초격차 패키지 Online.

안녕하세요.
시그니처 백엔드 강의
Course 2. 백엔드 웹 개발 입문/실전을
진행하는 예상국입니다.

[Course 2] 백엔드 웹 개발 입문/실전

PART1 | 웹 개발 입문과 데이터베이스

PART2 | 웹 서비스 개발 실전

PART3 | 최신/심화 웹 개발 실전

모니터링

Application Log

slf4j

Simple Logging Facade for Java

Java의 로깅을 위한 추상화 (인터페이스) 계층, 해당 추상화를 구현한(logback, log4j) 구현체를 개발자가 선택 하여서 로그 시스템을 일관적으로 적용할 수 있도록 해줍니다.

Logback

slf4j의 구현체로써 다음과 같은 장점이 있습니다.

비동기 로깅:
Logback은 비동기 로거를 지원합니다. 이는 별도의 스레드가 로그 메시지를 처리하도록 하여, 로그 작성에 대한 지연이 애플리케이션의 주요 스레드에 영향을 미치지 않도록 합니다. 이는 애플리케이션의 성능에 중요한 영향을 미칠 수 있습니다.

효율적인 객체 생성:
Logback은 가능한 한 적은 수의 객체를 생성하려고 노력합니다. 이는 특히 가비지 컬렉션(GC) 오버헤드를 줄이는 데 도움이 됩니다.

배치 작업:
Logback은 가능한 한 많은 로그 이벤트를 함께 처리하려고 노력합니다. 이는 I/O 작업의 효율성을 높이고, 애플리케이션의 성능을 향상시킵니다.

Logback

최적화된 문자열 처리:
Logback은 로그 메시지를 생성하는 데 필요한 문자열 연결을 최소화하려고 합니다. 이는 로그 메시지의 생성 비용을 줄이는 데 도움이 됩니다.

조건부 로깅:
Logback은 로그 레벨 설정과 같은 기능을 통해 필요한 로그 메시지만 기록하도록 할 수 있습니다. 이는 불필요한 로그 메시지의 생성을 방지하고, 따라서 애플리케이션의 성능을 향상시킵니다.

유연한 구성:
Logback은 XML이나 Groovy를 통해 설정을 구성할 수 있습니다. 로그의 출력 형식, 로그 레벨, 로그를 기록하는 대상 등을 쉽게 설정하고 조정할 수 있습니다.

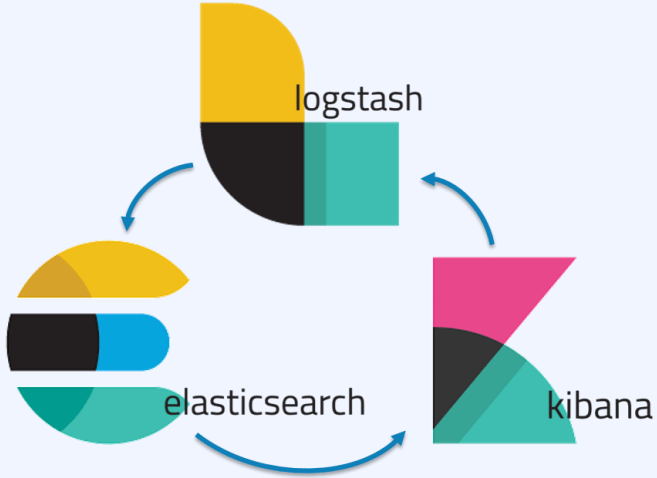
다양한 로그 대상 지원:
Logback은 콘솔, 파일, 데이터베이스 등 다양한 대상에 로그를 기록할 수 있습니다.

파일 로테이션:
로그 파일이 특정 크기에 도달하면 새 파일을 생성하고, 오래된 로그 파일을 삭제하는 로테이션 기능을 지원합니다.

Spring Application Log 모니터링

ELK Stack

ELK

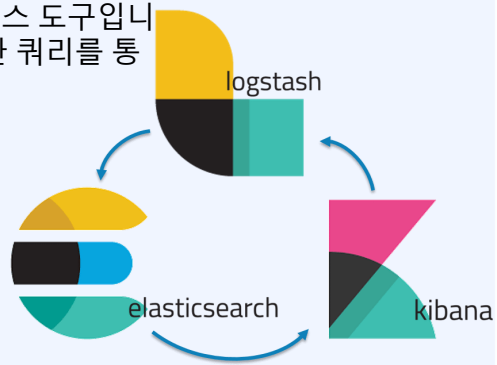


ELK

Elasticsearch
실시간 분산 검색 및 분석 엔진입니다. 복잡한 검색 쿼리를 처리하고, 대용량 데이터를 실시간으로 처리할 수 있는 능력이 있습니다. JSON 형태의 문서를 저장하고, 검색할 수 있으며, 강력한 분석 기능을 제공합니다. Elasticsearch는 내부적으로 Apache Lucene을 사용하며, 이는 강력한 풀 텍스트 검색 라이브러리입니다.

