



Projet Examen Final

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

Réalisé par :
ALLOUKH Ilyas

Table des matières

Table des matières

Outils et Technologies.....	3
1 Architecture Réseau	3
1.1 Création du VPC.....	3
1.2 Groupes de Sécurité.....	4
2 Architecture des Instances EC2.....	4
3 Déploiement du Serveur Zabbix.....	5
3.1 Installation de Docker	5
3.2 Lancement et Validation.....	7
3.3 Accès à l'Interface Web	7
4 Configuration des Clients.....	9
4.1 Client Linux (Ubuntu).....	9
Configuration de l'Agent	10
4.1.2 Déclaration de l'hôte dans Zabbix.....	11
4.2 Client Windows (Windows Server 2019)	11
4.2.1 Accès à l'instance	11
4.2.2 Installation et Configuration de l'Agent	12
5 Monitoring et Tableaux de Bord	13
5.1 Validation de la Connectivité	13
5.2 Visualisation des Données	14
6 Conclusion.....	15

Outils et Technologies

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

1 Architecture Réseau

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

1.1 Création du VPC

Le VPC a été créé avec le bloc CIDR 10.0.0.0/16.

The screenshot displays the AWS Management Console's VPC Dashboard. On the left, a sidebar lists navigation options under 'Virtual Private Cloud' and 'Security'. The main content area shows 'Vos VPC' (Your VPCs) with a table listing three VPCs. The third VPC, 'VPCC-Zabbix-vpc' with ID 'vpc-0ff47272d764b914d', is selected. Below the table, the details for this VPC are shown, including its CIDR block (10.0.0.0/16), DNS resolution status (Active), and DHCP options (dopt-069b101c7b3746c24).

Name	ID de VPC	État	ID de contrô...	Mode de contrôle d...	Bloquer l'acc...	CIDR
d	vpc-06de2c1a9bb78d9d4	Available	-	-	Désactivé	10.0.0.0/16
-	vpc-0c90ac8685d949eaa	Available	-	-	Désactivé	172.31.0.0/16
VPCC-Zabbix-vpc	vpc-0ff47272d764b914d	Available	-	-	Désactivé	10.0.0.0/16

