

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 : Ait Lamine Fatima Zahra

Encadré par : Prof. Azeddine KHIAT

Année universitaire : 2025/2026

Repository GitHub du projet :

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

Résumé Exécutif

Ce projet présente la mise en œuvre d'une solution de supervision centralisée dans le cloud AWS utilisant Zabbix déployé sous forme de conteneurs Docker.

L'infrastructure permet de surveiller en temps réel un parc informatique hybride composé de machines Linux et Windows.

Objectifs principaux :

- Déployer une infrastructure de monitoring hautement disponible dans le cloud AWS
- Utiliser la conteneurisation Docker pour une gestion simplifiée et portable
- Superviser des systèmes hétérogènes (Linux et Windows) depuis une interface centralisée
- Assurer la sécurité réseau via des Security Groups AWS configurés de manière restrictive

Technologies utilisées :

- Amazon Web Services (AWS) - Infrastructure cloud
- Docker & Docker Compose - Conteneurisation
- Zabbix - Solution de supervision et monitoring
- Ubuntu Server - Système d'exploitation Linux
- Windows Server - Système d'exploitation Windows

1. Introduction

1.1 Contexte du projet

Dans un contexte où la supervision des infrastructures informatiques est devenue critique pour assurer la disponibilité et la performance des systèmes, ce projet vise à mettre en place une solution de monitoring centralisée et évolutive. L'utilisation du cloud AWS permet de bénéficier d'une infrastructure flexible et hautement disponible, tandis que Zabbix offre des capacités de supervision complètes et personnalisables.

1.2 Objectifs du projet

Les objectifs principaux de ce projet sont :

- Concevoir et déployer une architecture réseau sécurisée sur AWS
- Installer et configurer Zabbix en mode conteneurisé avec Docker Compose
- Déployer des agents Zabbix sur des machines Linux et Windows
- Configurer la supervision temps réel des ressources système
- Mettre en place des tableaux de bord et des alertes

1.3 Contraintes et limitations

Le projet a été réalisé dans l'environnement AWS Learner Lab, qui impose certaines contraintes :

- Budget limité en termes de crédits AWS
- Accès restreint à certains services AWS
- Durée de session limitée nécessitant une architecture simple

2. Architecture Réseau

2.1 Conception de l'architecture

L'architecture réseau a été conçue de manière simple et efficace pour faciliter l'accès et la gestion. Elle repose sur les composants suivants :

- **VPC unique** : Virtual Private Cloud isolé
- **Sous-réseau public** : Un seul subnet pour toutes les instances
- **Internet Gateway** : Accès Internet bidirectionnel
- **Security Groups** : Règles de pare-feu

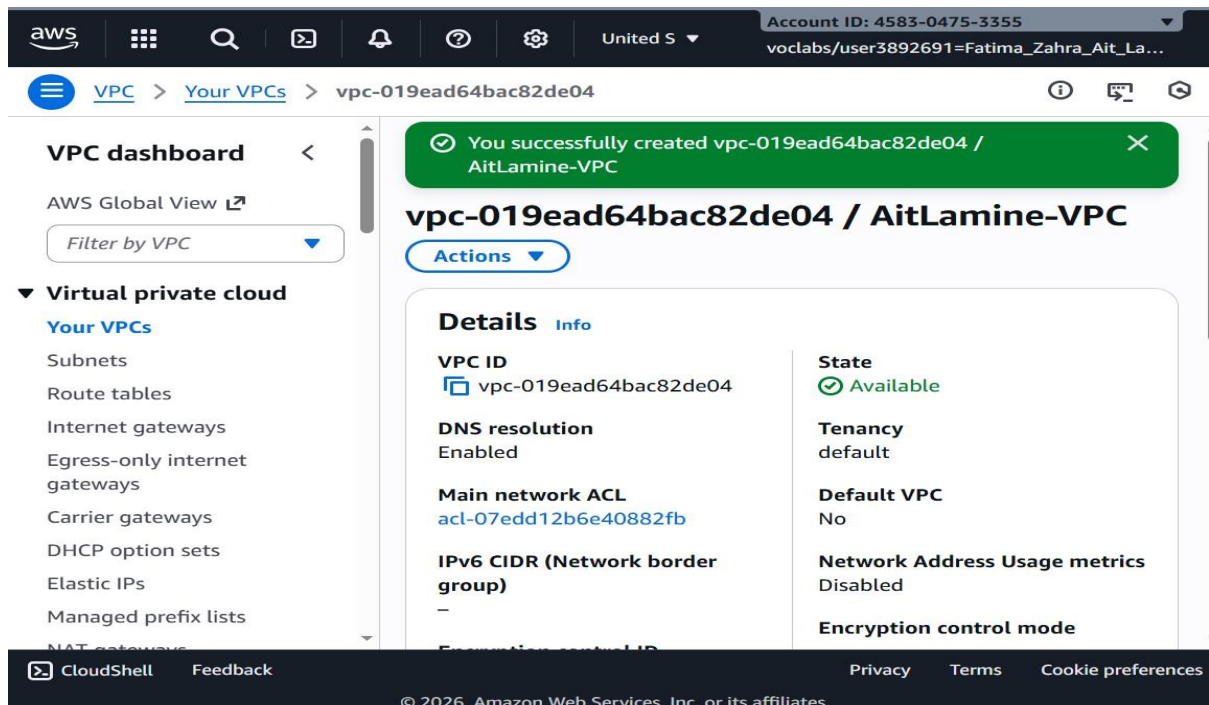


Figure 1 : VPC créé pour le projet (AitLamine-VPC)