Logstash
서버로부터 다양한 소스의 로그 또는 이벤트 데이터를 수집, 처리하여 Elasticsearch와 같은 "스토리지"로 전송하는 파이프라인 도구입니다. Logstash는 다양한 유형의 입력 (파일, 비트, syslog 등)에서 데이터를 수집하고, 필터를 사용해 이 데이터를 분석 및 변환한 다음, Elasticsearch와 같은 출력에 이 데이터를 색인화합니다.

Kibana
저장된 데이터를 시각화하고, 이 데이터를 기반으로 고급 데이터 분석을 수행하는 웹 인터페이스 도구입니다. Kibana를 사용하면 히스토그램, 파이 차트, 선 그래프 등 다양한 시각화를 생성하고, 복잡한 쿼리를 통해 데이터를 탐색하고, 사전 정의된 대시보드를 통해 데이터를 확인할 수 있습니다.



Application Monitoring

Spring Boot 의 공식 모니터링 라이브러리 이다.

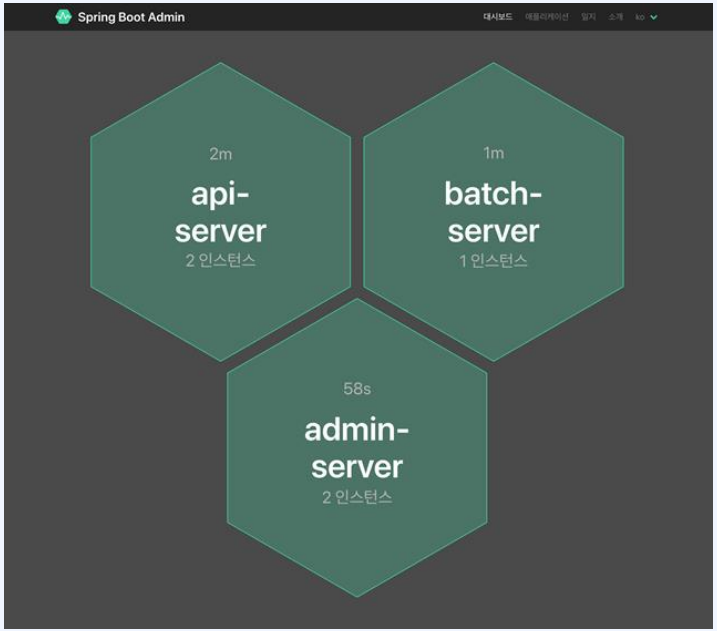
모니터링 대상은
Application 정보 : http server req, rest template req etc...
JVM 정보
Logback 정보
Process 정보
Statsd 정보
System 정보
Tomcat 정보

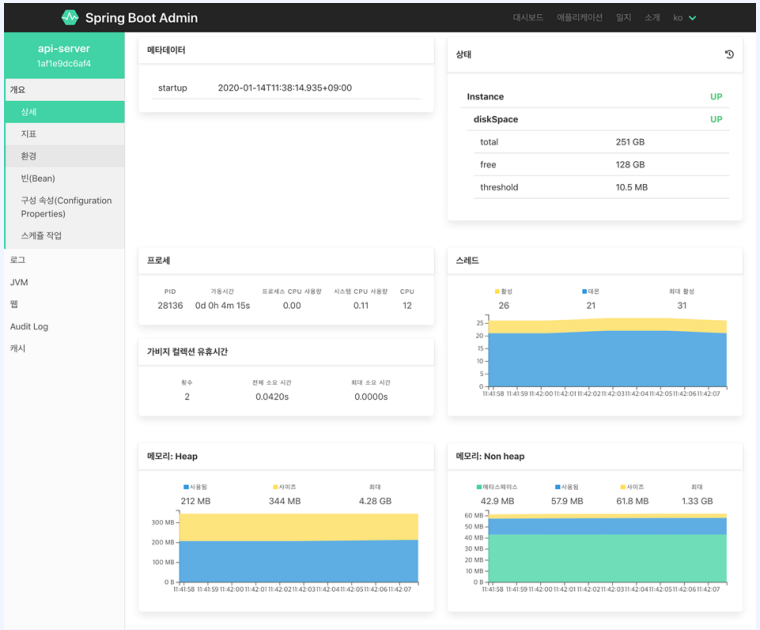
Application Monitoring

방식	장점	단점	비고	대표적 예
PUSH	모니터링 서버를 통해서 각 client 에 접근할 수 없음으로, 보안상 안전하다	Client가 추가 될 때마다, 각 Client에 모니터링 관련 설정을 적용해 주어야 한다	Client 에서 Server로 주기적으로 모니터링 데이터 발송	telegraf + influxDB+Chronograf
PULL	Client 가 추가 되어도 Server에만 모니터링 설정을 적용해주면 된다.	Server를 통해서 각 Client 에 접근이 가능하여 보안 취약점이 있으며, Client 서버가 바쁜경우 응답이 없을수도 있다.	Server에서 주기적으로 Client 로 모니터링 데이터 수집	Prometheus + Grafana

Spring Boot Admin

Spring Boot Admin





Prometheus + Grafana

Prometheus + Grafana

Prometheus

시스템 모니터링 및 경고를 위해 설계되었으며, 메트릭을 수집하고 저장하도록 설계된 플랫폼입니다.
Prometheus는 본래 컨테이너화 된 환경을 위해 개발되었으므로 Kubernetes와 같은 플랫폼과 잘 맞습니다.

