

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

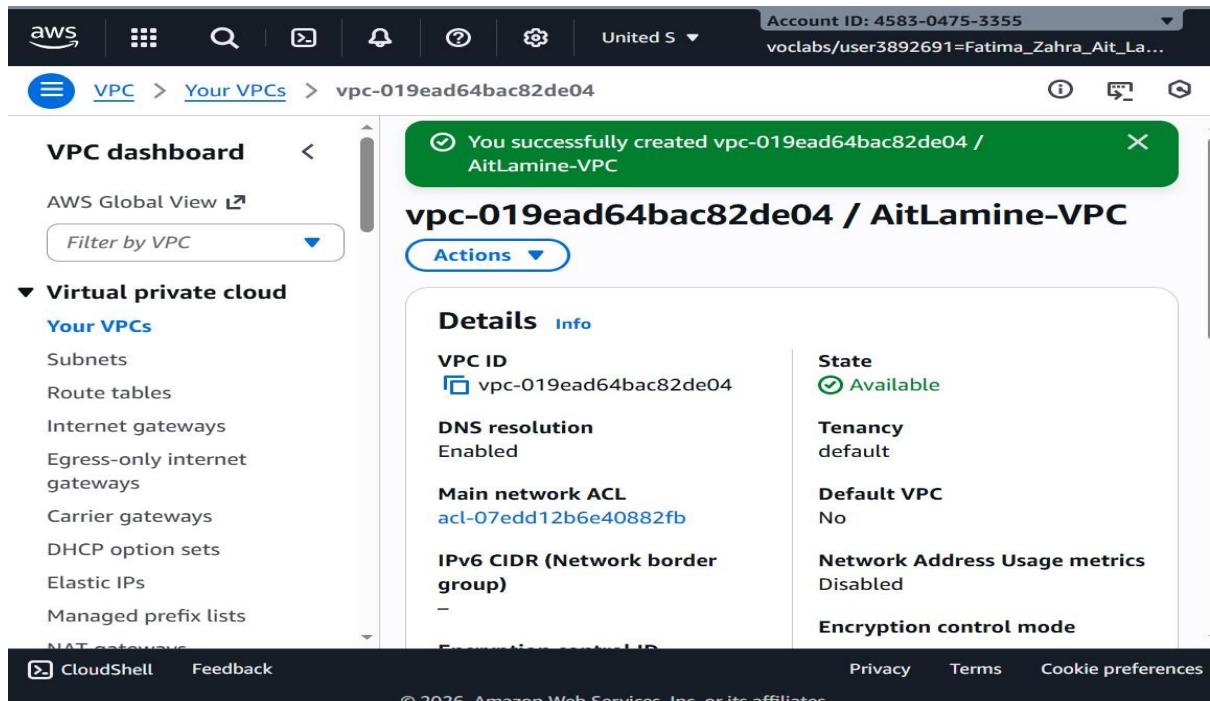


Figure 1 : VPC créé pour le projet (AitLamine-VPC)

