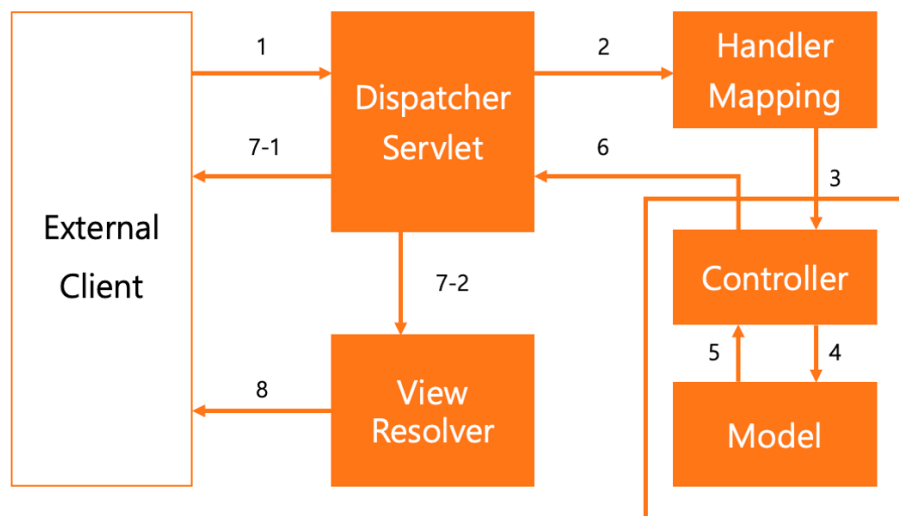
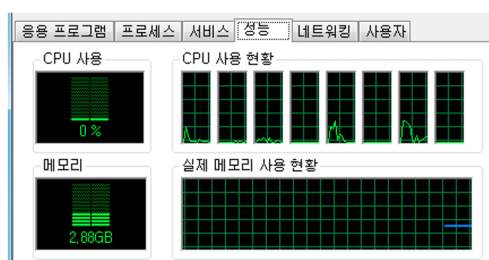


Chapter 7-4 Spring Boot Actuator

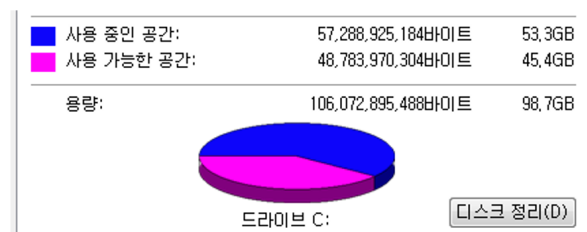
상용 서비스 준비



여태까지의 작업은 기능을 제공하는데 집중

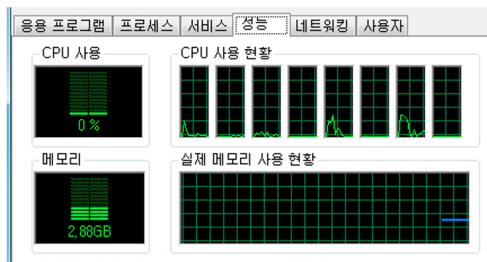


컴퓨터 메모리 사용량

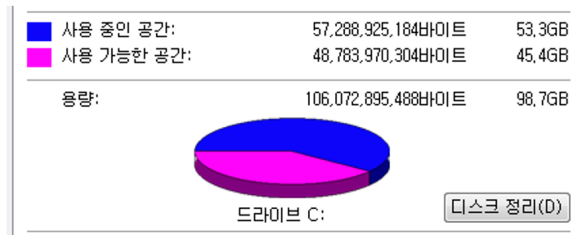


컴퓨터 디스크 공간

서비스가 안정적으로 돌아가는데 필요한 기능



컴퓨터 메모리 사용량



컴퓨터 디스크 공간

Spring Boot Actuator

Actuator와 Prometheus



Prometheus - 세계적으로 많이 사용하는 모니터링 도구

- 다양한 계측 정보를 HTTP 요청을 통해 받아온다.
- 계측 정보에 대한 내용을 GUI로 표시하고,
- 위험 상황에 대한 알림을 보여준다.