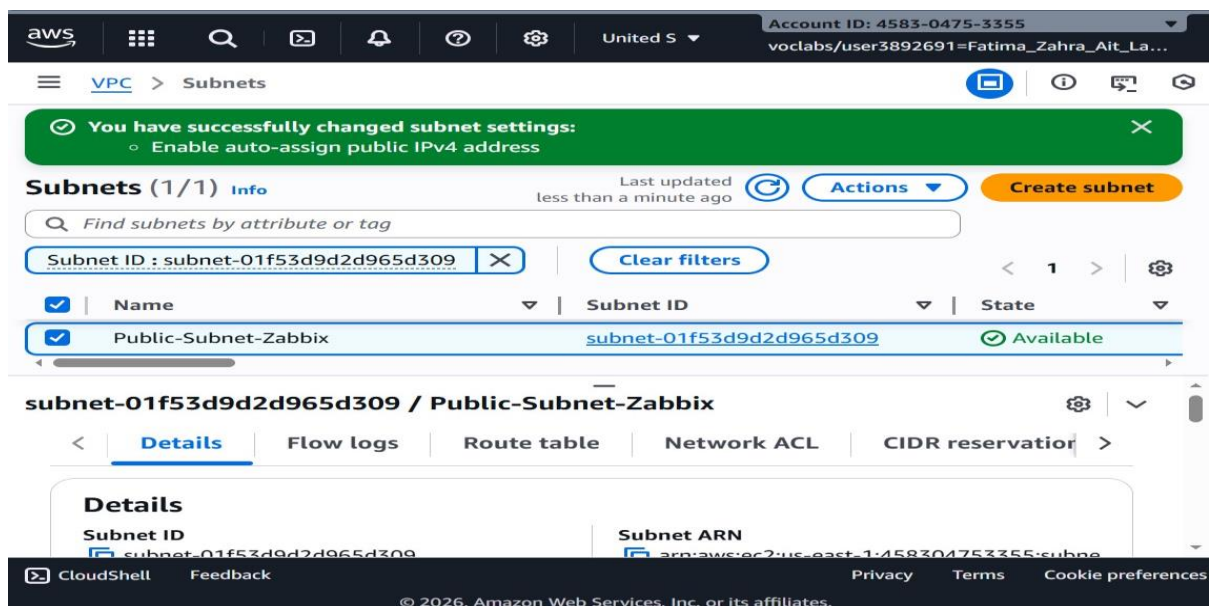


Figure 2 : Sous-réseau public configuré

2.2 Configuration de l'Internet Gateway

Une Internet Gateway a été créée et attachée au VPC pour permettre aux instances de communiquer avec Internet.

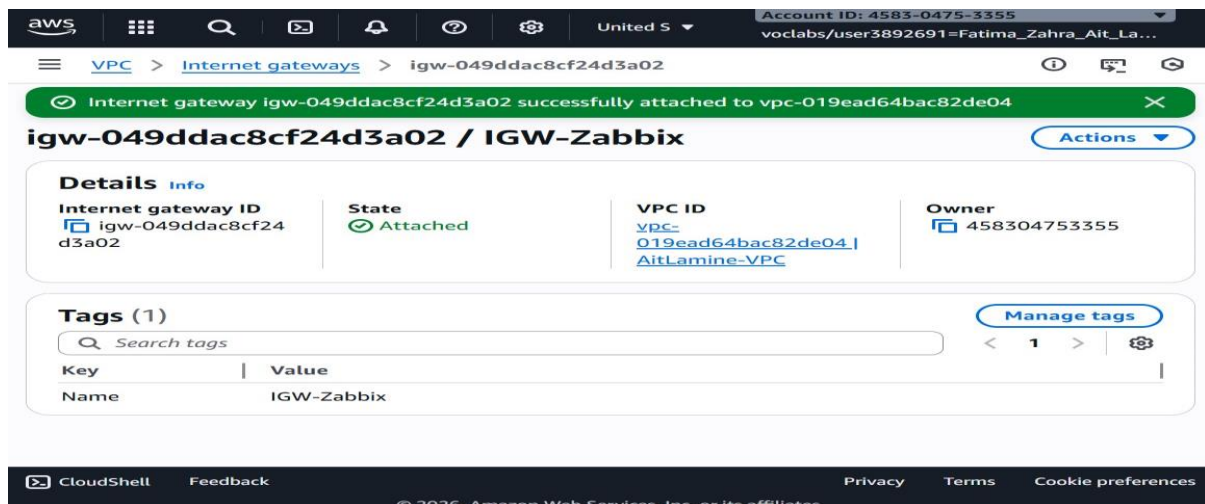


Figure 3 : Internet Gateway attachée au VPC

2.3 Configuration des tables de routage

Une table de routage a été configurée avec une route par défaut (0.0.0.0/0) vers l'Internet Gateway.