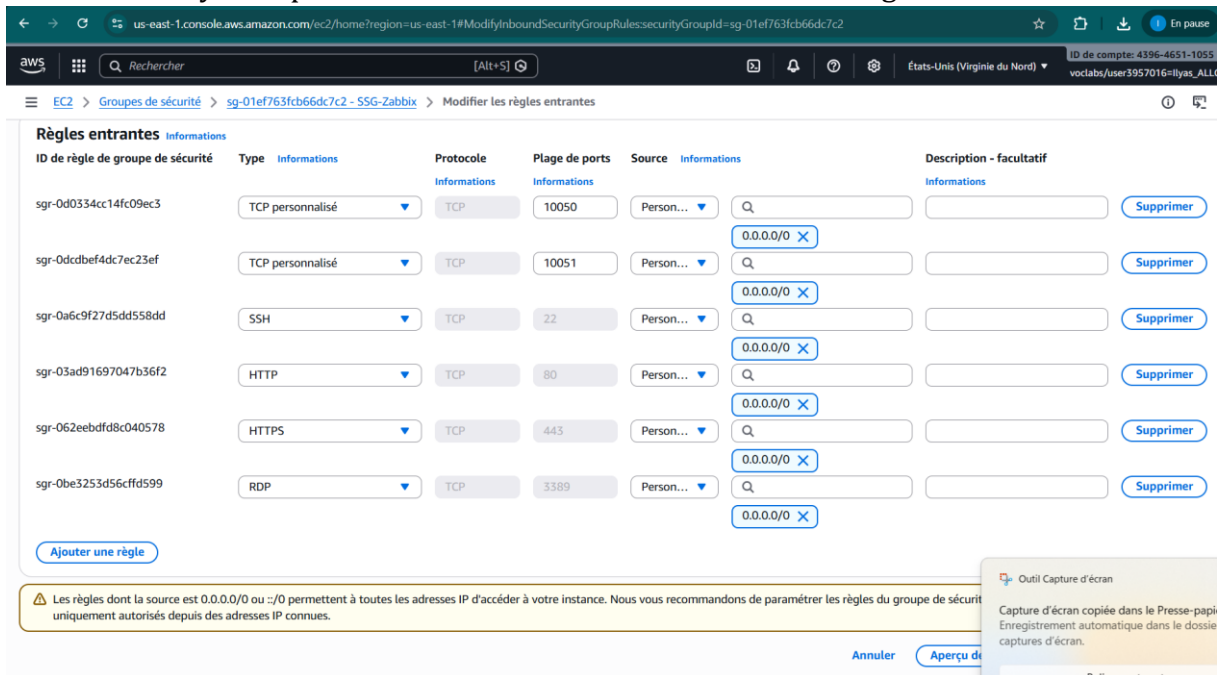
vpc-0ff47272d764b914d / VPCC-Zabbix-vpc

Détails

ID de VPC vpc-0ff47272d764b914d	État Available	Bloquer l'accès public Désactivé	Noms d'hôte DNS Activé
Résolution DNS Activé	Location default	Jeu d'options DHCP dopt-069b101c7b3746c24	Table de routage principale rtb-00ca7bd80f93b4abe
ACL réseau principal acl-0af6f4d93b14ee13006	VPC par défaut vpc-0ff47272d764b914d	CIDR IPv4 10.0.0.0/16	Groupe IPv6 -

1.2 Groupes de Sécurité

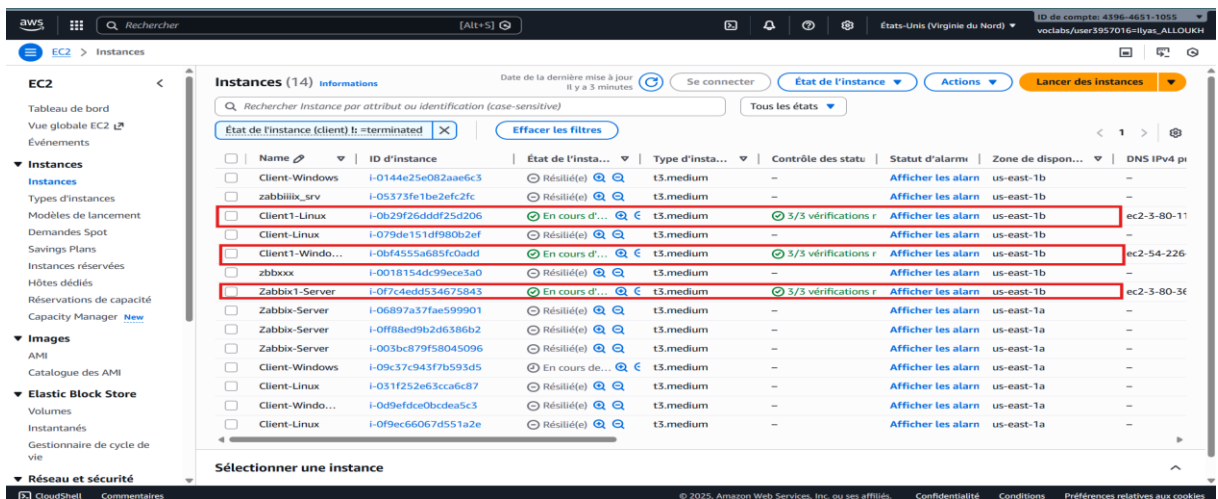
Le Security Group autorise les flux nécessaires au monitoring et à l'administration



ID de règle de groupe de sécurité	Type	Protocole	Plage de ports	Source	Description - facultatif
sgr-0d0334cc14fc09ec3	TCP personnalisé	TCP	10050	Person...	0.0.0.0/0
sgr-0dcdbe4dc7ec23ef	TCP personnalisé	TCP	10051	Person...	0.0.0.0/0
sgr-0a6c9f27d5dd558dd	SSH	TCP	22	Person...	0.0.0.0/0
sgr-03ad91697047b36f2	HTTP	TCP	80	Person...	0.0.0.0/0
sgr-062eebdf8dc040578	HTTPS	TCP	443	Person...	0.0.0.0/0
sgr-0be3253d56c5ff599	RDP	TCP	3389	Person...	0.0.0.0/0

