(/

Monitoreo autohospedado para aplicaciones Spring Boot

Última modificación: 1 de octubre de 2019

por Michael Pratt (https://www.baeldung.com/author/michael-pratt/) (https://www.baeldung.com/author/michael-pratt/)

DevOps (https://www.baeldung.com/category/devops/)
Spring Boot (https://www.baeldung.com/category/spring/spring-boot/)



1. Introducción

Una de las muchas características excelentes de Spring Boot es el conjunto de actuadores (https://www.baeldung.com/spring-boot-actuators) integrados . Estos actuadores proporcionan una manera fácil de monitorear y controlar casi todos los aspectos de una aplicación Spring Boot.

En este tutorial, veremos cómo usar el actuador de métricas (https://docs.spring.io/spring-boot/docs/current/reference/html/production-ready-metrics.html) para crear una solución de monitoreo autohospedada para las aplicaciones Spring Boot.

2. Base de datos de métricas

La primera parte de la supervisión de las aplicaciones Spring Boot es elegir una base de datos de **métricas**. De manera predeterminada, Spring Boot configurará un registro de métricas de micrómetro (https://micrometer.io/) en cada aplicación.

Esta implementación predeterminada recopila un conjunto predefinido de métricas de aplicaciones, como el uso de memoria y CPU, solicitudes HTTP y algunas otras. Pero **estas métricas se almacenan solo en la**Utilizan**mentaria**pallonquas significa reques expecterámicada vez rejuicio de <u>policación la applicación la appli</u>

Ok

(X)

X

Para crear una solución de monitoreo autohospedada, primero debemos elegir una base de datos de métricas que viva fuera de la aplicación Spring Boot. Las siguientes secciones analizarán solo algunas de las opciones disponibles de alojamiento propio.

Tenga en cuenta que cada vez que Spring Boot detecta otra base de datos de métricas en el classpath, deshabilita automáticamente el registro en memoria.

2.1. InfluxDB

InfluxDB (https://www.influxdata.com/products/influxdb-overview/) es una base de datos de series temporales de código abierto. La forma más rápida de comenzar con InfluxDB es ejecutarlo localmente como un contenedor Docker:

```
1 docker run -p 8086:8086 -v /tmp:/var/lib/influxdb influxdb
```

Tenga en cuenta que esto almacenará métricas en la partición local / tmp. Esto está bien para el desarrollo y las pruebas, pero no sería una buena opción para entornos de producción.

Una vez que InfluxDB se está ejecutando, podemos configurar nuestra aplicación Spring Boot para publicar métricas agregando la siguiente línea a *application.properties*:

```
1 management.metrics.export.influx.uri=https://influx.example.com:8086
```

InfluxDB no proporciona una herramienta de visualización nativa. Sin embargo, proporciona una herramienta separada llamada Cronógrafo (https://docs.influxdata.com/chronograf/) que funciona bien para visualizar datos de InfluxDB.

2.2. Pro

Prometheus (https://prometheus.io) es un kit de herramientas de monitoreo y alerta de código abierto creado originalmente en SoundCloud. Funciona ligeramente diferente de InfluxDB.

En lugar de configurar nuestra aplicación para publicar métricas en Prometheus, **configuramos Prometheus para sondear nuestra aplicación periódicamente** .

Primero, configuramos nuestra aplicación Spring Boot para exponer un nuevo punto final del actuador Prometheus. Hacemos esto incluyendo la dependencia micrometer-registro-prometeo (https://search.maven.org/search?q=g:io.micrometer%20a:micrometer-registry-prometheus):

Esto creará un nuevo punto final del actuador que produce datos de métricas en un formato especial que Prometheus entiende.

Luego, tenemos que configurar Prometheus para sondear nuestra aplicación agregando nuestra
Utilizar CONFIGUES SIÓN GENERAL DE PLANTA DE LA PROPOSE DE LA POLÍTICA DE PROVINCIA DE LA POLÍTICA DE LA POLÍTICA DE PROVINCIA DE LA POLÍTICA DE PROVINCIA DE LA POLÍTICA DE PROVINCIA DE LA POLÍTICA DEL POLÍTICA DE LA POLÍTICA DEL POLÍTICA DE LA POLÍTICA DE LA POLÍTICA DEL POLÍTICA DEL POLÍTICA DEL POLÍTICA DEL POLÍTICA DE LA POLÍTICA DEL POL

La siguiente configuración indica a Prometheus que sondee nuestra aplicación cada 5 segundos, utilizando el nuevo punto final del actuador:

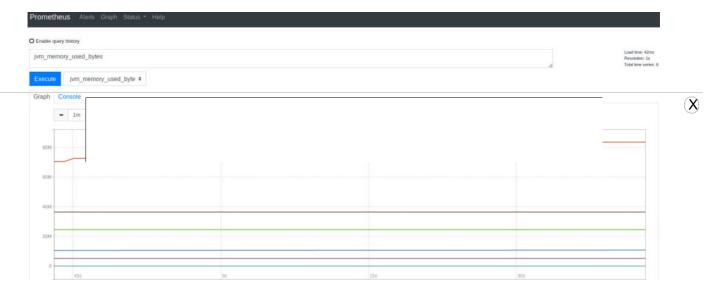
```
X
```

```
scrape_configs:
    job_name: 'spring-actuator'
metrics_path: '/actuator/prometheus'
scrape_interval: 5s
static_configs:
    targets: ['127.0.0.1:8080']
```

Finalmente, podemos iniciar un servidor Prometheus local con Docker. Esto supone que nuestro archivo de configuración personalizado se encuentra en el archivo local /etc/prometheus/prometheus.yml:

```
docker run -d \
--name=prometheus \
-p 9090:9090 \
-v /etc/prometheus/prometheus.yml:/etc/prometheus/prometheus.yml \
prom/prometheus \
--config.file=/etc/prometheus/prometheus.yml
```

Prometheus proporciona su propia herramienta de visualización para ver las métricas que se han recopilado. Se puede acceder en la URL http://localhost: 9090 /.



(http://inprogress.baeldung.com/wp-content/uploads/2019/09/self-hosted-monitoring-spring-boot-prometheus-dashboard.jpg)

2.3. Grafito

Graphite (https://graphiteapp.org) es otra base de datos de series temporales de código abierto. Su arquitectura es un poco más complicada que las otras bases de datos que hemos visto, pero con Docker, es sencillo ejecutar una instancia localmente:

```
1
   docker run -d \
2
    --name graphite \
3
     --restart=always \
4
    -p 80:80 \
     -p 2003-2004:2003-2004 \
6
     -p 2023-2024:2023-2024 \
7
     -p 8125:8125/udp \
8
     -p 8126:8126 \
    graphiteapp/graphite-statsd
```

Luego podemos configurar Spring Boot para publicar métricas en nuestra instancia agregando la dependencia micrometer-Registry-Graphite (https://search.maven.org/search?

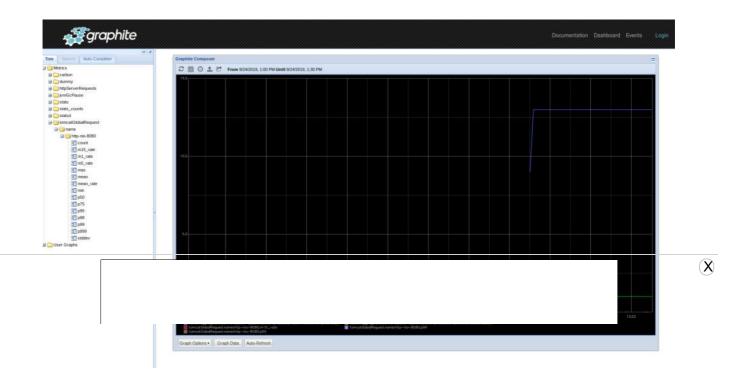
Utilizan psonie prometer % 20 merinorio poeteritire en stato privación, puede leer la Política de privacidad y cookies completa (/privacy-policy)



Además de agregar las propiedades de configuración a application.properties :

```
management.metrics.export.graphite.host=127.0.0.1
management.metrics.export.graphite.port=2004
```

Al igual que Prometheus, Graphite incluye su propio panel de visualización. Está disponible en la URL http://localhost/.



(http://inprogress.baeldung.com/wp-content/uploads/2019/09/self-hosted-monitoring-spring-boot-graphite-dashboard.jpg)

3. Herramientas de visualización

Una vez que tengamos una solución para almacenar métricas fuera de nuestra aplicación Spring Boot, **la siguiente decisión es cómo queremos visualizar los datos** .

Algunas de las bases de datos de métricas mencionadas anteriormente incluyen sus propias herramientas de visualización. Existe una herramienta de visualización independiente que vale la pena buscar para nuestra solución de monitoreo autohospedada.

3.1. Grafana

Grafana (https://grafana.com/) es una herramienta de análisis y monitoreo de código abierto. **Se puede conectar a todas las bases de datos mencionadas anteriormente, así como a muchas otras**.

Utilizamos cookies para mejorar su experiencia con el sitio. Para obtener más información, puede leer la Política de privacidad y cookies completa (/privacy-policy)



Grafana generalmente proporciona una mejor configuración y alertas superiores que la mayoría de las herramientas de visualización integradas. Se puede ampliar fácilmente mediante complementos, y hay muchos paneles precompilados que se pueden importar para crear rápidamente nuestras propias visualizaciones.

