**📊 PROMPT CONFIGURATION MONITORING PODMAN DESKTOP**

@codebase **CRÉER STACK MONITORING COMPLÈTE POUR PODMAN DESKTOP**

**MISSION : MONITORING PRODUCTION AUTOMATIQUE**

Créer une configuration monitoring complète Prometheus + Grafana optimisée pour Podman Desktop avec ports automatiques et déploiement sécurisé.

**CONFIGURATION PODMAN DESKTOP**

**Contraintes techniques :**

* Utiliser Podman Desktop (pas Docker Compose classique)
* Ports automatiques selon disponibilité
* Volumes Podman nommés
* Réseau bridge automatique
* Compatible interface graphique Podman Desktop

**Structure attendue :**

monitoring/

├── podman-compose.yml (ou docker-compose.yml compatible)

├── prometheus.yml

├── alert\_rules.yml

├── grafana/

│ ├── provisioning/

│ │ ├── datasources/

│ │ └── dashboards/

│ └── dashboards/

├── scripts/

│ ├── deploy-monitoring.sh

│ └── health-check-monitoring.sh

└── .env.monitoring

**EXIGENCES FONCTIONNELLES**

**Services à configurer :**

1. **Prometheus** - Collecte métriques
2. **Grafana** - Dashboards visuels
3. **Node Exporter** - Métriques système
4. **Postgres Exporter** - Métriques base de données
5. **Alertmanager** - Gestion alertes (optionnel)

**Métriques E-COMPTA-IA à surveiller :**

* Temps réponse API Spring Boot
* Nombre requêtes/seconde
* Erreurs HTTP (4xx, 5xx)
* Connexions base de données actives
* Utilisation CPU/RAM containers
* Status santé services
* Authentifications réussies/échouées

**Alertes automatiques :**

* Service down > 5 minutes
* CPU > 80% pendant 5 minutes
* Mémoire > 85%
* Temps réponse > 2 secondes
* Connexions DB > seuil maximum

**CONFIGURATION BACKEND SPRING BOOT**

**Dépendances Maven à ajouter :**

* spring-boot-starter-actuator
* micrometer-registry-prometheus

**Endpoints métriques à exposer :**

* /actuator/health
* /actuator/prometheus
* /actuator/metrics
* Configuration sécurisée production

**SCRIPTS AUTOMATISÉS**

**deploy-monitoring.sh :**

* Créer volumes Podman nécessaires
* Créer réseau monitoring
* Démarrer tous les services monitoring
* Vérifier santé de chaque service
* Afficher URLs d'accès

**health-check-monitoring.sh :**

* Vérifier status Prometheus
* Vérifier status Grafana
* Vérifier collecte métriques
* Tester alertes
* Rapport santé monitoring

**SÉCURITÉ PRODUCTION**

**Variables sensibles :**

* Mots de passe Grafana administrateur
* Clés secrètes diverses
* Configuration SMTP alertes
* Tokens d'accès

**Restrictions réseau :**

* Accès interfaces admin sécurisés
* Rate limiting si applicable
* Headers sécurité
* Logs d'accès monitoring

**DASHBOARDS GRAFANA**

**Dashboard principal E-COMPTA-IA :**

* Vue d'ensemble santé application
* Métriques temps réel
* Graphiques performance
* Status services
* Alertes actives

**Panels requis :**

* Status services (up/down)
* Temps réponse API P95
* Requêtes par seconde
* Utilisation CPU/RAM
* **Métriques base de données PostgreSQL :**
  + Connexions actives vs maximum
  + Transactions par seconde
  + Query duration (slow queries)
  + Cache hit ratio
  + Database size growth
  + Index efficiency
  + Lock wait time
  + Checkpoint frequency
* **Métriques cache Redis :**
  + Mémoire utilisée vs totale
  + Cache hit/miss ratio
  + Commandes par seconde
  + Clients connectés
  + Keys total et expirées
  + Latence moyenne
  + Throughput réseau
  + Persistence status
* Erreurs HTTP par type
* **Métriques métier :**
  + Utilisateurs actifs
  + Opérations comptables par heure

**FORMAT RÉPONSE ATTENDU**

**Créer automatiquement :**

1. Tous les fichiers de configuration
2. Scripts de déploiement
3. Configuration backend Spring Boot
4. Dashboards Grafana JSON
5. Documentation d'utilisation
6. Variables d'environnement exemple

**Résultat final :**

Un système monitoring clé-en-main qui se déploie avec une commande et fournit surveillance complète de E-COMPTA-IA en production.

**EXÉCUTION IMMÉDIATE**

Créer maintenant cette configuration monitoring complète optimisée Podman Desktop avec ports automatiques et sécurité production.

**OBJECTIF : Monitoring professionnel déployable en 5 minutes**