

TP :

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

*Filière d'ingénieur :*

*Ingénierie Informatique Big Data et Intelligence Artificielle*

Réalisé par :

*Najoua Mouaddab*

Encadré par :

*Prof. Azeddine KHIAT*

*Année universitaire : 2025/2026*

## CONTENTS

|  |    |
|--|----|
| 1. Introduction .....  | 3  |
| Présentation du projet .....   | 3  |
| Présentation des outils.....   | 3  |
| 2. Architecture Réseau.....  | 4  |
| 2.1. Création du VPC .....   | 4  |
| 2.2. Création du sous-réseau public.....                               | 5  |
| 2.3. Configuration des groupes de sécurité .....                       | 6  |
| 2.4. Passerelle Internet et Routage.....                               | 7  |
| 3. Architecture des Instances EC2.....                                 | 8  |
| 3.1. Serveur Zabbix (Ubuntu, t3.large) .....                           | 8  |
| 3.2. Client Linux (Ubuntu, t3.medium).....                             | 8  |
| 3.3. Client Windows (Windows Server, t3.large) .....                   | 8  |
| 4. Déploiement du Serveur Zabbix.....                                  | 9  |
| 4.1. Connexion au serveur Zabbix .....                                 | 9  |
| 4.2. Mise à jour des paquets .....                                     | 9  |
| 4.3. Installation de Docker .....                                      | 10 |
| 4.4. Création du fichier Docker Compose .....                          | 11 |
| 4.5. Vérification de l'installation .....                              | 12 |
| 5. Configuration des Clients (Agents).....                             | 14 |
| 5.1. Client Linux.....   | 14 |
| 5.2. Mise à jour des paquets .....                                     | 15 |
| 5.3. Installation de l'agent Zabbix .....                              | 15 |
| 6. Configuration du client Windows.....                                | 16 |
| 6.1Connexion au client Windows.....                                    | 16 |
| 6.2Téléchargement et installation de l'agent Zabbix pour Windows ..... | 17 |
| 7. Conclusion.....   | 18 |

## 1. INTRODUCTION

## PRÉSENTATION DU PROJET

Ce projet a pour objectif de déployer une infrastructure de supervision centralisée et conteneurisée sur Amazon Web Services (AWS). La solution de monitoring choisie est Zabbix, qui sera installée via Docker pour faciliter son déploiement et sa gestion. L'infrastructure doit permettre de surveiller un parc informatique hybride, composé d'un serveur client sous Linux (Ubuntu) et d'un autre sous Windows Server. Ce projet, réalisé dans le cadre du cycle d'ingénieur en Big Data et Intelligence Artificielle, met en pratique les compétences d'administration système et cloud, essentielles pour la gestion de parcs informatiques modernes et complexes.

## PRÉSENTATION DES OUTILS

- **Amazon Web Services (AWS)** : Le fournisseur de cloud public choisi pour héberger l'intégralité de l'infrastructure. Nous utilisons plusieurs de ses services fondamentaux :
  - **VPC (Virtual Private Cloud)** : Pour créer un réseau isolé et sécurisé.
  - **EC2 (Elastic Compute Cloud)** : Pour déployer les machines virtuelles (instances) qui serviront de serveur Zabbix et de clients.
  - **Security Groups** : Pour agir en tant que pare-feu virtuel et contrôler le trafic réseau.
- **Docker** : Une plateforme de conteneurisation qui permet d'emballer une application et ses dépendances dans un conteneur. Nous l'utilisons pour déployer Zabbix (serveur, interface web, base de données) de manière rapide, reproductible et isolée.
- **Zabbix** : Une solution open-source de supervision de réseaux et d'applications. Elle est composée de plusieurs éléments :
  - **Zabbix Server** : Le cœur du système, qui collecte et traite les données.
  - **Base de données (ex: PostgreSQL)** : Stocke toutes les configurations et les données de monitoring.

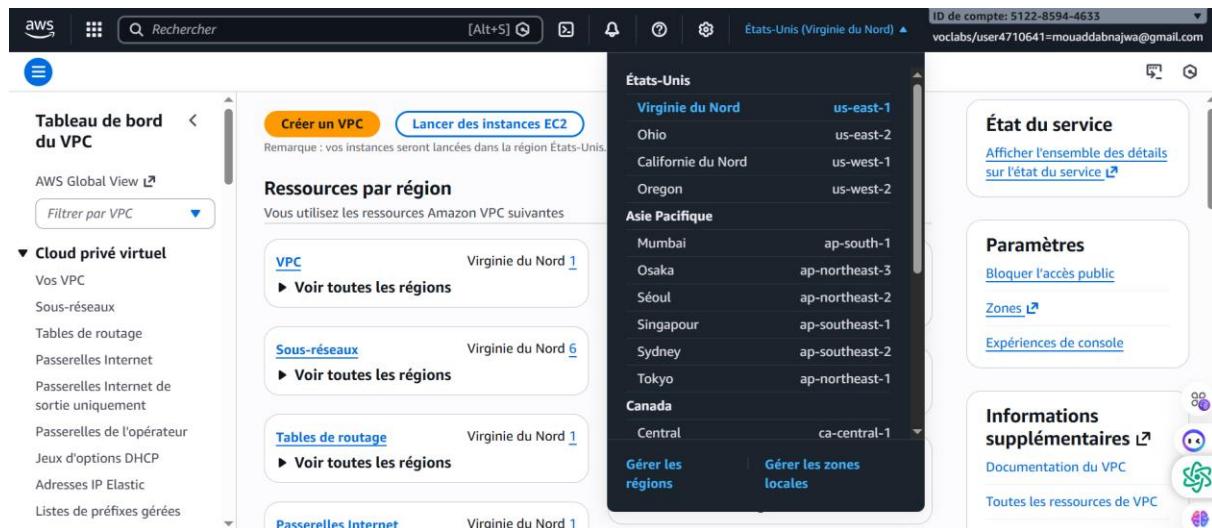
- **Interface Web (frontend)** : Permet de visualiser les données, configurer la surveillance et créer des graphiques.
- **Zabbix Agent** : Un petit programme installé sur les machines à surveiller (clients) qui remonte les informations au serveur.

