

Monitoring avec OpenTelemetry

Exposer des metrics pour Node Exporter avec un script Python.



Sommaire

1. Introduction
2. Prometheus
3. Exporters
4. Grafana
5. NodeExporter
6. Demo création de metrics avec Python
7. Conclusion

Introduction

Retour d'expérience sur la création de metrics avec Python.

Contexte :

- Générer des metrics dans un fichier pouvant être lu par NodeExporter
- Récupérer les données à partir d'un fichier xml
- Pas possibilité de mettre à jour l'application
- Utilisation de docker possible

Prometheus

Une solution de Monitoring pour :

- Collecter des metrics
- Stocker des metrics
- Superviser avec le service AlertManager
- Écrite principalement en Go.

À l'origine produit par SoundCloud en 2013 et rendu open source.

En 2016, Prometheus a été accepté par le CNCF (Cloud Native Computing Foundation).

Prometheus

Solution de métrologie

Collecter métriques à interval régulier.

4 types de metrics :

- Compteur
- Jauge
- Histogramme
- Résumé

Propriétés d'une metric :

- Nom
- labels (job, instance,...)

Prometheus

Exposer les metrics

- Bibliothèque client
- Exporters

Stockage des résultats :

- Dans une base de données timeseries
- Format TSDB

Accéder aux données :

- Langage de requête - PromQL (Prometheus Query Language)
- Interface web (port 9090)
- API restful

Prometheus

Solution de supervision

Lever des alertes.

Service prometheus :

- Publie des alertes en fonction des règles.
- Règles basées sur des requêtes au format PromQL

Service AlertManager pour l'émission d'alertes :

- Notifications
- mails
- API (slack...)

Prometheus

Superviser plusieurs infrastructures

Agréger les données de plusieurs instances de Prometheus :

- Fédérateur Prometheus
- Solution Thanos

Volumétrie de la base de données :

- Rétention des métriques
- Interval de scrape

Exporters

Exposer des métriques (système ou applicative)

Un exporter :

- HTTP Endpoint
- Métriques système (node exporter)
- Accès à un service externe

Exporters

Exemples :

- Node (système)
- cAdvisor (docker)
- Blackbox (DNS, HTTP, ...)
- JMX (Java)
- Pushgateway

Exemple service externe :

- Gitlab
- Harbor

Grafana

Plateforme de visualisation de données Grafana.

S'interfacer avec diverses bases et APIs.

S'appuie sur les métriques collectées par diverses solutions de métrologies.

Exemple de données qu'on veut récupérer (Golden Signals) :

- Latence
- Trafic
- Saturation
- Erreurs

Création de dashboards :

- Créer des dashboards from scratch
- Dashboards élaborés par la communauté

Grafana

Supervision :

Fonction d'alertes et de notifications :

- Alert rules
- Contact points
- Notification policies

NodeExporter

Récupérer les informations système d'une machine.

Exporter utilisé pour les systèmes Linux.

Exemple :

- CPU
- Mémoire
- Espace disques
- Réseau

NodeExporter

Exemples de collecteur pour NodeExporter :

- filesystem
- systemd
- textfile

Pour exposer des metrics provenant d'un dossier, ajouter l'option :

```
--collector.textfile.directory
```

Exemple de fichier produit :

```
# HELP store_total_rows Total rows in store
# TYPE store_total_rows gauge
store_total_rows {path="/tmp/store.xml"} 6
```

Demo création de metrics avec Python

Objectifs du projet Python

- Extraire les données d'un store
- Générer les metrics dans un fichier .prom
- Filtrer les données par type
- Afficher les données pour debugger

Conclusion

Retour d'expérience

Pour un cas d'usage simple, textfile collector de node exporter est une bonne solution.

Avantages :

- Simplicité de la mise en place d'un script Python
- Pas besoin de modifier la configuration dans prometheus si node exporter est déjà utilisé.

Pour des cas d'usage plus complexes :

- Utiliser la librairie Prometheus dans l'application si possible.
- Sinon développer un outil qui utilise la librairie Prometheus.