**Documentation : Intégration de Prometheus et Grafana dans une application Django**

**1. Introduction**

Cette documentation explique en détail comment intégrer Prometheus pour la collecte des métriques et Grafana pour la visualisation et la surveillance d'une application Django. Ce guide couvre l'installation, la configuration, et la mise en place des alertes, ainsi que les bonnes pratiques pour respecter les normes de gestion des données personnelles.

**2. Prérequis**

- Système d'exploitation : Ubuntu/Debian ou autre distribution Linux.

- Python : Version 3.6 ou supérieure avec Django installé.

- Accès root ou sudo pour installer les services Prometheus et Grafana.

**3. Installation de Prometheus**

3.1 Téléchargement de Prometheus

```bash

wget https://github.com/prometheus/prometheus/releases/download/v2.41.0/prometheus-2.41.0.linux-amd64.tar.gz

tar -xvf prometheus-2.41.0.linux-amd64.tar.gz

cd prometheus-2.41.0.linux-amd64

sudo mv prometheus /usr/local/bin/

sudo mv promtool /usr/local/bin/

```

3.2 Création de l'utilisateur Prometheus

```bash

sudo useradd --no-create-home --shell /bin/false prometheus

sudo mkdir /etc/prometheus /var/lib/prometheus

sudo chown prometheus:prometheus /etc/prometheus /var/lib/prometheus

```

3.3 Configuration de Prometheus

- Créez et éditez le fichier de configuration `/etc/prometheus/prometheus.yml` :

```bash

sudo nano /etc/prometheus/prometheus.yml

```

- Ajoutez la configuration suivante pour scraper les métriques de Django :

```yaml

global:

scrape\_interval: 15s

scrape\_configs:

- job\_name: 'django'

static\_configs:

- targets: ['127.0.0.1:8000'] Remplacez par l'URL de votre application Django

```

3.4 Création du service systemd pour Prometheus

- Créez le fichier `/etc/systemd/system/prometheus.service` :

```bash

sudo nano /etc/systemd/system/prometheus.service

```

- Ajoutez la configuration suivante :

```ini

[Unit]

Description=Prometheus Monitoring

Wants=network-online.target

After=network-online.target

[Service]

User=prometheus

ExecStart=/usr/local/bin/prometheus --config.file=/etc/prometheus/prometheus.yml --storage.tsdb.path=/var/lib/prometheus/

Restart=always

[Install]

WantedBy=multi-user.target

```

3.5 Démarrage de Prometheus

```bash

sudo systemctl daemon-reload

sudo systemctl start prometheus

sudo systemctl enable prometheus

```

- Accéder à Prometheus : `http://localhost:9090`

**4. Installation de Grafana**

4.1 Ajout du dépôt Grafana

```bash

sudo apt-get install -y software-properties-common

sudo add-apt-repository "deb https://packages.grafana.com/oss/deb stable main"

wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -

sudo apt-get update

```

4.2 Installation de Grafana

```bash

sudo apt-get install grafana

```

4.3 Démarrage de Grafana

```bash

sudo systemctl start grafana-server

sudo systemctl enable grafana-server

```

- Accéder à Grafana : `http://localhost:3000`

4.4 Connexion de Grafana à Prometheus

- Accédez à Grafana via `http://localhost:3000`.

- Connectez-vous avec les identifiants par défaut (`admin` / `admin`).

- Ajoutez Prometheus comme source de données :

- Allez dans Configuration > Data Sources > Add data source.

- Sélectionnez Prometheus et entrez `http://localhost:9090` comme URL.

- Cliquez sur Save & Test pour vérifier la connexion.

**5. Configuration des métriques et des alertes dans Grafana**

5.1 Création d'un tableau de bord pour surveiller Django

- Cliquez sur + > Dashboard > Add new panel.

- Sélectionnez votre source de données Prometheus.

- Entrez une requête Prometheus, par exemple pour le temps de réponse :

```prometheus

rate(django\_http\_requests\_latency\_seconds\_by\_view\_method\_sum[1m]) / rate(django\_http\_requests\_latency\_seconds\_by\_view\_method\_count[1m])

```

- Ajoutez des panneaux supplémentaires pour les autres métriques comme le taux de succès, le taux d'erreur, etc.

5.2 Configuration des alertes

- Dans chaque panneau, allez dans l'onglet Alert et configurez une nouvelle alerte :

- Condition : Par exemple, pour le temps de réponse :

```prometheus

avg(rate(django\_http\_requests\_latency\_seconds\_by\_view\_method\_sum[1m]) / rate(django\_http\_requests\_latency\_seconds\_by\_view\_method\_count[1m])) > 0.2

```

- Notification : Configurez l'envoi d'alertes par email ou Slack.

**6. Intégration des règles de journalisation dans Django**

6.1 Installation de `django-prometheus`

```bash

pip install django-prometheus

```

6.2 Configuration de `django-prometheus`

- Ajoutez `'django\_prometheus'` à `INSTALLED\_APPS` dans `settings.py`.

- Modifiez `MIDDLEWARE` pour inclure :

```python

'django\_prometheus.middleware.PrometheusBeforeMiddleware',

'django\_prometheus.middleware.PrometheusAfterMiddleware',

```

- Configurez les URLs pour les métriques :

```python

urlpatterns = [

...

path('', include('django\_prometheus.urls')),

]

```

6.3 Journalisation des métriques

- Configurez les métriques que vous souhaitez surveiller et collecter via Prometheus.

**7. Tests et validation**

7.1 Test des métriques

- Vérifiez que toutes les métriques sont correctement exposées via `/metrics`.

- Assurez-vous que Grafana affiche les métriques en temps réel.

7.2 Test des alertes

- Simulez des conditions pour déclencher les alertes et vérifiez que les notifications sont reçues.

**Commandes de gestion des services**

- Démarrer Prometheus : `sudo systemctl start prometheus`

- Arrêter Prometheus : `sudo systemctl stop prometheus`

- Redémarrer Prometheus : `sudo systemctl restart prometheus`

- Vérifier l'état de Prometheus : `sudo systemctl status prometheus`

- Démarrer Grafana : `sudo systemctl start grafana-server`

- Arrêter Grafana : `sudo systemctl stop grafana-server`

- Redémarrer Grafana : `sudo systemctl restart grafana-server`

- Vérifier l'état de Grafana : `sudo systemctl status grafana-server`

**Dépannage**

- Vérification des logs Prometheus : `sudo journalctl -u prometheus -f`

- Vérification des logs Grafana : `sudo journalctl -u grafana-server -f`