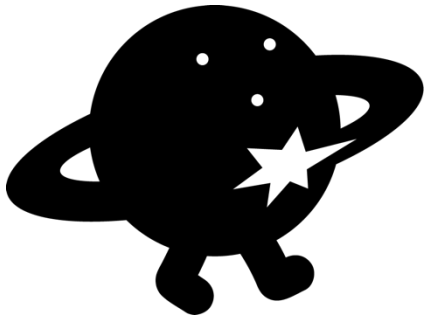
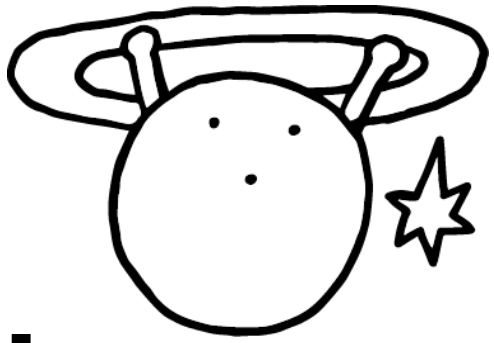
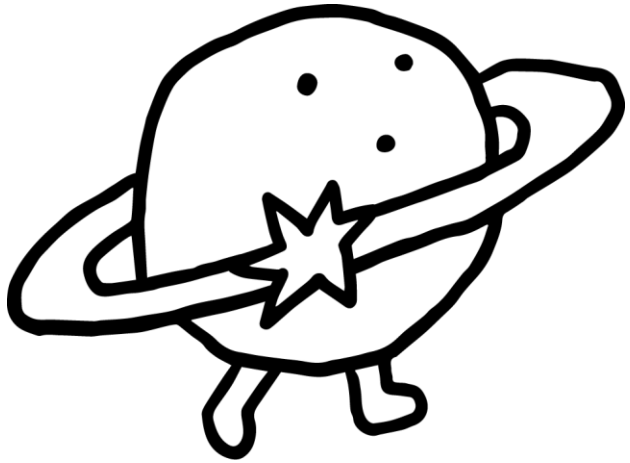