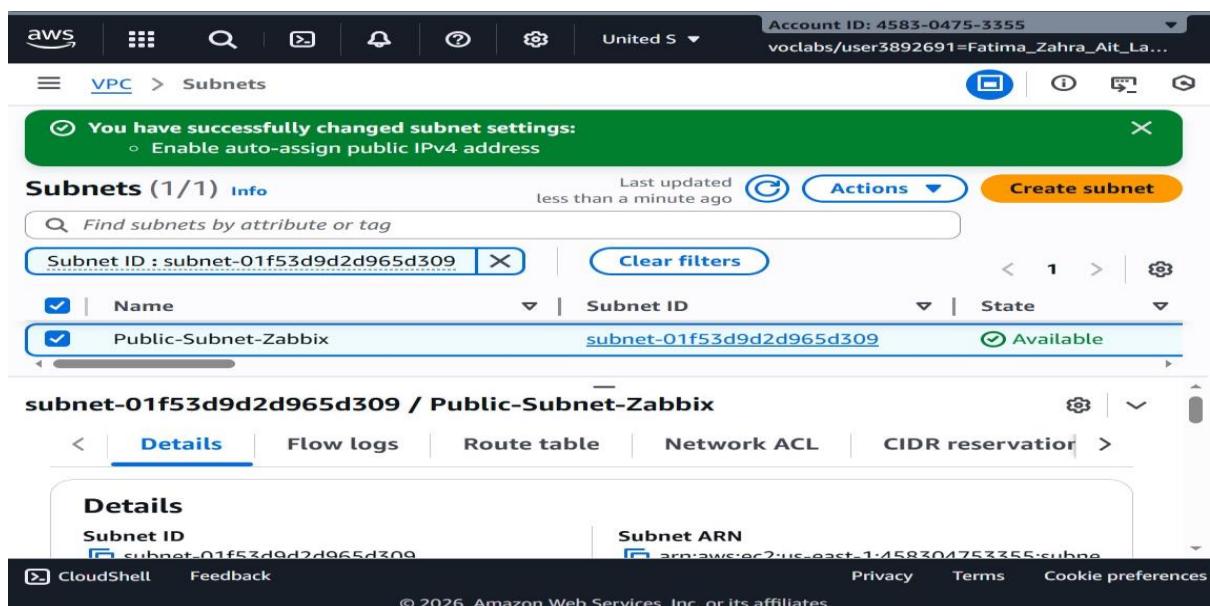


Figure 2 : Sous-réseau public configuré

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.

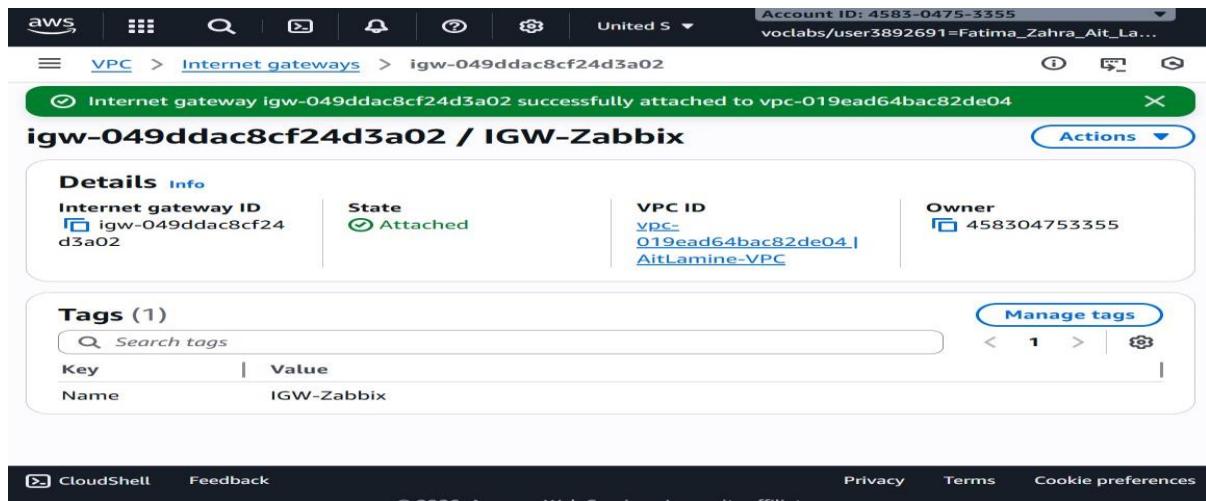


Figure 3 : Internet Gateway attachée au VPC

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.

