

# **BTS Services Informatiques aux Organisations**

Option SISR – Épreuve E6

## **Ressources documentaires – Réalisation professionnelle n°2**

Mise en place d'une plateforme d'administration et de supervision avec Docker

Candidat : EZZAMOURI Yanis

Établissement : Lycée Saint-Exupéry – Saint-Raphaël

Période : 03/04/2026

Lien vers la production : <https://yanis-ezzamouri.fr/?p=783>

### **1. Contexte et besoins**

#### **1.1 Présentation de l'organisation**

Le port de Cherbourg est une infrastructure portuaire française accueillant des activités variées : transport de passagers, ferries, fret maritime, croisières et opérations de maintenance navale. Il dispose d'un système d'information composé de plusieurs services informatiques mais ne possède pas de solution centralisée permettant la gestion du parc ni la supervision des services et journaux système.

#### **1.2 Problèmes identifiés**

- Absence d'outil centralisé pour la gestion du parc informatique et la supervision des services.
- Administration réalisée manuellement, ce qui rend la maintenance complexe et chronophage.
- Aucune solution permettant la centralisation et l'analyse des journaux système.

#### **1.3 Objectifs fixés**

Le directeur des systèmes d'information, M. Richard, souhaite moderniser l'infrastructure afin de :

- Faciliter l'administration du système informatique via une interface centralisée.
- Mettre en place une supervision centralisée des services.
- Améliorer la gestion et le suivi des équipements et des incidents.
- Mettre en place un système de collecte et d'analyse des journaux système.

## 2. Solutions envisagées et solution retenue

### 2.1 Comparatif des solutions

Solution	Avantages	Inconvénients
<b>GLPI</b>	Open-source, gestion de parc complète, suivi des incidents, riche en fonctionnalités	Configuration initiale assez longue
<b>NinjaOne</b>	Interface moderne, supervision centralisée, facilité d'utilisation	Solution payante, moins flexible
<b>Elastic Stack</b>	Centralisation puissante des logs, visualisations avancées avec Kibana, scalable	Consommation mémoire élevée
<b>Graylog</b>	Analyse des journaux, interface web intuitive, open-source	Moins complet qu'Elastic Stack
<b>Docker</b>	Conteneurisation légère, déploiement simplifié, maintenance facilitée	Nécessite une bonne maîtrise des commandes

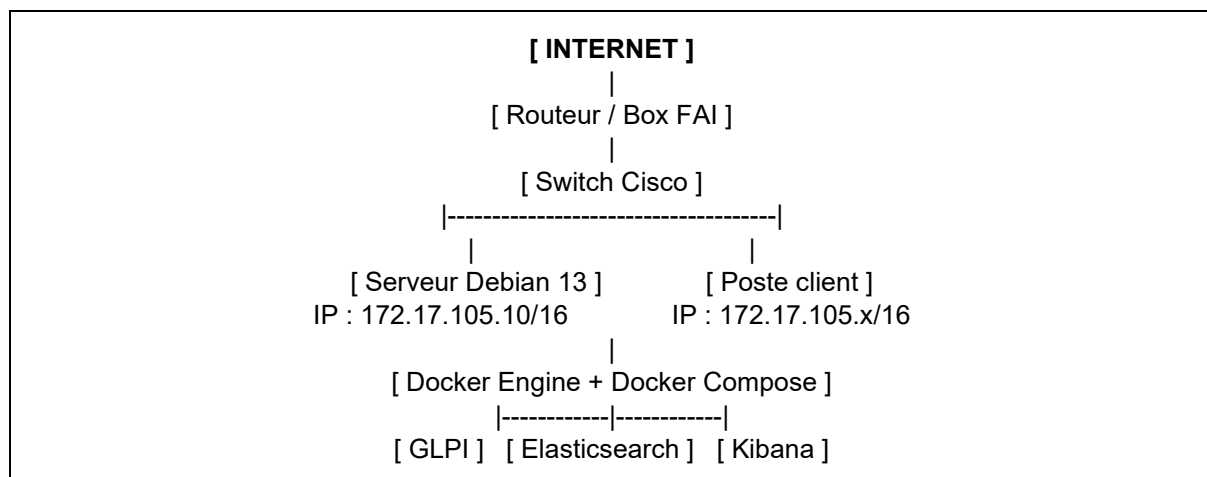
### 2.2 Solution retenue

- GLPI : retenu pour la gestion du parc informatique et le suivi des incidents, grâce à sa richesse fonctionnelle et son modèle open-source.
- Elastic Stack (Elasticsearch + Kibana) : retenu pour la centralisation et la visualisation des journaux système.
- Docker : retenu pour faciliter le déploiement, l'administration et la maintenance de l'ensemble des services. Docker Compose permet d'orchestrer les conteneurs simplement.