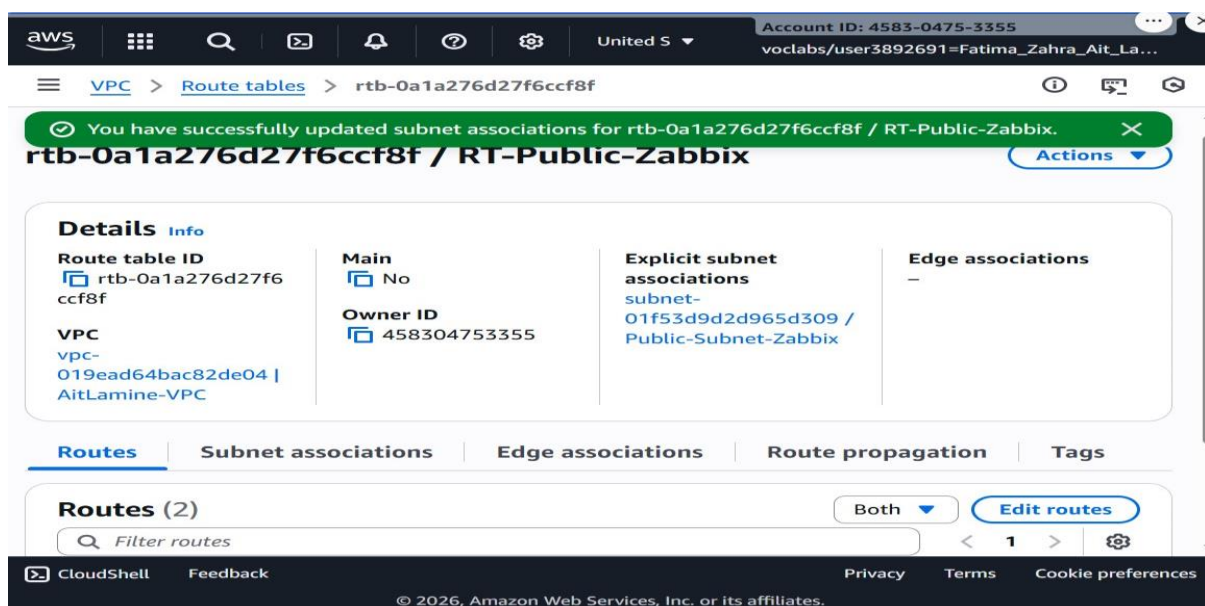


Figure 4 : Table de routage avec route vers Internet Gateway

2.4 Configuration des Security Groups

Des groupes de sécurité ont été configurés pour autoriser uniquement les ports nécessaires :

- Port 80/443 : Interface Web Zabbix
- Port 10050/10051 : Communication Zabbix
- Port 22 : SSH (Linux)
- Port 3389 : RDP (Windows)

The screenshot shows the AWS Management Console interface. At the top, there's a navigation bar with the AWS logo, search, and account information (Account ID: 4583-0475-3355). The breadcrumb trail indicates the path: EC2 > Security Groups > sg-02ae7ffe0dfe61dd4 - SG-Zabbix. A green success message at the top states: "Security group (sg-02ae7ffe0dfe61dd4 | SG-Zabbix) was created successfully". Below this, the title "sg-02ae7ffe0dfe61dd4 - SG-Zabbix" is displayed with an "Actions" dropdown. The "Details" section contains the following information:

Security group name SG-Zabbix	Security group ID sg-02ae7ffe0dfe61dd4	Description Access Zabbix+SSH +RDP	VPC ID vpc-08bf3b28997fe73a4
Owner 458304753355	Inbound rules count 6 Permission entries	Outbound rules count 1 Permission entry	

Below the details, there are tabs for "Inbound rules", "Outbound rules", "Sharing", "VPC associations", and "Tags". The "Inbound rules" tab is selected, showing "Inbound rules (6)". There are buttons for "Manage tags" and "Edit inbound rules". The footer includes "CloudShell", "Feedback", "Privacy", "Terms", and "Cookie preferences".

Figure 5 : Security Group SG-Zabbix créé

This screenshot shows the "Inbound rules" section of the Security Group "SG-Zabbix". It displays a table with 6 rules. The table has columns for Security group rule ID, IP version, Type, Protocol, Port range, and Source. The rules are as follows:

Security group rule ID	IP version	Type	Protocol	Port range	Source
sgr-0d428b831ee7e9616	IPv4	RDP	TCP	3389	0.0.0.0/0
sgr-05f8d5497d7c67000	IPv4	HTTP	TCP	80	0.0.0.0/0
sgr-06e8aacfcfe316ef8	IPv4	HTTPS	TCP	443	0.0.0.0/0
sgr-0a240ceb89e45079f	IPv4	Custom TCP	TCP	10051	0.0.0.0/0
sgr-0828992d174d48acd	IPv4	Custom TCP	TCP	10050	0.0.0.0/0
sgr-003690cba1e0f90b1	IPv4	SSH	TCP	22	0.0.0.0/0

The interface includes a search bar, pagination controls (showing 1 of 6), and buttons for "Manage tags" and "Edit inbound rules". The footer is identical to the previous screenshot.

Figure 6 : Règles entrantes (Inbound rules) du Security Group

3. Architecture des Instances EC2

Trois instances EC2 de type t3.medium ont été déployées pour constituer l'infrastructure de supervision.