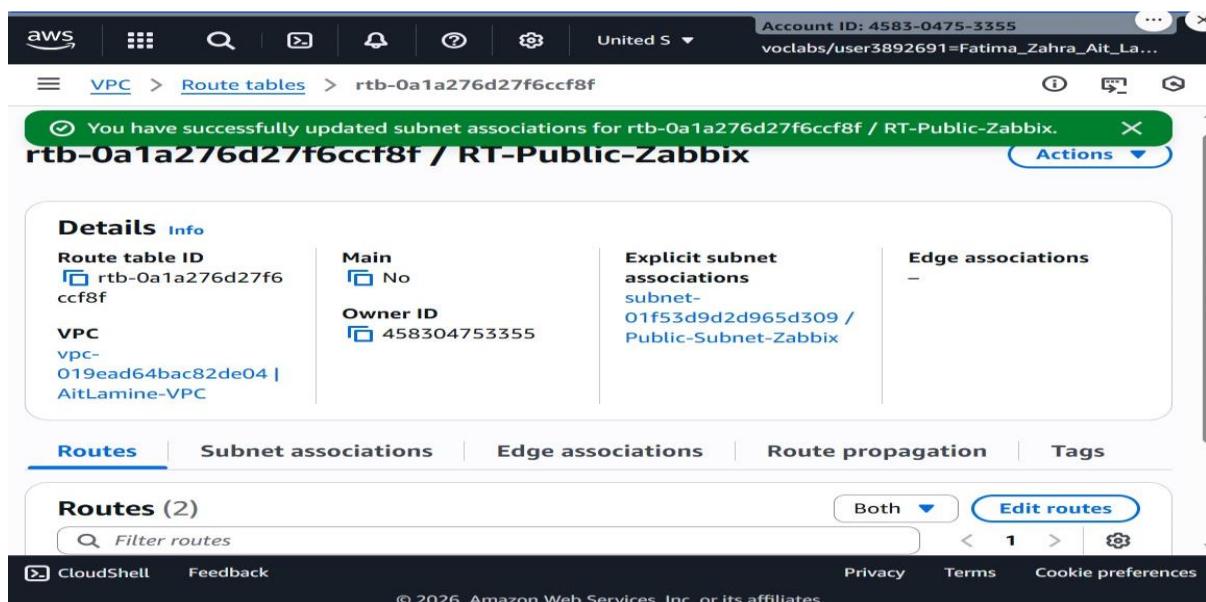


Figure 4 : Table de routage avec route vers Internet Gateway

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 EC2 Security Groups interface. At the top, a green success message states: "Security group (sg-02ae7ffe0dfe61dd4 | SG-Zabbix) was created successfully". Below this, the security group details are listed:

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 6 entries. A sub-table titled "Inbound rules (6)" lists the following rules:

Security group rule ID	IP version	Type	Protocol	Port range	Source
sgr-0d428b831ee7e9616	IPv4	RDP	TCP	3389	0.0.0.
sgr-05f8d5497d7c67000	IPv4	HTTP	TCP	80	0.0.0.
sgr-06e8aacfcf316ef8	IPv4	HTTPS	TCP	443	0.0.0.
sgr-0a240ceb89e45079f	IPv4	Custom TCP	TCP	10051	0.0.0.
sgr-0828992d174d48acd	IPv4	Custom TCP	TCP	10050	0.0.0.
sgr-003690cba1e0f90b1	IPv4	SSH	TCP	22	0.0.0.

Figure 5 : Security Group SG-Zabbix créé

The screenshot shows the detailed view of the "SG-Zabbix" security group. At the top, a green success message states: "Security group (sg-02ae7ffe0dfe61dd4 | SG-Zabbix) was created successfully". Below this, the security group details are listed:

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 6 entries. A sub-table titled "Inbound rules (6)" lists the following rules:

Security group rule ID	IP version	Type	Protocol	Port range	Source
sgr-0d428b831ee7e9616	IPv4	RDP	TCP	3389	0.0.0.
sgr-05f8d5497d7c67000	IPv4	HTTP	TCP	80	0.0.0.
sgr-06e8aacfcf316ef8	IPv4	HTTPS	TCP	443	0.0.0.
sgr-0a240ceb89e45079f	IPv4	Custom TCP	TCP	10051	0.0.0.
sgr-0828992d174d48acd	IPv4	Custom TCP	TCP	10050	0.0.0.
sgr-003690cba1e0f90b1	IPv4	SSH	TCP	22	0.0.0.

