



链滴

springboot 应用监控方案

作者: [2457081614](#)

原文链接: <https://ld246.com/article/1577185696409>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



简述

最近在学习 SpringBoot 应用监控方面知识，现在大多数的服务都是分布式架构，跟踪和监视这些服务非常重要，在此做下整理。

监控指标主要包括以下方面：

- 服务可用性
- 容错性
- 安全
- 数据收集

系统组成

1. 监控数据存储（时间序列数据库），有 [InfluxDB](#), [TimescaleDB](#), [Prometheus](#)等
2. 面板（显示），将存储的数据显示。
3. 数据拉取获推送组件。

这是最基本的组成，除此之外，我们还可以集成很多其他组件，如报警组件。

实现

在Spring Boot应用中，Spring Boot Actuator提供了监控数据的获取功能，开箱即用。因此我们只要在监控数据存储和可视化显示上选择方案。

监控数据存储方案

1.InfluxDB

Influxdb 是一个时间序列数据库，用于处理高负载读写。

1.安装InfluxDB

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

2.应用程序添加依赖

```
<dependency>  
  <groupId>io.micrometer</groupId>  
  <artifactId>micrometer-registry-influx</artifactId>  
</dependency>
```

3.配置文件

```
management:  
  metrics:  
    export:  
      influx:  
        db: customDbName  
        password: mySecret  
        uri: http://localhost:8086  
        user-name: myUserName
```

2.Prometheus

普罗米修斯是一个开源的监控和警报工具包，由SoundCloud编写。它的工作方式与influxdb略有不同。与其配置应用程序将度量发布到普罗米修斯，不如配置普罗米修斯定期轮询应用程序。

1、添加依赖

```
<dependency>  
  <groupId>io.micrometer</groupId>  
  <artifactId>micrometer-registry-prometheus</artifactId>  
</dependency>
```

接下来，我们必须将在Prometheus.yml文件中进行配置来轮询应用程序。

```
global:  
  scrape_interval: 60s  
  evaluation_interval: 60s  
scrape_configs:  
  - job_name: 'demo'  
    metrics_path: "/actuator/prometheus"  
    scrape_interval: 5s  
    static_configs:  
      - targets: ['129.204.148.xx:9999']
```

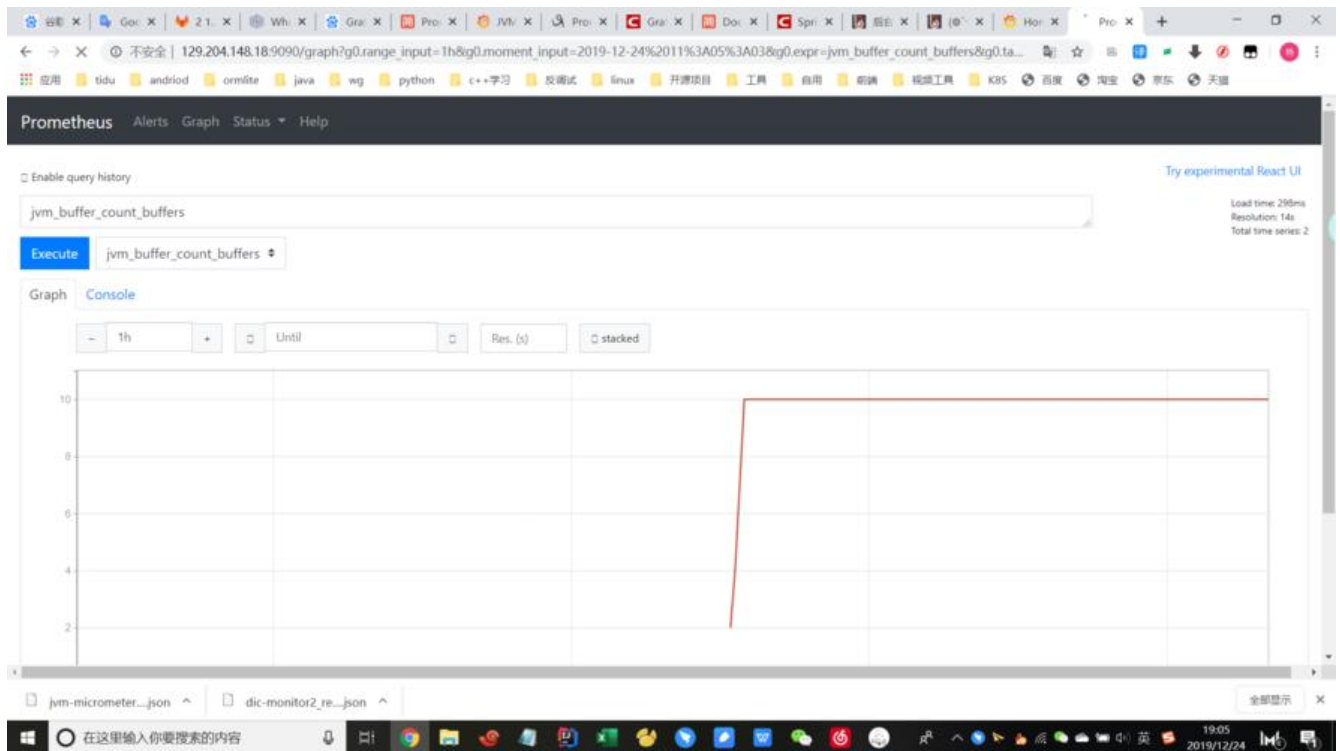
启动普罗米修斯，命令如下。

```
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 提供了自己的可视化工具来查看收集到的指标，通过URL <http://ip:9090/>访问。



3.Graphite

Graphite是另一个开源的时间序列数据库。它的体系结构比我们看到的其他数据库稍微复杂一些，但使用Docker，在本地运行一个实例很简单。

1、运行Graphite

