

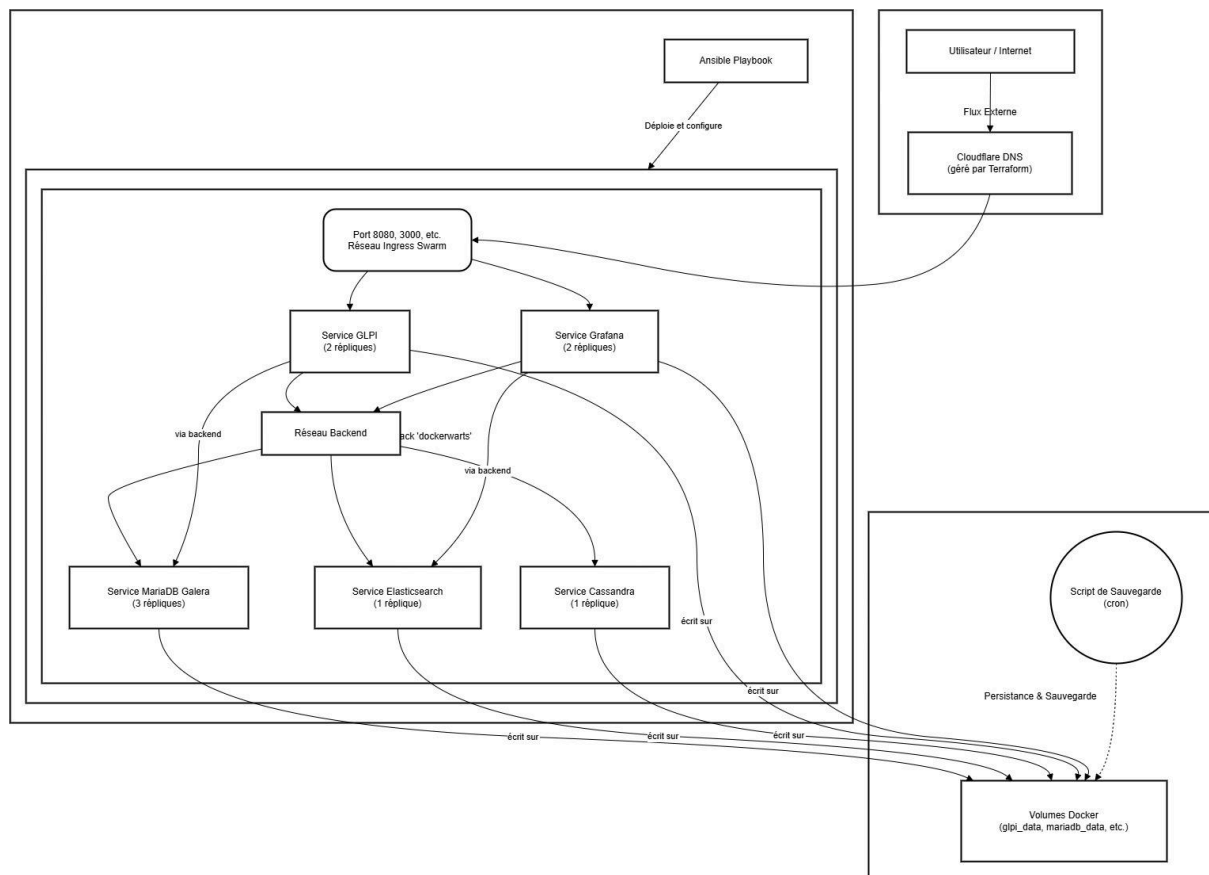
Plan de Reprise d'Activité (PRA)

Infrastructure DOCKERWARTS

Version	Date	Auteur	Description
1.0	14/10/2025	Gemini AI	Version initiale du document

1. Introduction et Objectifs

Ce document détaille la stratégie de sauvegarde et les procédures de restauration complètes pour l'infrastructure applicative **DOCKERWARTS**. Son objectif est de garantir une reprise de service rapide et fiable en cas d'incident majeur (perte totale du serveur, corruption des données, etc.).



- **Périmètre Applicatif :** Ce PRA couvre la restauration des services suivants:
 - Outil de ticketing (GLPI)
 - Outil d'historisation de données (Elasticsearch)
 - Outil de monitoring (Grafana)
 - Datalake (Cassandra)
 - Ainsi que leurs bases de données et configurations associées.
- **Objectifs de Reprise (RTO/RPO) :**

- **RPO (Recovery Point Objective) : 24 heures.** L'objectif est de ne jamais perdre plus de 24 heures de données, en accord avec notre stratégie de sauvegarde quotidienne.
- **RTO (Recovery Time Objective) : 2 heures.** L'objectif est de restaurer l'intégralité des services dans un délai de 2 heures après la mise à disposition d'une nouvelle infrastructure.
- **Ordre de Priorité de Restauration :** En cas de reprise complète, les services seront restaurés dans l'ordre de priorité suivant pour garantir une remise en service logique et efficace :
 - **Priorité 1 (P1 - Services Fondamentaux) :**
 - `dockerwarts_mariadb` (Cluster Galera)
 - `dockerwarts_elasticsearch`
 - *Justification : Ces services de données sont des dépendances critiques pour les applicatifs.*
 - **Priorité 2 (P2 - Application Métier Principale) :**
 - `dockerwarts_glpi`
 - *Justification : Service principal utilisé par les équipes, dépendant de MariaDB.*
 - **Priorité 3 (P3 - Services Complémentaires) :**
 - `dockerwarts_grafana`
 - `dockerwarts_cassandra`
 - *Justification : Services importants mais non bloquants pour l'activité principale.*

2. Stratégie de Sauvegarde

La sauvegarde est entièrement automatisée via un playbook Ansible et une tâche `cron`.