## 2.ARCHITECTURE RÉSEAU

La première étape a consisté à mettre en place l'infrastructure réseau sur AWS, en respectant les limitations du Learner Lab.

### 2.1. CRÉATION DU VPC

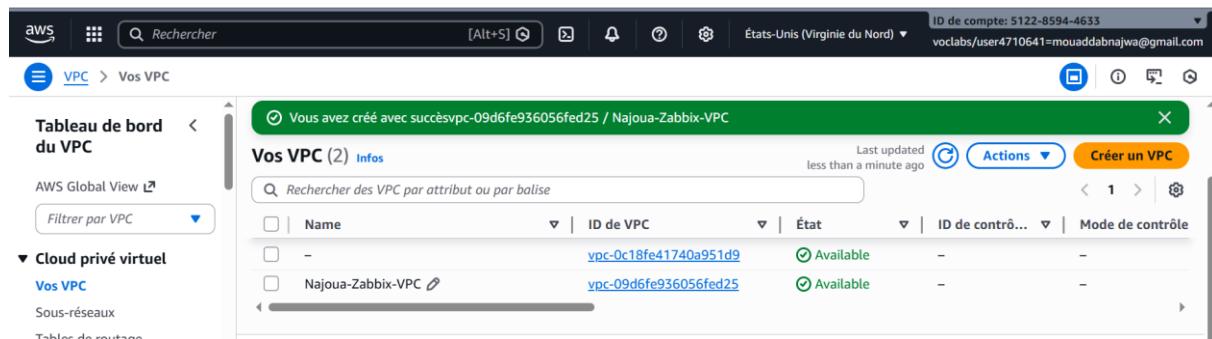
Nous avons commencé par créer un VPC (Virtual Private Cloud) dédié à ce projet. Ce VPC isole nos ressources dans le cloud.



The screenshot shows the AWS VPC Dashboard. On the left, there's a sidebar for 'Cloud privé virtuel' with options like 'Vos VPC', 'Sous-réseaux', 'Tables de routage', etc. The main area displays 'Ressources par région' with sections for 'VPC', 'Sous-réseaux', 'Tables de routage', and 'Passerelles Internet'. A modal window titled 'États-Unis' lists regions and their IDs:

| Région             | ID             |
|--------------------|----------------|
| Virginie du Nord   | us-east-1      |
| Ohio               | us-east-2      |
| Californie du Nord | us-west-1      |
| Oregon             | us-west-2      |
| Mumbai             | ap-south-1     |
| Osaka              | ap-northeast-3 |
| Séoul              | ap-northeast-2 |
| Singapour          | ap-southeast-1 |
| Sydney             | ap-southeast-2 |
| Tokyo              | ap-northeast-1 |
| Canada             | ca-central-1   |
| Central            | -              |

## VPC : Création du réseau virtuel



The screenshot shows the AWS VPC Dashboard under the 'Vos VPC' section. It displays a table with two entries:

| Name                  | ID de VPC             | État      | ID de contrôle | Mode de contrôle |
|-----------------------|-----------------------|-----------|----------------|------------------|
| vpc-0c18fe41740a951d9 | vpc-0c18fe41740a951d9 | Available | -              | -                |
| Najoua-Zabbix-VPC     | vpc-09d6fe936056fed25 | Available | -              | -                |

### 2.2. CREATION DU SOUS-RESEAU PUBLIC

À l'intérieur de ce VPC, nous avons créé un unique sous-réseau public pour simplifier l'accès à nos instances.

aws Rechercher [Alt+S] Pose une question à Amazon Q États-Unis (Virginie du Nord) ID de compte: 5122-8594-4633 vclabs/user4710641=mouaddabnajwa@gmail.com

☰ VPC > Sous-réseaux > Créer un sous-réseau



### Créer un sous-réseau Infos

#### VPC

##### ID de VPC

Créez des sous-réseaux dans ce VPC.

vpc-09d6fe936056fed25 (Najoua-Zabbix-VPC)



#### CIDR de VPC associés

##### CIDR IPv4

10.0.0.0/16

aws Rechercher [Alt+S] Pose une question à Amazon Q États-Unis (Virginie du Nord) ID de compte: 5122-8594-4633 vclabs/user4710641=mouaddabnajwa@gmail.com

☰ VPC > Sous-réseaux > Créer un sous-réseau



#### Sous-réseau 1 sur 1

##### Nom du sous-réseau (subnet)

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

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-az6 (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

10.0.1.0/24

256 IPs

## 2.3. CONFIGURATION DES GROUPES DE SECURITE

Nous avons créé deux groupes de sécurité. Le premier pour le serveur Zabbix, autorisant les ports SSH (22) et HTTP (80) depuis mon IP, ainsi que les ports agents (10050/10051) depuis les clients.

aws Rechercher [Alt+S] Pose une question à Amazon Q États-Unis (Virginie du Nord) ID de compte: 5122-8594-4633 vclabs/user4710641=mouaddabnajwa@gmail.com

☰ 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

SG-Zabbix-Project

Le nom ne peut pas être modifié après sa création.