## 3. Architecture réseau

### 3.1 Schéma de l'infrastructure

L'infrastructure repose sur un réseau local avec adressage IP fixe. Le schéma ci-dessous représente l'architecture déployée :



Port 80	Port 9200	Port 5601
---------	-----------	-----------

### 3.2 Plan d'adressage

Équipement	Adresse IP	Masque	Rôle
Serveur hôte (Debian 13)	172.17.105.10	/16	Hébergement Docker
Poste client	172.17.105.x	/16	Tests / accès web
Routeur / Passerelle	172.17.250.3	/16	Accès réseau

### 3.3 Ports des services

Service	Port	URL d'accès
GLPI	80	http://172.17.105.10/glpi
Elasticsearch	9200	http://172.17.105.10:9200
Kibana	5601	http://172.17.105.10:5601

## 4. Mise en oeuvre

### Phase 1 – Mise en place de l'environnement

#### 4.1 Installation de Debian 13

- Téléchargement de l'image ISO Debian 13 depuis debian.org.
- Installation du système en mode minimal (sans interface graphique).
- Attribution de l'adresse IP fixe 172.17.105.10/16 via /etc/network/interfaces.
- Mise à jour du système : apt update && apt upgrade -y

#### 4.2 Installation de Docker et Docker Compose

- Installation des dépendances : apt-transport-https, ca-certificates, curl, gnupg.
- Ajout du dépôt officiel Docker et import de la clé GPG.
- Installation : apt install docker-ce docker-compose-plugin
- Vérification : docker --version && docker compose version
- Ajout de l'utilisateur courant au groupe docker.

## Phase 2 – Déploiement des services

### 4.3 Fichier docker-compose.yml

Un fichier docker-compose.yml unique orchestre les trois services. Voici sa structure :

```
version: '3.8'
services:
  glpi:
    image: diouxx/glpi
    ports:
      - "80:80"
    volumes:
      - glpi_data:/var/www/html/glpi

  elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch:8.13.0
    environment:
      - discovery.type=single-node
      - xpack.security.enabled=false
    ports:
      - "9200:9200"

  kibana:
    image: docker.elastic.co/kibana/kibana:8.13.0
    environment:
      - ELASTICSEARCH_HOSTS=http://elasticsearch:9200
    ports:
      - "5601:5601"
    depends_on:
      - elasticsearch

volumes:
  glpi_data:
```

Démarrage de tous les services :

```
docker compose up -d
```

## Phase 3 – Configuration

### 4.4 Configuration de GLPI

- Accès à l'interface web : <http://172.17.105.10/glpi>
- Suivi de l'assistant d'installation (base de données, compte administrateur).
- Suppression du fichier install.php après installation pour la sécurité.
- Création des catégories d'équipements, des utilisateurs et des groupes.

### 4.5 Liaison Elasticsearch – Kibana

- Vérification d'Elasticsearch : `curl http://172.17.105.10:9200`
- Accès à Kibana : <http://172.17.105.10:5601>
- Création de l'index pattern dans Kibana.
- Mise en place de dashboards de visualisation des journaux système.

## Phase 4 – Tests

## 4.6 Tableau de recette

Test	Action réalisée	Résultat attendu	Résultat obtenu
Démarrage conteneurs	docker compose up -d	Tous UP	OK
Accès GLPI	Navigateur → http://172.17.105.10	Page d'accueil GLPI	OK
Accès Elasticsearch	curl http://172.17.105.10:9200	Réponse JSON cluster	OK
Accès Kibana	Navigateur → http://172.17.105.10:5601	Dashboard Kibana	OK
Collecte de logs	Envoi de logs → visualisation Kibana	Logs visibles	OK
Redémarrage service	docker compose restart glpi	Service redémarré	OK

## 5. Ressources matérielles et logicielles

### 5.1 Matériel utilisé

Équipement	Rôle	Caractéristiques
Machine physique (PC/Serveur)	Serveur hôte hébergeant Docker	CPU multi-cœur, RAM ≥ 8 Go
Switch Cisco	Interconnexion du réseau local	Commutateur manageable
Poste client	Tests et accès aux interfaces web	Navigateur web, accès réseau local

### 5.2 Logiciels utilisés

Logiciel	Version	Usage
Debian	13 (Bookworm)	Système d'exploitation du serveur
Docker Engine	Dernière stable	Conteneurisation des services
Docker Compose	v2.x	Orchestration multi-conteneurs
GLPI	Dernière stable	Gestion de parc et des incidents
Elasticsearch	8.13.0	Indexation et stockage des logs
Kibana	8.13.0	Visualisation des données

## **6. Bilan**

Le déploiement de cette plateforme a permis de moderniser l'administration du système informatique du port de Cherbourg. La solution retenue répond pleinement aux besoins identifiés en phase d'analyse.

Les principaux bénéfices sont les suivants :

- Centralisation des services : toutes les solutions sont déployées sur un seul serveur via Docker.
- Meilleure visibilité sur le système : Kibana permet de visualiser les journaux en temps réel.
- Simplification du déploiement et de la maintenance : Docker Compose permet de gérer tous les services en une seule commande.
- Gestion structurée du parc : GLPI assure un suivi précis des équipements et des incidents.

Compétences mobilisées :

- Concevoir une solution d'infrastructure réseau.
- Installer, tester et déployer une solution d'infrastructure réseau.
- Exploiter, dépanner et superviser une solution d'infrastructure réseau.