre les agents et le serveur Zabbix transitent de manière sécurisée et contrôlée.

Détails	Informations
ID de VPC	vpc-0e7813a227c90c97f
Résolution DNS	Activé
ACL réseau principal	ad-095a23ab14efb99
CIDR IPv6 (groupe de bordure réseau)	-
ID de contrôle de chiffrement	-
État	Available
Location	default
VPC par défaut	Non
Métriques d'utilisation d'adresses réseau	Désactivé
Mode de contrôle de chiffrement	-
Bloquer l'accès public	Désactivé
Jeu d'options DHCP	adpt-018cb9f6a0354a43b
CIDR IPv4	10.0.0.0/16
Groupe de règles du pare-feu DNS de Route 53	Resolver
Noms d'hôte DNS	Activé
Table de routage principale	rtb-0f8e8917da515b38b
Groupe IPv6	-
ID du propriétaire	219005156175

La structuration du réseau se poursuit avec la mise en place d'un sous-réseau au sein du VPC précédemment créé. Comme l'indique la figure au-dessous, cette étape est cruciale pour définir la plage d'adressage IP où seront déployées les instances EC2.

- **Nom du sous-réseau :** Le sous-réseau a été nommé my_pub_subnet afin de l'identifier facilement comme la zone publique de l'infrastructure.
- **Sélection du VPC :** Il est rattaché au bloc d'adresse CIDR IPv4 10.0.0.0/16, correspondant au VPC EL_FELLAH_Meryem_Zabbix_Server-vpc.

Cette configuration garantit que le serveur Zabbix et ses agents (Linux et Windows) disposent d'un espace d'adressage cohérent pour communiquer efficacement à l'intérieur du cloud AWS.

The screenshot shows the AWS Management Console interface for creating a new subnet. The top navigation bar includes the AWS logo, a search bar, and account information: ID de compte: 2190-0315-6175 and vclabs/user4543147=EL_FELLAH_Meryem. The main page title is "Sous-réseau 1 sur 1". The configuration fields are as follows:

- Nom du sous-réseau (subnet):** my_pub_subnet
- Zone de disponibilité:** Aucune préférence
- Bloc d'adresse CIDR IPv4 VPC:** 10.0.0.0/16
- Bloc d'adresse CIDR de sous-réseau IPv4:** 10.0.16.0/20 (4096 IPs)

Le Groupe de Sécurité, nommé **EL_FELLAH_Meryem_ZabbixServerSG**, fait office de pare-feu virtuel pour contrôler de manière granulaire les flux entrants vers le serveur et les agents. Comme illustré dans la figure au-dessous, les règles entrantes ont été rigoureusement configurées pour permettre le fonctionnement optimal de la solution Zabbix tout en assurant l'administration des instances :

- **Flux de Supervision (Zabbix) :** * L'ouverture du port **TCP 10051 (Zabbix Trapper)** permet au serveur de recevoir les données envoyées par les agents.
 - L'ouverture du port **TCP 10050 (Zabbix Agent)** autorise le serveur à interroger activement les clients.
- **Accès Interface Web :** Les ports **HTTP (80)** et **HTTPS (443)** ont été ouverts pour permettre la consultation du tableau de bord Zabbix via un navigateur web depuis n'importe quelle source.
- **Administration à distance :** * Le port **SSH (22)** est activé pour la gestion en ligne de commande de l'instance Linux.
 - Le port **RDP (3389)** a été configuré pour permettre l'accès au bureau à distance du client Windows Server.

Toutes ces règles sont appliquées avec une source ouverte (0.0.0.0/0) pour les besoins du Lab, garantissant une connectivité totale entre les différents composants du parc hybride situé dans le VPC.

The screenshot shows the AWS Management Console with the URL [https://console.aws.amazon.com/ec2/v2/home?#Groups%3Asg-05a28a9345b74aa5c-EL_FELLAH_Meryem_ZabbixServerSG:Rules](#). The page title is "Modifier les règles entrantes". The main content area displays a table of security group rules:

ID de règle de groupe de sécurité	Type	Protocole	Plage de ports	Source	Description - facultatif	Action
sgr-0819c980dbd4d2d55	HTTPS	TCP	443	Personn... ▾		Supprimer
sgr-03c30dababd9cb1eb	RDP	TCP	3389	Personn... ▾	0.0.0.0/0	Supprimer
sgr-014d2e3c25a992dac	SSH	TCP	22	Personn... ▾	0.0.0.0/0	Supprimer
sgr-0b05ea661b3abeb51	TCP personnalisé	TCP	10050	Personn... ▾	0.0.0.0/0	Supprimer
sgr-07086126e66d81943	HTTP	TCP	80	Personn... ▾	zabbix agent	Supprimer
sgr-0e51e6808efc838bd	TCP personnalisé	TCP	10051	Personn... ▾	zabbix traper	Supprimer

At the bottom left, there is a "Ajouter une règle" (Add a rule) button. The bottom right corner contains links for "CloudShell", "Commentaires", and other AWS navigation links.

2.2. Inventaire des Instances EC2

L'architecture hybride est composée de trois instances stratégiques, chacune répondant à un rôle précis :

a) Instance 1: Zabbix Server

Le cœur de l'infrastructure de supervision repose sur une instance robuste configurée pour héberger le serveur Zabbix via Docker. Comme illustré dans les étapes de lancement (**Figures au-dessous**), les paramètres suivants ont été appliqués :

- **Identification et Nommage :** L'instance a été nommée **EL_FELLAH_Meryem_ZabbixUbuntu**, respectant ainsi la nomenclature personnalisée requise pour le projet.
- **Système d'Exploitation (AMI) :** Le choix s'est porté sur une image **Ubuntu Server 24.04 LTS (Noble Numbat)** en architecture 64 bits (x86), offrant un environnement moderne et stable pour la conteneurisation.
- **Dimensionnement (Type d'instance) :** Une instance de type **t3.medium** (2 vCPU, 4 Gio de mémoire) a été sélectionnée pour garantir les ressources nécessaires au fonctionnement simultané du serveur Zabbix, de sa base de données et de l'interface Web.
- **Configuration Réseau et Sécurité :** * L'instance est intégrée au VPC **EL_FELLAH_Meryem_Zabbix_Server-vpc**.
 - Elle est déployée dans le sous-réseau public **my_pub_subnet** avec l'attribution automatique d'une adresse IP publique activée pour permettre l'accès à l'interface de monitoring.
 - La sécurité est assurée par le groupe de sécurité **EL_FELLAH_Meryem_ZabbixServerSG**, qui autorise les flux de supervision et l'administration distante.
- **Authentification :** Une paire de clés spécifique, nommée **cle_EL_FELLAH_Meryem_ZabbixUbuntu**, a été générée pour sécuriser les accès SSH à la machine.

Nom et balises [Informations](#)

Nom

EL_FELLAH_Meryem_ZabbixUbuntu

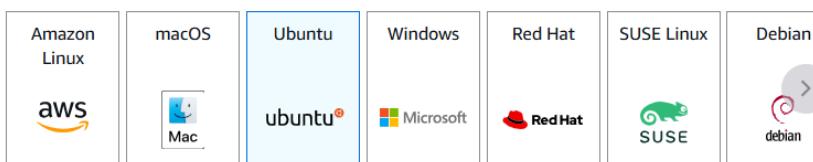
[Ajouter des balises supplémentaires](#)

▼ 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

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

Type d'instance

t3.medium
Famille: t3 2 vCPU 4 Gio Mémoire Génération actuelle: true
À la demande SUSE base tarification: 0.0979 USD par heure
À la demande Windows base tarification: 0.06 USD par heure
À la demande Linux base tarification: 0.0416 USD par heure
À la demande Ubuntu Pro base tarification: 0.0451 USD par heure
À la demande RHEL base tarification: 0.0704 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_EL_FELLAH_Meryem_ZabbixUbuntu

[Créer une paire de clés](#)

▼ Paramètres réseau [Informations](#)

VPC - *obligatoire* | [Informations](#)

vpc-0e7813a227c90c97f (EL_FELLAH_Meryem_Zabbix_Server-vpc)
10.0.0.0/16

☰ EC2 > Instances > Lancer une instance

▼ Paramètres réseau Informations

VPC - **obligatoire** | Informations
vpc-0e7813a227c90c97f (EL_FELLAH_Meryem_Zabbix_Server-vpc)
10.0.0.0/16

Sous-réseau | Informations
subnet-05adec32bfab33ef6
EL_FELLAH_Meryem_Zabbix_Server-subnet-public1-us-east-1a
VPC: vpc-0e7813a227c90c97f Propriétaire: 219003156175
Zone de disponibilité: us-east-1a (use1-az1) Type de zone: Zone de disponibilité
Adresses IP disponibles: 4091 CIDR: 10.0.0.0/20

Créer un nouveau sous-réseau ↗

Attribuer automatiquement l'adresse IP publique | Informations
Activer

Pare-feu (groupes de sécurité) | Informations
Un groupe de sécurité est un ensemble de règles de pare-feu qui contrôlent le trafic de votre instance. Ajoutez des règles pour autoriser un trafic spécifique à atteindre votre instance.