##### Description Informations

autorisation au serveur zabbix

##### VPC Informations

vpc-09d6fe936056fed25 (Najoua-Zabbix-VPC)



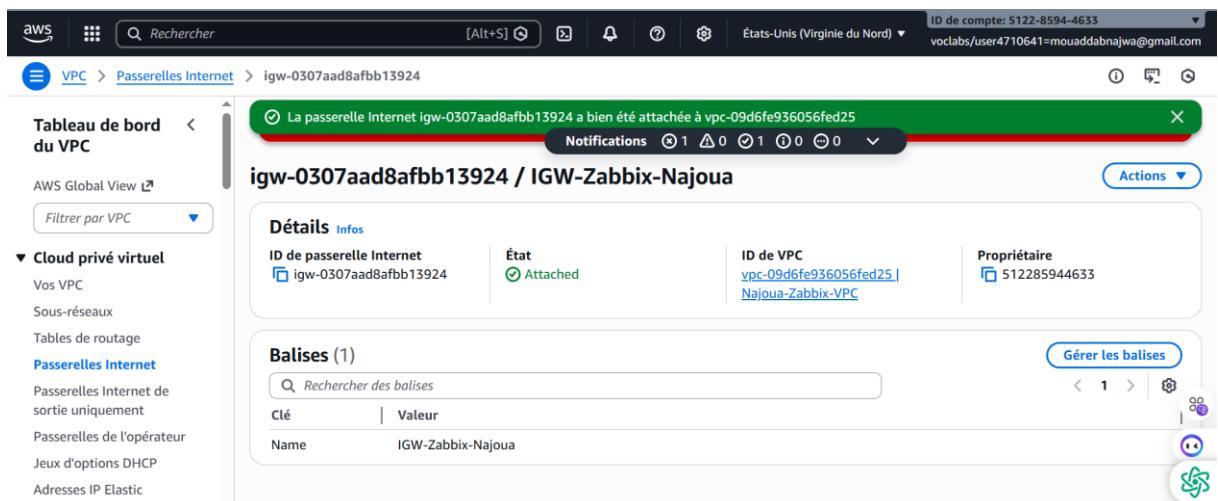
Le second groupe de sécurité est destiné aux clients, autorisant SSH (22) pour Linux, RDP (3389) pour Windows, et les ports agents (10050/10051) depuis le serveur Zabbix.

| Type             | Informations | Protocole    | Plage de ports | Source | Informations     | Description - facultatif | Informations              |
|------------------|--------------|--------------|----------------|--------|------------------|--------------------------|---------------------------|
| Informations     |              | Informations |                |        |                  |                          |                           |
| SSH              |              | TCP          | 22             | Mo...  | 41.143.10.124/32 |                          | <a href="#">Supprimer</a> |
| HTTP             |              | TCP          | 80             | N'i... | 0.0.0.0/0        |                          | <a href="#">Supprimer</a> |
| HTTPS            |              | TCP          | 443            | N'i... | 0.0.0.0/0        |                          | <a href="#">Supprimer</a> |
| TCP personnalisé |              | TCP          | 10050          | N'i... | 0.0.0.0/0        |                          | <a href="#">Supprimer</a> |
| RDP              |              | TCP          | 3389           | Mo...  | 41.143.10.124/32 |                          | <a href="#">Supprimer</a> |

[Ajouter une règle](#)

## 2.4. PASSERELLE INTERNET ET ROUTAGE

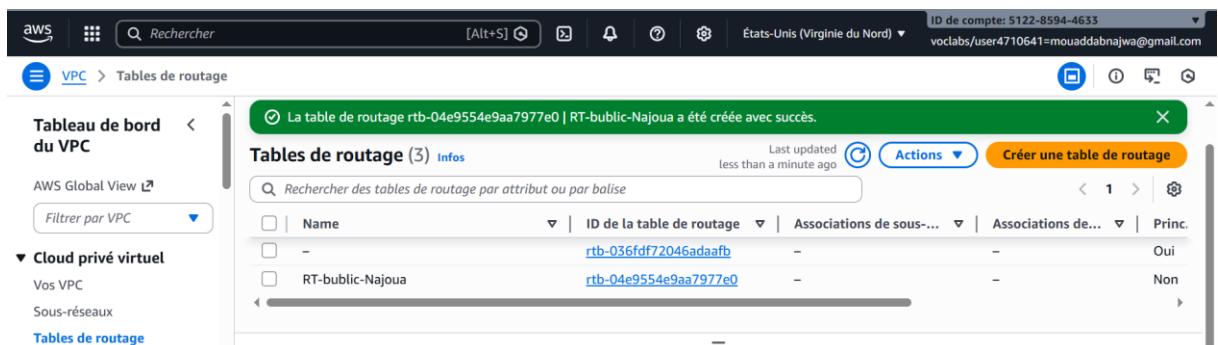
Pour permettre aux instances d'accéder à Internet et d'être accessibles depuis l'extérieur, nous avons créé et attaché une Passerelle Internet à notre VPC.



The screenshot shows the AWS VPC console with the following details:

- Tableau de bord du VPC**: Shows the VPC configuration.
- Cloud privé virtuel**: Shows the VPC network.
- Passerelles Internet**: Shows the creation of an Internet Gateway named "igw-0307aad8afbb13924". A success message indicates it has been attached to the VPC.
- Détails**: Shows the gateway's ID (igw-0307aad8afbb13924), its status as "Attached", its VPC ID (vpc-09d6fe936056fed25), and its owner (Najoua-Zabbix-VPC).
- Balises (1)**: Shows a single tag "Name: IGW-Zabbix-Najoua".

