

Projet : 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)

1. Cahier des Charges Technique

Objectif

Déployer une infrastructure de monitoring centralisée sur AWS en utilisant Zabbix (Docker) pour surveiller un parc hybride (Linux & Windows).

Architecture Proposée (Respectant les limitations Lab)

- **VPC** : 1 VPC avec un sous-réseau public (pour simplifier l'accès sans VPN).
- **Sécurité** : Groupes de sécurité (Security Groups) autorisant le port 80/443 (Zabbix Web), 10050/10051 (Agents Zabbix) et 3389 (RDP) / 22 (SSH).
- **Instances EC2** :
 1. **Serveur Zabbix** : Instance t3.large (recommandé pour Docker/Zabbix) sous Ubuntu.
 2. **Client Linux** : Instance t3. medium (Ubuntu).
 3. **Client Windows** : Instance t3.large (Windows Server nécessite au moins 4Go de RAM pour être fluide).

2. Structure du Compte Rendu (Livrable PDF)

Le rapport doit être **très clair, bien présenté et bien structuré**.

Il doit obligatoirement contenir une **page de garde** comportant les éléments suivants :

- Le **logo de l'établissement**
- Le **titre du projet**
- Le **nom de l'étudiant**
- Le **nom de l'encadrant** : Prof. Azeddine KHIAT
- **Année universitaire** : 2025/2026
- La **filière**

Le rapport doit être **riche en captures d'écran** afin d'illustrer les différentes étapes du travail réalisé.

Captures d'écran

- Chaque capture d'écran doit être **numérotée** et accompagnée d'une **légende**

descriptive placée sous l'image.

- Exemple :
 - **Figure 1 : Création du VPC**
 - **Figure 2 : Configuration du Subnet**

🔗 Dépôt GitHub (obligatoire pour valorisation)

Pour une meilleure valorisation du projet, il est obligatoire de créer un **dépôt GitHub** contenant :

- La documentation du projet (README détaillé)
- Les fichiers de configuration (Docker, scripts, fichiers Zabbix...)
- Les étapes d'installation et de déploiement
- Toute ressource utile (diagrammes, captures, etc.)

Le lien du dépôt GitHub doit être **inséré clairement dans le compte rendu** (ex : en introduction ou en fin de rapport).

Sommaire

1. **Introduction** : Présentation du projet et des outils (AWS, Docker, Zabbix).
2. **Architecture Réseau** : Schéma du VPC et configuration des Security Groups.
Capture d'écran
3. **Architecture des Instances EC2** :
Serveur Zabbix : Instance t3.large (recommandé pour Docker/Zabbix) sous Ubuntu.
Client Linux : Instance t3. medium (Ubuntu).
Client Windows : Instance t3.large (Windows Server nécessite au moins 4Go de RAM pour être fluide).
Capture d'écran
4. **Déploiement du Serveur Zabbix** :
 - Installation de Docker et Docker-Compose.
 - Lancement des conteneurs (Zabbix Server, Web Interface, DB).
 - *Capture d'écran* : Interface de connexion Zabbix réussie.
5. **Configuration des Clients (Agents)** :
 - Installation de l'agent sur la machine Linux.
 - Installation de l'agent sur la machine Windows.
 - *Capture d'écran* : Configuration du fichier zabbix_agentd.conf.
6. **Monitoring et Tableaux de Bord** :
 - Ajout des hôtes dans l'interface Zabbix.
 - *Capture d'écran* : Statut "Vert" (ZBX) des deux clients.
 - *Capture d'écran* : Graphique de charge CPU ou RAM d'un client.
7. **Conclusion** : Difficultés rencontrées et solutions apportées.

3. Conseils pour les Captures d'Écran

Pour obtenir une note d'excellence, vos captures doivent prouver que **c'est bien vous** qui avez fait le travail :

- **Identité** : Essayez d'inclure votre nom d'utilisateur AWS ou un tag spécifique (ex: NomEtudiant-Zabbix-Server) visible dans la console AWS.
- **Qualité** : Utilisez des outils comme l'outil "Capture" de Windows ou "Snipping Tool". Évitez les photos de l'écran avec un téléphone.
- **Annotations** : Entourez en rouge les éléments importants sur la capture (l'adresse IP publique, le statut de l'agent, etc.).

4. Préparation de la Séquence Vidéo (Présentation)

La vidéo doit être synthétique (5 à 10 minutes maximum).

- **Introduction**: Montrez votre visage (si possible) et présentez l'objectif sur votre page de garde.
- **Démonstration AWS**: Montrez rapidement vos 3 instances en état "Running" dans la console AWS.
- **Cœur de la démo**:
 1. Connectez-vous à l'interface Web de Zabbix.
 2. Allez dans **Configuration > Hosts**. Montrez que les agents Linux et Windows sont connectés.
 3. Allez dans **Monitoring > Latest Data**. Montrez que vous recevez des données en temps réel (ex: utilisation CPU du client Windows).
 4. Provoquez une petite alerte (ex: un ping test) pour montrer que Zabbix réagit.
- **Conclusion** : Résumé des acquis.

5. Limitations Critiques du Learner Lab à surveiller

- **Type d'instances** : Restez sur du t3.medium ou t3.large Ne tentez pas de grosses instances, elles sont bloquées.
- **Région** : Utilisez uniquement us-east-1 (N. Virginia), c'est la région la plus stable pour les Labs.
- **Arrêt automatique** : Le Lab s'arrête après un certain temps. **Pensez à redémarrer vos conteneurs Docker** (docker-compose up -d) si vous relancez l'instance après une pause.
- **Budget** : Surveillez votre barre de consommation. Éteignez vos instances (Stop) quand

vous ne travaillez pas pour économiser les 50\$.