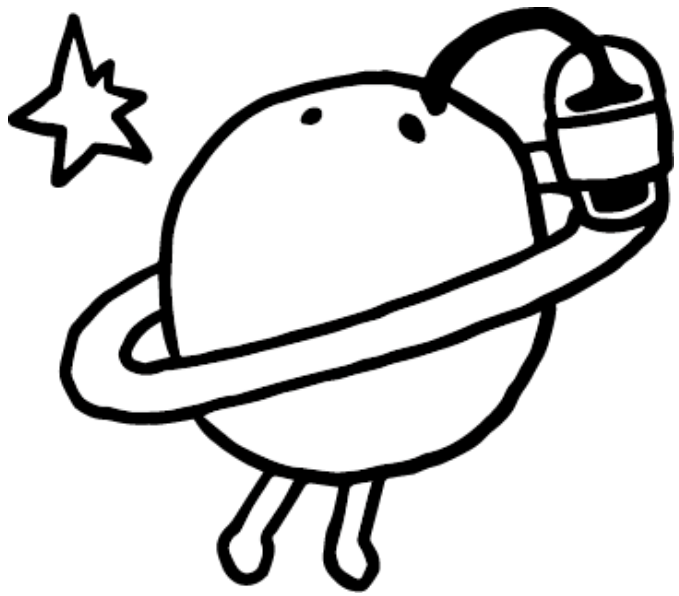


모니터링 이사하기: CloudWatch에서 Grafana로



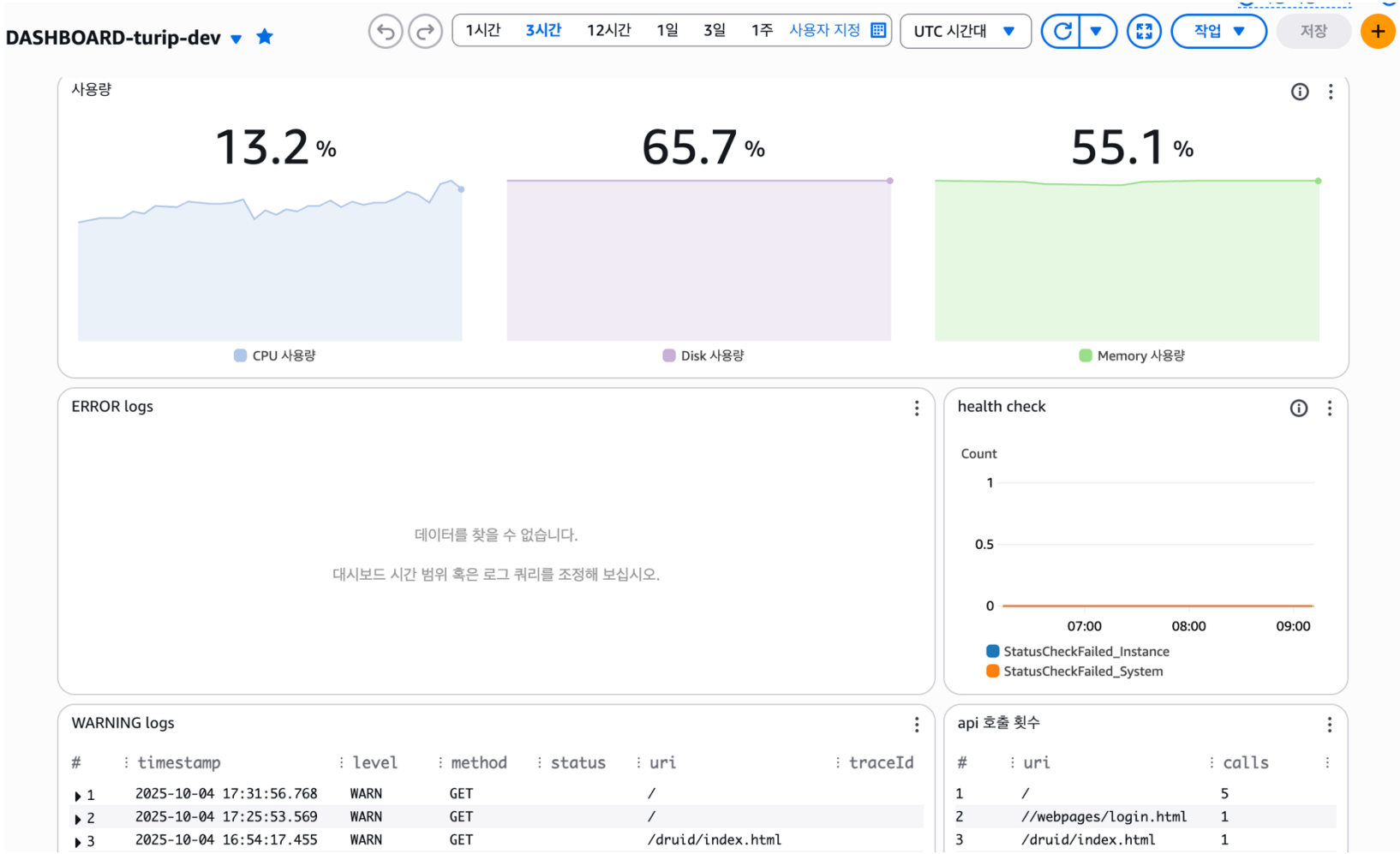
BE 7기 메이

현재
모니터링 방법



현재 모니터링

AWS CloudWatch



CloudWatch를 선택했던 이유

- Grafana, Prometheus 같은 거 쓰다가 설정만 3일 걸릴 수 있어요
  - AWS Cloudwatch로 빠르게 구성할 수도 있어요