Nous avons ensuite configuré une table de routage pour diriger le trafic Internet (0.0.0.0/0) vers cette passerelle.



The screenshot shows the AWS VPC console with the following details:

- Tableau de bord du VPC**: Shows the VPC configuration.
- Cloud privé virtuel**: Shows the VPC network.
- Tables de routage**: Shows the creation of a routing table named "RT-public-Najoua". A success message indicates it was created successfully.
- Tables de routage (3) Infos**: Shows the routing table configuration. It lists two routes: one for the local table (rtb-036fdf72046adaafb) and one for the public IP range (rtb-04e9554e9aa7977e0).

### 3. ARCHITECTURE DES INSTANCES EC2

Conformément au cahier des charges, nous avons lancé trois instances EC2.

#### 3.1. SERVEUR ZABBIX (UBUNTU, T3.LARGE)

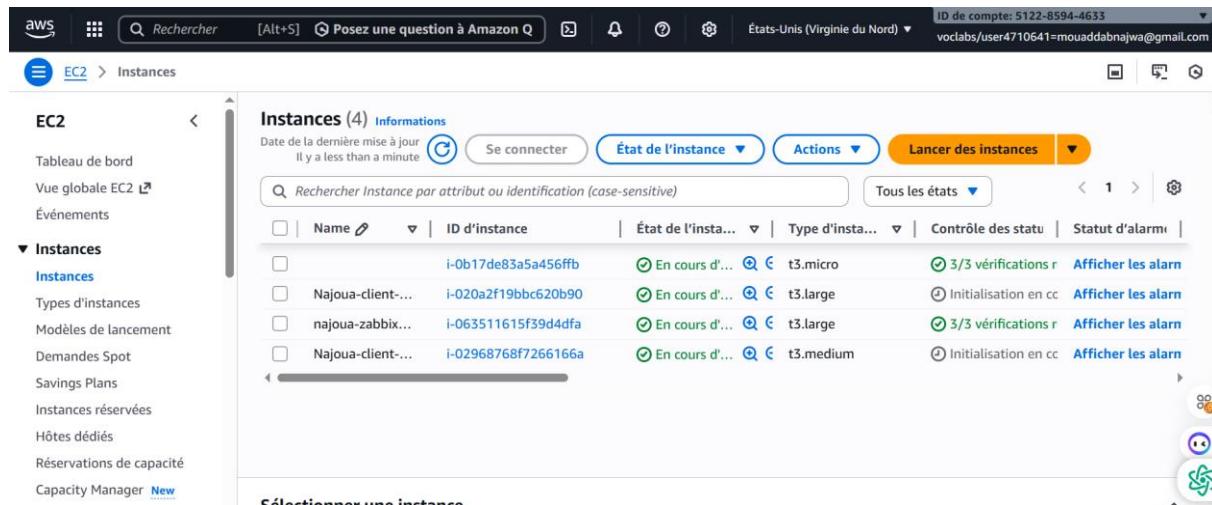
- **Nom :** Zabbix-Server
- **OS :** Ubuntu 22.04 LTS
- **Type :** t3.large (2 vCPU, 8 Go RAM)
- **Rôle :** Héberger tous les conteneurs Docker Zabbix.

#### 3.2. CLIENT LINUX (UBUNTU, T3.MEDIUM)

- Nom : Client-Linux
- OS : Ubuntu 22.04 LTS
- Type : t3.medium (2 vCPU, 4 Go RAM)

#### 3.3. CLIENT WINDOWS (WINDOWS SERVER, T3.LARGE)

- Nom : Client-Windows
- OS : Windows Server
- Type : t3.large (2 vCPU, 8 Go RAM)



| Name                | ID d'instance       | État de l'insta... | Type d'insta... | Contrôle des statu... | Statut d'alarm...     |
|---------------------|---------------------|--------------------|-----------------|-----------------------|-----------------------|
| i-0b17de83a5a456ffb | i-0b17de83a5a456ffb | En cours d'...     | t3.micro        | 3/3 vérifications r   | Afficher les alarm... |
| Najoua-client-...   | i-020a2f19bbc620b90 | En cours d'...     | t3.large        | Initialisation en cc  | Afficher les alarm... |
| najoua-zabbix...    | i-063511615f39d4dfa | En cours d'...     | t3.large        | 3/3 vérifications r   | Afficher les alarm... |
| Najoua-client-...   | i-02968768f7266166a | En cours d'...     | t3.medium       | Initialisation en cc  | Afficher les alarm... |

## 4. DEPLOIEMENT DU SERVEUR ZABBIX

Une fois l'instance Zabbix-Server opérationnelle, nous nous y sommes connectés en SSH pour procéder au déploiement de la solution de monitoring.

### 4.1. CONNEXION AU SERVEUR ZABBIX

Pour accéder au serveur, nous avons utilisé la commande SSH avec notre fichier de clé privée (zabbix-key.pem) et l'adresse publique de l'instance. La première connexion nous a demandé de confirmer l'authenticité de l'hôte.

```
C:\Users\lenovo\Downloads>ssh -i "zabbix-key.pem" ubuntu@ec2-34-235-118-162.compute-1.amazonaws.com
The authenticity of host 'ec2-34-235-118-162.compute-1.amazonaws.com (34.235.118.162)' can't be established.
ED25519 key fingerprint is SHA256:Ay8tjP510vVwdjPLiSE1fxC7VaoNMrW+bYAtEwaCAvI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-235-118-162.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1018-aws x86_64)

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

System information as of Wed Feb 11 09:42:21 UTC 2026

System load:  0.0          Temperature:      -273.1 C
Usage of /:   26.4% of 6.71GB  Processes:        113
Memory usage: 3%           Users logged in:  1
Swap usage:   0%           IPv4 address for ens5: 172.31.40.63

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

Last login: Wed Feb 11 09:36:50 2026 from 18.206.107.27
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-40-63:~$
```

Une fois connectés, nous avons pu constater les informations système de l'instance, notamment son adresse IP privée (172.31.40.63) qui sera utilisée pour la communication avec les agents.

