# Workshop 2025 – Dockerwarts & PRAcadabra

Maison : GPTaigle  
Nom : GPTpsi  
Projet réalisé intégralement avec IA (GPT 5 flash et pro).

## 1. Présentation du défi

Objectif : Concevoir une infrastructure entièrement dockerisée incluant GLPI (ticketing), MariaDB, ElasticSearch, Grafana, Prometheus, cAdvisor et un reverse proxy Nginx en HTTPS, avec un Plan de Reprise d’Activité (PRA) automatisé. Tout le travail (conception, scripts, documentation) est généré par l’IA.

## 2. Schéma organisationnel & Planning

Rôles IA :  
- Infra & Sécurité : GPT 5  
- Monitoring : GPT 5  
- Documentation & prompts : GPT 5  
Planning :  
Jour 1 – Architecture & Compose  
Jour 2 – Monitoring & HTTPS  
Jour 3 – PRA (backup/restore)  
Jour 4 – Tests & captures  
Jour 5 – Soutenance.

## 3. Prompts utilisés (extraits significatifs)

🎯 Architecture & Compose

### Prompt

Génère un docker-compose.yml professionnel comprenant GLPI, MariaDB, ElasticSearch, Grafana, Prometheus, cAdvisor et Nginx avec réseaux frontend/backend et volumes persistants.

🎯 Monitoring

### Prompt

Ajoute Prometheus et cAdvisor à la stack, fournis un fichier prometheus.yml et la procédure d’import d’un dashboard Grafana pour Docker.

🎯 HTTPS

### Prompt

Fournis un nginx.conf multi-vhosts en HTTPS avec certificats auto-signés montés depuis l’hôte et des hôtes locaux glpi.local, grafana.local, elastic.local, prometheus.local, cadvisor.local.

🎯 PRA Backup

### Prompt

Écris un script bash robuste qui sauvegarde chaque volume Docker en tar.gz avec SHA256 et gère les cas d’erreur.

🎯 PRA Restore

### Prompt

Écris un script bash qui restaure le dernier backup d’un volume (ex: root\_mariadb\_data), avec suppression/recréation du volume et extraction des fichiers.

## 4. Documentation technique

Système hôte : Debian 12  
Conteneurisation : Docker / Docker Compose v2  
Réseaux : frontend (public), backend (interne).

Services actifs : GLPI, MariaDB, ElasticSearch, Grafana, Prometheus, cAdvisor, Nginx (HTTPS).

Persistance : mariadb\_data, es\_data, grafana\_data, glpi\_data (bind), /etc/nginx/ssl (certificats).

## 5. Codes générés

### docker-compose.yml

version: '3.9'  
  
services:  
 glpi:  
 image: diouxx/glpi  
 container\_name: glpi  
 ports:  
 - "8080:80"  
 networks:  
 - frontend  
 - backend  
 depends\_on:  
 - mariadb  
 volumes:  
 - ./glpi\_data:/var/www/html/glpi  
  
 mariadb:  
 image: mariadb:11  
 container\_name: mariadb  
 environment:  
 - MYSQL\_ROOT\_PASSWORD=root  
 - MYSQL\_DATABASE=glpi  
 - MYSQL\_USER=glpi  
 - MYSQL\_PASSWORD=glpi  
 volumes:  
 - mariadb\_data:/var/lib/mysql  
 networks:  
 - backend  
 restart: always  
  
 elasticsearch:  
 image: docker.elastic.co/elasticsearch/elasticsearch:8.9.0  
 container\_name: elasticsearch  
 environment:  
 - discovery.type=single-node  
 volumes:  
 - es\_data:/usr/share/elasticsearch/data  
 networks:  
 - backend  
 restart: always  
  
 grafana:  
 image: grafana/grafana  
 container\_name: grafana  
 ports:  
 - "3000:3000"  
 networks:  
 - backend  
 depends\_on:  
 - prometheus  
 volumes:  
 - grafana\_data:/var/lib/grafana  
 environment:  
 - GF\_SECURITY\_ADMIN\_USER=admin  
 - GF\_SECURITY\_ADMIN\_PASSWORD=admin  
 - GF\_PATHS\_PROVISIONING=/etc/grafana/provisioning  
 restart: always  
  