시각화 도구

Actuator에서 제공

```
spring:
  profiles:
    active: test

management:
  endpoints:
    enabled-by-default:true
    web:
  exposure:
    include: health,info,loggers
  endpoint:
    health:
      show-details:always
      show-components:always
```

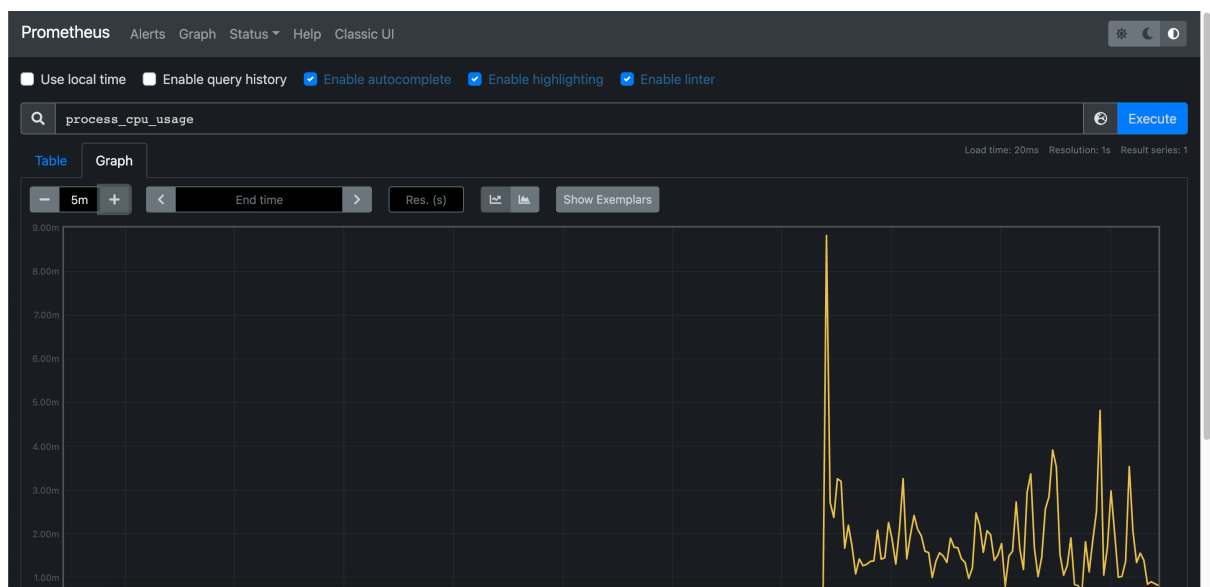
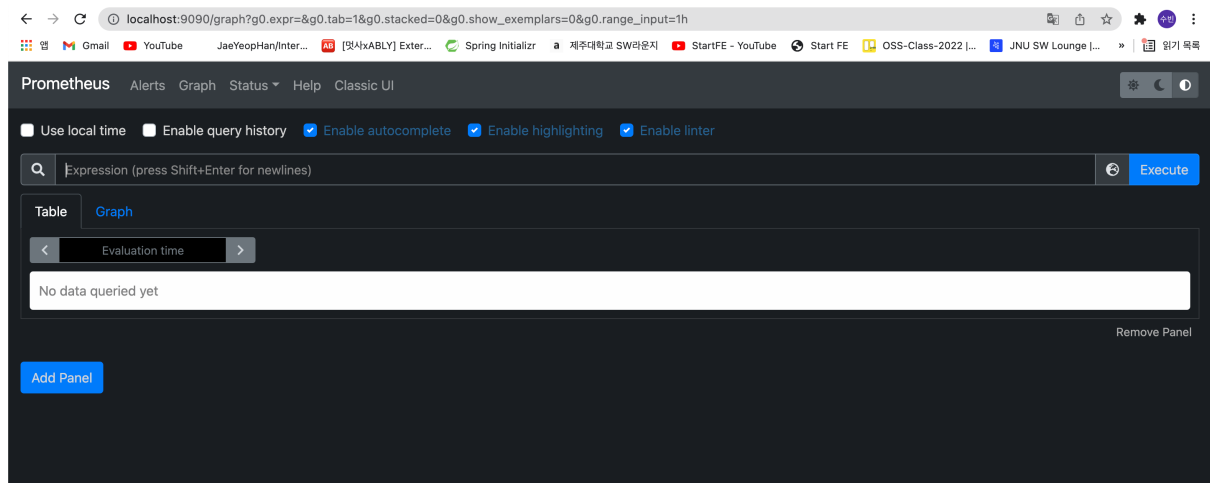
Prometheus

