

Rapport Technique : Système de Réservation d'Événements

Table des Matières

1. Lancement du Système
2. Tests et Validation
3. Scénarios de Panne
4. Procédures de Migration

Lancement du Système

Démarrage Initial

1. Préparation de l'environnement

```
# Vérifier que les ports requis sont disponibles  
netstat -an | grep "LISTEN" | grep -E "8080|18000|3306|8404"
```

```
# Nettoyer les anciens conteneurs et volumes si nécessaire  
docker-compose down -v  
docker system prune -f
```

2. Séquence de démarrage

```
# Démarrer le premier nœud MariaDB  
docker-compose up -d mariadb-1
```

```
# Attendre 30 secondes pour l'initialisation  
sleep 30
```

```
# Démarrer les autres services  
docker-compose up -d
```

3. Vérification du démarrage

```
# Vérifier l'état des conteneurs  
docker-compose ps
```

```
# Vérifier les logs  
docker-compose logs --tail=100
```

Configuration Post-démarrage

1. Vérification du cluster Galera

```
SHOW STATUS LIKE 'wsrep_%';  
-- Vérifier wsrep_cluster_size = 3  
-- Vérifier wsrep_cluster_status = 'Primary'
```

2. Vérification de HAProxy

- Accéder à `http://localhost:18404`
- Vérifier que tous les nœuds sont marqués UP

Tests et Validation

Tests Fonctionnels

1. Test de l'API

Test de l'endpoint événements

```
curl -X GET http://localhost:18000/events/
```

Test de création d'événement

```
curl -X POST http://localhost:18000/events/ \
```

```
-H "Content-Type: application/json" \
```

```
-d '{"name":"Test Event","description":"Test","venue":"Test Venue","categories":["
```

2. Test de l'Interface Utilisateur

- Accéder à `http://localhost:8080`
- Vérifier l'affichage des événements
- Tester le processus de réservation

3. Test de l'Interface Admin

- Accéder à `http://localhost:8080/admin.html`
- Créer un nouvel événement
- Ajouter une session
- Vérifier la mise à jour en temps réel

Tests de Performance

1. Test de charge avec Apache Benchmark

Test de l'API événements

```
ab -n 1000 -c 50 http://localhost:18000/events/
```

Test avec données POST

```
ab -n 500 -c 20 -p test_data.json http://localhost:18000/events/
```

2. Surveillance des Performances

Surveillance des conteneurs

```
docker stats
```

Logs en temps réel

```
docker-compose logs -f
```

Scénarios de Panne

1. Panne d'un Nœud MariaDB

Simulation :

```
docker-compose stop mariadb-2
```

Vérification :

```
# Vérifier HAProxy
```

```
curl http://localhost:18404/stats
```

```
# Vérifier le cluster
```

```
docker exec mariadb-node-1 mysql -uroot -prootpassword \  
-e "SHOW STATUS LIKE 'wsrep_cluster_size'"
```

Récupération :

```
docker-compose start mariadb-2
```

2. Panne de HAProxy

Simulation :

```
docker-compose stop haproxy
```

Impact : - L'API ne peut plus accéder à la base de données - Les nouvelles requêtes échouent

Récupération :

```
docker-compose start haproxy
```

```
docker-compose restart api # Si nécessaire
```

3. Panne Complète du Cluster

Procédure de Récupération :

```
# Arrêt complet
```

```
docker-compose down
```

```
# Nettoyage si nécessaire
```

```
docker volume rm $(docker volume ls -q | grep mariadb)
```

```
# Redémarrage séquentiel
```

```
docker-compose up -d mariadb-1
```

```
sleep 30
```

```
docker-compose up -d
```

Procédures de Migration

Migration de la Base de Données

1. Sauvegarde

```
# Dump de la base de données
docker exec mariadb-node-1 mysqldump -uroot -prootpassword \
    --single-transaction event_booking > backup.sql
```

2. Restauration

```
# Restauration sur un nouveau système
cat backup.sql | docker exec -i mariadb-node-1 mysql -uroot -prootpassword event_booking
```

Migration de Version

1. Sauvegarde des Données

```
# Données
docker-compose exec mariadb-1 mysqldump -uroot -prootpassword event_booking > backup.sql

# Configuration
cp -r config/mariadb config/mariadb.bak
cp -r config/haproxy config/haproxy.bak
```

2. Mise à Jour des Images

```
# Mettre à jour docker-compose.yml avec les nouvelles versions
# Reconstruire les images
docker-compose build --pull

# Redémarrer avec les nouvelles versions
docker-compose down
docker-compose up -d
```

Migration vers un Nouveau Serveur

1. Préparation

```
# Archive du projet
tar -czf event_booking.tar.gz ./*

# Transfert
scp event_booking.tar.gz newserver:/path/to/destination
```

2. Installation sur le Nouveau Serveur

```
# Décompression
tar -xzf event_booking.tar.gz

# Configuration des variables d'environnement
```

```
cp .env.example .env
nano .env # Ajuster les paramètres

# Démarrage
docker-compose up -d
```

Bonnes Pratiques et Recommandations

Maintenance Régulière

1. Nettoyage des Logs

```
# Rotation des logs Docker
/etc/logrotate.d/docker-container
```

2. Surveillance des Ressources

```
# Espace disque
df -h
```

```
# Utilisation mémoire
docker stats
```

Sécurité

1. Sauvegardes

- Automatiser les sauvegardes quotidiennes
- Tester régulièrement les restaurations
- Conserver les sauvegardes hors site

2. Mises à Jour

- Mettre à jour régulièrement les images Docker
- Suivre les bulletins de sécurité
- Maintenir une liste des dépendances à jour

Documentation

1. Logs

- Centraliser les logs (ELK Stack recommandé)
- Configurer des alertes sur les erreurs critiques
- Maintenir un historique des incidents

2. Monitoring

- Mettre en place Prometheus + Grafana
- Configurer des dashboards pour :
 - Performances du cluster
 - Métriques applicatives
 - Alertes système