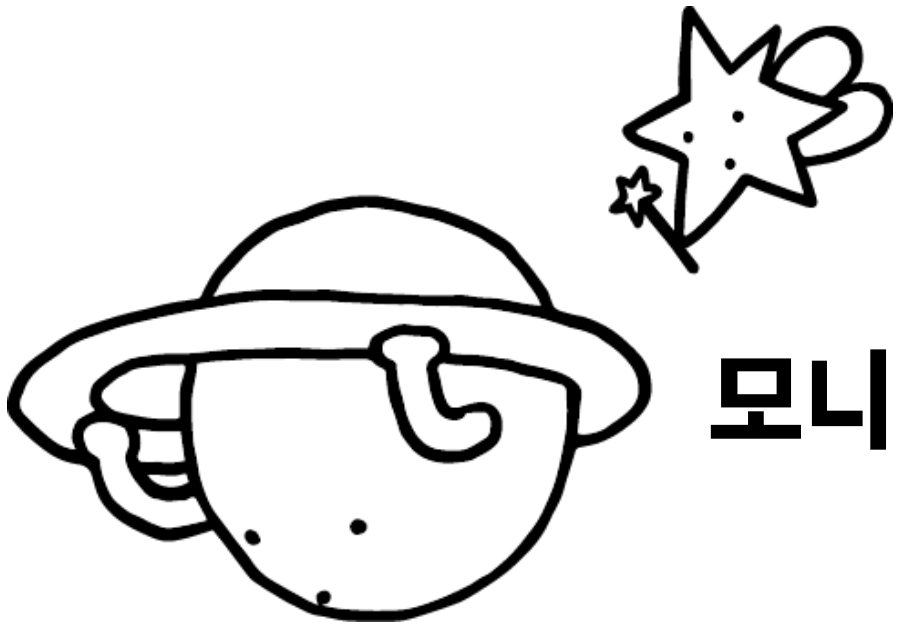
개선이 필요한 이유 1

AWS에 과도한 의존

개선이 필요한 이유 2

부하테스트

트레이스



모니터링 이사 시작!