Créer un groupe de sécurité Sélectionner un groupe de sécurité existant

Nom du groupe de sécurité - **obligatoire**
EL_FELLAH_Meryem_ZabbixServerSG

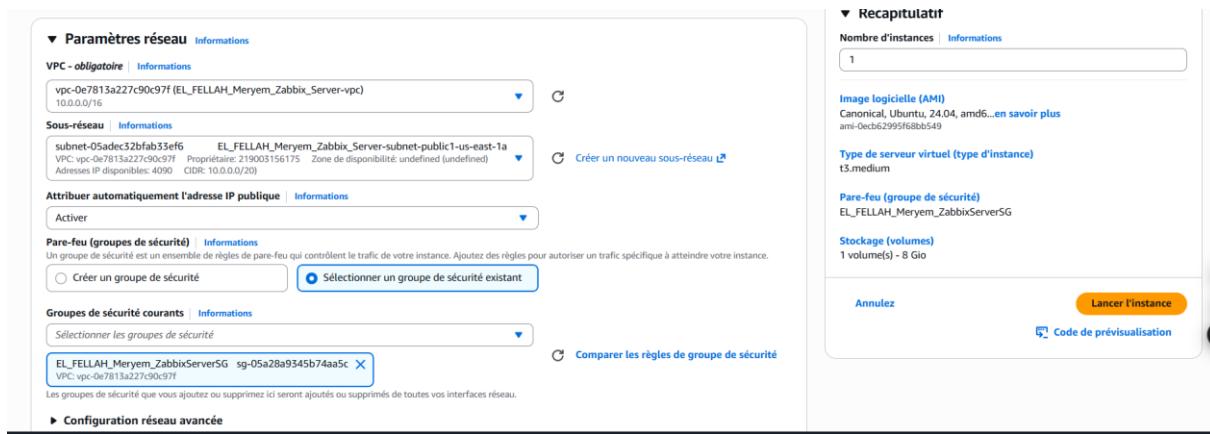
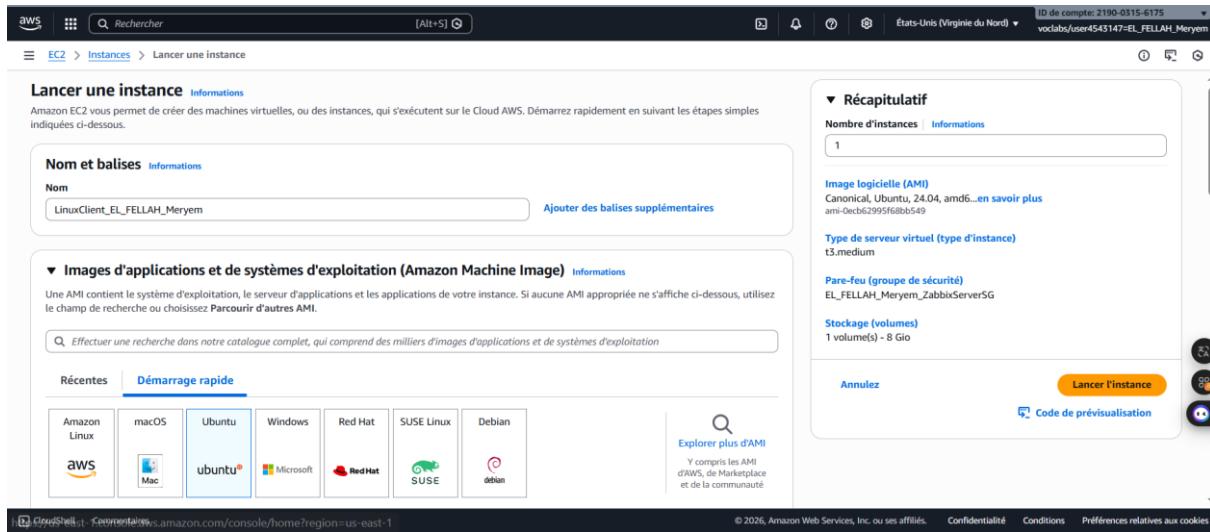
Ce groupe de sécurité sera ajouté à toutes les interfaces réseau. Le nom ne peut pas être modifié après la création du groupe de sécurité. La longueur maximale est de 255 caractères. Caractères valides : a-z, A-Z, 0-9, espaces et _ ; / ! #, @ [] = & ; ! \$ *

Description - **obligatoire** | Informations
launch-wizard-1 created 2026-01-04T18:45:49.472Z

b) Instance 2: Linux Ubuntu Server

Le second pilier de ce parc hybride est constitué par une instance Linux dédiée à la supervision. Bien que partageant une infrastructure réseau commune avec le serveur, elle joue le rôle de nœud supervisé. Comme illustré dans la phase de lancement (**Figures ci-dessous**), sa configuration repose sur les spécifications suivantes :

- **Identification** : L'instance est nommée **LinuxClient_EL_FELLAH_Meryem**, permettant une distinction claire lors de son enregistrement dans l'inventaire Zabbix.
- **Système d'Exploitation (AMI)** : Elle utilise l'image **Ubuntu Server 24.04 LTS**, garantissant une compatibilité totale avec les agents Zabbix de dernière génération (version 7.0.x).
- **Ressources (Type d'instance)** : Aussi **t3.medium**.
- **Intégration Réseau et Sécurité** : * L'instance est rattachée au même VPC et déployée dans le sous-réseau, ainsi que c'est le même Security Group.



c) Instance 3: Windows Server

Le dernier volet du parc hybride est représenté par une instance Windows Server, permettant de valider les capacités de supervision multi-plateforme de la solution Zabbix. Comme le démontrent les étapes de configuration (**Figures ci-dessous**), les paramètres suivants ont été appliqués :

- Identification et Nommage :** L'instance a été nommée **Windows_Server_Client_EL_FELLAH_Meryem** afin d'assurer une traçabilité parfaite au sein de la console AWS et de l'interface Zabbix.
- Système d'Exploitation (AMI) :** Pour ce client, l'image **Microsoft Windows Server 2025 Base** (version 2025.12.10) a été sélectionnée, offrant ainsi un environnement de test basé sur la dernière génération des systèmes serveurs de Microsoft.
- Dimensionnement (Type d'instance) :** Encore une fois c'est **t3.medium**.
- Intégration Réseau et Sécurité :** * L'instance est rattachée au même VPC et déployée dans le sous-réseau, ainsi que c'est le même Security Group.

EC2 > Instances > Lancer une instance

Nom: Windows_Server_Client_EL_FELLAH_Meryem 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

Récentes Démarrage rapide

Amazon Linux	macOS	Ubuntu	Windows	Red Hat	SUSE Linux	Debian

Explorer plus d'AMI Y compris les AMI d'AWS, de Marketplace et de la communauté

Amazon Machine Image (AMI)

Microsoft Windows Server 2025 Base
ami-06777e7e7f441def (64 bits (x86))
Virtualisation: hvm ENA activé: true Type de périphérique racine: ebs

Eligible à l'offre gratuite

Image logicielle (AMI)
Microsoft Windows Server 2025 ...en savoir plus
ami-06777e7e7f441def

Type de serveur virtuel (type d'instance)
t3.micro

Par-feu (groupe de sécurité)
Nouveau groupe de sécurité

Stockage (volumes)
1 volume(s) - 30 Gio

Annulez Lancer l'instance

Code de prévisualisation

EC2 > Instances > Lancer une instance

▼ Type d'instance Informations | Obtenez des conseils

Type d'instance

t3.medium
Famille: t3 2 vCPU 4 Go Mémoire Génération actuelle: true
À la demande SUSE base tarification: 0.0979 USD par heure À la demande Windows base tarification: 0.06 USD par heure
À la demande Linux base tarification: 0.0416 USD par heure
À la demande Ubuntu Pro base tarification: 0.0451 USD par heure À la demande RHEL base tarification: 0.0704 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

Sélectionnez

Créer une paire de clés

The screenshot shows the AWS EC2 'Launch Instance' wizard at step 3: 'Paramètres réseau'. It includes sections for VPC (selected VPC: vpc-0e7813a227c90c97f), Subnet (selected subnet: subnet-05adec32bfab33ef6), and Security Groups (selected group: EL_FELLAH_Meryem_ZabbixServerSG). The 'Attribuer automatiquement l'adresse IP publique' section is set to 'Activer'. The 'Pare-feu (groupes de sécurité)' section shows a choice between 'Créer un groupe de sécurité' (radio button) and 'Sélectionner un groupe de sécurité existant' (button). The 'Groupes de sécurité courants' section lists the selected security group. A sidebar on the right shows navigation steps: N, I, M, ai, T, t:, P, E, S, 1.

3. Installation du Docker et Deploiement du Zabbix Ubuntu Server

Cette étape consiste à transformer l'instance Ubuntu en une plateforme de supervision via la **conteneurisation**. L'installation de **Docker** et **Docker-Compose** permet de déployer l'intégralité de la pile Zabbix (Serveur, Base de données et Interface Web) de manière isolée, rapide et standardisée. Cette approche garantit une gestion agile des services et une portabilité optimale au sein de l'infrastructure AWS.

3.1. Connexion SSH:

La première étape consiste à établir une connexion sécurisée avec l'instance Ubuntu via le protocole SSH.

- **Description:** Cette capture d'écran illustre l'accès réussi à l'interface en ligne de commande de l'instance EL_FELLAH_Meryem_ZabbixUbuntu en utilisant la clé ssh présente sur AWS.
- **Détails techniques :** On y confirme l'utilisation d'**Ubuntu 24.04.3 LTS** et l'adresse IP privée interne **10.0.3.253**, ce qui valide la cohérence avec le plan d'adressage du VPC.
- **Analyse :** Le système est prêt pour l'administration, avec une charge système nulle et une température de processeur stable, offrant un environnement sain pour l'installation des paquets.

```
ubuntu@ip-10-0-3-253: ~
Warning: Permanently added 'ec2-44-211-120-215.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 Mon Jan  5 00:53:16 UTC 2026

System load: 0.0           Temperature:      -273.1 C
Usage of /: 27.6% of 6.71GB Processes:        110
Memory usage: 6%
Swap usage:  0%           Users logged in:   0
                           IPv4 address for ens5: 10.0.3.253

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-10-0-3-253:~$
```

3.2. Installer Docker:

On utilise le Docker pour lancer Zabbix sans installation manuelle compliquée.

```
ubuntu@ip-10-0-3-253: ~
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-10-0-3-253:~$ docker --version
Docker version 28.2.2, build 28.2.2-0ubuntu1~24.04.1
ubuntu@ip-10-0-3-253:~$ docker-compose --version
docker-compose version 1.29.2, build unknown
ubuntu@ip-10-0-3-253:~$
```

Comme l'image le montre, j'ai docker et docker-compose déjà installé et mis à jour en exécutant les commandes suivantes:

```
sudo apt update
sudo apt install -y docker.io docker-compose
sudo systemctl start docker
sudo systemctl enable docker
```

3.2. Deploiement du Zabbix:

Une fois Docker opérationnel, l'étape suivante consiste à structurer les fichiers nécessaires au déploiement multi-conteneurs de Zabbix.