2 Architecture des Instances EC2

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



Name	ID d'instance	État de l'inst...	Type d'insta...	Contrôle des statu	Statut d'alarme	Zone de dispon...	DNS IPv4 pi
Client-Windows	i-0144e25e082aae6c3	Résilié(e)	t3.medium	-	Afficher les alarme	us-east-1b	-
zabbiilix_srv	i-05373fe1be2efc2fc	Résilié(e)	t3.medium	-	Afficher les alarme	us-east-1b	-
Client1-Linux	i-0b29f26dddf25d206	En cours d'...	t3.medium	3/3 vérifications r	Afficher les alarme	us-east-1b	ec2-3-80-11
Client-Linux	i-079de151df980b2ef	Résilié(e)	t3.medium	-	Afficher les alarme	us-east-1b	-
Client1-Windo...	i-0b64555a685fc0add	En cours d'...	t3.medium	3/3 vérifications r	Afficher les alarme	us-east-1b	ec2-54-226
zbbxxx	i-0018154dc99ece3a0	Résilié(e)	t3.medium	-	Afficher les alarme	us-east-1b	-
Zabbix-Server	i-0f7c4edd534675843	En cours d'...	t3.medium	3/3 vérifications r	Afficher les alarme	us-east-1b	ec2-3-80-36
Zabbix-Server	i-06897a37fae599901	Résilié(e)	t3.medium	-	Afficher les alarme	us-east-1a	-
Zabbix-Server	i-0ff88ed9b2d6386b2	Résilié(e)	t3.medium	-	Afficher les alarme	us-east-1a	-
Zabbix-Server	i-003bc879f58045096	Résilié(e)	t3.medium	-	Afficher les alarme	us-east-1a	-
Client-Windows	i-09c37c943f7b593d5	En cours de...	t3.medium	-	Afficher les alarme	us-east-1a	-
Client-Linux	i-031f252e63cca6c87	Résilié(e)	t3.medium	-	Afficher les alarme	us-east-1a	-
Client-Windo...	i-0d9efdc0b0cdea5c3	Résilié(e)	t3.medium	-	Afficher les alarme	us-east-1a	-
Client-Linux	i-0f9ec66067d551a2e	Résilié(e)	t3.medium	-	Afficher les alarme	us-east-1a	-

3 Déploiement du Serveur Zabbix

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

3.1 Installation de Docker

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

```
1 # 1. Mise à jour du système
2 sudo apt update && sudo apt upgrade -y
3
4 # 2. Installation de Docker et Docker Compose
5 sudo apt install docker.io -y
6 sudo apt install docker-compose -y
7
8 # 3. Démarrage et activation du service Docker
9 sudo systemctl start docker
10 sudo systemctl enable docker
11
12 # 4. Ajout de l'utilisateur actuel au groupe docker
13 sudo usermod -aG docker $USER
```

```
ubuntu@ip-10-0-25-56: ~$ sudo apt update
sudo apt install -y docker.io docker-compose
sudo systemctl start docker
sudo systemctl enable docker
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [3161 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [484 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [19.0 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [5043 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [944 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [644 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1244 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [310 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [30.0 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [57.6 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [13.2 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [600 B]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [69.4 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [11.5 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [412 B]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
```