아키텍처

모니터링 서버 ec2

Amazon EC2

애플리케이션 서버 ec2

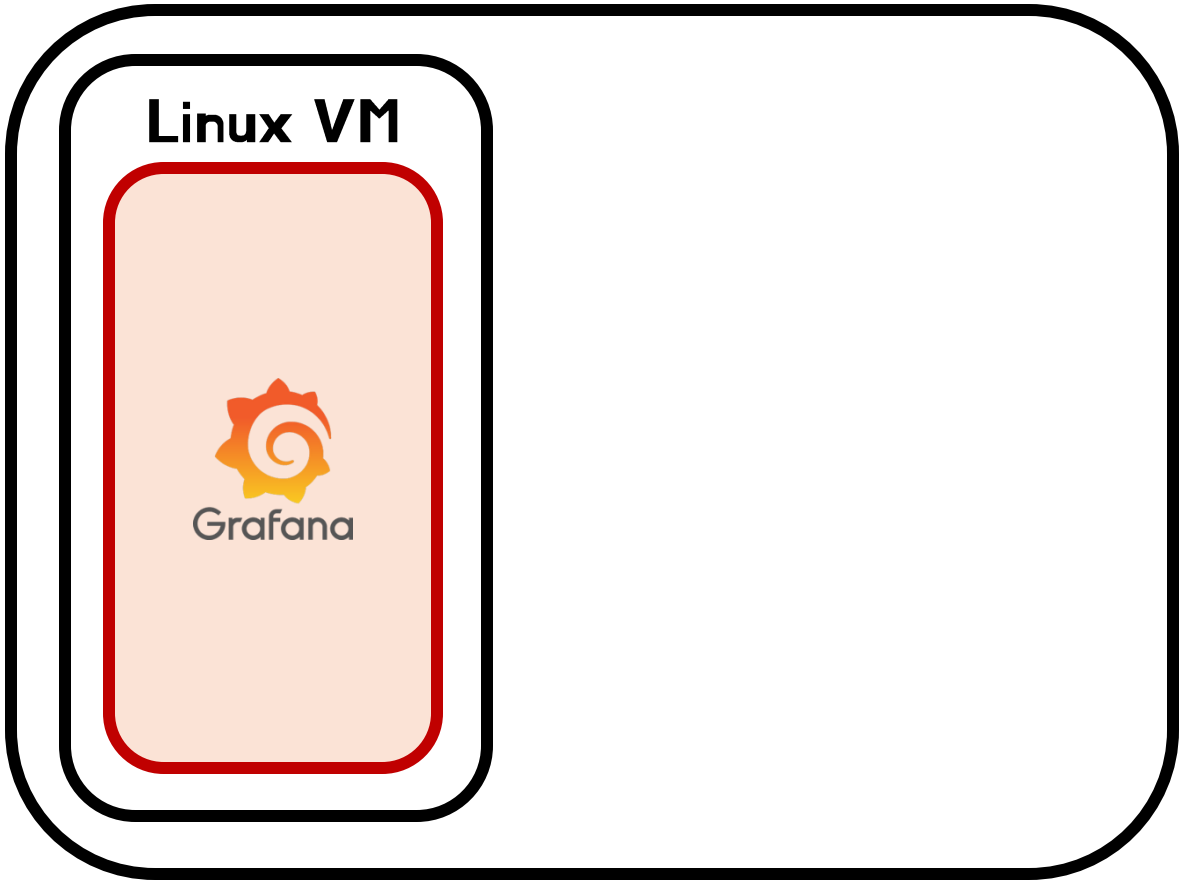
Amazon EC2

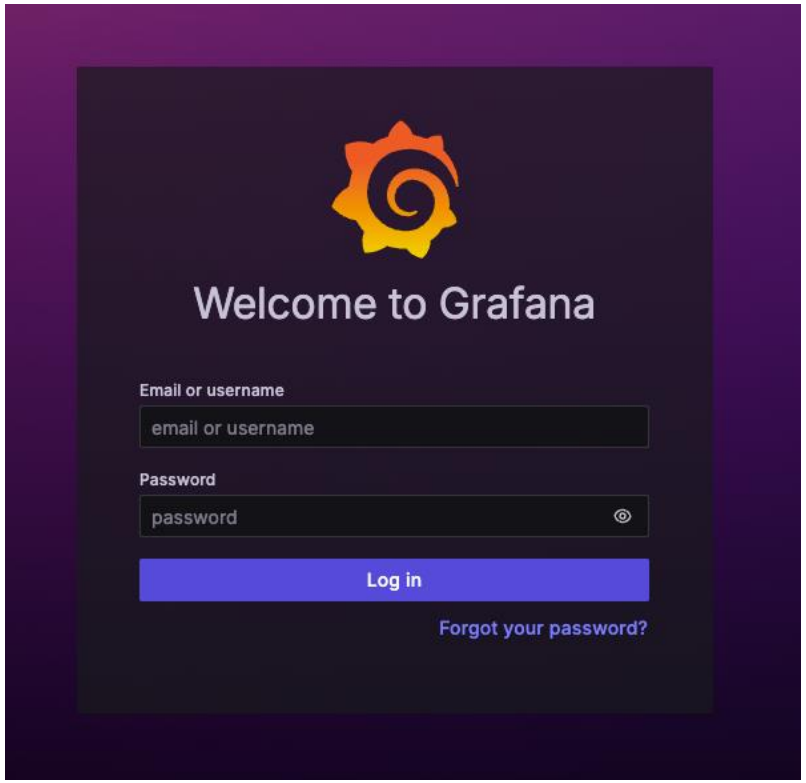
모니터링 ec2에 grafana 서버 띄우기

모니터링 서버
ec2

Amazon EC2

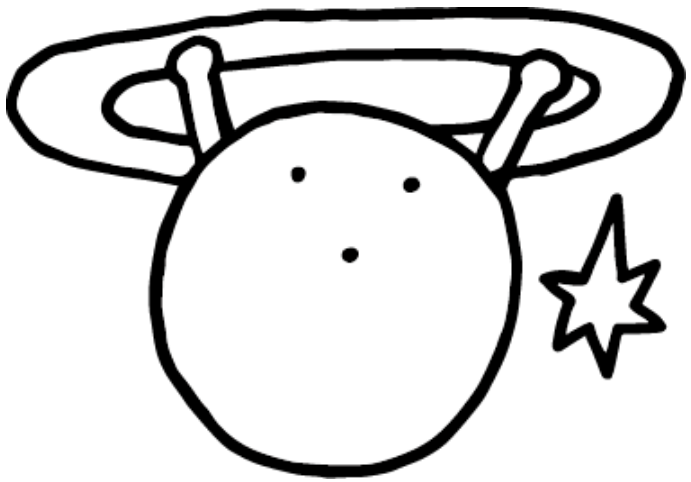
모니터링 EC2





http://<모니터링ip>:3000

메트릭 모니터링 이사하기



메트릭

애플리케이션
서버 ec2

<의존성 추가>

implementation "io.micrometer:micrometer-registry-prometheus"