Passons à la création du dossier zabix et fichier docker-compose.yml, en utilisant les commandes suivantes:

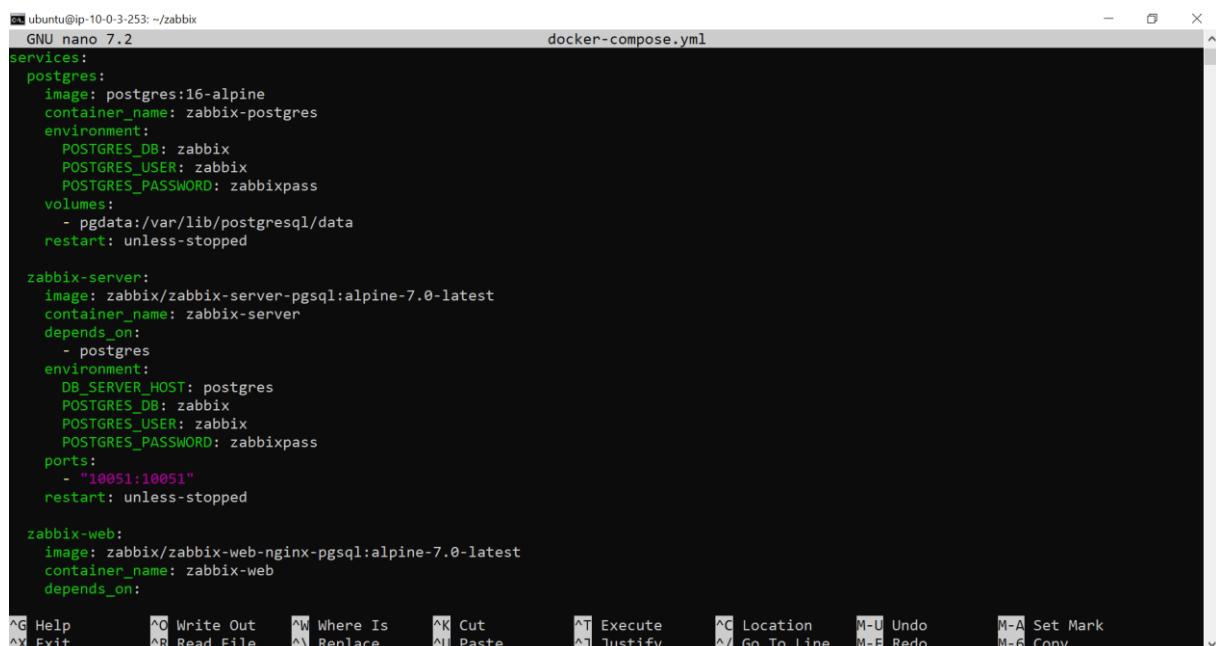
```
mkdir zabix  
cd Zabbix  
nano docker-compose.yml
```

Cette capture d'écran ci-dessous confirme la création et l'organisation du répertoire de travail sur l'instance serveur.

```
ubuntu@ip-10-0-3-253:~/zabbix/  
ubuntu@ip-10-0-3-253:~/zabbix$ ls  
docker-compose.yml  docker-compose_v3_alpine_pgsql_latest.yaml  
ubuntu@ip-10-0-3-253:~/zabbix$ nano docker-compose.yml  
ubuntu@ip-10-0-3-253:~/zabbix$
```

Un dossier dédié nommé `zabbix/` a été créé pour centraliser les fichiers de configuration.

- À l'intérieur de ce répertoire, on retrouve le fichier principal `docker-compose.yml`, édité via l'utilitaire `nano` pour définir l'orchestration des services.
- La présence du fichier `docker-compose_v3_alpine_pgsql_latest.yaml` indique l'utilisation d'une architecture moderne basée sur **Alpine Linux** et **PostgreSQL** pour optimiser la légèreté et les performances de la base de données.



```
ubuntu@ip-10-0-3-253:~/zabbix  
GNU nano 7.2                               docker-compose.yml  
services:  
  postgres:  
    image: postgres:16-alpine  
    container_name: zabbix-postgres  
    environment:  
      POSTGRES_DB: zabbix  
      POSTGRES_USER: zabbix  
      POSTGRES_PASSWORD: zabbixpass  
    volumes:  
      - pgdata:/var/lib/postgresql/data  
    restart: unless-stopped  
  
  zabbix-server:  
    image: zabbix/zabbix-server-pgsql:alpine-7.0-latest  
    container_name: zabbix-server  
    depends_on:  
      - postgres  
    environment:  
      DB_SERVER_HOST: postgres  
      POSTGRES_DB: zabbix  
      POSTGRES_USER: zabbix  
      POSTGRES_PASSWORD: zabbixpass  
    ports:  
      - "10051:10051"  
    restart: unless-stopped  
  
  zabbix-web:  
    image: zabbix/zabbix-web-nginx-pgsql:alpine-7.0-latest  
    container_name: zabbix-web  
    depends_on:
```

Le fichier docker-compose.yml est la pièce maîtresse du déploiement. Il permet d'automatiser le lancement et l'interconnexion des services nécessaires à la plateforme Zabbix. Comme le montre la **Figure ci-dessus**, la configuration s'articule autour de trois conteneurs interdépendants :

- **Service postgres** : Ce conteneur gère la base de données. Il utilise l'image **postgres:16-alpine**, une version légère basée sur Alpine Linux.
- **Service zabbix-server** : Il s'agit du cœur applicatif. Il utilise l'image **zabbix-server-pgsql:alpine-7.0-latest**. Ce service dépend directement du conteneur de base de données (depends_on: postgres) et expose le port **10051**, essentiel pour la réception des flux de monitoring.
- **Service zabbix-web** : Ce conteneur fournit l'interface graphique utilisateur. Il repose sur l'image **zabbix-web-nginx-pgsql:alpine-7.0-latest**, combinant un serveur web Nginx et le support PostgreSQL.

L'édition de ce fichier a été réalisée avec l'éditeur nano. Une fois la configuration finalisée, la sortie de l'éditeur s'effectue via le raccourci **Ctrl+X**, suivie de la commande docker-compose up -d pour lancer l'infrastructure en arrière-plan.

```
ubuntu@ip-10-0-3-253:~/zabbix$ nano docker-compose.yml
ubuntu@ip-10-0-3-253:~/zabbix$ docker-compose up -d
zabbix-postgres is up-to-date
zabbix-server is up-to-date
zabbix-web is up-to-date
ubuntu@ip-10-0-3-253:~/zabbix$
```

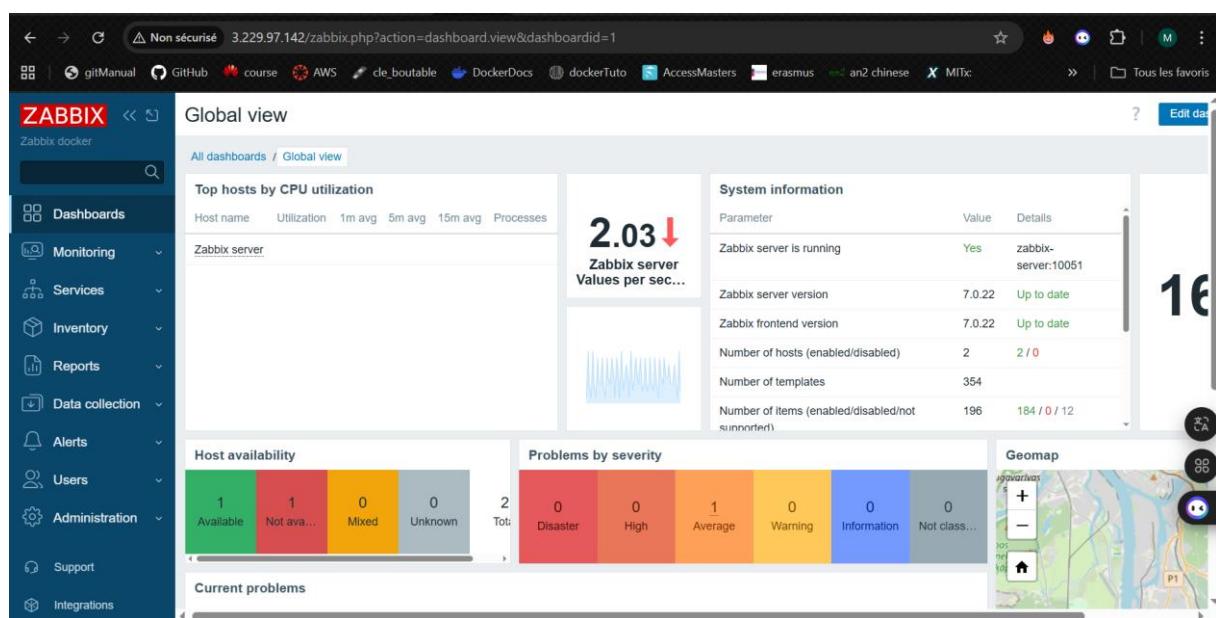
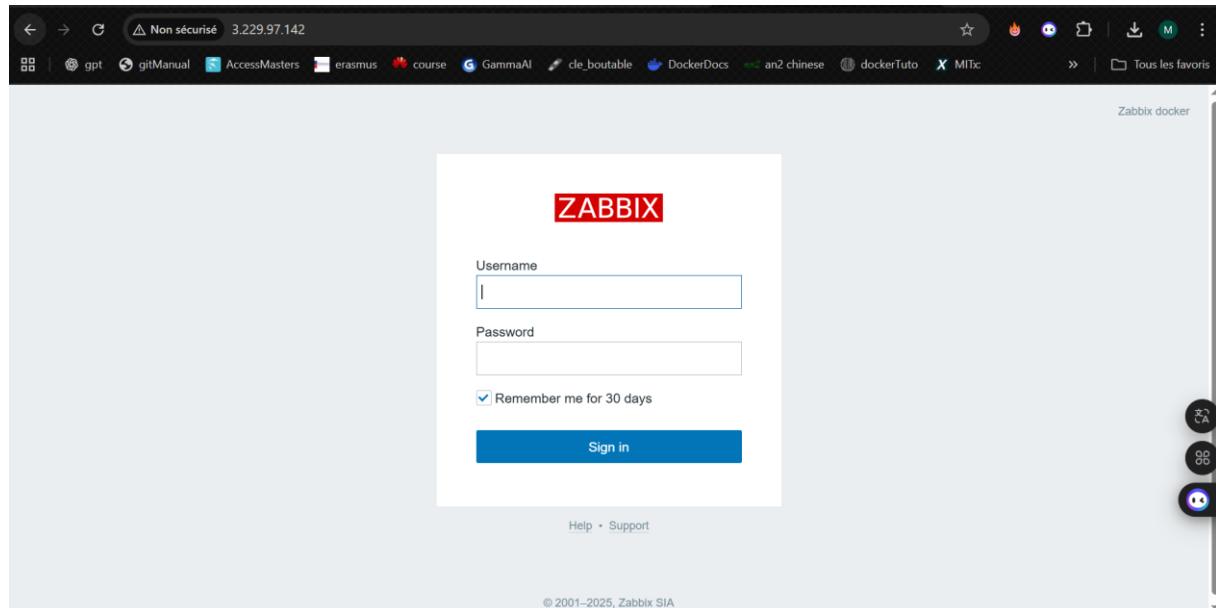
J'utilise le `docker ps` pour s'assurer que le Zabbix server est bien lancé.

```
ubuntu@ip-10-0-3-253:~/zabbix
ubuntu@ip-10-0-3-253:~/zabbix$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
 NAMES
3dda1283a88c        zabbix/zabbix-web-nginx-pgsql:alpine-7.0-latest   "docker-entrypoint.sh"   23 seconds ago    Up 22 seconds (healthy)   8443/tcp, 0.0.0.0:80->8080/tcp, [::]:80
->8080/tcp          zabbix-web
db455f5a1276        zabbix/zabbix-server-pgsql:alpine-7.0-latest   "/usr/bin/docker-ent..."  23 seconds ago    Up 23 seconds          0.0.0.0:10051->10051/tcp, [::]:10051->1
0051/tcp           zabbix-server
fcd2dce42cc4        postgres:16-alpine
                     zabbix-postgres
                     "docker-entrypoint.s..."  25 seconds ago    Up 23 seconds          5432/tcp
ubuntu@ip-10-0-3-253:~/zabbix$
```

3.3. Initialisation de l'interface web Zabbix:

Une fois les conteneurs déployés avec succès, l'étape finale de la mise en place du serveur consiste à configurer l'accès à la console d'administration.