### 4.2. MISE A JOUR DES PAQUETS

Avant d'installer Docker, nous avons commencé par mettre à jour la liste des paquets disponibles.

```
ubuntu@ip-172-31-40-63:~$ 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 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1446 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [234 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.6 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [9892 B]
Get:11 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [929 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [212 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [74.2 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [19.9 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [2445 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/restricted Translation-en [562 kB]
Get:18 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:19 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [536 B]
Get:20 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [28.8 kB]
Get:21 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [6492 B]
Get:22 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Get:23 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [396 B]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1741 kB]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [324 kB]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [16.5 kB]
Get:33 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1528 kB]
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [313 kB]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [386 kB]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [31.9 kB]
Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [2588 kB]
```

Cette commande a permis de récupérer les informations des dépôts Ubuntu nécessaires pour l'installation des logiciels.

#### 4.3. INSTALLATION DE DOCKER

Nous avons ensuite procédé à l'installation de Docker. La commande sudo apt install -y docker.io a installé Docker ainsi que tous ses composants nécessaires.

```
ubuntu@ip-172-31-40-63:~$ sudo apt install -y docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 83 not upgraded.
Need to get 76.1 MB of archives.
After this operation, 288 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/main amd64 bridge-utils amd64 1.7.1-1ubuntu2 [33.9 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble-updates/main amd64 runc amd64 1.3.3-0ubuntu1~24.04.3 [8815 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble-updates/main amd64 containerd amd64 1.7.28-0ubuntu1~24.04.2 [38.4 MB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble-updates/main amd64 dns-root-data all 2024071801-ubuntu0.24.04.1 [5918 B]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble-updates/main amd64 dnsmasq-base amd64 2.90-2ubuntu0.1 [376 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble-updates/universe amd64 docker.io amd64 28.2.2-0ubuntu1~24.04.1 [28.3 MB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/noble-updates/universe amd64 ubuntu-fan all 0.12.16+24.04.1 [34.2 kB]
Fetched 76.1 MB in 1s (91.4 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 71752 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.7.1-1ubuntu2_amd64.deb ...
Unpacking bridge-utils (1.7.1-1ubuntu2) ...
Selecting previously unselected package runc.
Preparing to unpack .../2-runc_1.3.3-0ubuntu1~24.04.3_amd64.deb ...
Unpacking runc (1.3.3-0ubuntu1~24.04.3) ...
Selecting previously unselected package containerd.
Preparing to unpack .../3-containerd_1.7.28-0ubuntu1~24.04.2_amd64.deb ...
```

L'installation a inclus plusieurs paquets supplémentaires comme runc, container, pigz, etc., qui sont les dépendances nécessaires au bon fonctionnement de Docker.

#### 4.4. CREATION DU FICHIER DOCKER COMPOSE

Après l'installation de Docker, nous avons créé un répertoire de travail et un fichier docker-compose.yml pour définir l'architecture conteneurisée de Zabbix.

Le fichier docker-compose.yml définit trois services :

- **zabbix-db** : La base de données MySQL qui stockera toutes les données de configuration et de monitoring.
- **zabbix-server** : Le cœur de Zabbix qui collecte et traite les données.
- **zabbix-web** : L'interface web Apache permettant de visualiser et configurer la supervision.

Les variables d'environnement configurées (comme MYSQL\_ROOT\_PASSWORD, MYSQL\_DATABASE, etc.) permettent aux conteneurs de communiquer entre eux et d'accéder à la base de données.

```
ubuntu@ip-172-31-40-63: ~/zabbix
GNU nano 7.2
version: '3'
services:
  zabbix-db:
    image: mysql:5.7
    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_DATABASE: zabbix
      MYSQL_USER: zabbix
      MYSQL_PASSWORD: zabbix

  zabbix-server:
    image: zabbix/zabbix-server-mysql
    ports:
      - "10051:10051"
    environment:
      DB_SERVER_HOST: zabbix-db
      MYSQL_USER: zabbix
      MYSQL_PASSWORD: zabbix
    depends_on:
      - zabbix-db

  zabbix-web:
    image: zabbix/zabbix-web-apache-mysql
    ports:
      - "80:8080"
    environment:
      DB_SERVER_HOST: zabbix-db
      MYSQL_USER: zabbix
      MYSQL_PASSWORD: zabbix
    depends_on:
      - zabbix-server
```