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

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

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

```
GNU nano 6.2 docker-compose.yml
version: '3.5'
services:
  zabbix-db:
    image: mysql:8.0
    command: --character-set-server=utf8 --collation-server=utf8_bin --default-authentication-plugin=mysql_native_password
    volumes:
      - ./zbx_db_data:/var/lib/mysql
    env_file:
      - .env_db_mysql

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

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

3.2 Lancement et Validation

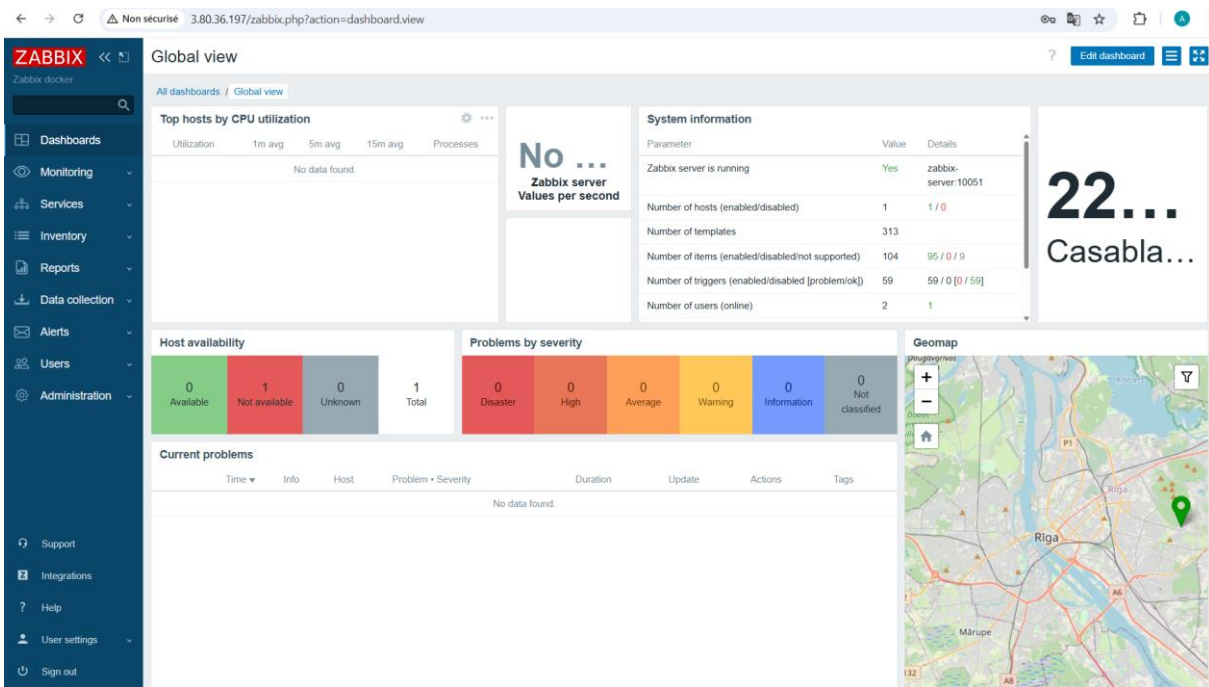
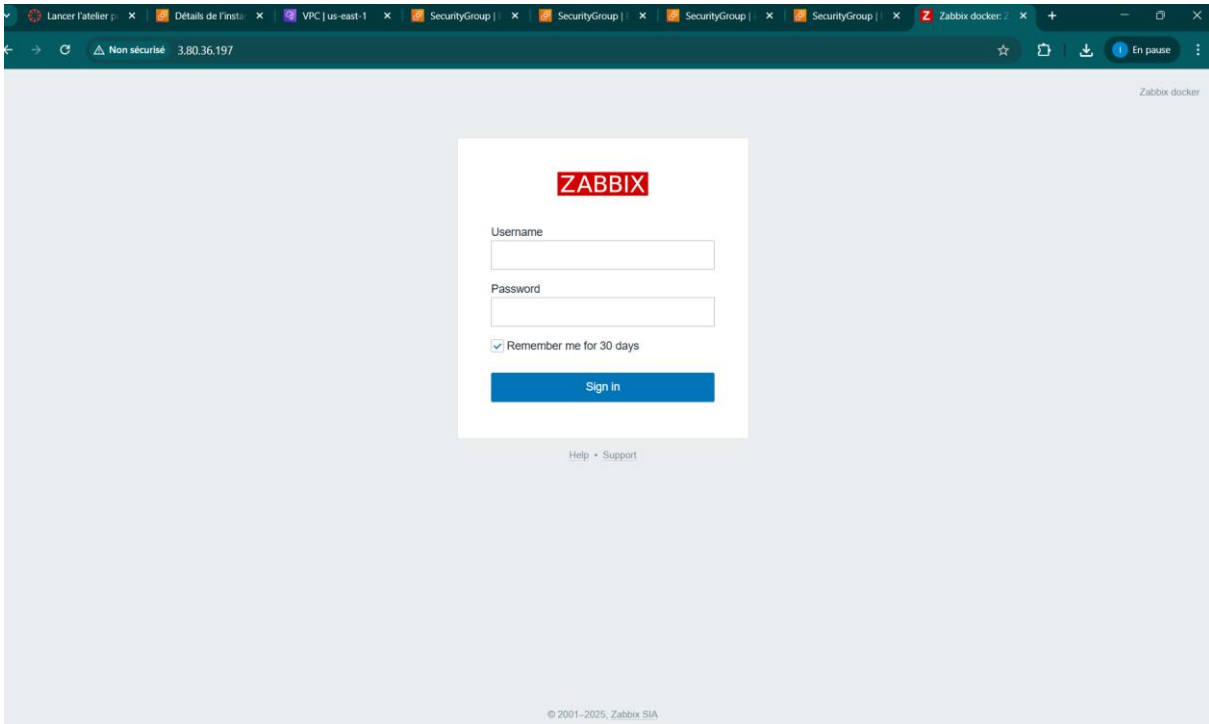
La vérification de l'état des conteneurs confirme que le déploiement est un succès :