3.1 Serveur Zabbix

Caractéristiques :

- Type : t3.medium (2 vCPUs, 4 Go RAM)
- Système : Ubuntu Server 22.04 LTS
- Rôle : Hébergement des conteneurs Docker Zabbix

The screenshot displays the AWS Management Console interface for an EC2 instance. At the top, the navigation bar shows the AWS logo, search, and account information (Account ID: 4583-0475-3355, user: voclabs/user3892691=Fatima_Zahra_Ait_Lamine). The breadcrumb trail indicates the path: EC2 > Instances > i-0c07a092c662d485b. The main content area is titled 'Instance summary for i-0c07a092c662d485b (FatimaZahra-Zabbix-Server)'. It features a 'Connect' button, an 'Instance state' dropdown menu showing 'Running', and an 'Actions' dropdown. Below this, a table-like layout presents various instance details:

Instance ID i-0c07a092c662d485b	Public IPv4 address 34.230.23.25 open address	Private IPv4 addresses 10.0.1.238
IPv6 address -	Instance state Running	Public DNS -
Hostname type IP name: ip-10-0-1-238.ec2.internal	Private IP DNS name (IPv4 only) ip-10-0-1-238.ec2.internal	Elastic IP addresses -
Answer private resource DNS name -	Instance type t3.medium	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address 34.230.23.25 [Public IP]	VPC ID vpc-019ead64bac82de04 (AitLamine-VPC)	Auto Scaling Group name
IAM Role	Subnet ID	

The footer of the console includes links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences, along with the copyright notice: © 2026, Amazon Web Services, Inc. or its affiliates.

Figure 7 : Instance EC2 du serveur Zabbix

3.2 Client Linux

- Type : t3.medium
- Système : Ubuntu Server 22.04 LTS
- Rôle : Machine supervisée via agent Linux

The screenshot displays the AWS Management Console interface for an EC2 instance. The top navigation bar shows the AWS logo, account ID (4583-0475-3355), and region (United States). The breadcrumb trail indicates the path: EC2 > Instances > i-0d3ad60be23a4c75c. The main content area is titled 'Instance summary for i-0d3ad60be23a4c75c (FatimaZahra-Linux-Client)' and includes buttons for 'Connect', 'Instance state', and 'Actions'. Below this, a table lists various instance attributes:

Instance ID i-0d3ad60be23a4c75c	Public IPv4 address 54.164.15.225 open address	Private IPv4 addresses 10.0.1.17
IPv6 address -	Instance state Running	Public DNS -
Hostname type IP name: ip-10-0-1-17.ec2.internal	Private IP DNS name (IPv4 only) ip-10-0-1-17.ec2.internal	Elastic IP addresses -
Answer private resource DNS name -	Instance type t3.medium	AWS Compute Optimizer finding -
Auto-assigned IP address -	VPC ID -	

The footer of the console shows 'CloudShell', 'Feedback', and links for 'Privacy', 'Terms', and 'Cookie preferences'. A copyright notice for Amazon Web Services is also present.

Figure 8 : Instance EC2 du client Linux

3.3 Client Windows

- Type : t3.medium
- Système : Windows Server 2022
- Rôle : Machine supervisée via agent Windows