#### 4.5. VÉRIFICATION DE L'INSTALLATION

Après avoir installé Docker et préparé le fichier docker-compose.yml, nous avons procédé à une vérification de l'état du serveur. Lors d'une reconnexion ultérieure au serveur Zabbix, nous avons pu

observer les informations système actualisées.

```
ubuntu@ip-172-31-40-63: ~
C:\Users\lenovo\Downloads>ssh -i zabbix-key.pem ubuntu@35.171.150.64
The authenticity of host '35.171.150.64 (35.171.150.64)' can't be established.
ED25519 key fingerprint is SHA256:Ay8tjP510vVwdjPLiSE1fxC7VaoNMrW+bYAtEwaCAvI.
This host key is known by the following other names/addresses:
  C:\Users\lenovo/.ssh/known_hosts:4: ec2-34-235-118-162.compute-1.amazonaws.com
  C:\Users\lenovo/.ssh/known_hosts:7: 54.86.124.165
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Host key verification failed.

C:\Users\lenovo\Downloads>ssh -i zabbix-key.pem ubuntu@35.171.150.64
The authenticity of host '35.171.150.64 (35.171.150.64)' can't be established.
ED25519 key fingerprint is SHA256:Ay8tjP510vVwdjPLiSE1fxC7VaoNMrW+bYAtEwaCAvI.
This host key is known by the following other names/addresses:
  C:\Users\lenovo/.ssh/known_hosts:4: ec2-34-235-118-162.compute-1.amazonaws.com
  C:\Users\lenovo/.ssh/known_hosts:7: 54.86.124.165
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '35.171.150.64' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1018-aws x86_64)

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

System information as of Fri Feb 13 20:03:22 UTC 2026

System load:  0.01           Temperature:      -273.1 C
Usage of /:   52.2% of 6.71GB Processes:          123
Memory usage: 3%            Users logged in:    0
Swap usage:   0%            IPv4 address for ens5: 172.31.40.63

Expanded Security Maintenance for Applications is not enabled.

43 updates can be applied immediately.
To see these additional updates run: apt list --upgradable
```

Cette capture montre plusieurs éléments importants :

- La connexion SSH à l'adresse 35.171.150.64 (l'IP publique de notre serveur Zabbix) a été établie avec succès.
- Le système d'exploitation est **Ubuntu 24.04.3 LTS**.
- L'adresse IP privée du serveur est **172.31.40.63**, qui sera utilisée pour la communication avec les agents Zabbix.
- L'utilisation mémoire n'est que de **3%**, ce qui est normal pour une instance fraîchement lancée.
- L'espace disque utilisé est de **52.2%** (environ 3.5 Go sur 6.71 Go), ce qui inclut l'installation de Docker et ses dépendances.
- **43 mises à jour** sont disponibles, ce qui est typique après une installation initiale.

```
43 updates can be applied immediately.  
To see these additional updates run: apt list --upgradable  
  
1 additional security update can be applied with ESM Apps.  
Learn more about enabling ESM Apps service at https://ubuntu.com/esm  
  
Last login: Fri Feb 13 15:56:45 2026 from 105.68.144.68  
ubuntu@ip-172-31-40-63:~$
```

Cette seconde capture confirme que :

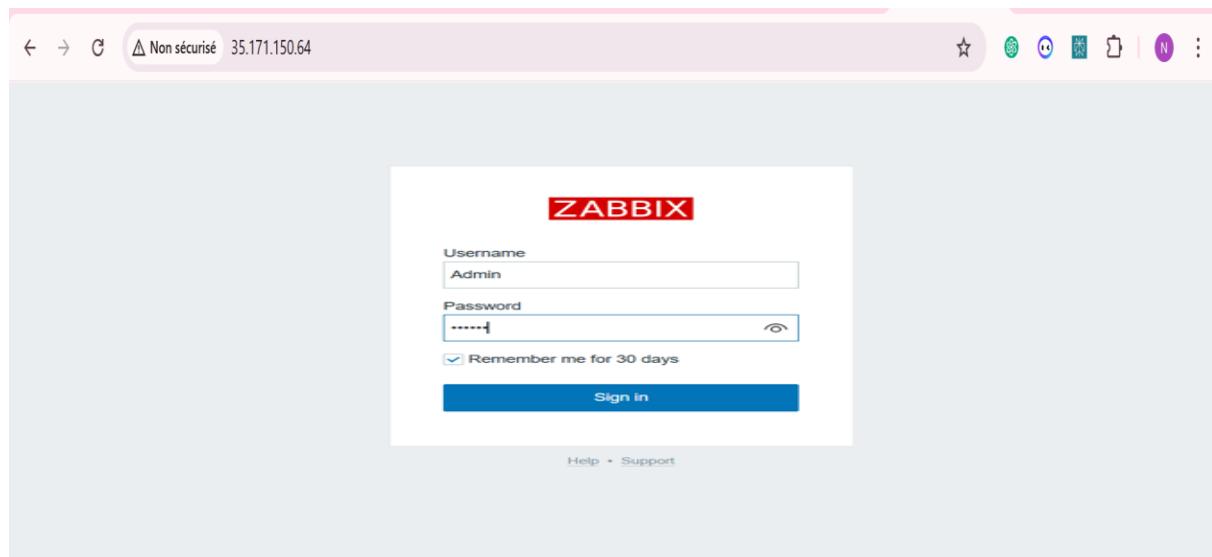
- La dernière connexion au serveur a eu lieu le **13 février 2026 à 15:56:45** depuis l'adresse IP 105.68.144.68.
- Une mise à jour de sécurité supplémentaire est disponible via ESM Apps (Expanded Security Maintenance).
- L'invite de commande `ubuntu@ip-172-31-40-63:~$` nous indique que nous sommes prêts à exécuter les commandes suivantes pour lancer les conteneurs Zabbix.