```
ubuntu@ip-10-0-13-208: ~/zabbix-infra$ docker ps
```

CONTAINER ID	IMAGE	NAMES	COMMAND	CREATED	STATUS	PORTS
568424759c14	zabbix/zabbix-web-apache-mysql:ubuntu-6.4-latest		"docker-entrypoint.sh"	34 seconds ago	Up 34 seconds (healthy)	8443/tcp, 0.0.0.0:80->
8080/tcp, [::]:80->8080/tcp		zabbix-infra_zabbix-web_1				
d27d94458133	zabbix/zabbix-server-mysql:ubuntu-6.4-latest		"/usr/bin/tini -- /u_"	35 seconds ago	Up 34 seconds	0.0.0.0:10051->10051/t
cp, [::]:10051->10051/tcp		zabbix-infra_zabbix-server_1				
7ec795a6b021	mysql:8.0		"docker-entrypoint.s_"	37 seconds ago	Up 34 seconds	3306/tcp, 33060/tcp
		zabbix-infra_zabbix-db_1				

3.3 Accès à l'Interface Web

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



4 Configuration des Clients

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

4.1 Client Linux (Ubuntu)

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

Configuration de l'Agent

Nous avons édité les paramètres suivants :

- **Server=10.0.25.56** : Adresse IP privée de notre serveur Zabbix (autorisé à faire du polling).
- **ServerActive=10.0.25.56** : Adresse IP pour les vérifications actives.
- **Hostname=Client-Linux** : Nom unique identifiant la machine.

```
GNU nano 6.2 /etc/zabbix/zabbix_agentd.conf *
# IPv6 addresses must be enclosed in square brackets if port for that host is specified.
# If port is not specified, square brackets for IPv6 addresses are optional.
# If this parameter is not specified, active checks are disabled.
# Example: ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
#
# Mandatory: no
# Default:
# ServerActive=

ServerActive=10.0.25.56

### Option: Hostname
# Unique, case sensitive hostname.
# Required for active checks and must match hostname as configured on the server.
# Value is acquired from HostnameItem if undefined.
#
# Mandatory: no
# Default:
# Hostname=