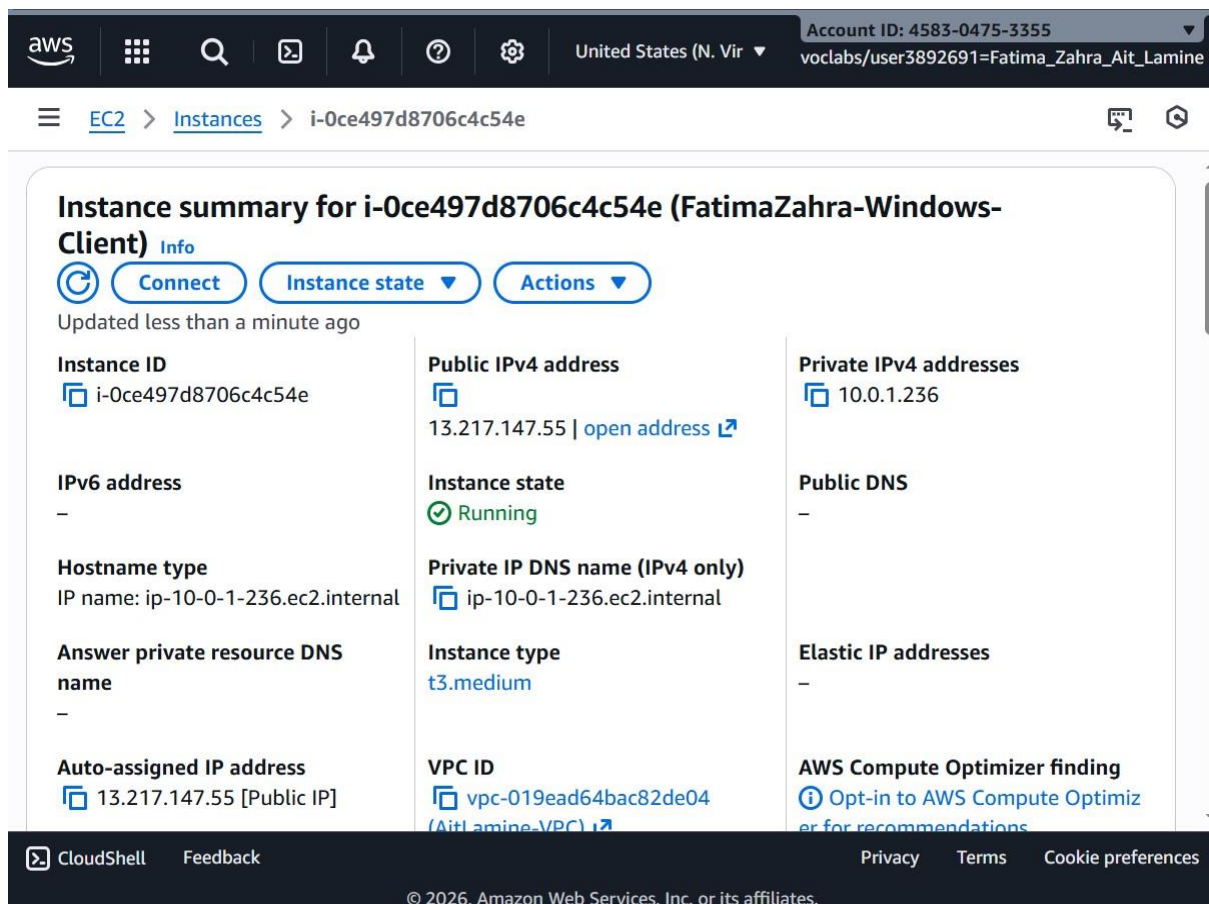


Figure 9 : Instance EC2 du client Windows

4. Déploiement du Serveur Zabbix

4.1 Installation de Docker et Docker Compose

Docker et Docker Compose ont été installés sur le serveur Ubuntu pour déployer Zabbix sous forme de conteneurs.

4.2 Lancement des conteneurs

Les conteneurs Zabbix Server, Web Interface et MySQL ont été lancés avec Docker Compose.

```
ubuntu@ip-10-0-1-238: ~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
d64a27f7f5a1	zabbix/zabbix-web-apache-mysql	"docker-entrypoint.sh"	About a minute ago	Up About a minute (healthy)	8443/tcp, 0.0.0.0:80->8080/tcp, [::]:80->8080/tcp	zabbix-web
ccaa08f932f4	zabbix/zabbix-server-mysql	"/usr/bin/docker-ent..."	About a minute ago	Restarting (1) 4 seconds ago		zabbix-server
9d54cc4055e9	mysql:8.0	"docker-entrypoint.s..."	About a minute ago	Up About a minute	3306/tcp, 33060/tcp	zabbix-mysql

Figure 10 : Conteneurs Docker en cours d'exécution

4.3 Accès à l'interface Web Zabbix

L'interface web Zabbix est accessible via l'IP publique du serveur sur le port 80.

The screenshot shows the Zabbix web interface in the 'Global view' dashboard. The browser address bar indicates the URL: `34.230.23.25/zabbix.php?action=dashboard.view&dashboardid=1&from=now-1h&to=now`. The interface features a sidebar with navigation links, a top navigation bar, and a main content area with various widgets.

Global view

All dashboards / Global view

From: To:

Time range selection:

- Last 2 days
- Last 7 days
- Last 30 days
- Last 3 months
- Last 6 months
- Last 1 year
- Last 2 years
- Yesterday
- Day before yesterday
- This day last week
- Previous week
- Previous month
- Previous year
- Today
- Today so far
- This week
- This month
- This year
- This year so far

Top hosts by CPU utilization

Host name	Utilization	1m avg	5m avg	15m avg	Process
Zabbix server					

System information

Parameter	Value	Details
Zabbix server is running	Yes	zabbix-server:10051
Zabbix server version	7.4.6	
Zabbix frontend version	7.4.6	
Number of hosts (enabled/disabled)	1	1 / 0
Number of templates	357	

Memory utilization

0% 20% 40% 60% 80% 100%

No data

Figure 11 : Interface web Zabbix - Dashboard principal

5. Configuration des Agents Zabbix

5.1 Installation de l'agent Linux

L'agent Zabbix a été installé et configuré sur la machine Ubuntu cliente.

```
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

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

System information as of Thu Jan  1 13:50:09 UTC 2026

System load:  0.08          Temperature:    -273.1 C
Usage of /:   25.8% of 6.71GB Processes:      110
Memory usage: 6%           Users logged in: 0
Swap usage:   0%           IPv4 address for ens5: 10.0.1.218

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

https://ubuntu.com/aws/pro

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-10-0-1-218: $
```

Figure 12 : Connexion SSH au client Linux

```
ubuntu@ip-10-0-1-17: $ sudo systemctl status zabbix-agent
● zabbix-agent.service - Zabbix Agent
   Loaded: loaded (/lib/systemd/system/zabbix-agent.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2026-01-01 14:55:29 UTC; 11s ago
     Docs: man:zabbix_agentd
   Main PID: 2146 (zabbix_agentd)
    Tasks: 6 (limit: 4580)
   Memory: 3.3M
      CPU: 12ms
   CGroup: /system.slice/zabbix-agent.service
           └─2146 /usr/sbin/zabbix_agentd --foreground
             └─2149 "/usr/sbin/zabbix_agentd: collector [idle 1 sec]"
               └─2150 "/usr/sbin/zabbix_agentd: listener #1 [waiting for connection]"
                 └─2151 "/usr/sbin/zabbix_agentd: listener #2 [waiting for connection]"
                   └─2152 "/usr/sbin/zabbix_agentd: listener #3 [waiting for connection]"
                     └─2153 "/usr/sbin/zabbix_agentd: active checks #1 [idle 1 sec]"