```
docker run -d \  
--name graphite \  
--restart=always \  
-p 80:80 \  
-p 2003-2004:2003-2004 \  
-p 2023-2024:2023-2024 \  
-p 8125:8125/udp \  
-p 8126:8126 \  
graphiteapp/graphite-statsd
```

2、添加依赖、配置文件

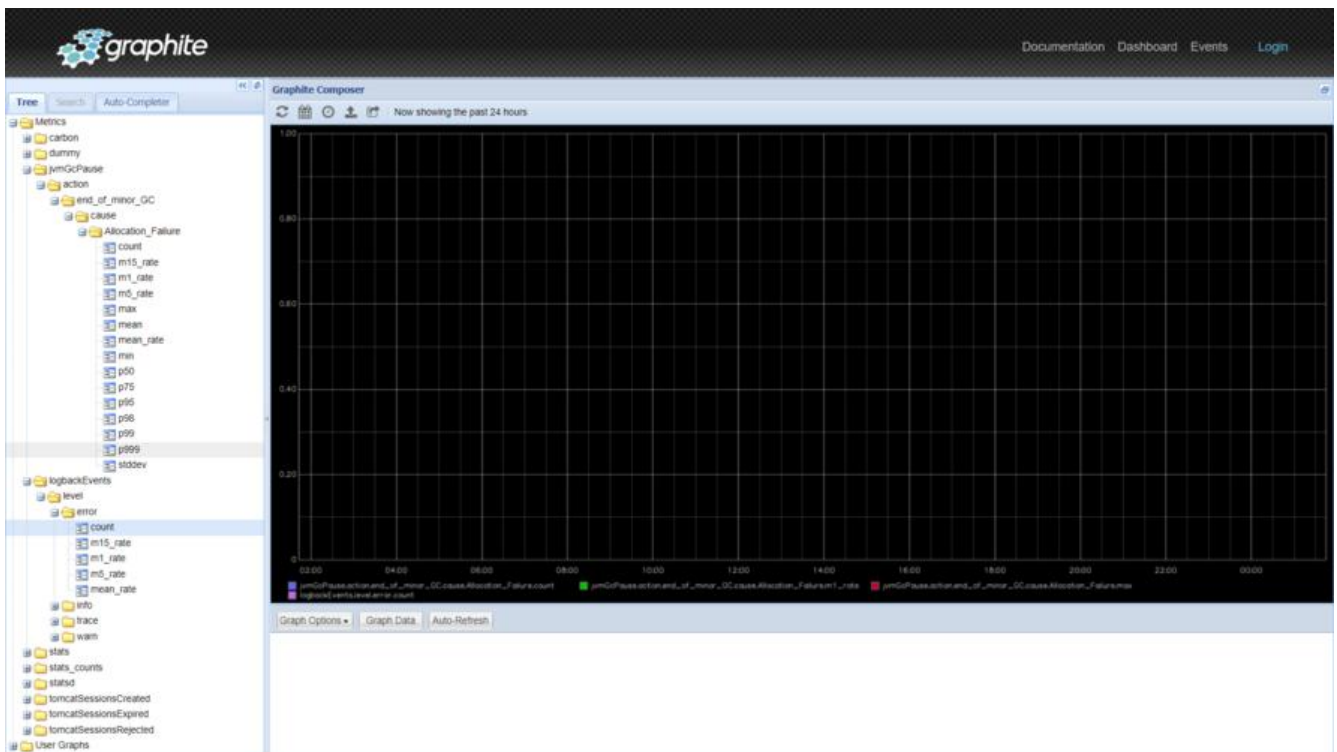
pom添加依赖：