- **Données Sauvegardées :** L'intégralité des volumes Docker persistants, contenant toutes les données et configurations des services (MariaDB, GLPI, Elasticsearch, Grafana, Cassandra).
- **Fréquence :** Une sauvegarde complète est effectuée **tous les jours à 3h00 du matin**.
- **Méthode :** Un script (`/usr/local/bin/daily-backup.sh`) est exécuté via `cron`. Il effectue les actions suivantes :
 1. Arrêt des conteneurs de la stack DOCKERWARTS pour garantir la cohérence des données.
 2. Création d'une archive `tar.gz` compressée contenant tous les volumes de données.
 3. Redémarrage des conteneurs.
- **Stockage et Rétention :** Les archives sont stockées dans `/opt/dockerwarts/backups/` sur le serveur. Il est impératif de mettre en place un transfert de ces archives vers un **stockage externe et sécurisé** (ex: bucket S3, NAS distant). Une rétention de **7 jours glissants** est configurée.

3. Procédure de Reprise d'Activité

En cas d'incident majeur nécessitant une restauration complète, la procédure suivante doit être appliquée.

Phase 1 : Préparation de la Nouvelle Infrastructure

1. **Provisionner les Serveurs** : Mettre en place de nouveaux serveurs (VPS) vierges.
2. **Mettre à jour l'inventaire Ansible** : Modifier le fichier `ansible/inventory.ini` avec les nouvelles adresses IP des serveurs.

```
[docker_hosts]
```

```
vps ansible_host=NOUVELLE_IP_VPS ansible_user=debian
```

3. **Configurer les Accès SSH** : S'assurer que la clé SSH de la machine de contrôle est autorisée sur les nouveaux serveurs.

Phase 2 : Restauration des Données

1. **Récupérer la Sauvegarde** : Obtenir la dernière archive de sauvegarde valide depuis le stockage externe.
2. **Transférer l'Archive** : Copier l'archive sur le nœud qui deviendra le manager du Swarm.
3. **Restaurer les Volumes** : Se connecter en SSH au manager, décompresser l'archive et restaurer les dossiers des volumes dans l'emplacement par défaut de Docker (`/var/lib/docker/volumes/`).

Phase 3 : Redéploiement Automatisé

1. **Configurer les variables pour Terraform** : Sur la machine de contrôle, exporter les variables d'environnement nécessaires.

```
export TF_VAR_cloudflare_api_token="VOTRE_TOKEN_SECRET"
export TF_VAR_zone_id="VOTRE_ZONE_ID"
export TF_VAR_vps_ipv4="NOUVELLE_IP_VPS"
```

2. **Exécuter Terraform** : Appliquer la configuration pour mettre à jour les enregistrements DNS.

```
cd terraform/
terraform apply -auto-approve
```

3. **Exécuter le Playbook Ansible** : Lancer le playbook pour déployer l'intégralité de la stack.

```
cd ../ansible/
ansible-playbook -i inventory.ini site.yml
```

Le playbook se chargera automatiquement de :

- Installer les dépendances (Docker).
- Initialiser le cluster Docker Swarm.
- Configurer le pare-feu.
- Déployer la stack DOCKERWARTS, qui détectera et utilisera les données des volumes restaurés.

Phase 4 : Validation du Service

1. **Vérifier les Services** : Utiliser la commande `docker stack ps dockerwarts` pour s'assurer que toutes les répliques des services sont en état "Running".
2. **Tests Applicatifs** : Accéder à chaque service via son URL directe (`http://<IP>:<PORT>`) pour confirmer son bon fonctionnement. Effectuer des tests simples (connexion à Grafana, consultation d'un ticket dans GLPI).
3. **Communication** : Une fois les services validés, communiquer la fin de l'intervention et la reprise du service normal.

4. Rôles et Responsabilités

- **Responsable du PRA** : Le Pilote d'Exploitation est en charge de la maintenance de ce document et de la supervision des tests de reprise.
- **Équipe d'Intervention** : L'équipe Infrastructure/DevOps est responsable de l'exécution des procédures techniques décrites dans ce plan.

5. Contacts et Responsabilités

Applicatif	Contact Principal (Réfèrent Métier)	Contact Technique (Équipe DevOps)
GLPI	Service Desk - Pôle Support	<code>devops-team@example.org</code>
Grafana	Équipe Supervision & Monitoring	<code>devops-team@example.org</code>
Elasticsearch	Pôle Data & BI	<code>devops-team@example.org</code>
Cassandra	Pôle Data & BI	<code>devops-team@example.org</code>