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

### Option: HostMetadata

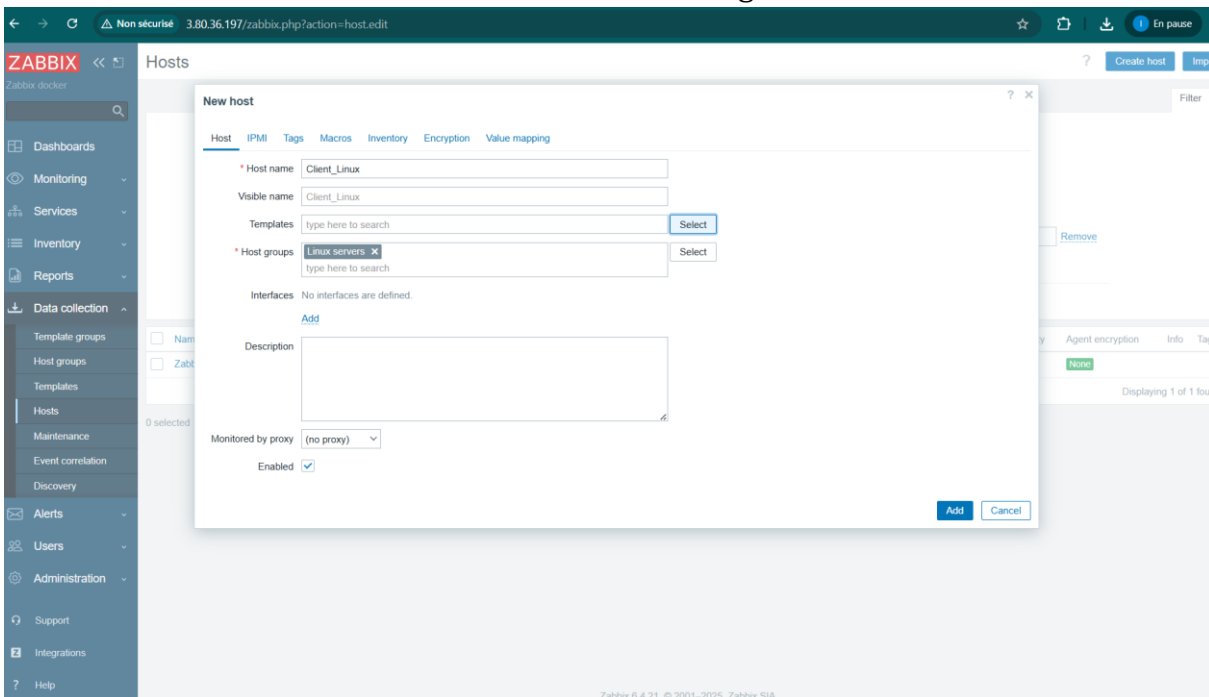
^G Help      ^O Write Out  ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo     M-A Set Mark
^X Exit      ^R Read File  ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line M-E Redo     M-6 Copy
```

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