- **Connexion au portail** : L'accès s'effectue via un navigateur Web en utilisant l'adresse IP publique de l'instance serveur. Dans ce projet, la connexion a été établie à l'URL : <http://3.229.97.142/>.
- **Authentification (Figure issue de l'image_959746.png)** : La page de connexion officielle de Zabbix confirme que le conteneur zabbix-web est correctement exposé sur Internet. L'accès initial a été réalisé avec les identifiants administrateur par défaut :
 - **Username** : Admin
 - **Password** : zabbix



4. Installation des agents Zabbix

Revenons vers notre ligne de commande pour continuer avec l'installation des 2 agents Zabbix, qui seront 2 machines surveillées. On a le client Linux Ubuntu Server et le client Windows Server.

Pour se déconnecter de la machine "EL_FELLAH_Meryem_ZabbixUbuntu", je fais exit.

4.1. La machine cliente Linux:

- ✓ Comme d'habitude je me connecte sur la machine "LinuxClient_EL_FELLAH_Meryem" en utilisant la clé ssh fournie sur AWS platform.
- ✓ J'installe l'agent zabbix pour le client Linux Server `sudo apt install zabbix-agent`, c'est ce que j'ai déjà fait, donc juste pour montrer qu'il existe déjà et tourne bien, voici la figure suivante.

```
Last login: Mon Jan  5 22:59:30 2026 from 105.68.138.171
ubuntu@ip-10-0-3-160:~$ zabbix_agentd -V
zabbix_agentd (daemon) (Zabbix) 7.0.22
Revision 70c23564978 16 December 2025, compilation time: Dec 16 2025 19:12:34

Copyright (C) 2025 Zabbix SIA
License AGPLv3: GNU Affero General Public License version 3 <https://www.gnu.org/licenses/>.
This is free software: you are free to change and redistribute it according to
the license. There is NO WARRANTY, to the extent permitted by law.

This product includes software developed by the OpenSSL Project
for use in the OpenSSL Toolkit (http://www.openssl.org/).

Compiled with OpenSSL 3.0.13 30 Jan 2024
Running with OpenSSL 3.0.13 30 Jan 2024
ubuntu@ip-10-0-3-160:~$ sudo systemctl status zabbix-agent
Warning: The unit file, source configuration file or drop-ins of zabbix-agent.service changed.
● zabbix-agent.service - Zabbix Agent
    Loaded: loaded (/usr/lib/systemd/system/zabbix-agent.service; enabled; preset: enabled)
      Active: active (running) since Tue 2026-01-06 12:46:23 UTC; 3h 19min ago
        Main PID: 614 (zabbix_agentd)
          Tasks: 13 (limit: 4525)
         Memory: 13.8M (peak: 17.3M)
            CPU: 13.415s
       CGroup: /system.slice/zabbix-agent.service
               └─614 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf
                  ├─622 "/usr/sbin/zabbix_agentd: collector [idle 1 sec]"
                  ├─623 "/usr/sbin/zabbix_agentd: listener #1 [waiting for connection]"
                  ├─629 "/usr/sbin/zabbix_agentd: listener #2 [waiting for connection]"
                  ├─630 "/usr/sbin/zabbix_agentd: listener #3 [waiting for connection]"
                  ├─635 "/usr/sbin/zabbix_agentd: listener #4 [waiting for connection]"
                  ├─641 "/usr/sbin/zabbix_agentd: listener #5 [waiting for connection]"
                  ├─642 "/usr/sbin/zabbix_agentd: listener #6 [waiting for connection]"
                  ├─658 "/usr/sbin/zabbix_agentd: listener #7 [waiting for connection]"
```

- ✓ Je rentre maintenant à l'intérieur du fichier de configuration du ZabbixAgent pour modifier les 3 lignes:
 - Server= IP_PRIVÉE_ZABBIX (10.0.3.253)
 - ServerActive= IP_PRIVÉE_ZABBIX
 - Hostname= LinuxClient_EL_FELLAH_Meryem

```

ubuntu@ip-10-0-3-160: ~                               /etc/zabbix/zabbix_agentd.conf
GNU nano 7.2

##### Passive checks related

### Option: Server
#      List of comma delimited IP addresses, optionally in CIDR notation, or DNS names of Zabbix servers and Zabbix proxies.
#      Incoming connections will be accepted only from the hosts listed here.
#      If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' are treated equally
#      and '::/0' will allow any IPv4 or IPv6 address.
#      '0.0.0.0/0' can be used to allow any IPv4 address.
#      Example: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
#
# Mandatory: yes, if StartAgents is not explicitly set to 0
# Default:
# Server=

Server=10.0.3.253

### Option: ListenPort
#      Agent will listen on this port for connections from the server.
#
# Mandatory: no
# Range: 1024-32767
# Default:
# ListenPort=10050

### Option: ListenIP
#      List of comma delimited IP addresses that the agent should listen on.
#      First IP address is sent to Zabbix server if connecting to it to retrieve list of active checks.
#
# Mandatory: no
# Default:

^G Help      ^O Write Out    ^W Where Is     ^K Cut          ^T Execute      ^C Location     M-U Undo      M-A Set Mark
^X Exit      ^R Read File   ^Y Replace      ^U Paste        ^J Justify     ^L Go To Line   M-E Redo      M-B Copy

```

```

#
# Mandatory: no
# Default:
# ServerActive=

ServerActive=10.0.3.253

### 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=LinuxClient_EL_FELLAH_Meryem

### 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

### Option: HostMetadata
#      Optional parameter that defines host metadata.
#      Host metadata is used at host auto-registration process.
#      An agent will issue an error and not start if the value is over limit of 2034 bytes.

^G Help      ^O Write Out    ^W Where Is     ^K Cut          ^T Execute      ^C Location     M-U Undo      M-A Set Mark
^X Exit      ^R Read File   ^Y Replace      ^U Paste        ^J Justify     ^L Go To Line   M-E Redo      M-B Copy

```

- ✓ Revenons vers l'interface Zabbix dans laquelle j'ai déjà accédé, parce que maintenant j'aurai besoin de créer un Host pour le client LinuxServer en utilisant le même nom "LinuxClient_EL_FELLAH_Meryem".

Dans Configuration → Hosts → Create host :

- Host name = exactement le même que dans zabbix_agentd.conf
- Type = Agent
- IP = IP privée du client Linux (10.0.3.160)
- Port = 10050
- Template = Linux by Zabbix agent
- Host group = Linux servers

Après ça, si l'agent tourne, le statut passe en ZBX vert.

Non sécurisé 3.229.97.142/zabbix.php?action=host.edit

ZABBIX

New host

Host IPMI Tags Macros Inventory Encryption Value mapping

* Host name: LinuxClient_EL_FELLAH_Meryem

Visible name: LinuxClient_EL_FELLAH_Meryem

Templates: Linux by Zabbix agent

* Host groups: Linux servers

Interfaces:

Type	IP address	DNS name	Connect to	Port	Default
Agent	10.0.3.160		IP	10050	<input checked="" type="radio"/> Remove

Add Description:

Add Cancel Displaying 2 of 2 found

ZABBIX

Hosts

Name:

Status: Any Enabled Disabled

Host groups: Select

Tags: And/Or Or

IP:

DNS:

Port:

Show hosts in maintenance:

Show suppressed problems:

Severity: Not classified Warning High
Information Average Disaster

Save as Apply Reset

Name	Interface	Availability	Tags	Status	Latest data	Problems	Graphs
LinuxClient_EL_FELLAH_Meryem	10.0.3.160:10050	ZBX	class: os target: linux	Enabled	Latest data 75	Problems 1	Graphs 16
Zabbix server	127.0.0.1:10050	ZBX	class: os class: software target: linux ***	Enabled	Latest data 121	1	Graphs 8

4.2. La machine cliente Windows:

- ✓ Comme d'habitude je me connecte sur la machine "LinuxClient_EL_FELLAH_Meryem" mais cette fois-ci en utilisant le client RDP et non pas le client SSH.

ID d'instance
i-034bc506f8dff5c44 (Windows_Server_Client_EL_FELLAH_Meryem)

Type de connexion

Connexion à l'aide du client RDP
Téléchargez un fichier à utiliser avec votre client RDP et récupérez votre mot de passe.

Connexion à l'aide du gestionnaire de parc
Pour vous connecter à l'instance à l'aide du Bureau à distance du gestionnaire de parc, l'agent SSM doit être installé et en cours d'exécution sur l'instance. Pour plus d'informations, consultez Utilisation de l'agent SSM.

Vous pouvez également vous connecter à votre instance Windows en utilisant un client Bureau à distance de votre choix et en téléchargeant et en exécutant le fichier de raccourci RDP ci-dessous :

[Télécharger le fichier bureau à distance](#)

À l'étape correspondante, connectez-vous à votre instance à l'aide du nom d'utilisateur et du mot de passe suivants :

Public DNS
ec2-44-201-7-8.compute-1.amazonaws.com

Nom d'utilisateur Informations
 Administrator

Mot de passe [Obtenir le mot de passe](#)

Si vous avez joint votre instance à un répertoire, vous pouvez utiliser vos informations d'identification pour vous connecter à votre instance.