## 5. CONFIGURATION DES CLIENTS (AGENTS)

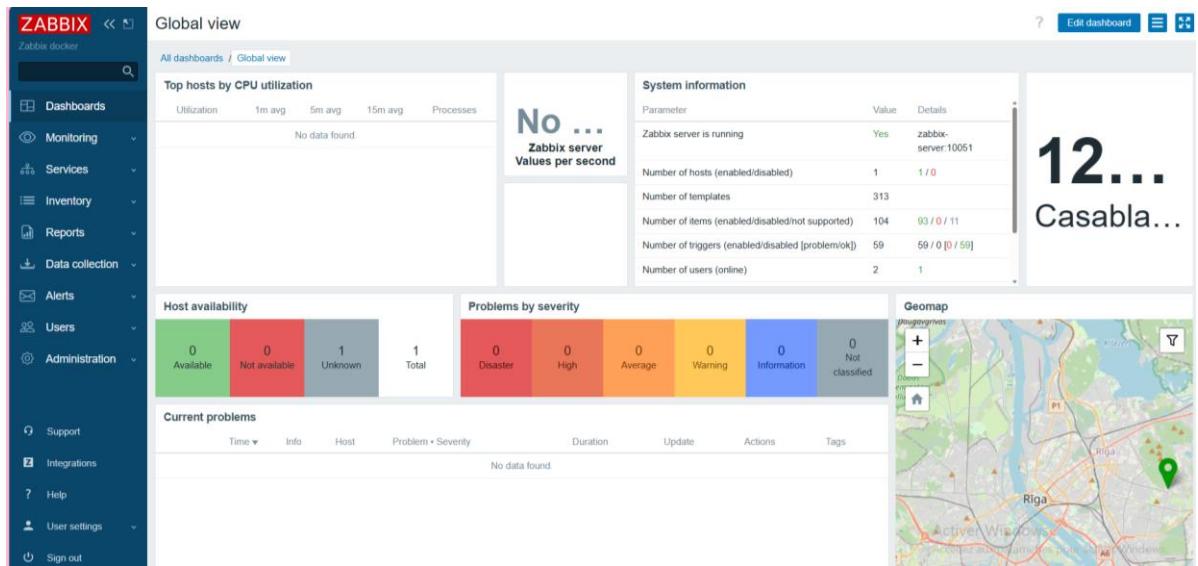
Après le déploiement du serveur Zabbix, nous avons procédé à l'installation et à la configuration des agents sur les machines clientes (Linux et Windows)

### 5.1. CLIENT LINUX

Nous nous sommes connectés en SSH à l'instance Client-Linux en utilisant la même clé privée (`zabbix-key.pem`).



La connexion a été établie avec succès



## 5.2. MISE A JOUR DES PAQUETS

Avant d'installer l'agent Zabbix, nous avons mis à jour la liste des paquets disponibles.

```
ubuntu@ip-172-31-40-63:~$ 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://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
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 [1739 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 [174 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://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1431 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [313 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [386 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [31.9 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [2582 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]
```

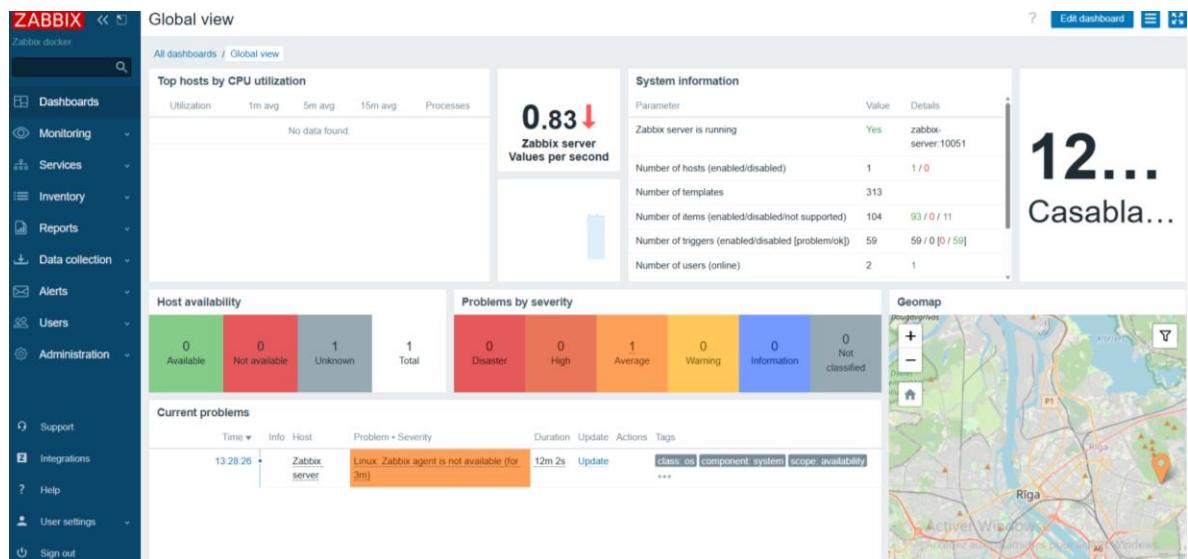
La commande sudo apt update a rafraîchi les dépôts Ubuntu, préparant le système pour l'installation.

## 5.3. INSTALLATION DE L'AGENT ZABBIX

Nous avons ensuite procédé à l'installation de l'agent Zabbix.

```
c:~$ ubuntu@ip-172-31-40-63:~$ sudo apt install mysql-client-core-8.0 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libmodbus5
The following NEW packages will be installed:
  libmodbus5 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 libmodbus5 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 (478 kB/s)
Selecting previously unselected package libmodbus5:amd64.
(Reading database ... 71758 files and directories currently installed.)
Preparing to unpack .../libmodbus5_3.1.10-1ubuntu1_amd64.deb ...
Unpacking libmodbus5: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 libmodbus5: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...
```

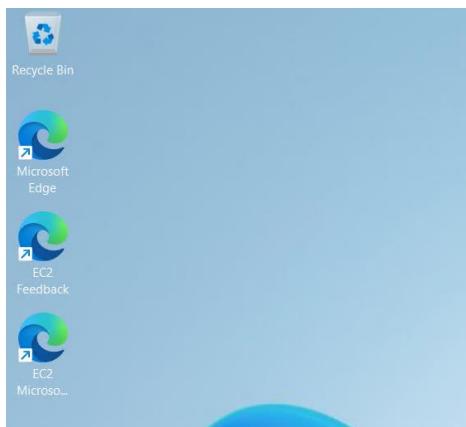
La commande `sudo apt install zabbix-agent -y` a installé l'agent ainsi que ses dépendances (notamment `libmodbus5`). L'installation a automatiquement créé un lien symbolique pour démarrer l'agent au boot du système .