```
ubuntu@ip-10-0-25-56:~/zabbix$ sudo systemctl restart zabbix-agent
ubuntu@ip-10-0-25-56:~/zabbix$ sudo systemctl enable zabbix-agent
Synchronizing state of zabbix-agent.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable zabbix-agent
ubuntu@ip-10-0-25-56:~/zabbix$
```

4.1.2 Déclaration de l'hôte dans Zabbix

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

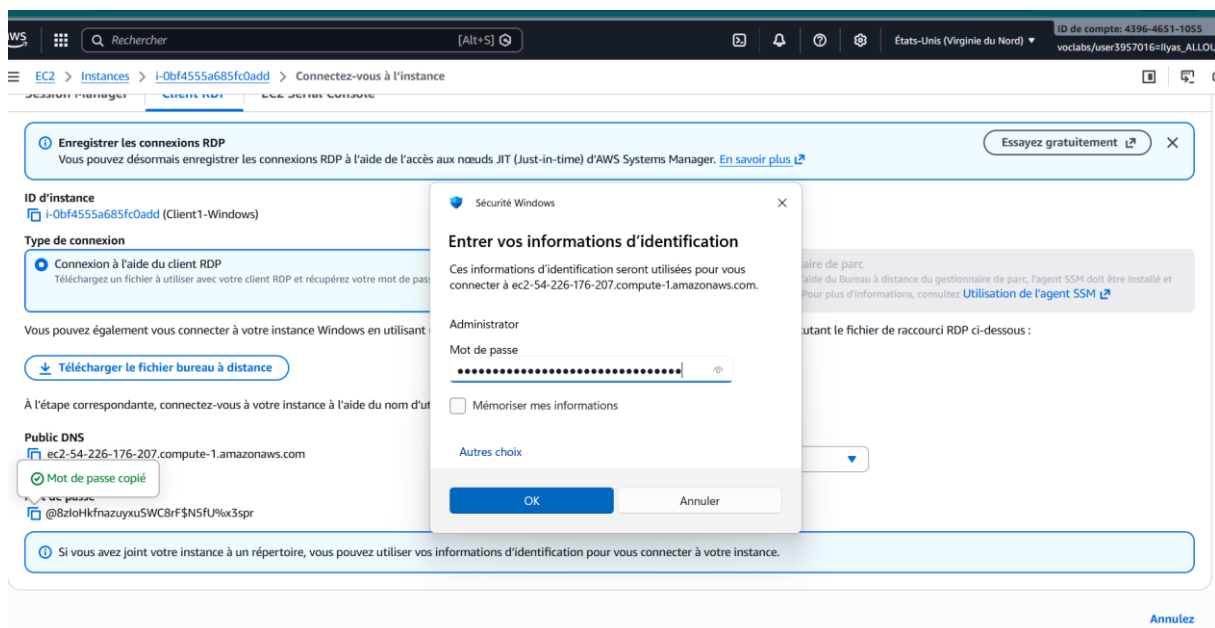


4.2 Client Windows (Windows Server 2019)

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

4.2.1 Accès à l'instance

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

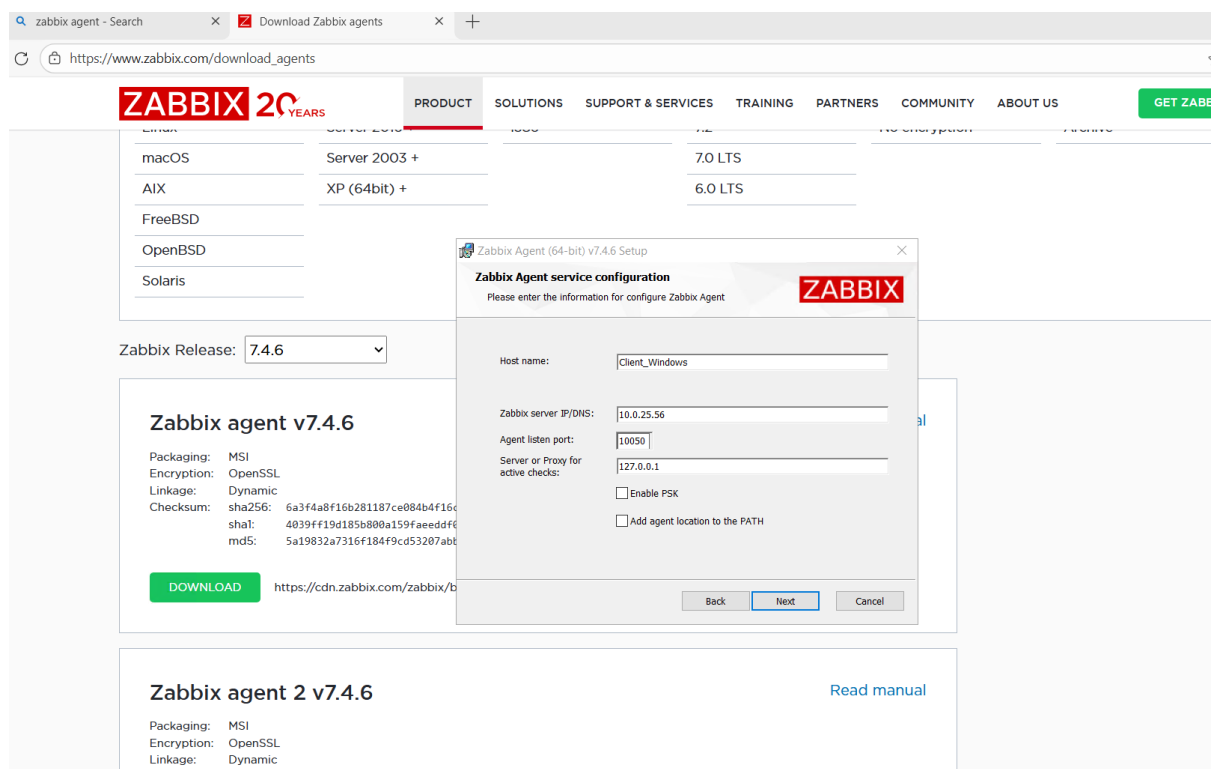


4.2.2 Installation et Configuration de l'Agent

Depuis le navigateur de la VM, nous avons téléchargé l'installateur MSI officiel de l'agent Zabbix.

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

- **Host name** : Client-Windows
- **Zabbix server IP** : 10.0.25.56
- **Agent listen port** : 10050



5 Monitoring et Tableaux de Bord

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

5.1 Validation de la Connectivité (Disponibilité)

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

ZABBIX << Zabbix docker

Hosts

Details Host disabled

Host groups: type here to search Select

Templates: type here to search Select

Name:

DNS:

IP:

Port:

Status: Any Enabled Disabled

Monitored by: Any Server Proxy

Proxy: Select

Tags: And/Or Or tag Contains value Remove

Add

Apply Reset

Name	Items	Triggers	Graphs	Discovery	Web	Interface	Proxy	Templates	Status	Availability	Agent encryption	Info	Tags
Client-Linux	Items	Triggers	Graphs	Discovery	Web	10.0.25.137.10050			Enabled	ZBX	None		
Client-Windows	Items	Triggers	Graphs	Discovery	Web	10.0.21.8.10050			Enabled	ZBX	None		
Client_Linux	Items	Triggers	Graphs	Discovery	Web	127.0.0.1.10050			Disabled		None		
Zabbix server	Items 104	Triggers 59	Graphs 19	Discovery 5	Web	127.0.0.1.10050		Linux by Zabbix agent, Zabbix server health	Enabled	ZBX	None		

0 selected Enable Disable Export Mass update Delete

Displaying 4 of 4 found

Zabbix 6.4.21. © 2001–2025, Zabbix SIA

5.2 Visualisation des Données (Graphiques)

ZABBIX << Zabbix docker

Monitoring

Problems

