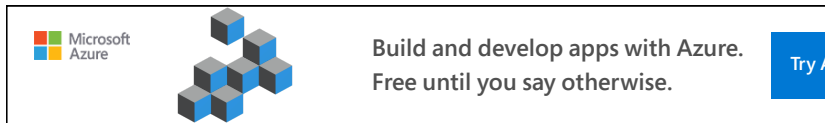


▲	HP 301XL... 38,99 €	HP 12A O...	HP 85A O...	HP 302XL...	HP 205A...	HP 953XL...	Cartucho...	HP 17A O...
▼		97,90 €	86,90 €	35,99 €	61,90 €	143,99 €	104,89 €	77,90 €

Base de conocimientos Recursos Ofertas Únete a nosotros Acerca de Iniciar sesión Registro



ANDROIDE JAVA IDIOMAS JVM DESARROLLO DE SOFTWARE ÁGIL CARRERA COMUNICACIONES DEVOPS META JCG

Inicio » Desarrollo de software » Microservicios para desarrolladores de Java: monitoreo y alertas

SOBRE ANDREY REDKO



Andriy es un desarrollador de software bien fundamentado con más de 12 años de experiencia práctica utilizando Java / EE, C # / .NET, C ++, Groovy, Ruby, programación funcional (Scala), bases de datos (MySQL, PostgreSQL, Oracle) y soluciones NoSQL (MongoDB, Redis).



Microservicios para desarrolladores de Java: monitoreo y alertas

Publicado por: Andrey Redko en Desarrollo de software 24 de octubre de 2019 0 577 reproducciones

1. Introducción

En esta última parte del tutorial, vamos a hablar sobre el tema donde se unen todos los pilares de observabilidad : monitoreo y alerta. Para muchos, este tema pertenece estrictamente a las operaciones y la única forma en que sabes que de alguna manera funciona es cuando estás de guardia y te atrapan.

Tabla de contenido

1. Introducción
2. Filosofía de seguimiento y alerta
3. Monitoreo de infraestructura
4. Monitoreo de aplicaciones

- 4.1. Prometeo y administrador de alertas
- 4.2. TICK Stack: Chronograf
- 4.3. Netfix Atlas
- 4.4. Hawkular
- 4.5. Stagemonitor
- 4.6. Grafana
- 4.7. Alerta adaptativa
5. Orquestación
6. nube
7. Sin servidor
8. Las alertas no son solo sobre métricas
9. Conclusiones
10. Al final

El objetivo de nuestra discusión es desmitificar al menos algunos aspectos del monitoreo, aprender acerca de las alertas y comprender cómo las métricas , los rastreos distribuidos y, a veces, incluso los registros se están utilizando para observar continuamente el estado del sistema y notificar sobre los próximos problemas, anomalías, posibles interrupciones o mal comportamiento.

2. Filosofía de seguimiento y alerta

Hay toneladas de métricas diferentes que podrían (y deberían) recopilarse mientras se opera un sistema de software más o menos realista, particularmente diseñado según los principios de la arquitectura de microservicios . En este contexto, el proceso para recopilar y almacenar dichos datos de estado generalmente se conoce como monitoreo.

Entonces, ¿qué exactamente debes monitorear? Para ser justos, no es fácil presentar por adelantado todos los aspectos posibles del sistema que deben ser monitoreados y, como tal, decidir qué métricas (y otras señales) necesita recopilar y cuáles no. pero la regla de oro "más datos es mejor que no tener datos" ciertamente se aplica aquí. El objetivo final es cuando las cosas salen mal, el subsistema de monitoreo debe informarle de inmediato. De esto se trata la alerta.

La mensajería de alerta (o notificación de alerta) es una comunicación de máquina a persona que es importante o urgente.

48,99 €	22,99 €	95,99 €
75,99 €	13,99 €	28,99 €

BOLETIN INFORMATIVO

¡Los iniciados ya disfrutan de ac...
semanales y libros blancos grati...