Jan 01 14:55:29 ip-10-0-1-17 systemd[1]: Started Zabbix Agent.
Jan 01 14:55:29 ip-10-0-1-17 zabbix_agentd[2146]: Starting Zabbix Agent [Linux-Client]. Zabbix 5.0.17 (revision 0a4
Jan 01 14:55:29 ip-10-0-1-17 zabbix_agentd[2146]: Press Ctrl+C to exit.
lines 1-19/19 (END)
```

Figure 13 : Service Zabbix Agent actif sur Linux

5.2 Installation de l'agent Windows

L'agent Zabbix Windows a été installé via l'assistant MSI.

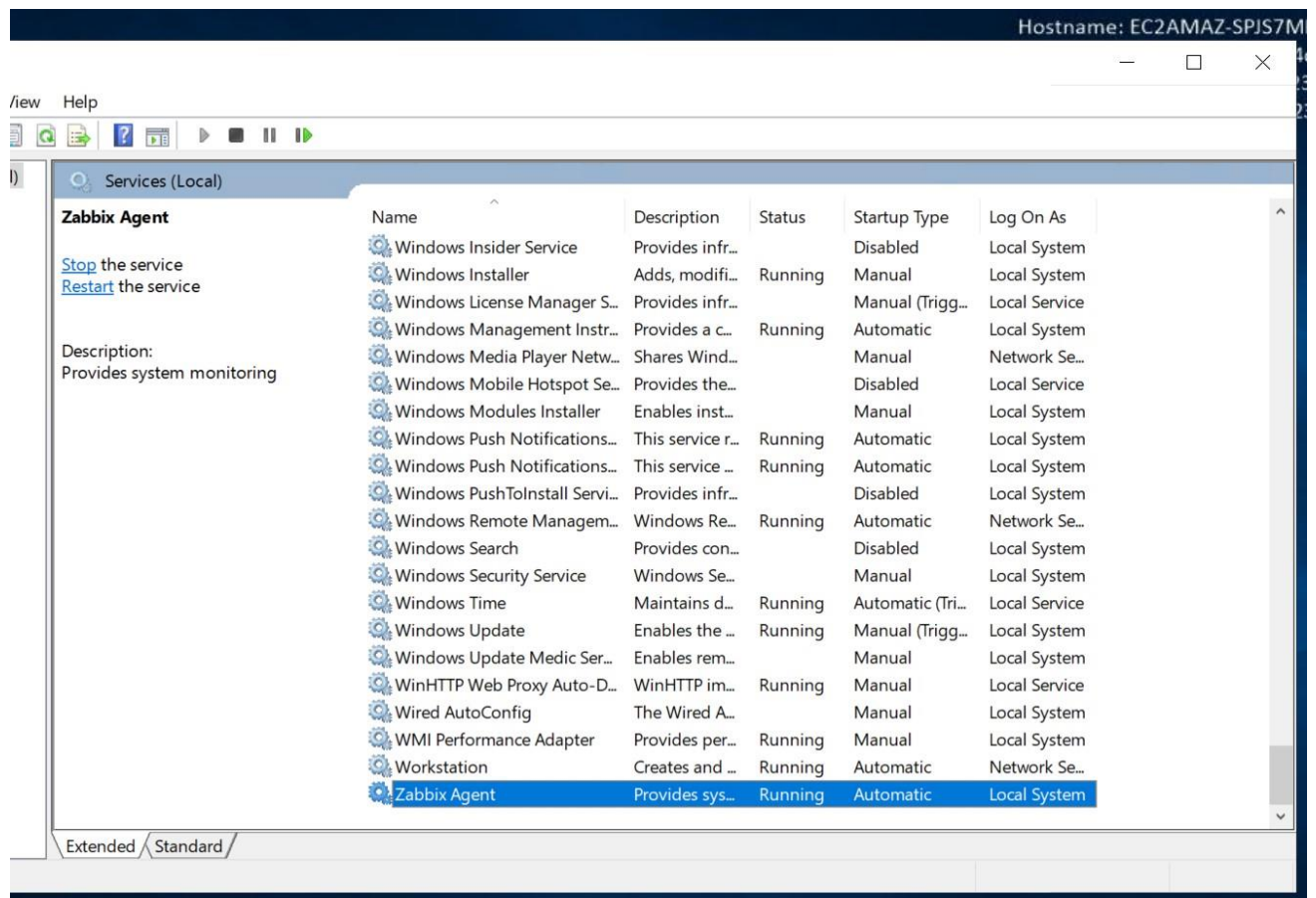


Figure 14 : Service Zabbix Agent fonctionnel sur Windows

6. Monitoring et Tableaux de Bord

6.1 Ajout des hôtes dans Zabbix

Les machines Linux et Windows ont été ajoutées comme hôtes dans l'interface Zabbix avec les templates appropriés.

Name ▲	Interface	Availability	Tags	Status	Latest data	Problems	Graphs
Linux-Client	10.0.1.17:10050	ZBX	class: os target: linux	Enabled	Latest data 43	Problems	Graphs 8

Figure 15 : Client Linux ajouté et disponible

Name ▲	Interface	Availability	Tags	Status	Latest data
Linux-Client	10.0.1.17:10050	ZBX	class: os target: linux	Enabled	Latest data 68
Windows-Client	10.0.1.236:10050	ZBX	class: os target: windows	Enabled	Latest data 100

Figure 16 : Les deux clients (Linux et Windows) supervisés

6.2 Visualisation des données en temps réel

L'interface Zabbix permet de visualiser toutes les métriques collectées en temps réel.

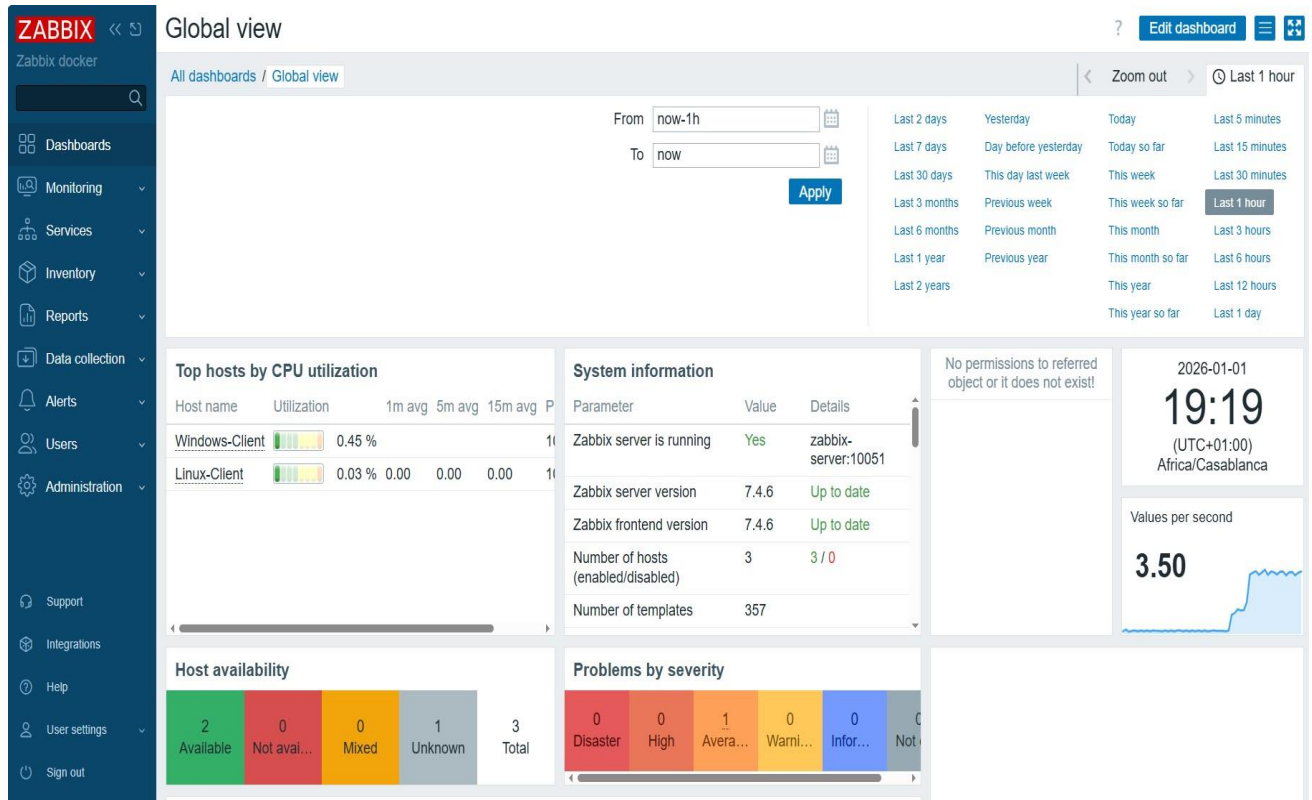


Figure 17 : Dashboard avec monitoring en temps réel des 3 hôtes

ZABBIX

Zabbix docker

Dashboards

Monitoring

Problems

Hosts

Latest data

Maps

Discovery

Services

Inventory

Reports

Data collection

Alerts

Users

Administration

Support

Integrations

Help

User settings

Sign out

DATA

With data Without data

<input type="checkbox"/>	Host	Name ▲	Last check	Last value	Change	Tags	Info
<input type="checkbox"/>	Windows-Client	0 C.: Average disk read queue length ?	9s	0.000101	+0.000101	component: storage disk: 0 C.	Graph
<input type="checkbox"/>	Windows-Client	0 C.: Average disk write queue length ?	6s	0.001179	+0.000685	component: storage disk: 0 C.	Graph
<input type="checkbox"/>	Windows-Client	0 C.: Disk average queue size (avgqu-sz) ?	5s	0		component: storage disk: 0 C.	Graph
<input type="checkbox"/>	Windows-Client	0 C.: Disk read rate ?	4s	0.1477 r/s	+0.1477 r/s	component: storage disk: 0 C.	Graph
<input type="checkbox"/>	Windows-Client	0 C.: Disk read request avg waiting time ?	8s	0.022ms	+0.022ms	component: storage disk: 0 C.	Graph
<input type="checkbox"/>	Windows-Client	0 C.: Disk utilization by idle time ?	10s	0.1118 %	+0.06948 %	component: storage disk: 0 C.	Graph
<input type="checkbox"/>	Windows-Client	0 C.: Disk write rate ?	3s	1.2636 w/s	+0.722 w/s	component: storage disk: 0 C.	Graph
<input type="checkbox"/>	Windows-Client	0 C.: Disk write request avg waiting time ?	7s	0.28ms	+0.042ms	component: storage disk: 0 C.	Graph