<메트릭 수집 api 설정>

```
management:
  endpoints:
    web:
      exposure:
        include: prometheus
```

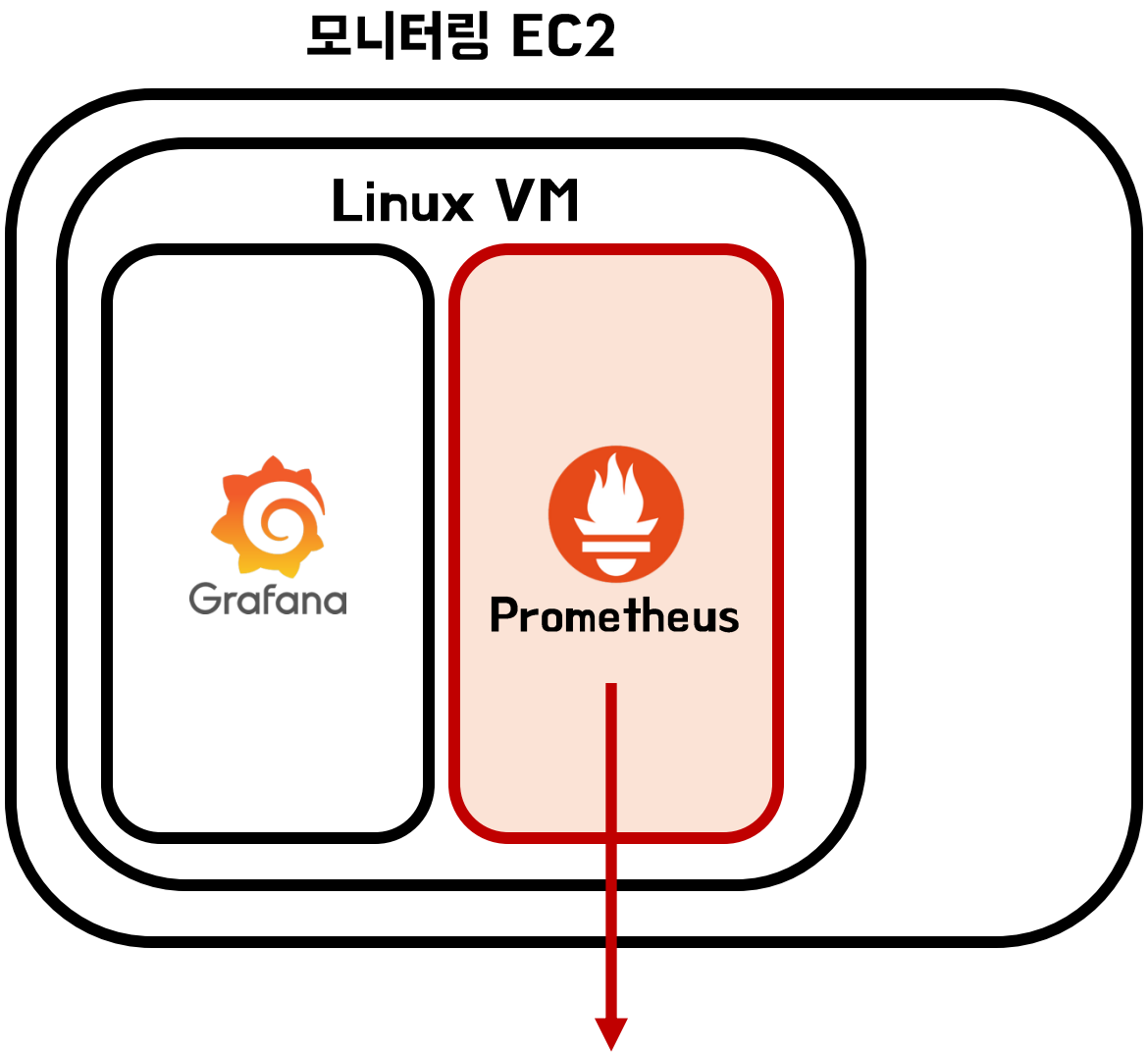

/actuator/prometheus ..... JVM 관련 메트릭 (CPU, Heap 사용량, GC 횟수 ...)