- ✓ Je clique sur le bouton "Télécharger le fichier bureau a distance".
- ✓ Je clique sur obtenir le mot de passe, puis je fournis la clé de cette instance que j'avais Déjà dans le dossier Downloads, ce qui me donne un mot de passe chiffré que je dois le déchiffrer pour l'utiliser après.
- ✓ Par la suite je clique sur ce fichier quand il est téléchargé.

Name	Date modified	Type	Size
Windows_Server_Client_EL_FELLAH_Meryem	1/6/2026 1:49 AM	Remote Desktop Con...	1 KB
cle_Windows_Server_Client_EL_FELLAH_Meryem...	1/6/2026 1:35 AM	PEM File	2 KB
OBS-Studio-32.0.4-Windows-x64-Installer	1/5/2026 9:13 PM	Application	153,813 KB
Examen_Cloud_Fellah_Youssef	1/5/2026 7:43 PM	Chrome HTML Docu...	1,664 KB
cle_LinuxClient_EL_FELLAH_Meryem.pem	1/5/2026 1:41 AM	PEM File	2 KB
cle_EL_FELLAH_Meryem_ZabbixUbuntu.pem	1/4/2026 7:47 PM	PEM File	2 KB

- ✓ Apres cliquer, la boite de texte grise suivante s'affiche, dans laquelle on colle le mot de passe qu'on a deja dechiffree et copiee.

EC2 > Instances > i-034bc506f8dff5c44 > Connectez-vous à l'instance

Enregistrer les connexions RDP
Vous pouvez désormais enregistrer les connexions RDP à l'aide de l'accès aux instances.

ID d'instance
i-034bc506f8dff5c44 (Windows_Server_Client_EL_FELLAH_Meryem)

Type de connexion
Connexion à l'aide du client RDP
Téléchargez un fichier à utiliser avec votre client RDP et récupérez votre mot de passe.

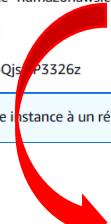
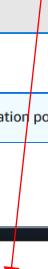
Vous pouvez également vous connecter à votre instance Windows en utilisant un client RDP.

[Télécharger le fichier bureau à distance](#)

À l'étape correspondante, connectez-vous à votre instance à l'aide du nom d'utilisateur et du mot de passe suivants :

Public DNS
ec2-44-201-7-8.compute-1.amazonaws.com
 Mot de passe copié
Mot de passe...
R81uU0iGib0EIN!%39bQjs...P3326z

Si vous avez joint votre instance à un répertoire, vous pouvez utiliser vos informations d'identification pour vous connecter à votre instance.

Windows Security

Enter your credentials

These credentials will be used to connect to ec2-44-201-7-8.compute-1.amazonaws.com.

Administrator

Password

Remember me

[More choices](#)

OK Cancel

[Allgemeine] Remote Desktop Connection

The identity of the remote computer cannot be verified. Do you want to connect anyway?

The remote computer could not be authenticated due to problems with its security certificate. It may be unsafe to proceed.

Certificate name
Name in the certificate from the remote computer:
EC2AMAZ-FP2BIGM

Certificate errors
The following errors were encountered while validating the remote computer's certificate:
⚠ The certificate is not from a trusted certifying authority.

Do you want to connect despite these certificate errors?

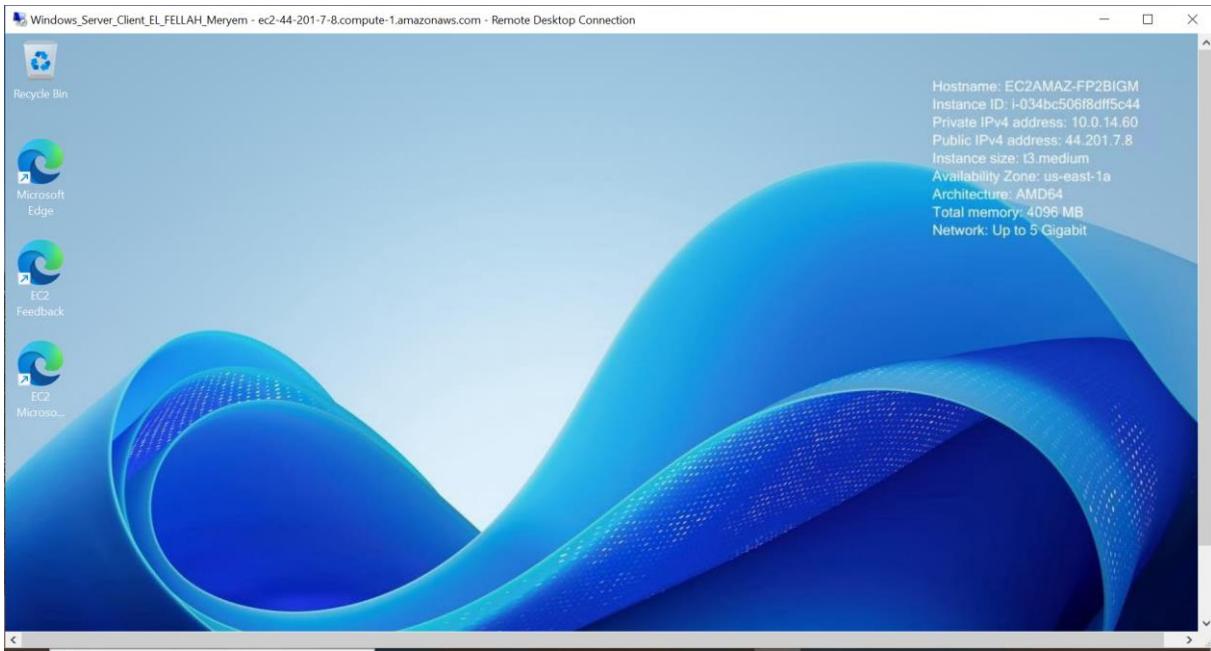
Don't ask me again for connections to this computer

[View certificate...](#) Yes No

Le parc
Bureau à distance du gestionnaire de parc
Informations, consultez Utilisation de l'agent

Le raccourci RDP ci-dessous :

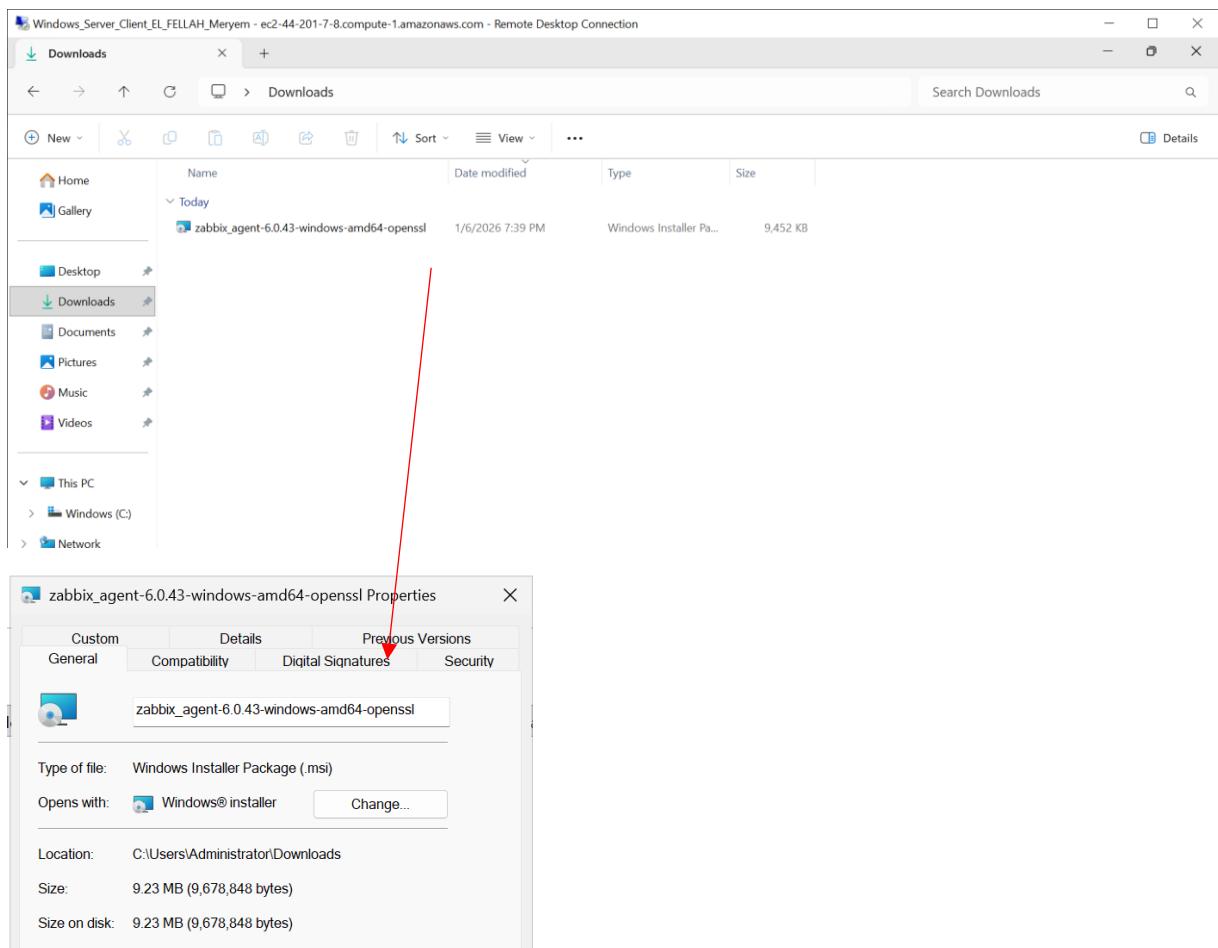
✓ Finalement, session Windows Server est ouverte:



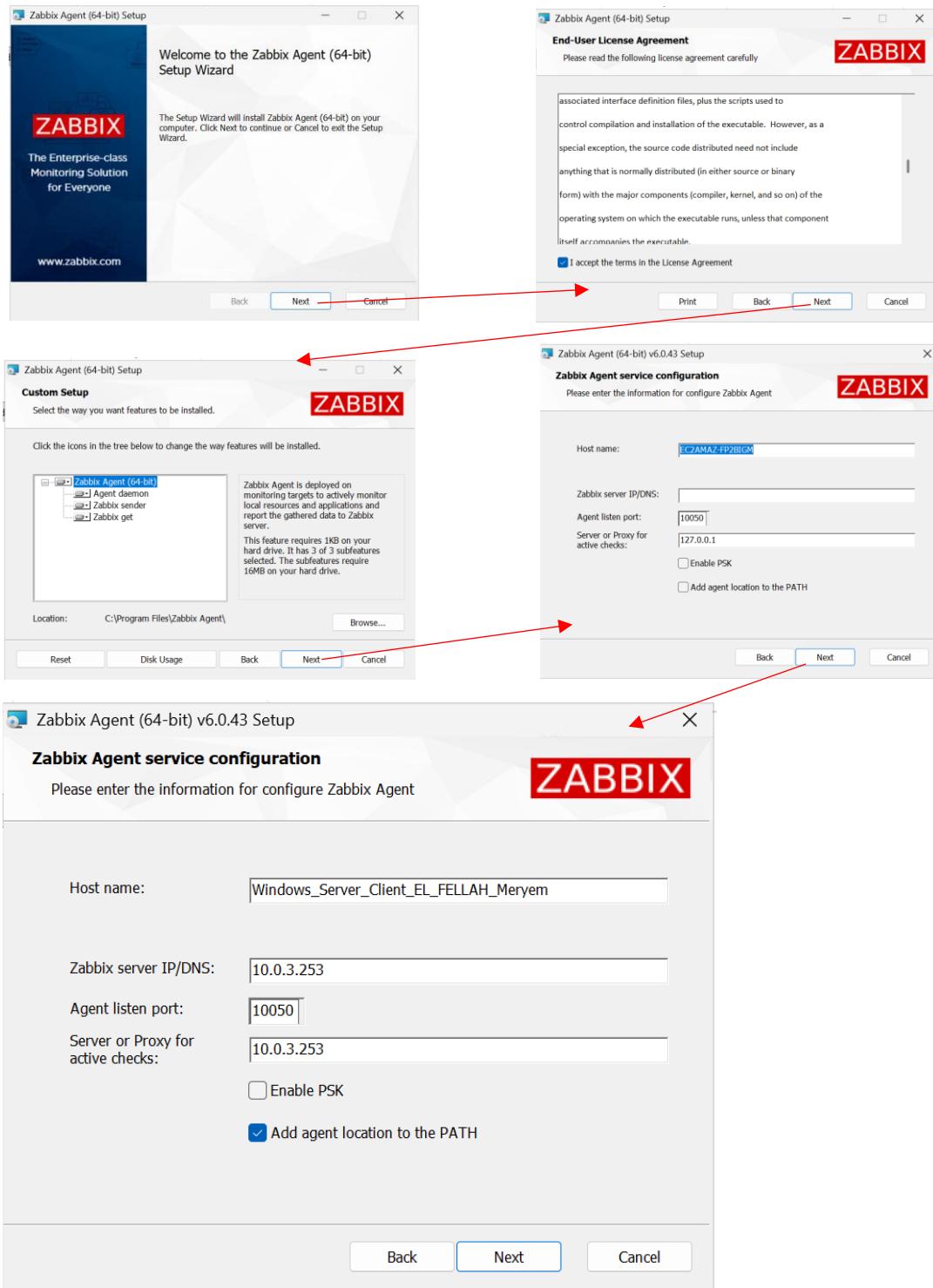
- ✓ Pour télécharger l'agent Zabbix, j'ouvre le navigateur Microsoft Edge. Je vais sur le site officiel Zabbix, exactement ici https://www.zabbix.com/download_agents. Je Choisis le bon agent et je telecharge le fichier .msi.

A screenshot of a Microsoft Edge browser window showing the Zabbix website at https://www.zabbix.com/download_agents. The page title is "Download pre-compiled Zabbix agent binaries". It features a "ZABBIX 20 YEARS" banner. Below the banner, there's a table for selecting OS distribution, version, hardware, and encryption. The table shows options for Windows (selected), Linux, macOS, AIX, FreeBSD, OpenBSD, and Solaris. The selected row for Windows shows OS Version 11, 10, Hardware amd64, Zabbix Version 7.4, Encryption OpenSSL, and Packaging MSI. Other rows show Linux (Server 2016+), macOS (Server 2003+), AIX (XP (64bit)+), FreeBSD, OpenBSD, and Solaris. At the bottom of the table, there's a "GET ZABBIX" button.

A screenshot of a Zabbix agent download page for Windows. The title is "Zabbix agent v6.0.43". It includes a "Read manual" link. Below the title, it lists packaging (MSI), encryption (OpenSSL), and linkage (Dynamic). It also shows checksums: sha256, sha1, and md5. At the bottom, there's a "DOWNLOAD" button with the URL https://cdn.zabbix.com/zabbix/binaries/stable/6.0/6.0.43/zabbix_agent-6.0.43-windows-amd64-openssl.msi.



- ✓ Dans la continuite, pour installer l'agent Zabbix, je double-clique sur le fichier .msi, je continue a cliquer sur Next .

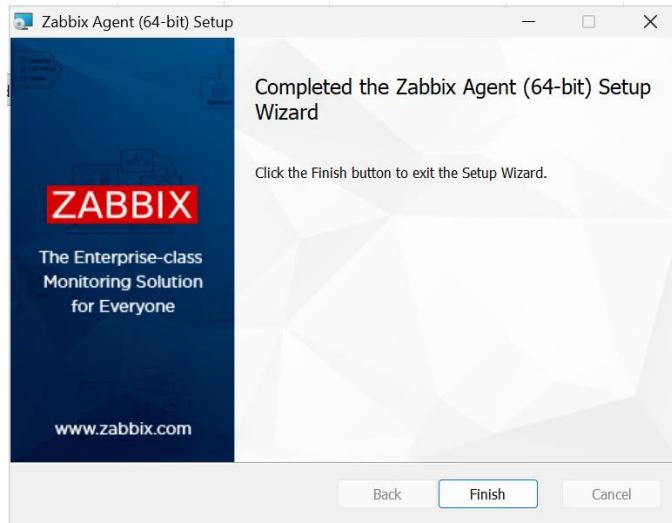


- ✓ Cette écran est très importante à faire attention, parce qu'elle contient les configurations les plus importantes:
- HostName = je dois mettre le même nom que j'utilisera lors de la création du host WindowsServer dans l'interface web Zabbix.
 - Zabbix server IP/DNS = IP_PRIVEE du “EL_FELLAH_Meryem_ZabbixUbuntu”
 - Server or Proxy for active checks = IP_PRIVEE du “EL_FELLAH_Meryem_ZabbixUbuntu”. Parce que le serveur Zabbix est sur une autre machine, les active checks doivent parler au serveur Zabbix.

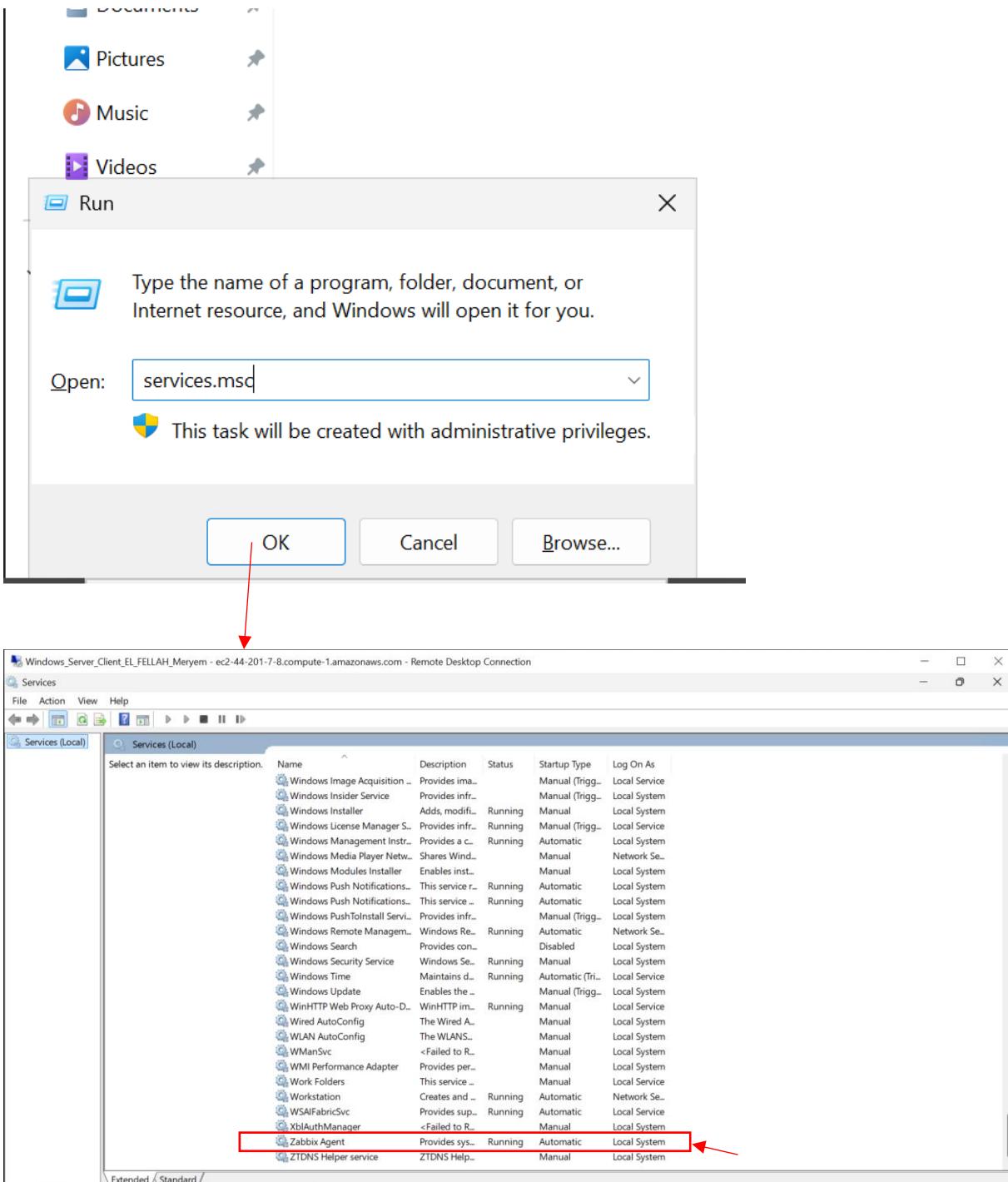
- Agent listen port = 10050. Parce que c'est le port par défaut Zabbix Agent autorisé dans mon Security Group

N.B: Lors de l'installation de l'agent Zabbix sur Windows Server, les paramètres Server et ServerActive ont été configurés avec l'adresse IP privée du serveur Zabbix afin d'assurer la communication interne au sein du VPC.”