Únase a ellos ahora para obtene...
a las últimas noticias en el munc...
como información sobre Android...
otras tecnologías relacionadas.

Enter your e-mail...

☐ Acepto los Términos y la P...
privacidad

Regístrate

ÚNETE A NOSOTROS

With **1,** unique v...
500 ai...
placed a...
related !...
Constan...
lookout...
encoura...
So If yo...
unique and interesting content t...
check out our **JCG** partners pro...
be a **guest writer** for JavaCod...
your writing skills!

Rob Ewaschuk . Trate de encontrar el tiempo para repasar estos recursos, las ideas presentadas allí no tienen precio.

Para resumir algunas de las mejores prácticas, cuando se activa una alerta, debería ser fácil entender por qué, por lo que es una buena idea mantener las reglas de alertas lo más simples posible. Una vez que la alerta se activa, alguien debe ser notificado y examinarlo. Como tal, las alertas deben indicar la causa real, ser procesables y significativas, las ruidosas deben evitarse a toda costa (y serán ignoradas de todos modos).

Por último, pero no menos importante, no importa cuántas métricas recopile, cuántos paneles y alertas haya configurado, siempre se perderá algo. Considere que este proceso es una mejora continua, reevalúe periódicamente sus decisiones de monitoreo, registro, rastreo distribuido, recopilación de métricas y alertas.

3. Monitoreo de infraestructura

El monitoreo de los componentes y capas de la infraestructura es un problema resuelto. Desde la perspectiva de código abierto, los nombres bien establecidos como Nagios , Zabbix , Riemann , OpenNMS e Icinga están gobernando allí y es muy probable que su equipo de operaciones ya esté apostando por uno de esos.

4. Monitoreo de aplicaciones

La infraestructura ciertamente cae en la categoría de "debe ser monitoreado", pero el lado de la aplicación del monitoreo es posiblemente mucho más interesante y más cercano al tema. Entonces, dirijamos la conversación hacia eso.

4.1 Prometeo y administrador de alertas

Ya hemos hablado de Prometheus , principalmente como un almacenamiento de métricas, pero el hecho es que también incluye el componente de alerta llamado AlertManager hace que regrese.

El AlertManager maneja las alertas enviadas por las aplicaciones cliente como el servidor de Prometeo. Se encarga de deduplicar, agrupar y enrutarlos a las integraciones correctas del receptor, como correo electrónico, PagerDuty u OpsGenie. También se encarga de silenciar e inhibir las alertas. - <https://prometheus.io/docs/alerting/alertmanager/>

En realidad, AlertManager es un proceso binario independiente que maneja las alertas enviadas por la instancia del servidor Prometheus . Dado que la plataforma **JCG Car Rentals** ha elegido a Prometheus como la métrica y la plataforma de monitoreo, también se convierte en una opción lógica para administrar las alertas.

Básicamente, hay algunos pasos a seguir. El procedimiento consiste en configurar y ejecutar la instancia de AlertManager , configurar Prometheus para hablar con esta instancia de AlertManager y finalmente definir las reglas de alerta en Prometheus . Dando un paso a la vez, comencemos primero con la configuración de AlertManager .

```
01 global:
02   resolve_timeout: 5m
03   smtp_smarthost: 'localhost:25'
04   smtp_from: 'alertmanager@jcg.org'
05 route:
06   receiver: 'jcg-ops'
07   group_wait: 30s
08   group_interval: 5m
09   repeat_interval: 1h
10   group_by: [cluster, alertname]
11   routes:
12   - receiver: 'jcg-db-ops'
13     group_wait: 10s
14     match_re:
15       service: postgresql|cassandra|mongodb
dieciséis receivers:
17 - name: 'jcg-ops'
18   email_configs:
19   - to: 'ops-alerts@jcg.org'
20   - name: 'jcg-db-ops'
21     email_configs:
22     - to: 'db-alerts@jcg.org'
```