프로메테우스가 메트릭을 얻기 위해 접근하는 Api

메트릭

모니터링 서버
ec2

Amazon EC2

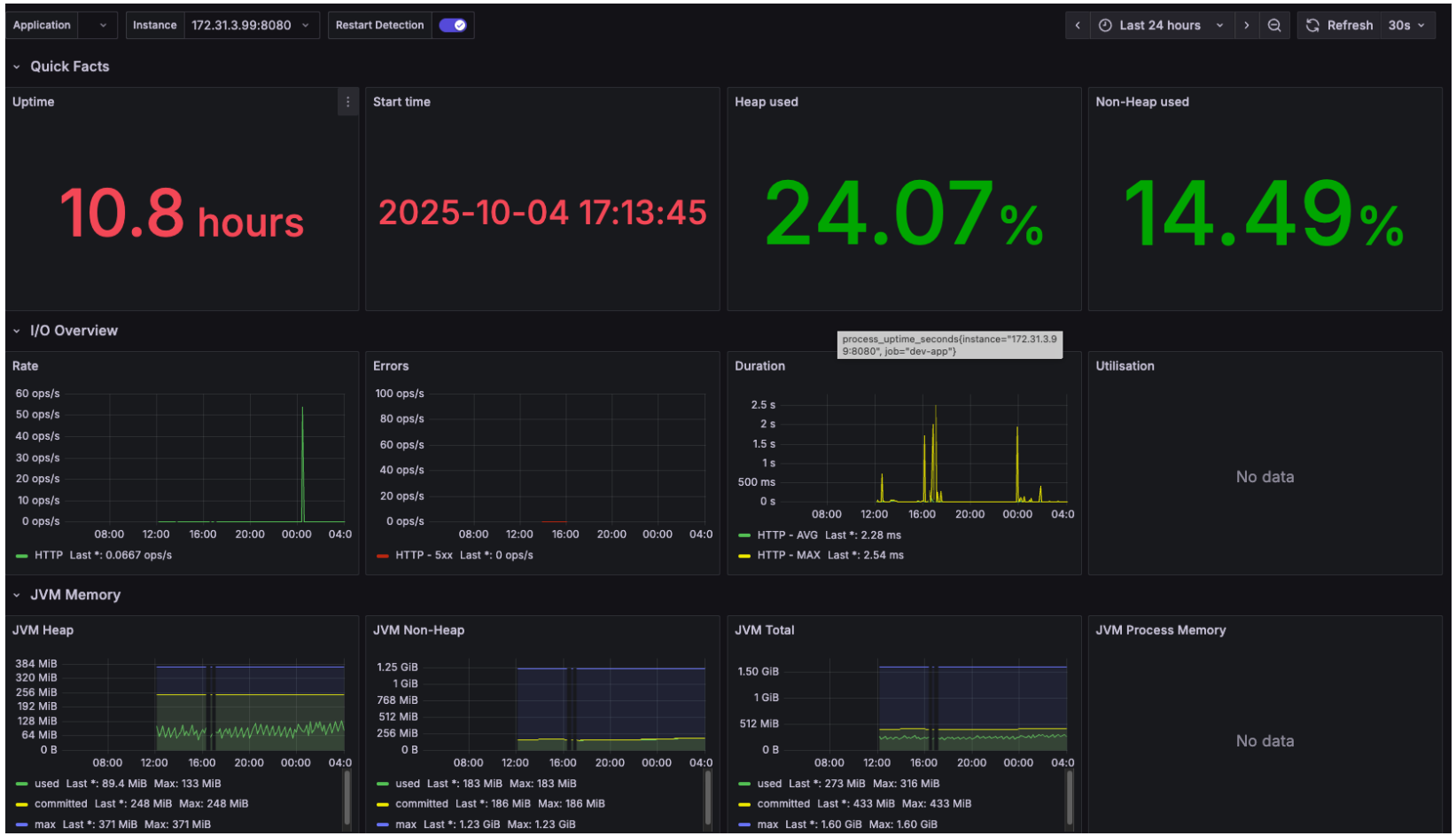


<prometheus.yml>

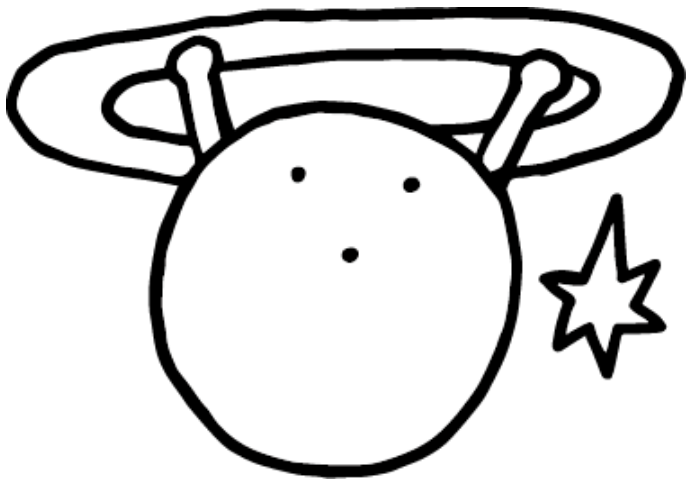
```
global:
  scrape_interval: 15s  # 15초마다 수집

scrape_configs:
  - job_name: 'dev-app'
    metrics_path: '/actuator/prometheus'
    static_configs:
      - targets: ['<스프링 서버 ec2 IP>:8080']
```

http://<스프링 서버 ec2 IP>:8080/actuator/prometheus

메트릭


Application
Instance
172.31.3.99:8080
Restart Detection
Last 24 hours
Refresh
30s
Quick Facts
Uptime
10.8 hours
Start time
2025-10-04 17:13:45
Heap used
24.07%
Non-Heap used
14.49%
I/O Overview
process_uptime_seconds{instance="172.31.3.99:8080", job="dev-app"}
Rate
60 ops/s
50 ops/s
40 ops/s
30 ops/s
20 ops/s
10 ops/s
0 ops/s
08:00 12:00 16:00 20:00 00:00 04:0
HTTP Last *: 0.0667 ops/s
Errors
100 ops/s
80 ops/s
60 ops/s
40 ops/s
20 ops/s
0 ops/s
08:00 12:00 16:00 20:00 00:00 04:0
HTTP - 5xx Last *: 0 ops/s