 prometheus:  
 image: prom/prometheus  
 container\_name: prometheus  
 ports:  
 - "9090:9090"  
 volumes:  
 - ./prometheus.yml:/etc/prometheus/prometheus.yml  
 networks:  
 - backend  
 restart: always  
  
 cadvisor:  
 image: gcr.io/cadvisor/cadvisor:v0.47.2  
 container\_name: cadvisor  
 ports:  
 - "8081:8080"  
 volumes:  
 - /:/rootfs:ro  
 - /var/run:/var/run:ro  
 - /sys:/sys:ro  
 - /var/lib/docker/:/var/lib/docker:ro  
 networks:  
 - backend  
 restart: always  
  
 nginx:  
 image: nginx:latest  
 container\_name: nginx  
 ports:  
 - "80:80"  
 - "443:443"  
 volumes:  
 - ./nginx.conf:/etc/nginx/nginx.conf  
 - /etc/nginx/ssl:/etc/nginx/ssl  
 networks:  
 - frontend  
 - backend  
 restart: always  
  
networks:  
 frontend:  
 backend:  
  
volumes:  
 es\_data:  
 mariadb\_data:  
 grafana\_data:

### prometheus.yml

global:  
 scrape\_interval: 10s  
  
scrape\_configs:  
 - job\_name: 'prometheus'  
 static\_configs:  
 - targets: ['prometheus:9090']  
  
 - job\_name: 'cadvisor'  
 static\_configs:  
 - targets: ['cadvisor:8080']

### nginx.conf

user nginx;  
worker\_processes auto;  
  
events {  
 worker\_connections 1024;  
}  
  
http {  
 include mime.types;  
 default\_type application/octet-stream;  
 sendfile on;  
 keepalive\_timeout 65;  
  
 ssl\_protocols TLSv1.2 TLSv1.3;  
 ssl\_ciphers HIGH:!aNULL:!MD5;  
  
 server {  
 listen 443 ssl;  
 server\_name glpi.local;  
 ssl\_certificate /etc/nginx/ssl/nginx-selfsigned.crt;  
 ssl\_certificate\_key /etc/nginx/ssl/nginx-selfsigned.key;  
 location / {  
 proxy\_pass http://glpi:80;  
 proxy\_set\_header Host $host;  
 proxy\_set\_header X-Real-IP $remote\_addr;  
 }  
 }  
  
 server {  
 listen 443 ssl;  
 server\_name grafana.local;  
 ssl\_certificate /etc/nginx/ssl/nginx-selfsigned.crt;  
 ssl\_certificate\_key /etc/nginx/ssl/nginx-selfsigned.key;  
 location / {  
 proxy\_pass http://grafana:3000;  
 proxy\_set\_header Host $host;  
 proxy\_set\_header X-Real-IP $remote\_addr;  
 }  
 }  
  
 server {  
 listen 443 ssl;  
 server\_name elastic.local;  
 ssl\_certificate /etc/nginx/ssl/nginx-selfsigned.crt;  
 ssl\_certificate\_key /etc/nginx/ssl/nginx-selfsigned.key;  
 location / {  
 proxy\_pass http://elasticsearch:9200;  
 proxy\_set\_header Host $host;  
 }  
 }  
  
 server {  
 listen 443 ssl;  
 server\_name prometheus.local;  
 ssl\_certificate /etc/nginx/ssl/nginx-selfsigned.crt;  
 ssl\_certificate\_key /etc/nginx/ssl/nginx-selfsigned.key;  
 location / {  
 proxy\_pass http://prometheus:9090;  
 proxy\_set\_header Host $host;  
 }  
 }  
  
 server {  
 listen 443 ssl;  
 server\_name cadvisor.local;  
 ssl\_certificate /etc/nginx/ssl/nginx-selfsigned.crt;  
 ssl\_certificate\_key /etc/nginx/ssl/nginx-selfsigned.key;  
 location / {  
 proxy\_pass http://cadvisor:8080;  
 proxy\_set\_header Host $host;  
 }  
 }  
  
 server {  
 listen 80;  
 server\_name \_;  
 return 301 https://$host$request\_uri;  
 }  
}

