

Rapport d'Analyse de Performance - Nextcloud

Contexte du projet

Test de performance d'une application Nextcloud dockerisée avec JMeter. Monitoring temps réel via InfluxDB et Grafana.

Architecture testée : Nextcloud + MySQL en conteneurs Docker avec stack de monitoring intégrée.

Méthodologie appliquée

Scénario utilisateur

Parcours de navigation : Login → Dashboard → Photos → Logout avec corrélation dynamique des tokens CSRF.



Login

Authentification utilisateur



Dashboard

Affichage tableau de bord



Photos

Navigations galerie photos

Logout

Déconnexion sécurisée

Approche de test

Tests de montée en charge progressifs de 10 et 15 utilisateurs virtuels simultanés. 9 minutes par tests. 3 tests par quantité de vu.

Monitoring en temps réel des métriques JMeter et ressources système Docker.

Résultats obtenus

Métriques de performance

3.2s

Temps moyen (10 VU)

CPU app ~50% (pic 177%)

4.1s

Temps moyen (15 VU)

CPU app ~120% (pic 267%)

0%

Taux d'erreur

Sur tous les tests

Throughput stable avec très peu de requêtes étant donné la faible charge

Observations techniques

Augmentation de l'utilisation du CPU progressive par le conteneur Nextcloud à partir de 15 utilisateurs virtuels.

Corrélation directe entre montée en charge et consommation CPU de l'application.

Analyse des goulots d'étranglement

Bottleneck identifié

Le conteneur applicatif (Apache/PHP) pourrait représenter le goulot principal avec saturation CPU >200% sous charge.

La base de données MySQL reste stable avec utilisation CPU <5%.

Limites de l'analyse actuelle

Absence de SLA définis

Sans seuils de performance prédéfinis, impossible de qualifier objectivement les résultats comme acceptables ou critiques.

Recommandation : définir des SLA métier (ex: temps de réponse <4s, disponibilité >99%).

Point d'inflexion non précisé

Tests supplémentaires nécessaires pour identifier le point de basculement.

Optimisation possible via allocation de ressources CPU supplémentaires au conteneur applicatif.

Enseignements techniques

Bonnes pratiques validées

Corrélation automatique des tokens de session

Monitoring système en temps réel

Approche méthodologique progressive

Identification précise des ressources critiques

Axes d'amélioration futurs

Optimisation configuration PHP-FPM

Mise en place de cache applicatif

Tests de montée en charge plus fins

Définition de seuils SLA métier

Recommandations

Actions possibles

01

Définir des SLA de performance basés sur les besoins métier

02

Augmenter les ressources CPU du conteneur Nextcloud

03

Effectuer des tests de validation post-optimisation

Conclusion

Le projet démontre une approche méthodologique rigoureuse du test de performance avec identification précise du goulot d'étranglement principal.

La capacité actuelle supportée est d'environ 15 utilisateurs simultanés avant dégradation notable des performances.

L'architecture de monitoring mise en place permet un suivi en temps réel efficace pour les optimisations futures.