BTS Services informatiques aux organisations - SISR

Session 2026

E4 – Support et mise à disposition de services informatiques

Coefficient 4

DESCRIPTION DE LA REALISATION PROFESSIONNELLE

NOM et prénom du candidat : Aoudani Noeh	

Contexte de la réalisation professionnelle

L'infrastructure actuelle repose sur un serveur Debian hébergeant plusieurs services conteneurisés (WordPress, Nextcloud) via Docker. Ces services sont exposés sur Internet via le reverse-proxy **Traefik**, qui gère également les certificats SSL Let's Encrypt. Jusqu'à présent, aucune surveillance n'était en place, rendant difficile la détection proactive des pannes de base de données ou de saturation des ressources (CPU/RAM). La problématique était de mettre en place une solution de supervision centralisée, capable de s'intégrer dynamiquement à l'existant sans perturber le routage Traefik. Pour y répondre, j'ai déployé une stack **Zabbix** pour la collecte de données et **Grafana** pour la visualisation.

Intitulé de la réalisation professionnelle

Déploiement d'une solution de supervision Zabbix pour l'entreprise

Période de réalisation : Novembre 2024 **Lieu : Longny-Les-Villages**
Modalité : Individuelle

Principale(s) activité(s) concernée(s) :

- Mise à disposition d'un service de supervision
- Installation et configuration serveur Zabbix (server, frontend, agent)
- Tests et validation des métriques et des alertes
- Documentation et transfert de compétence

Conditions de réalisation
Ressources présentes (état initial) :

- 1 VM dédiée (4 vCPU, 8–16 GB RAM, 20 GB disque) pour le serveur Zabbix
- Serveur de base de données : MariaDB
- Réseau interne (VLAN supervision) accessible depuis agents
- Accès administratif (SSH, console hyperviseur)

Résultats attendus :

- Serveur Zabbix opérationnel avec interface Web accessible
- Agents installés sur au moins 10 hôtes (serveurs et VMs) et remontée des métriques de base (CPU, mémoire, disque, interfaces réseau)
- Alertes configurées pour seuils critiques et tests d'envoi (email/snmp trap) opérationnels

Durée de réalisation : 5 à 10 jours (installation, configuration, tests, documentation)

Modalités d'accès à cette réalisation professionnelle.
https://site.jjba.fr. Compte d'accès : aucun. Mot de passe : BTSsiosisr

Partie 1 – Procédure de mise en œuvre.

Partie 1 - Procédure de mise en œuvre

Cette partie détaille les étapes clés de l'installation et de la configuration de la solution de supervision. L'objectif était d'intégrer Zabbix et Grafana dans une infrastructure existante pilotée par Docker et Traefik.

1. Architecture et environnement technique

La solution est déployée sous forme de micro-services afin d'assurer une haute disponibilité et une facilité de mise à jour. Les composants utilisés sont :

- **Système d'exploitation :** Debian 12 (Host Docker).
- **Orchestration :** Docker Compose.
- **Services :** Zabbix Server (Collecte), Zabbix Web (Interface PHP/Nginx), Grafana (Dashboards) et MySQL 8.0 (Base de données).
- **Réseau :** Intégration au réseau proxy existant pour la communication avec le reverse-proxy Traefik.

2. Déploiement de la stack via Docker Compose

Le déploiement a été réalisé en créant un fichier docker-compose.yml structuré pour répondre aux contraintes de routage de Traefik.

- **Configuration MySQL :** Utilisation de l'image officielle MySQL 8.0. Une commande spécifique a été ajoutée pour forcer le charset (--character-set-server=utf8mb4 --collation-server=utf8mb4_bin) afin de garantir la compatibilité avec le schéma Zabbix.
- **Gestion des réseaux :** Les conteneurs Web ont été rattachés à deux réseaux : un réseau interne pour communiquer avec la base de données et le réseau proxy pour être vus par Traefik.