<input type="checkbox"/>	Windows-Client	Cache bytes ?	8s	40.57 MB	+768 KB	component: memory	Graph
<input type="checkbox"/>	Windows-Client	Context switches per second ?	58s	217.604	-13.7589	component: cpu	Graph
<input type="checkbox"/>	Windows-Client	CPU DPC time ?	2s	0 %		component: cpu	Graph
<input type="checkbox"/>	Windows-Client	CPU interrupt time ?	1m 1s	0 %		component: cpu	Graph
<input type="checkbox"/>	Windows-Client	CPU privileged time ?	1m	0 %		component: cpu	Graph
<input type="checkbox"/>	Windows-Client	CPU queue length ?	57s	0		component: cpu	Graph
<input type="checkbox"/>	Windows-Client	CPU user time ?	59s	0 %		component: cpu	Graph
<input type="checkbox"/>	Windows-Client	CPU utilization ?	54s	0.1419 %	-0.01395 %	component: cpu	Graph
<input type="checkbox"/>	Windows-Client	Free swap space ?	49s	1.18 GB		component: memory component: storage	Graph
<input type="checkbox"/>	Windows-Client	Free swap space in % ?	3s	86.0793 %	+0.000277 %	component: memory component: storage	Graph
<input type="checkbox"/>	Windows-Client	Free system page table entries ?	7s	12299046	-51	component: memory	Graph
<input type="checkbox"/>	Windows-Client	FS [\\#SLABEL\\C.): Get data ?	45s	[\\#sname"C:","byt...		component: raw component: storage filesystem: C: ***	History
<input type="checkbox"/>	Windows-Client	FS [\\#SLABEL\\C.): Space: Available ?	45s	9.88 GB		component: storage filesystem: C: fstype: NTFS	Graph
<input type="checkbox"/>	Windows-Client	FS [\\#SLABEL\\C.): Space: Total ?	45s	30 GB		component: storage filesystem: C: fstype: NTFS	Graph
<input type="checkbox"/>	Windows-Client	FS [\\#SLABEL\\C.): Space: Used ?	45s	20.12 GB		component: storage filesystem: C: fstype: NTFS	Graph

7. Difficultés Rencontrées et Solutions

7.1 Configuration réseau et Security Groups

Difficulté :

Les agents ne parvenaient pas à communiquer avec le serveur malgré une configuration apparemment correcte.

Solutions appliquées :

- Vérification des règles inbound/outbound des Security Groups
- Utilisation des adresses IP privées AWS pour la communication interne
- Tests de connectivité avec telnet et netcat

7.2 Communication entre agents et serveur

Difficulté :

Les agents apparaissaient comme déconnectés dans l'interface Zabbix.

Solutions appliquées :

- Vérification de la cohérence des hostnames
- Redémarrage des agents après modification
- Analyse des logs pour identifier les erreurs

7.3 Gestion des conteneurs Docker

Difficulté :

Problèmes de persistance des données lors du redémarrage des conteneurs.

Solutions appliquées :

- Configuration de volumes Docker pour la persistance
- Ajout de depends_on dans docker-compose.yml
- Utilisation de docker logs pour le diagnostic

8. Compétences Développées