Si suministramos este fragmento de configuración al proceso AlertManager (generalmente almacenándolo en el

```
alertmanager.yml
```

), debería comenzar con éxito, exponiendo su interfaz web en el puerto

```
9093
```

22,99 €

48,99 €

Custom matcher, e.g. `env="production"`

Active Pending Expired

No silences found

Interfaz web AlertManager

Excelente, ahora tenemos que decirle a Prometheus dónde buscar la instancia de AlertManager. Como de costumbre, se realiza a través del archivo de configuración.

```

1 rule_files:
2   - alert.rules.yml
3
4 alerting:
5   alertmanagers:
6     - static_configs:
7       - targets:
8         - 'alertmanager:9093'
```

El fragmento anterior también incluye la parte más interesante, las reglas de alerta, y esto es lo que vamos a saquear a continuación. Entonces, ¿cuál sería un ejemplo bueno, simple y útil de alerta significativa en el contexto de la plataforma **JCG Car Rentals**? Dado que la mayoría de los servicios de **JCG Car Rentals** se ejecutan en JVM, el primero que viene a la mente es el uso del montón: acercarse demasiado al límite es una buena indicación de un problema y una posible pérdida de memoria.

```

01 groups:
02 - name: jvm
03   rules:
04   - alert: JvmHeapIsFillingUp
05     expr: jvm_memory_used_bytes{area="heap"} / jvm_memory_max_bytes{area="heap"} > 0.8
06     for: 5m
07     labels:
08       severity: warning
09     annotations:
10       description: 'JVM heap usage for {{ $labels.instance }} of job {{ $labels.job }} is close to 80% for 1
11       summary: 'JVM heap for {{ $labels.instance }} is filling up'
```

Las mismas reglas de alerta se pueden ver en Prometheus usando la vista **Alertas**, confirmando que la configuración se ha recogido correctamente.

Prometheus Alerts Graph Status Help

Alerts

Show annotations

/prometheus-config/alert.rules.yml > jvm

JvmHeapIsFillingUp (0 active)

```

alert: JvmHeapIsFillingUp
expr: jvm_memory_used_bytes{area="heap"}
    / jvm_memory_max_bytes{area="heap"} > 0.8
for: 5m
labels:
  severity: warning
annotations:
  description: JVM heap usage for {{ $labels.instance }} of job {{ $labels.job }}
    is close to 80% for last 5 minutes.
  summary: JVM heap for {{ $labels.instance }} is filling up
```

Vista de alertas en Prometeo

Una vez que se activa la alerta, aparecerá en AlertManager de inmediato, al mismo tiempo que se notifica a todos los destinatarios afectados (los receptores). En la siguiente imagen, puede ver el ejemplo de la

JvmHeapIsFillingUp

alerta activada .

Alertmanager Alerts Silences Status Help

New Silence

Filter Group Receiver: All Silenced Inhibited

alertname="JvmHeapIsFillingUp" Silence

Custom matcher, e.g. `env="production"`

Expand all groups

alertname="JvmHeapIsFillingUp" 1 alert

17:56:17, 2019-10-13 (UTC) Info Source Silence

application="reservation-service" area="heap" id="G1 Old Gen" instance="192.168.0.104:58900" job="jcg-car-rentals" severity="warning"

Alerta activada en AlertManager

Netflix Chronograf

Si la pila TICK te suena familiar es porque apareció en nuestro radar en la parte anterior del tutorial. Uno de los componentes de la pila TICK (que corresponde a la letra **C** en la abreviatura) es Chronograf .

Chronograf proporciona una interfaz de usuario para Kapacitor , un motor de procesamiento de datos nativo que puede procesar tanto datos de flujo como lotes de InfluxDB . Puede crear alertas con una simple interfaz de usuario paso a paso y ver su historial de alertas en Chronograf .

