# Introduction

Qu’est-ce qu’Uptime Kuma ?

Uptime Kuma est un outil puissant conçu pour superviser et surveiller en continu la disponibilité et les performances des sites web, des applications et des services en ligne. Grâce à ses fonctionnalités avancées, Uptime Kuma offre aux développeurs et aux administrateurs système une solution complète pour détecter rapidement les problèmes et garantir un temps de fonctionnement optimal.

Fonction de Uptime Kuma :

La fonction principale de Uptime Kuma est de fournir une surveillance proactive de la disponibilité des sites web et des services en ligne. En collectant des données en temps réel à partir de différentes sources, telles que des requêtes HTTP, des vérifications de ports et des tests de ping, Uptime Kuma offre une visibilité précise sur l'état de fonctionnement de vos ressources en ligne. Cela permet de détecter les problèmes dès qu'ils se produisent, minimisant ainsi les temps d'arrêt non planifiés et les perturbations pour les utilisateurs.

Possibilités offertes par Uptime Kuma :

1. Surveillance avancée des performances : Uptime Kuma va au-delà de la simple vérification de disponibilité en fournissant des métriques détaillées sur les performances de vos sites et services. Il mesure les temps de réponse, les délais de chargement des pages, les latences réseau et d'autres indicateurs clés, vous permettant ainsi d'identifier les goulots d'étranglement et d'optimiser les performances de vos applications.

2. Alertes personnalisées : Uptime Kuma permet de configurer des alertes personnalisées pour être informé immédiatement lorsque des problèmes surviennent. Vous pouvez définir des seuils spécifiques pour les temps de réponse, les codes d'erreur HTTP ou toute autre métrique pertinente, et recevoir des notifications par e-mail, SMS ou intégrations avec des outils de communication populaires.

3. Suivi de l'historique et rapports détaillés : Uptime Kuma conserve un historique complet de toutes les vérifications effectuées, ce qui vous permet d'analyser les tendances, de repérer les problèmes récurrents et d'évaluer les améliorations au fil du temps. Vous pouvez également générer des rapports détaillés pour partager les informations sur les performances avec les parties prenantes concernées.

Objectif : Surveillance automatisée d’un serveur Apache ou Nginx par Uptime Kuma, à l’aide d’une base de données.

# Base de données

* Aller dans la DB mysql
* Créer un utilisateur avec tous les droits et host %, tel que

CREATE USER 'user'@'%' IDENTIFIED BY 'pwd';

GRANT ALL PRIVILEGES ON \* . \* TO 'user'@'%';

* Configurer bind-address = 0.0.0.0

avec sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf

Bind adress est par défaut en 127.0.0.1, correspondant à localhost, le passage en 0.0.0.0 autorise la connexion depuis n’importe quelle ip.

Notez que permettre les connexions distantes peut présenter des risques de sécurité. Assurez-vous de prendre des mesures appropriées pour sécuriser votre serveur MySQL, telles que l'utilisation de mots de passe forts et la limitation des autorisations accordées aux utilisateurs distants.

Après avoir suivi ces étapes, l'utilisateur devrait pouvoir se connecter à distance à la base de données "test" en utilisant le nom d'utilisateur et le mot de passe spécifiés.

* Créer une base de données
* Créer une table serveur(id int, nom varchar , etat bool)
* Insérer des données en fonction du/des serveur(s) à analyser

Exemple : INSERT INTO serveur(nom, etat) VALUES (‘Apache’, 0)

* En ce qui concerne le type BOOL de la colonne etat, 0 correspond à DOWN, et 1 correspond à UP

# Uptime Kuma

* Créer des sondes MySQL et Push
* Sonde MySQL :

Ce type de sonde permet de requêter une base de données à intervalles réguliers, ici, il permet de vérifier la colonne état et nous informe ainsi de la fonctionnalité du/des service(s).