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 shows the AWS EC2 Instance Summary page for the instance **i-0c07a092c662d485b (FatimaZahra-Zabbix-Server)**. The instance is currently running. Key details displayed include:

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

At the bottom of the page, there are links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

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 shows the AWS EC2 Instances page for an instance named "i-0d3ad60be23a4c75c". The instance summary table provides the following details:

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

At the bottom of the page, there are links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

Figure 8 : Instance EC2 du client Linux

3.3 Client Windows

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

The screenshot shows the AWS EC2 Instances page for an instance named 'i-0ce497d8706c4c54e'. The instance is labeled '(FatimaZahra-Client)' and is in the 'Running' state. Key details include:

- Instance ID:** i-0ce497d8706c4c54e
- Public IPv4 address:** 13.217.147.55
- Private IPv4 addresses:** 10.0.1.236
- IPv6 address:** -
- Instance state:** Running
- Public DNS:** -
- Hostname type:** IP name: ip-10-0-1-236.ec2.internal
- Private IP DNS name (IPv4 only):** ip-10-0-1-236.ec2.internal
- Answer private resource DNS name:** -
- Instance type:** t3.medium
- Elastic IP addresses:** -
- Auto-assigned IP address:** 13.217.147.55 [Public IP]
- VPC ID:** vpc-019ead64bac82de04 (AitLamine-VPC)
- AWS Compute Optimizer finding:** Opt-in to AWS Compute Optimizer for recommendations.

At the bottom, there are links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences. A copyright notice for 2026, Amazon Web Services, Inc. or its affiliates, is also present.

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
ccaa8f932f4 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 with the following details:

- Header:** Shows the URL as "Not secure 34.230.23.25/zabbix.php?action=dashboard.view&dashboardid=1&from=now-1h&to=now".
- Left Sidebar:** A dark sidebar with the ZABBIX logo at the top, followed by navigation links: Dashboards, Monitoring (Services, Inventory), Reports, Data collection, Alerts, Users, Administration, Support, Integrations, and Help.
- Top Bar:** Includes a search bar and various browser icons.
- Dashboard Content:**
 - Global view:** A title bar with "All dashboards / Global view".
 - Time Range:** From "now-1h" to "now" with an "Apply" button.
 - Time Selection:** A dropdown menu for time ranges: Last 2 days, Yesterday, Today, Last 7 days, Day before yesterday, Today so far, Last 30 days, This day last week, This week, Last 3 months, Previous week, This week so far, Last 6 months, Previous month, This month, Last 1 year, Previous year, This month, Last 2 years, This year, and This year so far.
 - Host Utilization:** A table titled "Top hosts by CPU utilization" showing "Zabbix server" with utilization values: 1m avg, 5m avg, 15m avg, and Process.
 - System Information:** A table showing system parameters and their values, including "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 / 0), and "Number of templates" (357).
 - Memory Utilization:** A donut chart titled "Memory utilization" showing "No data" across all segments (0%, 20%, 40%, 60%, 80%, 100%).

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(8)
   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

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.