<https://www.influxdata.com/time-series-platform/chronograf/>

El InfluxDB 2.0 (aún en alfa), el futuro de la pila InfluxDB y TICK en general, incorporará Chronograf en su plataforma de series de tiempo.

4.3 Atlas de Netflix

Netflix Atlas , el último de los viejos lectores de los que hemos hablado antes, también tiene soporte para alertas integradas en la plataforma.

4.4 Hawkular

Comenzando desde Hawkular , uno de los proyectos de la comunidad de Red Hat , estamos apagando los cambios a las soluciones de monitoreo de código abierto todo en uno dedicadas.

Hawkular es un conjunto de proyectos de código abierto (licencia Apache v2) diseñados para ser una solución genérica para problemas de monitoreo comunes. Los proyectos Hawkular brindan servicios REST que pueden utilizarse para todo tipo de necesidades de monitoreo.

<https://www.hawkular.org/overview/>

La lista de los componentes de Hawkular incluye soporte para alertas, recopilación de métricas y rastreo distribuido (basado en Jaeger).

4.5 Stagemonitor

Stagemonitor es un ejemplo de la solución de monitoreo dedicada específicamente a aplicaciones de servidor basadas en Java.

Stagemonitor es un agente de monitoreo de Java que se integra estrechamente con bases de datos de series temporales como Elasticsearch , Graphite e InfluxDB para analizar métricas graficadas y Kibana para analizar solicitudes y pilas de llamadas. Incluye paneles preconfigurados de Grafana y Kibana que se pueden personalizar.

<https://github.com/stagemonitor/stagemonitor>

De manera similar a Hawkular , viene con rastreo distribuido, métricas y soporte de alertas listo para usar . Además, dado que se dirige solo a aplicaciones Java, también se respaldan muchas de las ideas específicas de Java en la plataforma.

4.6 Grafana

Puede sonar menos esperado, pero Grafana no solo es una herramienta de visualización impresionante, sino que a partir de la versión

4.0

viene con un motor de alerta y reglas de alerta propias . Las alertas en Grafana están disponibles en el nivel de panel por panel (solo gráficos en este momento) y al guardar, las reglas de alerta se extraerán en un almacenamiento separado y se programarán para su evaluación. Para ser honesto, existen ciertas restricciones que hacen que las alertas de Grafana sean de uso limitado.

4.7 Alertas adaptativas

Hasta ahora hemos hablado de enfoques más o menos tradicionales de alerta, basados en métricas, reglas, criterios o expresiones. Sin embargo, técnicas más avanzadas como la detección de anomalías están llegando lentamente a los sistemas de monitoreo. Uno de los pioneros en este espacio es Adaptive Alerting by Expedia .

El objetivo principal de las alertas adaptativas es ayudar a reducir el tiempo medio de detección (MTTD). Lo hace escuchando la transmisión de datos métricos, identificando anomalías candidatas, validando para evitar falsos positivos y finalmente pasándolos a los sistemas de enriquecimiento y respuesta posteriores. -
<https://github.com/ExpediaDotCom/adaptive-alerting/wiki/Architectural-Overview>

El Adaptive Alerting está detrás del subsistema de detección de anomalías en el Haystack , un sistema de rastreo y análisis resistente y escalable del que hemos hablado en la parte anterior del tutorial.

5. Orquestación

Los orquestadores de contenedores regidos por las mallas de servicio son probablemente el modelo de implementación de microservicios más extendido en la actualidad. De hecho, la malla de servicio desempeña el papel del "cardenal en la sombra" que está a cargo y lo sabe todo. Al

Kiali proporciona métricas detalladas, y una integración básica de Grafana está disponible para consultas avanzadas. El rastreo distribuido se proporciona integrando Jaeger .

