

Mise en place d'une synchronisation Zabbix - Uptime Kuma

- [Contexte et objectif](#)
- [Préparatifs](#)
- [Mise en service](#)

Contexte et objectif

Dans le cadre de mon travail, nous utilisons Zabbix comme outil de supervision afin de surveiller la disponibilité et la santé de plusieurs sites distants. Bien que Zabbix soit très complet, il n'est pas toujours adapté pour partager directement son interface avec des utilisateurs finaux ou des équipes non techniques.

Problème : comment fournir une **page de statut** simple et claire à ces utilisateurs (leur indiquant quels sites sont opérationnels ou en panne) **sans** devoir leur donner accès à Zabbix (et donc aux détails complets de la supervision) ?

La solution que j'ai trouvée était de déployer [Uptime Kuma](#) pour générer et publier une page de statut, puis le synchroniser avec Zabbix via un script Python qui récupère automatiquement la liste des hôtes et leur état.

Pour installer Uptime Kuma, vous pouvez vous rendre sur [cette page](#)

Préparatifs

Création d'un utilisateur et d'un groupe "API" dans Zabbix

Pour laisser le script accéder à Zabbix sans exposer d'autres informations :

1. Créer un utilisateur dédié :

- Nom : `API`
- Groupe : `API`
- Rôle : `API` (lecture seule).

2. Permissions : limitées uniquement au groupe d'hôtes que je souhaite récupérer

- L'utilisateur `API` peut lire la liste des hôtes et des déclencheurs (*Triggers*) pour ces hôtes spécifiques.

Cela me permettra notamment de voir les hôtes ayant une alerte "critique" et donc de détecter quel hôte est réellement "Down" ou non.

Script Python de synchronisation (Zabbix -> Uptime Kuma)

J'ai ensuite créé un script Python qui effectue les opérations suivantes :

- **Connexion** à l'API de Zabbix avec l'utilisateur `API`.
- **Récupération** de tous les hôtes du groupe que je souhaite récupérer
- **Nettoyage** du nom de l'hôte en retirant le préfixe "Site <ip>" spécifique à mon entreprise pour obtenir un libellé plus lisible.
- **Création ou mise à jour** du moniteur correspondant dans Uptime Kuma via la librairie python [uptime-kuma-api](#)
- **Vérification** s'il existe un déclencheur de niveau 5 (critique) actif pour chaque hôte :
 - S'il y a un déclencheur critique → le moniteur sera forcé en mode *Down* dans Uptime Kuma.
 - Sinon → le moniteur est en mode "inversé" (toujours Up en apparence).

Pourquoi "Mode Inversé" ?

Parce que nous pointons chaque moniteur vers une URL inexistante, ce qui le mettrait automatiquement en échec (Down). Activer le "mode inversé" dans Uptime Kuma permet de considérer ce faux échec comme un Up "normal".

- En clair, **pas de problème** = *Mode inversé* activé → Affiché Up.
- **Problème critique** = on désactive le *Mode inversé* → Affiché Down.

Intervalle de vérification (3 jours)

Pour éviter de “polluer” inutilement le réseau et le serveur Uptime Kuma avec un check qui n’a pas de raison d’être réel, on a défini un intervalle de **260000 secondes (environ trois jours)**. Ainsi, Uptime Kuma ne fera pratiquement jamais de requête significative (car la cible `http://nepaschanger.no/` est fictive), mais cela reste suffisant pour maintenir la configuration voulue.

Mise en service

Mise en service automatique (systemd)

Pour exécuter le script à chaque démarrage du serveur Zabbix :

- Créer un fichier de service `zabbix-kuma-sync.service` dans `/etc/systemd/system/`, par exemple :

```
[Unit]
Description=Synchronisation Zabbix → Uptime Kuma
After=network.target

[Service]
Type=simple
ExecStart=/usr/bin/python3 [Chemin vers le script]
Restart=on-failure
User=appliance

[Install]
WantedBy=multi-user.target
```

- Activer et lancer le service :

```
sudo systemctl enable zabbix-kuma-sync.service
sudo systemctl start zabbix-kuma-sync.service
```

- Vérifier qu'il tourne correctement :

```
systemctl status zabbix-kuma-sync.service
```

Configuration de la page de statut Uptime Kuma

Une fois les moniteurs créés automatiquement :

1. Moniteurs :

- URL pointée : `http://nepaschanger.no/` (inexistante).
- Intervalle de vérification : 260000 secondes.

- Par défaut, le script active le mode inversé, donc l'hôte s'affiche "Up" même si l'URL est inaccessible.

2. Status Page :

- Uptime Kuma propose la création d'une page de statut publique ou interne, consultable par les utilisateurs finaux.
- Les hôtes y apparaissent Up ou Down suivant leur *mode inversé*.

3. Personnalisation CSS :

- Pour n'afficher que les hôtes en **Down**, un CSS personnalisé masque ceux en **Up**.
 - Cette personnalisation se fait dans les paramètres de la page de statut de Kuma (Settings → Status Page → Custom CSS).

```
.monitor-list > .item:has(.bg-primary) {  
    display: none !important;  
}
```

De cette manière, les utilisateurs qui consultent la page de statut n'ont **pas** besoin d'accéder à Zabbix : ils voient uniquement la liste des hôtes impactés (en panne), ce qui remplit l'objectif d'une interface simple et dépouillée.

Résultat et avantages

- **Page de statut** : indique clairement quels sites distants sont en panne sans exposer l'interface complexe de Zabbix.
- **Sécurité** : l'utilisateur "API" de Zabbix n'a accès qu'au groupe d'hôtes que je souhaite et à leurs déclencheurs, pas à l'ensemble du monitoring ou de la configuration.
- **Automatisation** : le script se lance au démarrage, met à jour Uptime Kuma sans intervention manuelle.
- **Performance** : les checks étant inversés et programmés sur un intervalle très long, la surcharge réseau est quasi inexistante.

Ce système répond parfaitement au besoin de "vitrine" ou de "tableau de bord" : une **page de statut** sur laquelle ne figurent que les pannes, gérée de manière automatique, tout en évitant la complexité de l'interface Zabbix et en protégeant les données sensibles de l'outil de supervision.