- ✓ En cliquant sur Next et Install, je me mène vers la page suivante:



- ✓ Maintenant, pour vérifier le service je clique Win + R et je tape services.msc, puis je cherche Zabbix Agent.



✓ Et voilà, la figure au-dessus montre que Service Zabbix Agent Windows est actif.

- ✓ Revenons vers l'interface Zabbix, parce que maintenant j'aurai besoin de créer un Host pour le client WindowsServer en utilisant le même nom "Windows_Server_Client_EL_FELLAH_Meryem".

Dans Configuration → Hosts → Create host :

- Host name = Windows_Server_Client_EL_FELLAH_Meryem
- Type = Agent
- IP = IP privée du client Windows (10.0.14.60)
- Port = 10050
- Template = Windows by Zabbix agent
- Host groups = Windows servers

The screenshot shows the 'New host' creation dialog in the Zabbix interface. The 'Host' tab is selected. The 'Visible name' field contains 'Windows_Server_Client_EL_FELLAH_Meryem'. The 'Templates' field shows 'Windows by Zabbix agent'. The 'Host groups' field shows 'Windows servers (new)'. In the 'Interfaces' section, there is one entry: 'Agent' with IP '10.0.14.60', port '10050', and 'Default' selected. Below the interfaces, there is a 'Description' field and 'Add' and 'Cancel' buttons.

- ✓ Après ça, si l'agent tourne, le statut passe en ZBX vert.

The screenshot shows the 'Hosts' list in the Zabbix interface. It lists three hosts: 'LinuxClient_EL_FELLAH_Meryem', 'Windows_Server_Client_EL_FELLAH_Meryem', and 'Zabbix server'. The 'Windows_Server_Client_EL_FELLAH_Meryem' host has a red box drawn around its status icon, which is currently red (indicating an error or warning). The other two hosts have green status icons. The table columns include Name, Interface, Availability, Tags, Status, Latest data, Problems, and Graphs.

Name	Interface	Availability	Tags	Status	Latest data	Problems	Graphs
LinuxClient_EL_FELLAH_Meryem	10.0.3.160:10050	ZBX	class: os target: linux	Enabled	Latest data 75	Problems 0	Graphs 1
Windows_Server_Client_EL_FELLAH_Meryem	10.0.14.60:10050	ZBX	class: os target: windows	Enabled	Latest data 105	Problems 2	Graphs 10
Zabbix server	127.0.0.1:10050	ZBX	class: os class: software target: linux ...	Enabled	Latest data 121	Problems 1	Graphs 8

5. Monitoring et Dashboards

Après l'ajout des hôtes Linux et Windows, Zabbix permet de visualiser en temps réel les métriques système telles que l'utilisation CPU, la mémoire et l'espace disque. Les graphiques confirment la bonne communication entre les agents et le serveur Zabbix.

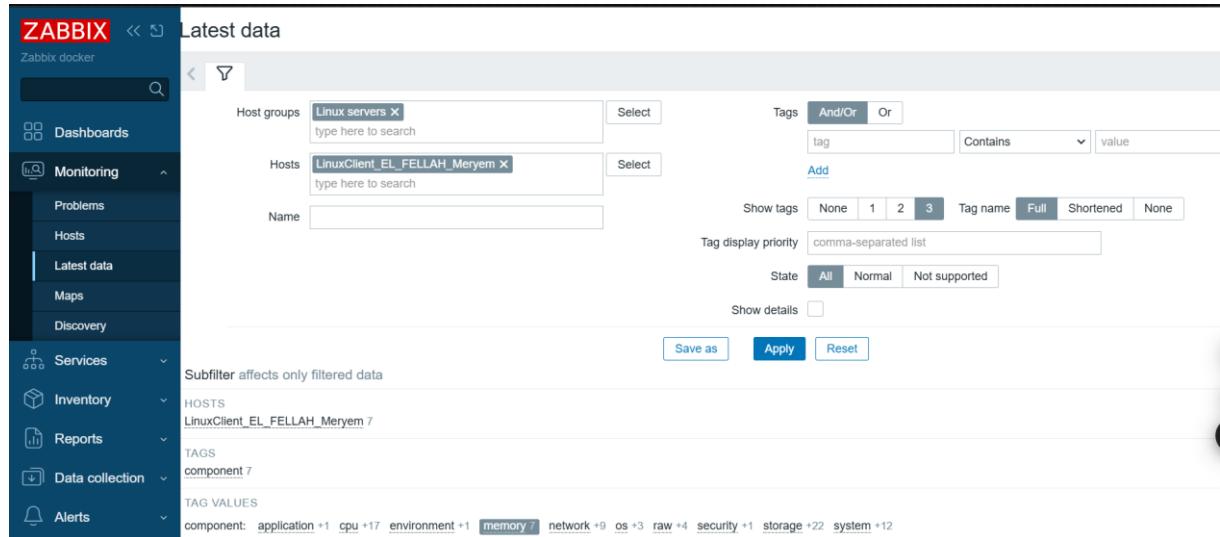
Objectifs de cette étape

À la fin de cette étape, je dois pouvoir :

1. Voir les données en temps réel (CPU, RAM, Disk)
2. Montrer que Zabbix reçoit bien les métriques

9.1) Vérifier la réception des données (Latest data)

a) Filtrer par Linux Client:

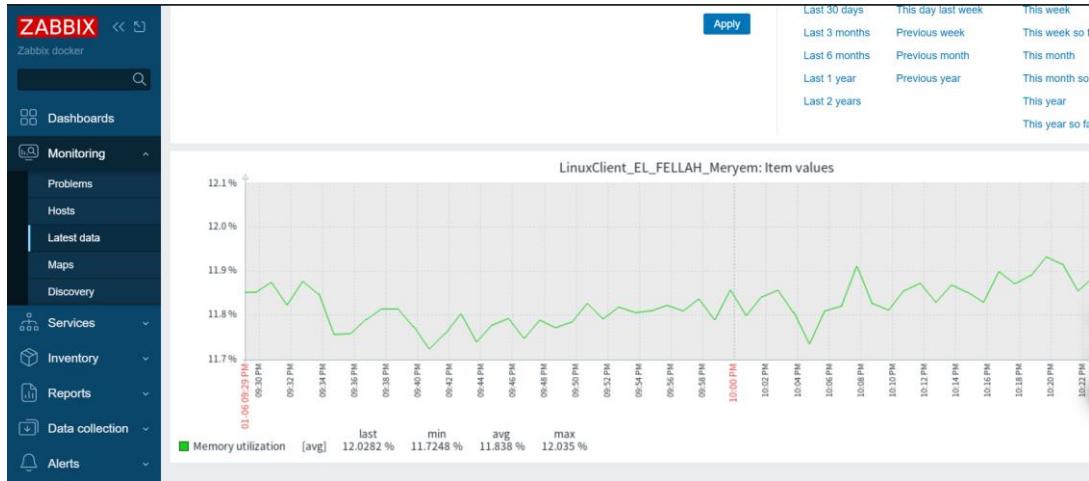


The screenshot shows the Zabbix interface with the 'Latest data' module selected. On the left, there's a sidebar with navigation links like Dashboards, Monitoring, Problems, Hosts, Latest data (which is highlighted), Maps, Discovery, Services, Inventory, Reports, Data collection, and Alerts. The main area is titled 'Latest data' and contains a search bar and several filter fields. Under 'Host groups', 'Linux servers' is selected. Under 'Hosts', 'LinuxClient_EL_FELLAH_Meryem' is selected. There are also fields for 'Name' and 'Tags'. Below these filters, there are buttons for 'Save as', 'Apply', and 'Reset'. A note says 'Subfilter affects only filtered data'. At the bottom, there are sections for 'HOSTS' (listing 'LinuxClient_EL_FELLAH_Meryem'), 'TAGS' (listing 'component'), and 'TAG VALUES' (listing various metrics like 'application', 'cpu', 'environment', etc.).

Je choisis par exemple le Tag Value memory.

The screenshot shows the Zabbix interface for monitoring a host named "LinuxClient_EL_FELLAH_Meryem". The left sidebar is the navigation menu. The main area displays "TAG VALUES" for the host, including component (memory), disk (nvme0n1), filesystem (/), and interface (ens5). Below this is a "DATA" section with two tabs: "With data" (selected) and "Without data". A table lists various metrics for the host, such as Available memory, Available memory in %, Free swap space, Free swap space in %, Memory utilization, Total memory, and Total swap space. The table includes columns for Host, Name, Last check, Last value, Change, and Tags. At the bottom of the table are several icons for filtering and displaying data.

Puis je fais check du host selon le name “Memory Utilization”, puis cliquer sur Display Graph, ce qui me donne tant que Admin l’opportunité à faire du Tracking de la Ressource memory en visualisant son changement dans un graph bien Claire.



b) Filtrer par Windows Client:

Pour pour montrer que ça réagit, dans session windows, j'ai ouvert 3 onglets youtube 8k, et le powershell, et file explorer, ça fait 15 minutes ou plus. Quand je reviens vers zabbix, Monitoring → Latest data:

Host groups

Tags And/Or Or

Hosts

Name

Show tags None 1 2 3 Tag name Full Shortened None

Tag display priority

State All Normal Not supported

Show details

Subfilter affects only filtered data

HOSTS
Windows_Server_Client_EL_FELLAH_Meryem 110

TAGS
component 110 description 9 disk 8 filesystem 5 filetype 5 interface 9 name 54 service 54

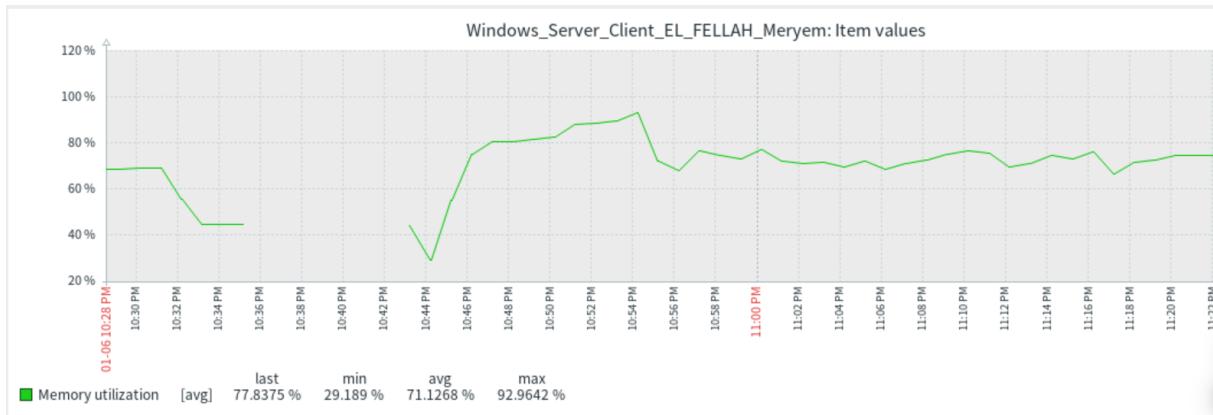
TAG VALUES
component: application 1 cpu 8 memory 12 network 9 os 4 raw 3 storage 17 system 61
description: Ethernet 9
disk: 0 C: 8
filesystem: C: 5
interface: NTEC 2

Je choisis le Tag Value memory.