8.1 Compétences techniques

Cloud Computing et AWS :

- Conception d'architectures VPC sur AWS
- Gestion des instances EC2
- Configuration des Security Groups

Conteneurisation et Docker :

- Installation et configuration de Docker
- Orchestration avec Docker Compose
- Gestion des volumes et persistance

Monitoring et supervision :

- Configuration de Zabbix Server
- Déploiement d'agents multi-plateformes
- Création de dashboards personnalisés

8.2 Compétences transversales

- **Méthodologie de projet** : Planification et documentation
- **Résolution de problèmes** : Analyse méthodique des erreurs
- **Autonomie** : Recherche de documentation

9. Conclusion

Ce projet a permis de mettre en œuvre avec succès une infrastructure complète de supervision centralisée dans le cloud AWS. L'utilisation de Zabbix déployé sous forme de conteneurs Docker a démontré l'efficacité de cette approche pour créer une solution de monitoring flexible et évolutive.

9.1 Objectifs atteints

- Déploiement réussi d'une architecture réseau sécurisée
- Installation de Zabbix en mode conteneurisé
- Supervision opérationnelle de machines Linux et Windows
- Collecte de métriques en temps réel

9.2 Perspectives d'évolution

- **Haute disponibilité** : Cluster Zabbix avec load balancing
- **Sécurité renforcée** : VPN, SSL/TLS, authentification MFA
- **Automatisation** : Terraform, Ansible pour le déploiement
- **Monitoring applicatif** : Extension aux applications et bases de données

9.3 Conclusion personnelle

Ce projet a permis de développer une expertise technique solide dans le domaine de la supervision d'infrastructures cloud. L'expérience acquise sur AWS, Docker et Zabbix constitue une base solide pour aborder des projets d'infrastructure plus ambitieux.

Les compétences développées sont directement transférables au monde professionnel et ouvrent de nombreuses perspectives de carrière dans l'administration système et le DevOps.

Annexes

Annexe A : Commandes Docker utiles

- `docker-compose up -d` : Lancer les conteneurs
- `docker-compose ps` : Vérifier l'état
- `docker logs [container]` : Consulter les logs

Annexe B : Commandes Zabbix Agent

- `systemctl status zabbix-agent` : Statut du service
- `systemctl restart zabbix-agent` : Redémarrer l'agent

Annexe C : Références

- AWS Documentation : <https://docs.aws.amazon.com/>
- Docker Documentation : <https://docs.docker.com/>
- Zabbix Documentation : <https://www.zabbix.com/documentation/>

Annexe D : Ressources du projet

Repository GitHub du projet :

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

Ce repository contient :

- Configurations Docker Compose complètes
- Scripts de configuration des agents Zabbix
- Documentation détaillée du projet
- Captures d'écran de l'infrastructure déployée

--- Fin du rapport ---