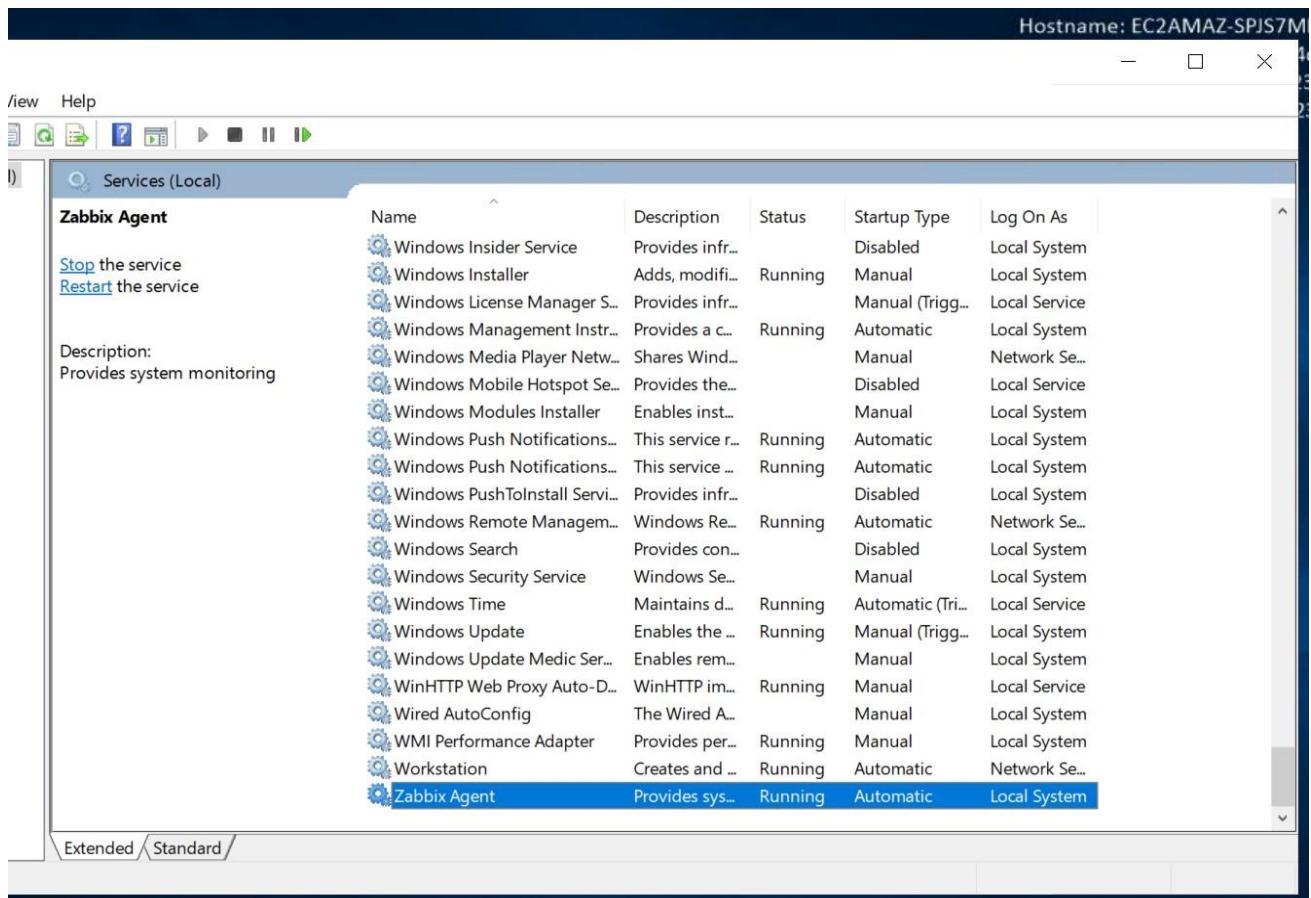


Figure 14 : Service Zabbix Agent fonctionnel sur Windows

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 0	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.