3. Publication sécurisée via Traefik

Pour exposer les interfaces de supervision sur les domaines zabbix.jjba.fr et grafana.jjba.fr, j'ai configuré des labels spécifiques :

- **Règles de Host :** Définition des FQDN pour chaque service.
- **Entrypoints :** Forçage du trafic sur le port 443 (HTTPS) avec l'utilisation du certificat SSL généré automatiquement par Let's Encrypt via Traefik.
- **Load Balancing :** Indication précise des ports internes (8080 pour Zabbix-Web, 3000 pour Grafana) via le label traefik.http.services.[service].loadbalancer.server.port.

4. Initialisation de la base de données (Phase délicate)

Lors du premier lancement, le conteneur Zabbix-Web peut renvoyer une erreur si les tables ne sont pas présentes.

- **Procédure d'import :** Extraction du schéma SQL depuis l'image Zabbix et injection manuelle dans le conteneur MySQL via une commande docker exec.
- **Persistance :** Montage de volumes persistants pour /var/lib/mysql (base de données) et /var/lib/grafana (données de visualisation) afin de ne pas perdre la configuration lors d'un redémarrage des conteneurs.

5. Sécurisation de l'infrastructure

La sécurité a été au centre de la mise en œuvre :

- **Isolation des flux :** Le conteneur de base de données (MySQL) n'est pas exposé sur le réseau proxy, il est uniquement accessible par le serveur Zabbix en interne.
- **Gestion des droits :** Pour Grafana, les permissions du volume ont été restreintes à l'utilisateur ID 472 pour respecter le principe de moindre privilège.
- **Chiffrement :** L'intégralité des flux d'administration passe par un tunnel TLS (HTTPS) géré par Traefik, empêchant l'interception des identifiants Admin.

Outils utilisés :

- **Docker / Docker Compose :** Pour l'isolation des services.
- **Traefik v2.11 :** Pour le routage et le SSL.
- **Zabbix 7.0 :** Pour la collecte de données.
- **Portainer (optionnel) :** Pour la surveillance visuelle de l'état des conteneurs.

Partie 2 – Validation.

Cette phase permet d'apporter la preuve que la solution de supervision est totalement opérationnelle et qu'elle répond aux besoins de visibilité de l'infrastructure.

1. Test d'accessibilité et de routage Traefik

La première validation consiste à vérifier que le reverse-proxy redirige correctement les flux HTTPS vers les bons conteneurs.

- **Action :** Accès aux URLs `https://zabbix.jjba.fr` et `https://grafana.jjba.fr`.
- **Résultat :** Les deux interfaces s'affichent avec un certificat SSL valide (Let's Encrypt). Le log de Traefik confirme la détection des "Routers" et des "Services" associés au réseau `proxy`.

2. Vérification de la persistance et de l'état de la base de données

Suite aux difficultés initiales d'import, une vérification directe dans le conteneur de base de données a été effectuée.

- **Commande :** `docker exec -it zabbix-mysql mysql -uzabbix -pzabbixpass zabbix -e "SELECT COUNT(*) FROM users;"`
- **Preuve :** Le retour de la commande affiche un nombre de tables cohérent (plus de 170 tables après import), confirmant que le schéma SQL a été correctement injecté et que les données survivront au redémarrage des conteneurs.

3. Validation de la remontée des métriques (Zabbix Agent)

Pour valider la collecte, j'ai vérifié la communication entre le **Zabbix Server** et le conteneur **Zabbix Web** (auto-monitoring).

- **Action :** Consultation du menu *Monitoring > Latest Data* dans Zabbix.
- **Observation :** Les graphiques de charge CPU et d'utilisation de la mémoire RAM se remplissent en temps réel. Cela valide que le serveur Zabbix parvient à communiquer avec ses agents via le port interne `10051`.

4. Analyse de trames et flux réseaux

Afin de valider la sécurité des flux entre les conteneurs, une capture de trafic a été réalisée sur le réseau Docker `proxy`.