Kiali consolida la mayoría de los pilares de observabilidad en un solo lugar, combinándolo con la vista de topología en tiempo real de su flota de microservicios . Si no está usando Istio , entonces Kiali puede no serle de gran ayuda, pero otras mallas de servicio se están poniendo al día, por ejemplo, Linkerd también viene con funciones de telemetría y monitoreo .

¿Y qué hay de las alertas? Parece que las capacidades de alerta se omiten en este momento, y es posible que deba conectarse a Prometheus y / y Grafana usted mismo para configurar las reglas de alerta.

6. nube

La historia de la nube para alertas es una continuación lógica de la discusión que hemos comenzado al hablar sobre las métricas . Las mismas ofertas que se encargan de recopilar los datos operativos son las que administran las alertas.

En el caso de AWS , Amazon CloudWatch permite configurar las alarmas (la noción de alertas de AWS) y acciones automatizadas basadas en umbrales predefinidos o en algoritmos de aprendizaje automático (como la detección de anomalías, por ejemplo).

RF Spectrum Monitoring - N Germany / up to 20GHz

Ad

Remote-controlled Spectrum Ana
to 20GHz incl. Software for Win o
aaronia.com

Learn more

El Azure monitor , que respalda las métricas y los registros de recolección de Microsoft Azure , permite configurar diferentes tipos de alertas basadas en registros, métricas o actividades.

En la misma línea, Google Cloud agrupa las alertas en Stackdriver Monitoring , que proporciona la forma de definir la política de alertas: las circunstancias a las que se debe alertar y cómo recibir una notificación.

7. Sin servidor

Las alertas son tan importantes en el mundo de los servidores sin servidor como en cualquier otro lugar. Pero como ya nos dimos cuenta, las alertas relacionadas con los hosts, por ejemplo, ciertamente no están en su horizonte. Entonces, ¿qué está sucediendo en el universo sin servidor con respecto a las alertas?

En realidad no es una pregunta fácil de responder. Obviamente, si está utilizando la oferta sin servidor de los proveedores de la nube, debería estar cubierto (¿o limitado?) Por sus herramientas . En el otro extremo del espectro, tenemos marcos independientes que toman sus propias decisiones.

Por ejemplo, OpenFaas usa Prometheus y AlertManager, por lo que puede definir las alertas que necesite. Del mismo modo, Apache OpenWhisk expone una serie de métricas que podrían publicarse en Prometheus y decorarse aún más con reglas de alerta. El marco sin servidor viene con un conjunto de alertas preconfiguradas pero hay restricciones asociadas con su capa gratuita.

8. Las alertas no son solo sobre métricas

En la mayoría de los casos, las métricas son la única entrada alimentada en las reglas de alerta. En general, tiene sentido, pero hay otras señales que puede aprovechar. Consideremos los registros, por ejemplo. ¿Qué sucede si desea recibir una alerta si aparece algún tipo específico de excepción en los registros?

Desafortunadamente, ni Prometeo ni Grafana , Netflix Atlas , Chronograf o Stagemonitor lo ayudarían aquí. En una nota positiva, tenemos Hawkular, que puede examinar los registros almacenados en Elasticsearch y activar las alertas mediante la coincidencia de patrones. Además, Grafana Loki está haciendo un buen progreso hacia el soporte de alertas basadas en registros. Como último recurso, es posible que deba implementar su propia solución.

RF Spectrum Monitoring - N Germany / up to 20GHz

Ad

Remote-controlled Spectrum Ana
to 20GHz incl. Software for Win o
aaronia.com

Learn more

9. Microservicios: Monitoreo y Alerta - Conclusiones

10. Al final

¡Es cierto que fue un largo viaje! En el camino, revisamos tantos temas diferentes que es posible que tenga miedo de la arquitectura de microservicios. No temas más, hay enormes beneficios que trae sobre la mesa, sin embargo, también requiere que pienses en los sistemas que estás construyendo de una manera diferente. Esperemos que el final de este tutorial sea solo el comienzo de su viaje al apasionante mundo de la arquitectura de microservicios.