Hosts

Latest data

Maps

Discovery

Services

Inventory

Reports

Data collection

Alerts

Users

Administration

Support

Integrations

Help

User settings

Sign out

Host	Item	Value	Unit	Graph
Zabbix server	Linux: Number of running processes			Graph
Zabbix server	Linux: Operating system			History
Zabbix server	Linux: Operating system architecture			History
Zabbix server	Linux: System boot time			Graph
Zabbix server	Linux: System description			History
Zabbix server	Linux: System local time			Graph
Zabbix server	Linux: System name			History
Zabbix server	Linux: System uptime			Graph
Zabbix server	Linux: Total memory			Graph
Zabbix server	Linux: Total swap space			Graph
Zabbix server	Linux: Version of Zabbix agent running			History
Zabbix server	Linux: Zabbix agent availability	30s	not available (0)	Graph
Zabbix server	Linux: Zabbix agent ping			Graph
Zabbix server	Zabbix server: Configuration cache, % used	3s	21.8408 %	Graph
Zabbix server	Zabbix server: Connector queue			Graph
Zabbix server	Zabbix server: History index cache, % used	1m 1s	0.4195 %	Graph
Zabbix server	Zabbix server: History write cache, % used	2s	0 %	Graph
Zabbix server	Zabbix server: LLD queue	22s	0	Graph
Zabbix server	Zabbix server: Number of processed character values p...	2s	0	Graph
Zabbix server	Zabbix server: Number of processed log values per sec...	1m 1s	0	Graph

1 2 3

Displaying 1 to 50 of 104 found

6 selected Display stacked graph Display graph Execute now

Zabbix 6.4.21. © 2001–2025, Zabbix SIA

6 Conclusion

Ce projet nous a permis de concevoir, mettre en place et déployer avec succès une infrastructure de supervision centralisée hébergée sur le cloud AWS. Cette solution assure une visibilité globale et un contrôle efficace des ressources, tout en répondant pleinement aux exigences techniques, opérationnelles et de sécurité propres à un environnement hybride moderne. Grâce à l'utilisation des services cloud, l'infrastructure mise en œuvre offre une meilleure évolutivité, une haute disponibilité et une gestion optimisée des systèmes, facilitant ainsi la supervision continue des différents composants, qu'ils soient locaux ou hébergés dans le cloud..