오픈소스 모니터링 툴로 지표 수집을 통한 모니터링이 주요 기능이다. 쿠버네티스 뿐만 아니라 애플리케이션이나 서버, OS 등 다양한 대상으로부터 **지표(Metric)**를 수집하여 모니터링할 수 있다. 기본적으로 **Pull 방식**으로 데이터를 수집하는데, 이 말은 모니터링 대상이 되는 자원이 지표정보를 프로메테우스로 보내는 것이 아니라, 프로메테우스가 주기적으로 모니터링 대상에서 지표를 읽어온다는 뜻이다(Push 방식으로 지표를 수집하는 모니터링 툴은 ELK스택 또는 Telegraf & InfluxDB 등이 있다). Pull 방식으로 지표정보를 읽어올 때는 각 서버에 설치된 Exporter를 통해서 정보를 읽어오며, 배치나 스케줄 작업의 경우에는 필요한 경우에만 떠 있다가 작업이 끝나면 사라지기 때문에 Pull 방식으로 데이터 수집이 어렵다. 그럴 경우 Push방식을 사용하는 Push gateway를 통해 지표정보를 받아오는 경우도 있다. 서버의 갯수가 정해져 있다면 프로메테우스에서 모니터링 대상을 관리하는데 어려움이 없지만, 오토스케일링이 많이 사용되는 클라우드 환경이나 쿠버네티스 클러스터에서는 모니터링 대상의 IP가 동적으로 변경되기 때문에 이를 일일이 설정파일에 넣는데 한계가 있다. 이러한 문제를 해결하기 위해 프로메테우스는 DNS나 Consul, etcd와 같은 다양한 서비스 디스커버리 서비스와 연동을 통해 모니터링 목록을 가지고 모니터링을 수행한다.

```
(anaconda3-2020.11) [X] ~/Documents/JavaSpringBoot/The-Origin-Java-Spring-Boot/Week7/jpa [main] docker run -d
-v /Users/d0dduck/Documents/JavaSpringBoot/The-Origin-Java-Spring-Boot/Week7/jpa/prometheus.yml:/etc/prometheus/promet
heus.yml -p 9090:9090 --name=prom-stub prom/prometheus
Unable to find image 'prom/prometheus:latest' locally
latest: Pulling from prom/prometheus
009932687766: Pull complete
ff9264fbb6f4: Pull complete
d0064201bd32: Pull complete
cbba1ac3d83b: Pull complete
ac0a3bd8622c: Pull complete
baa8a9fde418: Pull complete
efad3cab0966: Pull complete
Static_configs:
1fb8337d8a2d: Pull complete
517ed5dd8146: Pull complete
6d1fbb412502: Pull complete
349df8ccd572: Pull complete
f0b7fd883192: Pull complete
Digest: sha256:91100b06e86d724d5757e8f5a8920d1e22981067a5acf7eb2f47c3ae62be4ed2
Status: Downloaded newer image for prom/prometheus:latest
2aa398dfc28d514191c972f9c92242dc7cc0d1af24ec81cbe6dcb93017ed5a65
```

```
(anaconda3-2020.11) [main] docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
2aa398dfc28d	prom/prometheus	"/bin/prometheus --c..."	38 seconds ago	Up 33 seconds	0.0.0.0:9090->9090/tcp
270044b01491	mysql:8	"docker-entrypoint.s..."	3 weeks ago	Up 37 minutes	0.0.0.0:3306->3306/tcp, 33060/tcp
	some-mysql				



Targets

[All](#) [Unhealthy](#) [Collapse All](#)

spring boot scrape (1/1 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://host.docker.internal:8080/actuator/prometheus	UP	instance="host.docker.internal:8080" job="spring boot scrape"	635.000ms ago	6.003ms	