```
<dependency>  
  <groupId>io.micrometer</groupId>  
  <artifactId>micrometer-registry-graphite</artifactId>  
</dependency>
```

application.properties添加如下配置：

```
management:
  metrics:
    export:
      graphite:
        host: 127.0.0.1
        port: 2004
```

和普罗米修斯一样，Graphite 也有自己的可视化界面。访问地址：<http://localhost/>。



可视化工具

一旦我们有了一个在我们的Spring Boot应用程序之外存储度量的解决方案，下一个决定就是我们如何可视化数据。Grafana 通常比大多数内置的可视化工具提供更好的配置和更好的警报。使用插件以很容易地扩展它，并且有许多预先构建的指示板可以导入以快速创建我们自己的可视化。

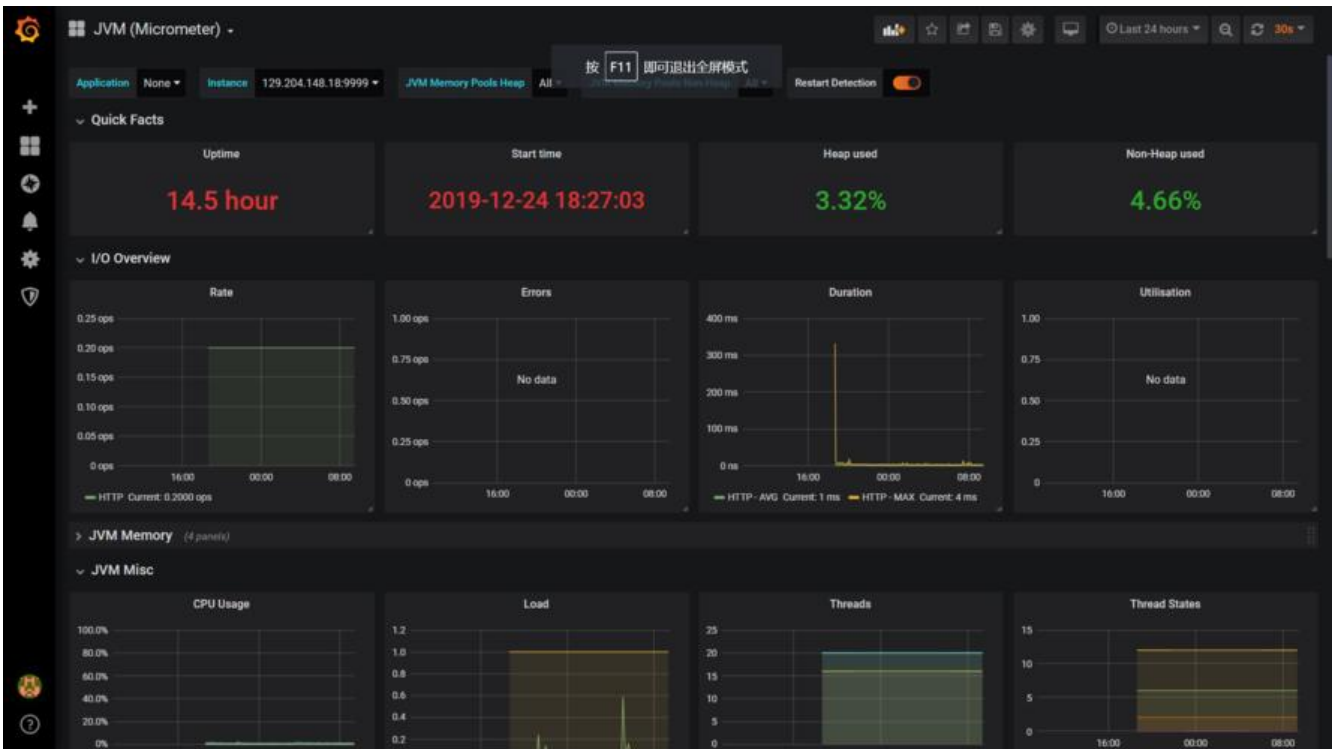
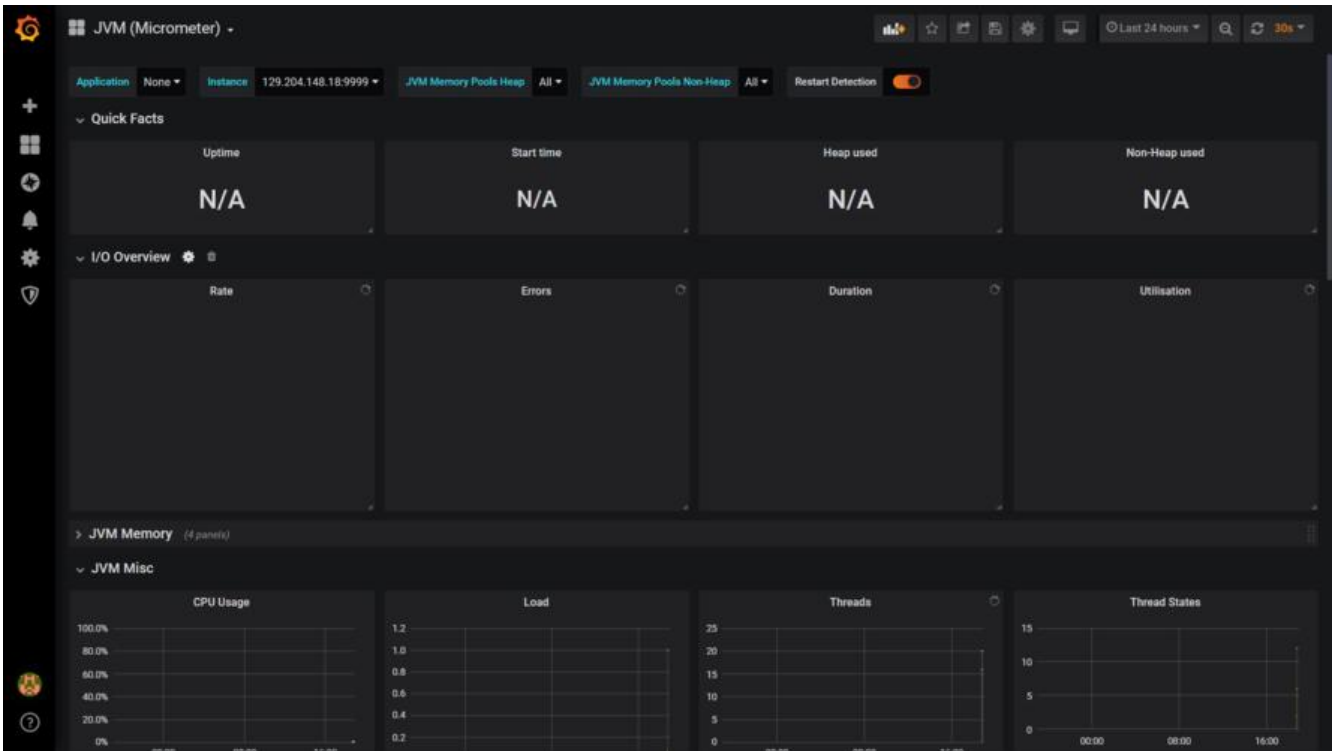
1.Grafana

Grafana 是一个开源分析和监控工具。它可以连接到前面提到的所有数据库，以及许多其他数据库。

1、安装

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

我们现在可以通过 URL <http://localhost:3000/> 访问 Grafana 主页。随后只需我们配置数据源就OK 我们可以导入面板JSON文件，无需自己手动添加配置。如下图所示。 [链接地址](#)



其他方案

Spring Boot Admin

具体实现--》 [springboot Admin](#)

spring boot admin是Springcloud全家桶的一员，它连接到spring boot实例、聚合节点等。但是，metrics端点并不是沿着时间线监视的，因此没有时间序列的监控数据。