<input type="checkbox"/> Host	Name ▲	Last check	Last value	Change	Tags
<input type="checkbox"/>	Windows_Server_Clie... Cache bytes	40s	58.16 MB	+252 KB	component: memory
<input type="checkbox"/>	Windows_Server_Clie... Free swap space	58s	862.3 MB	+1.09 MB	component: memory component: storag...
<input type="checkbox"/>	Windows_Server_Clie... Free swap space in %	45s	69.5892 %	+8.3466 %	component: memory component: storag...
<input type="checkbox"/>	Windows_Server_Clie... Free system page table entries	41s	4294006839	-92	component: memory
<input type="checkbox"/>	Windows_Server_Clie... Memory page faults per second	42s	9882.0026	+3771.9937	component: memory
<input type="checkbox"/>	Windows_Server_Clie... Memory pages per second	43s	0	-450.4885	component: memory
<input type="checkbox"/>	Windows_Server_Clie... Memory pool non-paged	44s	91.55 MB	-56 KB	component: memory
<input checked="" type="checkbox"/>	Windows_Server_Clie... Memory utilization	4s	74.5156 %	-0.3133 %	component: memory
<input type="checkbox"/>	Windows_Server_Clie... Total memory	2s	3.83 GB		component: memory
<input type="checkbox"/>	Windows_Server_Clie... Total swap space	59s	1.38 GB		component: memory component: storag...
<input type="checkbox"/>	Windows_Server_Clie... Used memory	3s	2.92 GB	+67.58 MB	component: memory
<input type="checkbox"/>	Windows_Server_Clie... Used swap space in %	45s	30.4108 %	-8.3466 %	component: memory component: storag...

1 selected

Puis je fais check du host selon le name “Memory Utilization”, puis cliquer sur Display Graph, afin de me permettre tant que Admin de surveiller l'utilisation de la memoire par le serveur Windows. Le graphe ci-dessous montre que le pourcentage de l'utilisation de la memoire a augmenter rapidement entre 11:44PM – 11:48PM, parce que à ce temps j'étais entrain de lancer les 3 vidéos sur youtube, ainsi que d'autres programmes comme le powershell et le file explorer.



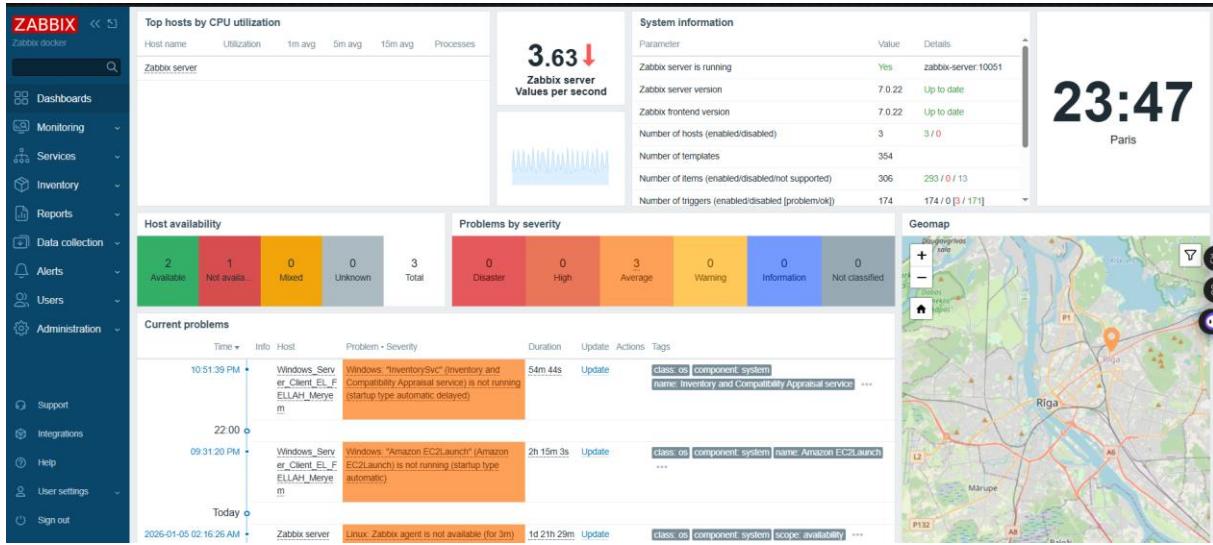
9.2) Vérifier les problèmes / alertes

Time	Severity	Recovery time	Status	Info	Host	Problem	Duration	Update	Actions	Tags
11:31:22 PM	Average	11:33:22 PM	RESOLVED		Windows_Server_Client_EL_FELLAH_Meryem	Windows: "AppXSvc" (AppX Deployment Service (AppXSVC)) is not running (startup type automatic)	2m	Update	...	class: os component: s name: AppX Deployme
23:00	Average		PROBLEM		Windows_Server_Client_EL_FELLAH_Meryem	Windows: "InventorySvc" (Inventory and Compatibility Appraisal service) is not running (startup type automatic delayed)	46m 38s	Update	...	class: os component: s name: Inventory and Co
22:00	Average		PROBLEM		Windows_Server_Client_EL_FELLAH_Meryem	Windows: "Amazon EC2Launch" (Amazon EC2Launch) is not running (startup type automatic)	2h 6m	Update	...	class: os component: s name: Amazon EC2La
Today	Average		PROBLEM	Zabbix server		Linux: Zabbix agent is not available (for 3m)	1d 21h	Update	...	class: os component: s scope: availability
2026-01-05 02:16:26 AM	Average		PROBLEM	Zabbix server		Linux: Zabbix agent is not available (for 3m)	21m	Update	...	class: os component: s scope: availability

La puissance de la solution Zabbix réside dans sa capacité à fournir une visibilité immédiate sur l'état de santé du parc informatique. Comme l'illustre le widget "**Current problems**" du tableau de bord, le système assure une détection proactive des anomalies en temps réel.

- Monitoring des incidents :** Le tableau de bord affiche un total de quatre événements détectés sur le client Windows, démontrant l'efficacité de la communication entre l'agent et le serveur.
- Détails et Granularité :** Pour chaque alerte, Zabbix fournit des informations précises : l'heure exacte de l'incident, le niveau de严重性 (marqué ici en orange comme "Average") et la nature du problème, tel que l'arrêt inattendu de services système comme le *Windows Update Service*.
- Résolution et Historique :** On observe qu'un des problèmes a déjà été marqué comme résolu, ce qui permet à l'administrateur de suivre le cycle de vie complet d'un incident directement depuis la console centrale.
- Aide à l'administration :** Cette centralisation des alertes simplifie considérablement la tâche de l'administrateur, lui permettant d'intervenir rapidement sur des composants critiques sans avoir à inspecter manuellement chaque instance du parc hybride.

9.3) Dashboards:



Cette capture d'écran présente le tableau de bord centralisé après la configuration réussie du parc hybride. On y observe plusieurs indicateurs clés de performance (KPI) qui valident le travail réalisé :

- Host Availability (Disponibilité des hôtes) :** Le widget indique que **2 hôtes sont "Available" (en vert)**. Cela confirme que le **Serveur Zabbix** communique correctement avec le **Client Linux** et le **Client Windows** via leurs agents respectifs.
- System Information :** On confirme ici que le serveur Zabbix est en cours d'exécution ("Zabbix server is running: Yes") sur le port **10051**, conformément à l'architecture réseau définie.
- Current Problems (Problèmes actuels) :** Cette section montre la réactivité du système en temps réel. On y voit des alertes spécifiques provenant de l'hôte **Windows_Server_Client_EL_FELLAH_Meryem**, prouvant que les modèles (templates) de surveillance Windows ont été correctement appliqués et remontent des informations précises sur l'état des services du système.
- Statut Global :** Avec un total de **3 hôtes** enregistrés (le serveur lui-même et les deux clients), cette interface démontre la réussite du déploiement conteneurisé sous Docker pour superviser un environnement cloud AWS complexe.

Conclusion

La réalisation de ce projet a permis de mettre en place une infrastructure de supervision robuste et évolutive sur **AWS**. En combinant la puissance de **Zabbix** et la flexibilité de la conteneurisation avec **Docker**, nous avons réussi à unifier le monitoring d'un parc hybride composé d'instances **Linux** et **Windows**.

L'implémentation réussie du réseau (VPC, Security Groups) et le déploiement des agents ont permis d'établir une remontée de données précise et une détection d'incidents en temps réel. Ce projet confirme que l'automatisation et la centralisation des métriques sont des leviers essentiels pour garantir la haute disponibilité et la sécurité des services cloud modernes.

Difficultés Rencontrées et Solutions

- ✿ **Confusion sur le protocole de connexion à distance :** Lors de la phase d'accès aux clients, il y a eu une confusion initiale entre l'accès SSH et l'accès RDP pour l'instance Windows.
 - **Solution :** Il a fallu identifier que l'instance Windows_Server_Client_EL_FELLAH_Meryem nécessitait le protocole **RDP** et le déchiffrement d'un mot de passe administrateur via la clé .pem, contrairement au client Linux qui s'administre via SSH.
- ✿ **Ajustement de la communication Agent-Serveur au sein du réseau privé :** La configuration des paramètres de l'agent (Server et ServerActive) représentait un point critique pour assurer la remontée des données sans exposer inutilement les flux sur l'internet public.
 - **Solution :** La solution a consisté à utiliser exclusivement les **adresses IP privées** du VPC (comme 10.0.3.253) dans le fichier de configuration de l'agent et à ouvrir précisément le port **10050** dans le Groupe de Sécurité pour permettre au serveur Zabbix de communiquer avec ses clients en toute sécurité.