특징
다차원 데이터 모델, 강력한 쿼리 언어, 자가 발견 또는 서비스 디스커버리, 효과적인 스토리지 기능이 포함.

Grafana
시계열 분석을 위한 오픈 소스 플랫폼이며, 대시보드 생성, 시각화, 경고 설정, 그리고 여러 데이터베이스와의 통합 등 다양한 기능을 제공합니다. Prometheus는 메트릭 수집에 맞춰진 플랫폼인 반면, Grafana는 수집된 데이터를 시각화하는데 더 중점을 두고 있습니다.

특징
다양한 데이터 소스를 지원합니다. (Prometheus, Graphite, Elasticsearch, InfluxDB 등)
유연하고 높은 수준의 시각화 옵션, 다양한 형식의 대시보드와 패널, 알림과 알림 채널을 설정

Prometheus + Grafana

[Monitoring 설정 방법]
1. Application Gradle dependencies 추가

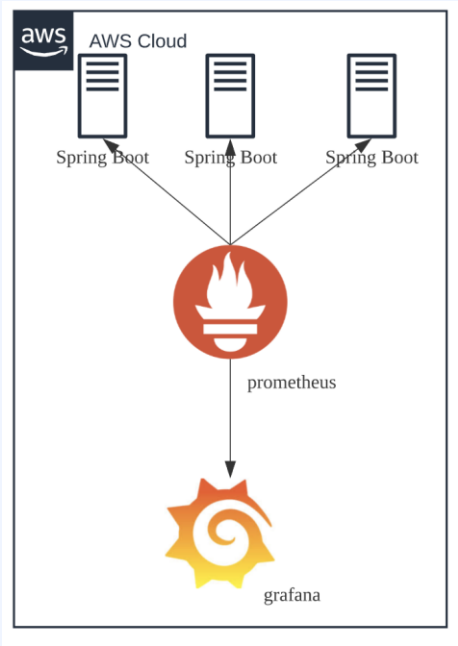
```
implementation("org.springframework.boot:spring-boot-starter-actuator")
implementation("io.micrometer:micrometer-registry-prometheus")
```

2. Application properties에 내용 추가

```
management.endpoint.metrics.enabled=true
management.endpoints.web.exposure.include=*
management.endpoint.prometheus.enabled=true
management.metrics.export.prometheus.enabled=true
```

3. Prometheus config 추가

```
global:
  scrape_interval:     10s
  evaluation_interval: 10s
  scrape_configs:
    - job_name: 'spring-app-demo'
    metrics_path: '/actuator/prometheus'
    static_configs:
      - targets: [‘${spring application host:port}']
```



TICK Stack

Tick Stack

Telegraf: InfluxData에서 개발한 플러그인 기반 서버 에이전트입니다. 이 에이전트는 데이터를 수집하고, 변환하고, 여러 소스로 보내는 역할을 합니다. 수백 개의 미리 만들어진 플러그인을 사용하여 여러 소스로부터 데이터를 수집하거나 자신의 플러그인을 작성할 수 있습니다.

InfluxDB: TICK 스택의 핵심 요소로, 고성능의 시계열 데이터베이스입니다. 이 데이터베이스는 대량의 시간 간격 데이터를 수집, 저장하며, 이러한 데이터의 고속 쿼리를 지원합니다.

Chronograf: InfluxDB 데이터를 시각화하고 대시보드를 만드는 도구입니다. 이 또한 사용자가 InfluxDB를 관리하고, 데이터베이스 쿼리를 생성하고, 경고를 설정할 수 있게 해줍니다.

Kapacitor: Kapacitor는 데이터 처리 엔진으로, InfluxDB에 저장된 데이터에 대한 복잡한 ETL 작업을 수행하거나 경고를 생성하는 역할을 합니다.

TICK 스택은 서버 모니터링, IoT 데이터 분석, 실시간 애플리케이션 모니터링 등 다양한 용도로 사용될 수 있습니다. 이 도구들은 각각 독립적으로 사용할 수 있지만, 함께 사용하면 강력한 시계열 데이터 플랫폼을 구축할 수 있습니다.

Tick Stack

1. Application Gradle dependencies 추가

```
implementation("io.micrometer:micrometer-registry-statsd")
implementation("org.springframework.boot:spring-boot-starter-actuator")
```

2. Application properties 내용 추가

```
management.endpoint.metrics.enabled=true
management.endpoints.web.exposure.include=*
management.metrices.export.statsd.enabled=true
management.metrices.export.statsd.flavor=telegraf
management.metrices.export.statsd.port=${telegraf port}
management.metrices.export.statsd.host=${telegraf host}
```