Spring Boot的第三次课程

Starter组件

//详见源码

自动装配

- importSlector
- SpringFactoriesLoader
- @Configuration
- Conditional

Actuator(监控)

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-actuator</artifactId>
</dependency>
```

- https://docs.spring.io/spring-boot/docs/current/reference/html/production-ready-features.html#production-ready-endpoints

health 健康检测

Metrics

- JVM（垃圾收集器 / 内存/堆）
- 系统（运行时间、平均负载、处理器的信息）
- 线程池信息
- tomcat会话信息
- Pheuthous / Grafana(图标展示)

loggers

info

Actuator有两种形态的监控

- http（web）
- jmx

JMX（Java Management Extensions）

JMX全称是Java Management Extensions。 Java 管理扩展。 它提供了对Java应用程序和JVM的监控和管理功能。通过JMX，我们可以监控

1. 服务器中的各种资源的使用情况，CPU、内存
2. JVM内存的使用情况
3. JVM线程使用情况

SpringBoot的信息 可以发布到Prometheus+Grafana

Prometheus

开源的监控系统 。

- 数据采集 : http://localhost:8080/actuator/prometheus
- time-series 存储metrics
- 可视化 : http://locahost:9090

安装Prometheus

- 下载Prometheus，https://github.com/prometheus/prometheus/releases
- tar -zxvf prometheus-2.19.1.linux-amd64
- 修改prometheus.yml，增加需要监控的应用节点

```
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries
scraped from this config.
  - job_name: 'prometheus'

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
    - targets: ['localhost:9090']
  - job_name: 'spring-actuator'
    metrics_path: '/actuator/prometheus'
    scrape_interval: 5s
    static_configs:
    - targets: ['192.168.8.174:8080']  #需要监控的应用节点
```

  - job_name：任务名称
  - metrics_path： 指标路径
  - targets：实例地址/项目地址，可配置多个
  - scrape_interval： 多久采集一次
  - scrape_timeout： 采集超时时间
- 执行`./prometheus --config.file=prometheus.yml`启动prometheus应用，访问：http://HOST_IP:9090

- `nohup prometheus &`

Spring Boot集成

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

一旦你增加上述的依赖，Spring Boot会自动配置一个`PrometheusMeterRegistry`和`CollectorRegistry`来收集和输出格式化的metrics数据，使得Prometheus服务器可以爬取。

所有应用的metrics数据是根据一个叫`http://localhost:8762/actuator/prometheus`的endpoint来设置是否可用。

```
prometheus: {
  href: "http://localhost:8762/actuator/prometheus",
  templated: false
}
```

Prometheus服务器可以周期性的爬取这个endpoint来获取metrics数据。

安装Grafana

- 下载Grafana:https://grafana.com/grafana/download

```
wget https://dl.grafana.com/oss/release/grafana-7.0.3-1.x86_64.rpm
sudo yum install grafana-7.0.3-1.x86_64.rpm
```

- 启动Grafana，`service grafana-server start`
- 访问Grafana，`http://localhost:3000`，默认的帐号密码`admin/admin`

Grafana的配置

- 菜单选择`Configuration`->`Data Source`->`Add Data Source`
- 配置Prometheus作为数据源

下载别人配置好的面板直接导入

Grafana的面板配置过程还是比较繁琐的，如果我们不想自己去配置，那我们可以去Grafana官网上去下载一个dashboard。

> 推荐： https://grafana.com/grafana/dashboards/6756

下载完成后，在"+"这个菜单中，点击"import"，导入下载好的json文件即可。

根据ID进行load

- 模板地址：https://grafana.com/dashboards
- 在搜索框中搜索`Spring Boot`会检索出相关的模板，选择一个自己喜欢。
- 这里可以采用: `https://grafana.com/grafana/dashboards/10280`这个，看起来比较清晰
- 复制下图所示的dashboard的ID号