* Sélectionner Type de sonde = MySQL/MariaDB

Chaîne de connexion=mysql://username:password@host:port/database

Adapter cette chaîne de connexion avec vos données, exemple :

mysql://kuma:x248h@1\*.1\*\*.2\*.1\*2:3\*0\*/testh

L’outil requête sera utile plus tard, sauvegarder

* Sonde Push :

Ce type de sonde permet de recevoir des informations par le système, permettant d’afficher l’état du serveur sur Uptime Kuma, UP ou DOWN.

Sélectionner Type de sonde = Push

Copier l’URL Push et le conserver pour la partie système

Sauvegarder

# Système

* Créer un fichier .sh et y inscrire un script tel que :

#!/bin/bash

# Vérifier l'état du serveur Apache

apache\_status=$(sudo systemctl status apache2.service | grep "Active:")

# Vérifier l'état du serveur Nginx

nginx\_status=$(sudo systemctl status nginx.service | grep "Active:")

#Apache

if [[ $apache\_status == \*"active (running)"\* ]]; then

echo "Le serveur Apache fonctionne."

# Modifier la base de données pour mettre à jour l’état du serveur Apache (id=2 dans cet exemple), donner la valeur 1 à etat puisque le serveur Apache est actif

mysql -u nom\_utilisateur -pmot\_de\_passe -e "USE testh; UPDATE service SET etat = 1 WHERE id = 2;"

#Envoyer un Push avec curl puis l’URL de votre sonde push, permettant d’afficher en temps réel l’état du serveur sur Uptime Kuma, ici UP

curl http://lien?status=up&msg=OK&ping=

else

echo "Le serveur Apache n'est pas en cours d'exécution."

# Modifier la base de données pour mettre à jour l’état du serveur Apache (id=2 dans cet exemple), donner la valeur 0 à etat puisque le serveur Apache est inactif

mysql -u nom\_utilisateur -pmot\_de\_passe -e "USE testh; UPDATE service SET etat = 0 WHERE id = 2;"

#Envoyer un Push avec curl puis l’URL de votre sonde push, permettant d’afficher en temps réel l’état du serveur sur Uptime Kuma, ici DOWN

curl http://lien?status=down&msg=OK&ping=

fi

#Nginx

if [[ $nginx\_status == \*"active (running)"\* ]]; then

echo "Le serveur Nginx fonctionne."

# Modifier la base de données pour mettre à jour l’état du serveur Nginx (id=3 dans cet exemple), donner la valeur 1 à etat puisque le serveur Nginx est actif

mysql -u nom\_utilisateur -pmot\_de\_passe -e "USE testh; UPDATE service SET etat = 1 WHERE id =3 ;"

#Envoyer un Push avec curl puis l’URL de votre sonde push, permettant d’afficher en temps réel l’état du serveur sur Uptime Kuma, ici UP

curl http://lien?status=up&msg=OK&ping=

else

echo "Le serveur Nginx n'est pas en cours d'exécution."

# Modifier la base de données pour mettre à jour l’état du serveur Nginx (id=3 dans cet exemple), donner la valeur 0 à etat puisque le serveur Nginx est inactif

mysql -u nom\_utilisateur -pmot\_de\_passe -e "USE testh; UPDATE service SET etat = 0 WHERE id = 3;"

#Envoyer un Push avec curl puis l’URL de votre sonde push, permettant d’afficher en temps réel l’état du serveur sur Uptime Kuma, ici DOWN

curl http://lien?status=down&msg=OK&ping=

fi

* Se rendre sur Uptime Kuma, sonde MySQL et rédiger une requête telle que

select nom from service where etat=0 ;

* Exécuter le script tout juste rédigé, sudo bash nom\_script.sh
* Vérifier si la valeur de la colonne etat dans la base de données correspond à l’état du serveur, puis vérifier sur Uptime Kuma si la réponse envoyée est la bonne
* Automatiser l’exécution du script par le système chaque minute, permettant d’actualiser l’état du serveur dans la base de données chaque minute, et donc sur Uptime Kuma chaque minute.
* Se rendre dans sudo crontab -e et écrire tout en bas du fichier :

\* \* \* \* \* bash /home/install/vserv\_push.sh >> /home/install/cronlog

(cronlog accessible via tail -f cronlog)