Para ejecutar Grafana localmente, podemos iniciarlo usando Docker:

1 docker run -d -p 3000:3000 grafana/grafana

Ahora podemos acceder a la página de inicio de Grafana en la URL http://localhost: 3000 / .

En este punto, necesitaríamos configurar una o más fuentes de datos . Esta puede ser cualquiera de las bases de datos de métricas discutidas anteriormente o una variedad de otras herramientas compatibles.

Una vez que se configura una fuente de datos, podemos construir un nuevo tablero o importar uno que haga lo que queramos.



(http://inprogress.baeldung.com/wp-content/uploads/2019/09/self-hosted-monitoring-spring-boot-grafana-dashboard.jpg)

4. Conclusión

En este artículo, hemos analizado la creación de una solución de monitoreo autohospedada para las aplicaciones Spring Boot.

Analizamos tres bases de datos de métricas que Spring Boot admite fácilmente y vimos cómo ejecutarlas localmente.

Utilizamando en analizamos do come contesta de una variedad de fuentes.



Acabo de anunciar el nuevo curso *Learn Spring*, centrado en los fundamentos de Spring 5 y Spring Boot 2:

>> VER EL CURSO (/ls-course-end)





¿Estás aprendiendo a construir tu API con Spring?

Enter your email address

>> Obtenga el libro electrónico

▲ el más nuevo ▲ más antiguo ▲ más votado



Alejandro a mejorar su experiencia con el sitio. Para obtener más información, puede leer la <u>Política de privacidad y cookies completa (/privacy-policy)</u>

(1) Hace 20 dias

() Hace 19 dias

ଡ

io.micrometer
micrometer- Registry -influx
para aquellos que estén interesados, el nombre predeterminado de la base de datos es "mydb"
iGracias!

+ o
Eric Martin
Hola Alexander,

gracias por tu comentario, actualizaremos el artículo pronto.

iLos comentarios están cerrados en este artículo!

+ 0 **-**

CATEGORÍAS

PRIMAVERA (HTTPS://WWW.BAELDUNG.COM/CATEGORY/SPRING/)
DESCANSO (HTTPS://WWW.BAELDUNG.COM/CATEGORY/REST/)
JAVA (HTTPS://WWW.BAELDUNG.COM/CATEGORY/JAVA/)
SEGURIDAD (HTTPS://WWW.BAELDUNG.COM/CATEGORY/SECURITY-2/)
PERSISTENCIA (HTTPS://WWW.BAELDUNG.COM/CATEGORY/PERSISTENCE/)
JACKSON (HTTPS://WWW.BAELDUNG.COM/CATEGORY/JSON/JACKSON/)
HTTP DEL LADO DEL CLIENTE (HTTPS://WWW.BAELDUNG.COM/CATEGORY/HTTP/)
KOTLIN (HTTPS://WWW.BAELDUNG.COM/CATEGORY/KOTLIN/)

SERIE

TUTORIAL DE JAVA 'VOLVER A LO BÁSICO' (/JAVA-TUTORIAL)

JACKSON JSON TUTORIAL (/JACKSON)

HTTPCLIENT 4 TUTORIAL (/HTTPCLIENT-GUIDE)

RESTO CON SPRING TUTORIAL (/REST-WITH-SPRING-SERIES)

TUTORIAL SPRING PERSISTENCE (/PERSISTENCE-WITH-SPRING-SERIES)

SEGURIDAD CON PRIMAVERA (/SECURITY-SPRING)

ACERCA DE

SOBRE BAELDUNG (/ABOUT)
LOS CURSOS (HTTPS://COURSES.BAELDUNG.COM)

Utilizamos Teò Riès para mejoral son en la Política de privacidad y cookies completa (/privacy-policy)

META BAELDUNG (HTTP://META.BAELDUNG.COM/)

Ok

X

EL ARCHIVO COMPLETO (/FULL_ARCHIVE)
ESCRIBIR PARA BAELDUNG (/CONTRIBUTION-GUIDELINES)
EDITORES (/EDITORS)
NUESTROS COMPAÑEROS (/PARTNERS)
ANUNCIE EN BAELDUNG (/ADVERTISE)

TÉRMINOS DE SERVICIO (/TERMS-OF-SERVICE)

POLÍTICA DE PRIVACIDAD (/PRIVACY-POLICY)

INFORMACIÓN DE LA COMPAÑÍA (/BAELDUNG-COMPANY-INFO)

CONTACTO (/CONTACT)

Utilizamos cookies para mejorar su experiencia con el sitio. Para obtener más información, puede leer la Política de privacidad y cookies completa (/privacy-policy)