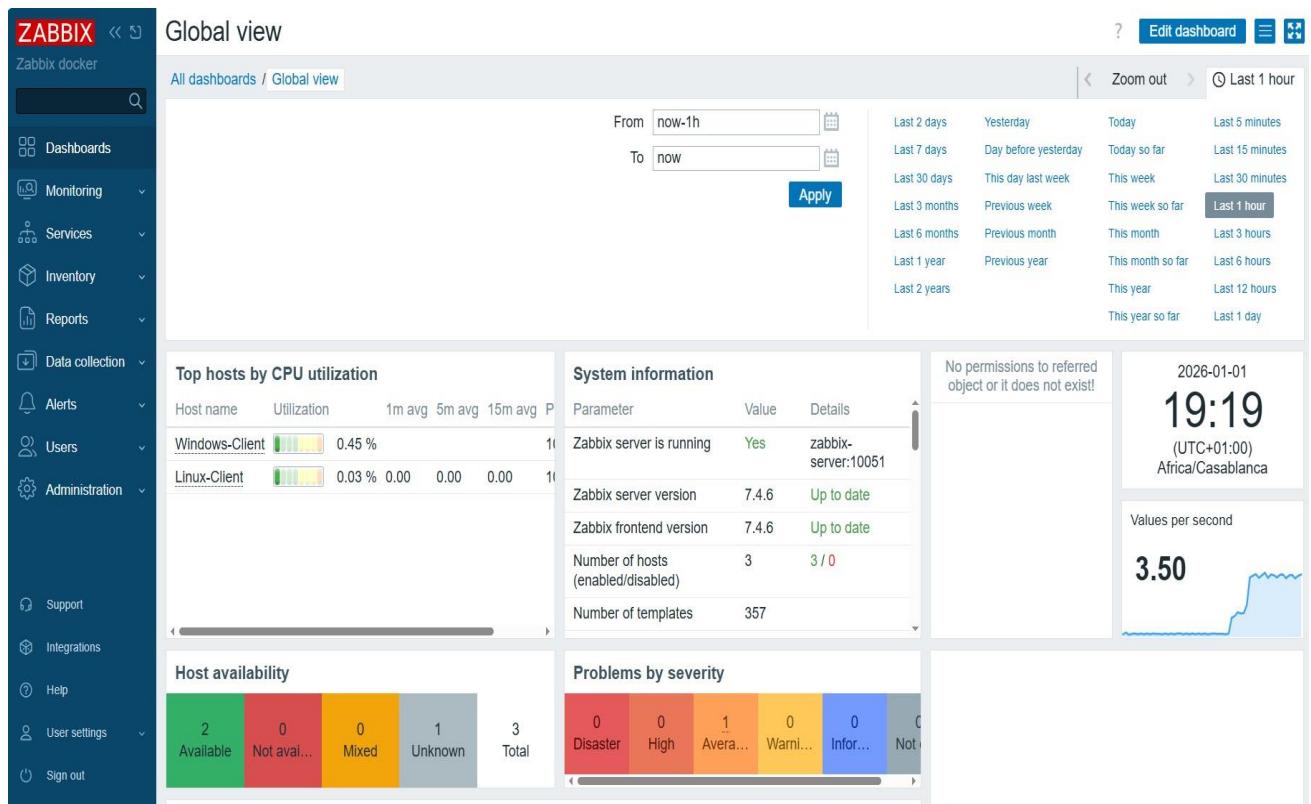


Figure 17 : Dashboard avec monitoring en temps réel des 3 hôtes

DATA						
	Name	Last check	Last value	Change	Tags	Info
	Host					
	Windows-Client	0 C:\ Average disk read queue length	9s	0.000101	+0.000101 component: storage disk: 0 C:\	Graph
	Windows-Client	0 C:\ Average disk write queue length	8s	0.001179	+0.000685 component: storage disk: 0 C:\	Graph
	Windows-Client	0 C:\ Disk average queue size (avgqu-sz)	5s	0	component: storage disk: 0 C:\	Graph
	Windows-Client	0 C:\ Disk read rate	4s	0.1477 r/s	+0.1477 r/s component: storage disk: 0 C:\	Graph
	Windows-Client	0 C:\ Disk read request avg waiting time	8s	0.022ms	+0.022ms component: storage disk: 0 C:\	Graph
	Windows-Client	0 C:\ Disk utilization by idle time	10s	0.1118 %	+0.06948 % component: storage disk: 0 C:\	Graph
	Windows-Client	0 C:\ Disk write rate	3s	1.2636 w/s	+0.722 w/s component: storage disk: 0 C:\	Graph
	Windows-Client	0 C:\ Disk write request avg waiting time	7s	0.28ms	+0.042ms component: storage disk: 0 C:\	Graph
	Windows-Client	Cache bytes	8s	40.57 MB	+768 KB component: memory	Graph
	Windows-Client	Context switches per second	58s	217.604	-13.7589 component: cpu	Graph
	Windows-Client	CPU DPC time	2s	0 %	component: cpu	Graph
	Windows-Client	CPU interrupt time	1m 1s	0 %	component: cpu	Graph
	Windows-Client	CPU privileged time	1m	0 %	component: cpu	Graph
	Windows-Client	CPU queue length	57s	0	component: cpu	Graph
	Windows-Client	CPU user time	59s	0 %	component: cpu	Graph
	Windows-Client	CPU utilization	54s	0.1419 %	-0.01395 % component: cpu	Graph
	Windows-Client	Free swap space	49s	1.18 GB	component: memory component: storage	Graph
	Windows-Client	Free swap space in %	3s	86.0793 %	+0.000277 % component: memory component: storage	Graph
	Windows-Client	Free system page table entries	7s	12299046	-51 component: memory	Graph
	Windows-Client	FS #[#FSLABEL](C:).Get data	45s	{"fsname": "C:", "byte...}	component: raw component: storage filesystem: C: ...	History
	Windows-Client	FS #[#FSLABEL](C:).Space.Available	45s	9.88 GB	component: storage filesystem: C: fstype: NTFS	Graph
	Windows-Client	FS #[#FSLABEL](C:).Space.Total	45s	30 GB	component: storage filesystem: C: fstype: NTFS	Graph
	Windows-Client	FS #[#FSLABEL](C:).Space.Used	45s	20.12 GB	component: storage filesystem: C: fstype: NTFS	Graph
	Windows-Client	FS #[#FSLABEL](C:).Space.Used_in %	45s	67.04%	component: storage filesystem: C: fstype: NTFS	Graph