### pra\_dockerwarts.sh (backup)

#!/bin/bash  
# Script PRA Dockerwarts - Version corrigée (grpespi 2025)  
BACKUP\_DIR="/backup"  
DATE=$(date +'%Y-%m-%d\_%H-%M-%S')  
mkdir -p "$BACKUP\_DIR"  
  
echo "=== [INFO] Début de la sauvegarde : $DATE ==="  
  
VOLUMES=$(docker volume ls -q)  
  
if [ -z "$VOLUMES" ]; then  
 echo "[WARN] Aucun volume Docker détecté. Rien à sauvegarder."  
 exit 0  
fi  
  
for VOL in $VOLUMES; do  
 echo "[INFO] Sauvegarde du volume : $VOL"  
 if ! docker volume inspect "$VOL" >/dev/null 2>&1; then  
 echo "[WARN] Volume $VOL introuvable, passage au suivant."  
 continue  
 fi  
  
 docker run --rm \  
 -v "${VOL}":/data:ro \  
 -v "${BACKUP\_DIR}":/backup \  
 alpine sh -c "tar -czf /backup/${VOL}\_${DATE}.tar.gz -C /data ."  
  
 if [ $? -eq 0 ]; then  
 sha256sum "$BACKUP\_DIR/${VOL}\_${DATE}.tar.gz" > "$BACKUP\_DIR/${VOL}\_${DATE}.sha256"  
 echo "[OK] Volume $VOL sauvegardé avec succès."  
 else  
 echo "[ERROR] Échec de la sauvegarde du volume $VOL."  
 fi  
done  
  
echo "=== [OK] Sauvegarde terminée ==="

### pra\_restore.sh (restore)

#!/bin/bash  
# Script de restauration PRA Dockerwarts (grpespi 2025)  
BACKUP\_DIR="/backup"  
VOLUME\_NAME="root\_mariadb\_data"  
  
echo "=== [INFO] Début de la restauration pour le volume : $VOLUME\_NAME ==="  
  
LAST\_BACKUP=$(ls -t $BACKUP\_DIR/${VOLUME\_NAME}\_\*.tar.gz | head -n 1)  
  
if [ -z "$LAST\_BACKUP" ]; then  
 echo "[ERROR] Aucun backup trouvé pour $VOLUME\_NAME"  
 exit 1  
fi  
  
echo "[INFO] Fichier de sauvegarde trouvé : $LAST\_BACKUP"  
  
docker volume rm -f "$VOLUME\_NAME"  
docker volume create "$VOLUME\_NAME"  
  
tar -xzf "$LAST\_BACKUP" -C /var/lib/docker/volumes/${VOLUME\_NAME}/\_data/  
  
echo "[OK] Restauration terminée depuis $LAST\_BACKUP"

## 6. Supervision, validations & preuves

Dashboards Grafana : ID 193 (Docker Monitoring). Preuves attendues :  
- Capture docker ps (7 conteneurs actifs)  
- Capture Grafana (dashboard)  
- Capture GLPI en HTTPS  
- Capture ElasticSearch (auth en HTTPS)  
- Extrait de /backup (tar.gz + sha256)  
- nginx -t successful

## 7. Remplacement de Cassandra

Cassandra (datalake) a été remplacé par une combinaison plus cohérente et légère :  
- ElasticSearch : historisation et recherche  
- Prometheus : time-series des métriques  
- cAdvisor : export des métriques Docker  
Visualisation centralisée via Grafana.

## 8. Conclusion

L’architecture livrée répond au défi Dockerwarts : sécurité (HTTPS), persistance des données, supervision temps réel et PRA automatisé. Le tout a été conçu et documenté intégralement par IA (GPT 5) pour GEMEpsi.