Pero mantén tus oídos abiertos. Sí, la arquitectura de microservicios no es una bala de plata. No lo compre como un argumento de venta ni caiga en la trampa exagerada. Resuelve los problemas reales, pero en realidad debería encontrarse con ellos antes de elegir microservicios como la solución. Por favor, no invierta esta fórmula simple.

¡Mucha suerte y con eso, feliz microservicio!

Etiquetado con: MICROSERVICIOS DEVOPS

(0 calificación, 0 votos)

Debe ser un miembro registrado para calificar esto. 🗨 Inicie el debate 👁 577 reproducciones i 🐦 Tweet!

¿Quieres saber cómo desarrollar tus habilidades para convertirte en un Rockstar de Java?



¡Suscríbete a nuestro boletín para comenzar a rockear

Para comenzar, ¡te ofrecemos nuestros eBooks más vendidos

GRATIS!

1. JPA Mini libro
2. Guía de resolución de problemas de JVM
3. Tutorial JUnit para pruebas unitarias
4. Tutorial de anotaciones de Java
5. Preguntas de la entrevista Java
6. Preguntas de la entrevista de primavera
7. Diseño de la interfaz de usuario de Android

y muchos más

Enter your e-mail...

☐ Acepto los Términos y la Política de privacidad

Registrate



¿TE GUSTA ESTE ARTÍCULO? LEER MÁS DE JAVA CODE GEEKS

RF Spectrum Monitoring - Made in Germany / up to 20GHz

Ad aaronia.com

Microservices for Java Developers: Metrics

javacodegeeks.com

API Development Simplified

Ad Postman

What are Microservices?

javacodegeeks.com

Benefits of Microservices

Ad Hazelcast

MicroServices using Spring Boot & Spring Cloud - Part 1 : Overview

javacodegeeks.com

Local Microservices: Breaking up the Jigsaw

javacodegeeks.com

Microservice Java Development Distributed

javacodegeeks.com

Deja una respuesta



Start the discussion...

Este sitio usa Akismet para reducir el spam. Aprenda cómo se procesan sus datos de comentarios .

✉ Suscribirse ▼

KNOWLEDGE BASE

[Courses](#)
[Examples](#)
[Minibooks](#)
[Resources](#)
[Tutorials](#)

PARTNERS

[Mkyong](#)

THE CODE GEEKS NETWORK

[.NET Code Geeks](#)
[Java Code Geeks](#)
[System Code Geeks](#)
[Web Code Geeks](#)

HALL OF FAME

["Android Full Application Tutorial" series](#)
[11 Online Learning websites that you should check out](#)
[Advantages and Disadvantages of Cloud Computing – Cloud computing pros and cons](#)
[Android Google Maps Tutorial](#)
[Android JSON Parsing with Gson Tutorial](#)
[Android Location Based Services Application – GPS location](#)
[Android Quick Preferences Tutorial](#)
[Difference between Comparator and Comparable in Java](#)
[GWT 2 Spring 3 JPA 2 Hibernate 3.5 Tutorial](#)
[Java Best Practices – Vector vs ArrayList vs HashSet](#)

ABOUT JAVA CODE GEEKS

JCGs (Java Code Geeks) is an independent online community focused on ultimate Java to Java developers resource center; targeted at the technical team lead (senior developer), project manager and junior developers. JCGs serve the Java, SOA, Agile and Telecom communities with daily news, domain experts, articles, tutorials, reviews, announcements, code snippets, source projects.

DISCLAIMER

Todas las marcas comerciales y marcas comerciales registradas que aparecen en Java Code Geeks son propiedad de sus respectivos dueños. Java es una marca comercial registrada de Oracle Corporation en los Estados Unidos y en otros países. Ejemplos Java Code Geeks no está conectado a Oracle Corporation patrocinado por Oracle Corporation.