- **Analyse :** En utilisant `tcpdump` sur l'interface réseau du bridge Docker, on observe que :
    - Le trafic HTTP (port 80) est bien redirigé par Traefik vers le port 8080 du conteneur Zabbix.
    - Les flux MySQL (port 3306) sont totalement absents du réseau `proxy`, prouvant qu'ils restent isolés sur le réseau privé de la stack.

5. Couplage Grafana - Zabbix

La validation finale repose sur l'affichage des données Zabbix dans l'interface Grafana.

- **Test :** Configuration de la "Data Source" dans Grafana en pointant sur l'API interne : `http://zabbix-web:8080/api_jsonrpc.php`.
- **Résultat :** Le test de connexion affiche "Zabbix API version 7.0.x found". Un tableau de bord affiche désormais les statistiques de performance de la machine Debian hôte et des conteneurs (WordPress, Nextcloud), confirmant le succès de l'interconnexion.

Partie 3 – Veille technologique.

Cette partie présente l'étude comparative réalisée pour valider le choix de la stack Zabbix/Grafana face aux autres solutions de supervision du marché, ainsi que les axes d'évolution possibles.

1. Recherche de solutions alternatives

Lors de la phase de conception, plusieurs outils de monitoring ont été mis en concurrence pour s'intégrer à l'infrastructure Docker/Traefik :

- **Prometheus & Alertmanager :**
  - **Points forts :** Conçu nativement pour les micro-services et les conteneurs. Très performant pour le "Service Discovery".
  - **Points faibles :** Configuration plus complexe (fichiers YAML ardus) et moins adapté pour le monitoring réseau traditionnel (SNMP) que Zabbix.
- **Netdata :**
  - **Points forts :** Visualisation en temps réel (seconde par seconde) extrêmement granulaire et installation ultra-rapide.
  - **Points faibles :** Historisation des données limitée par défaut et moins de flexibilité pour la création de tableaux de bord personnalisés à long terme.
- **Uptime Kuma :**
  - **Points forts :** Interface moderne, très simple pour surveiller la disponibilité HTTP/Ping.
  - **Points faibles :** Ne permet pas de faire de la métrologie (mesure de RAM, CPU, entrées/sorties disque). C'est un outil de "Health Check" et non de supervision complète.

2. Justification du choix de la solution

Le choix s'est porté sur le couple **Zabbix & Grafana** pour les raisons suivantes :

- **Polyvalence :** Capacité à surveiller aussi bien les conteneurs Docker (via l'agent) que le pare-feu pfSense (via SNMP) et les serveurs Linux/Windows.
- **Interopérabilité :** L'existence d'une API JSON-RPC robuste permet à Grafana d'extraire les données de Zabbix sans latence.
- **Visualisation professionnelle :** Grafana permet de créer des dashboards "Executive" (pour la direction) et "Technique" (pour l'administrateur) sur une seule et même interface.

- **Open Source :** Pas de coût de licence, tout en bénéficiant d'une communauté très active pour les templates de monitoring.

3. Axes d'évolution et amélioration

La veille technologique a permis d'identifier des pistes pour durcir et améliorer la solution :

- **Alerting déporté :** Intégration de webhooks pour envoyer les alertes critiques directement sur un canal **Discord** ou **Telegram** via les fonctions natives de Zabbix.
- **Auto-discovery :** Automatisation de l'ajout des nouveaux conteneurs WordPress ou Nextcloud dès leur création dans Docker via le "Low Level Discovery" (LLD) de Zabbix.
- **Sécurisation des agents :** Mise en place de certificats TLS pour chiffrer les communications entre les agents distants et le serveur de collecte.

4. Sources de la veille

- *Documentation officielle Zabbix 7.0 (LTS) :* https://www.zabbix.com/documentation/7.0/en
- *Grafana Labs - Zabbix Integration :* https://grafana.com/solutions/zabbix/
- *Traefik Proxy - Docker Guide :* https://doc.traefik.io/traefik/providers/docker/
- *Comparatif Supervision 2024 (IT-Connect) :* https://www.it-connect.fr