## 6. CONFIGURATION DU CLIENT WINDOWS

### 6.1 CONNEXION AU CLIENT WINDOWS

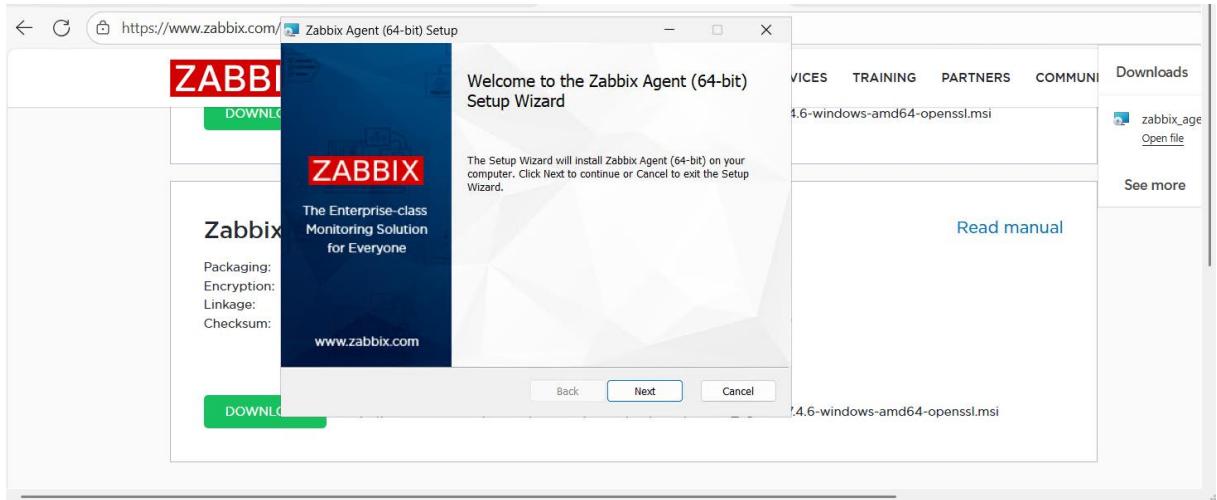
Nous nous sommes connectés en RDP (Remote Desktop Protocol) à l'instance Client-Windows en utilisant son adresse IP publique et le mot de passe administrateur.



Le bureau Windows affiche les icônes habituelles (Corbeille, Microsoft Edge, etc.), confirmant que la connexion à distance fonctionne correctement.

## 6.2 TELECHARGEMENT ET INSTALLATION DE L'AGENT ZABBIX POUR WINDOWS

Depuis le client Windows, nous avons ouvert un navigateur et téléchargé l'installateur de l'agent Zabbix pour Windows depuis le site officiel ([www.zabbix.com/download](https://www.zabbix.com/download)).



L'assistant d'installation de Zabbix Agent (64-bit) pour Windows a été lancé. Nous avons suivi les étapes de l'assistant en renseignant :

- **Adresse du serveur Zabbix** : L'adresse IP privée du serveur Zabbix (172.31.40.63)
- **Nom de l'hôte** : Client-Windows

## 7. CONCLUSION

### Bilan du projet

Ce projet a été mené avec succès. Nous avons déployé une infrastructure de monitoring complète et fonctionnelle sur AWS. Un serveur Zabbix conteneurisé avec Docker a été installé et configuré pour superviser deux clients, l'un sous Linux et l'autre sous Windows. L'interface web permet de visualiser l'état de santé du parc en temps réel.

### Difficultés rencontrées et solutions apportées

- **Limitations du Learner Lab :** L'arrêt automatique des instances a nécessité de redémarrer les conteneurs Docker à chaque reconnexion avec la commande docker-compose up -d.
- **Communication entre le serveur et les agents :** Le problème le plus fréquent a été l'utilisation d'adresses IP incorrectes. La solution a été de s'assurer que dans la configuration des agents et lors de l'ajout des hôtes dans Zabbix, on utilise bien les **adresses IP privées** des instances.
- **Groupes de sécurité :** Il a fallu configurer précisément les règles pour autoriser le trafic sur les ports 10050/10051 entre le serveur et les clients.

## 8. LIVRABLE GITHUB

Conformément aux exigences du projet et pour assurer une meilleure valorisation du travail réalisé, un dépôt GitHub a été créé. Il centralise l'ensemble des ressources et de la documentation nécessaires à la reproduction de cette infrastructure.

### Objectifs du dépôt

- **Reproductibilité** : Permettre à toute personne de reproduire l'infrastructure pas à pas
- **Transparence** : Montrer l'ensemble des configurations utilisées
- **Portfolio** : Valoriser les compétences techniques acquises
- **Veille technique** : Servir de base pour de futurs projets similaires

### Accès au dépôt

Le dépôt est accessible publiquement à l'adresse suivante :

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