Figure 18 : Métriques détaillées du client Windows

DATA						
	Name	Last check	Last value	Change	Tags	Info
	Host					
	Linux-Client	Available memory	5s	3.3 GB	+16 KB component: memory	Graph
	Linux-Client	Available memory in %	4s	88.1065 %	+0.000407 % component: memory	Graph
	Linux-Client	Checksum of /etc/passwd			component: security	History
	Linux-Client	Context switches per second	29s	42.9161	-0.8842 component: cpu	Graph
	Linux-Client	CPU guest nice time	28s	0 %	component: cpu	Graph
	Linux-Client	CPU guest time	27s	0 %	component: cpu	Graph
	Linux-Client	CPU idle time	26s	99.9667 %	+0.01666 % component: cpu	Graph
	Linux-Client	CPU interrupt time	25s	0 %	component: cpu	Graph
	Linux-Client	CPU iowait time	24s	0 %	component: cpu	Graph
	Linux-Client	CPU nice time	23s	0 %	component: cpu	Graph
	Linux-Client	CPU softirq time	22s	0 %	-0.008335 % component: cpu	Graph
	Linux-Client	CPU steal time	21s	0.01667 %	+0.000005 % component: cpu	Graph
	Linux-Client	CPU system time	20s	0.008336 %	-0.008333 % component: cpu	Graph
	Linux-Client	CPU user time	19s	0.008336 %	+0.000001 % component: cpu	Graph
	Linux-Client	CPU utilization	26s	0.03334 %	-0.01666 % component: cpu	Graph
	Linux-Client	Free swap space	13s	0 B	component: memory component: storage	Graph
	Linux-Client	Free swap space in %	12s	100 %	component: memory component: storage	Graph
	Linux-Client	FS [/].Get data	6s	{"fsname": "/", "byte...}	component: raw component: storage filesystem: / ...	History
	Linux-Client	FS [/].Inodes_Free.in %	6s	92.645 %	component: storage filesystem: / fstype: ext4	Graph
	Linux-Client	FS [/].Option: Read-only			component: storage filesystem: / fstype: ext4	Graph
	Linux-Client	FS [/].Space: Available	6s	5.36 GB	component: storage filesystem: / fstype: ext4	Graph
	Linux-Client	FS [/].Space: Total	6s	7.57 GB	component: storage filesystem: / fstype: ext4	Graph
	Linux-Client	FS [/].Space: Used	6s	2.2 GB	component: storage filesystem: / fstype: ext4	Graph
	Linux-Client	FS [/].Space: Used, in %	6s	29.0971 %	component: storage filesystem: / fstype: ext4	Graph

Figure 19 : Métriques détaillées du client Linux

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 ---