Duration
2.5 s
2 s
1.5 s
1 s
500 ms
0 s
08:00 12:00 16:00 20:00 00:00 04:0
HTTP - AVG Last *: 2.28 ms
HTTP - MAX Last *: 2.54 ms
Utilisation
No data
JVM Memory
JVM Heap
384 MiB
320 MiB
256 MiB
192 MiB
128 MiB
64 MiB
0 B
08:00 12:00 16:00 20:00 00:00 04:0
used Last *: 89.4 MiB Max: 133 MiB
committed Last *: 248 MiB Max: 248 MiB
max Last *: 371 MiB Max: 371 MiB
JVM Non-Heap
1.25 GiB
1 GiB
768 MiB
512 MiB
256 MiB
0 B
08:00 12:00 16:00 20:00 00:00 04:0
used Last *: 183 MiB Max: 183 MiB
committed Last *: 186 MiB Max: 186 MiB
max Last *: 1.23 GiB Max: 1.23 GiB
JVM Total
1.50 GiB
1 GiB
512 MiB
0 B
08:00 12:00 16:00 20:00 00:00 04:0
used Last *: 273 MiB Max: 316 MiB
committed Last *: 433 MiB Max: 433 MiB
max Last *: 1.60 GiB Max: 1.60 GiB
JVM Process Memory
No data

로그 모니터링 이사하기



로그

Spring Boot (Logback → 로그 파일)

↓ tail

Promtail (로그 수집기, 스프링 서버 EC2에 설치)

↓ push

Loki (로그 저장소, 모니터링 서버 EC2에 설치)

↓

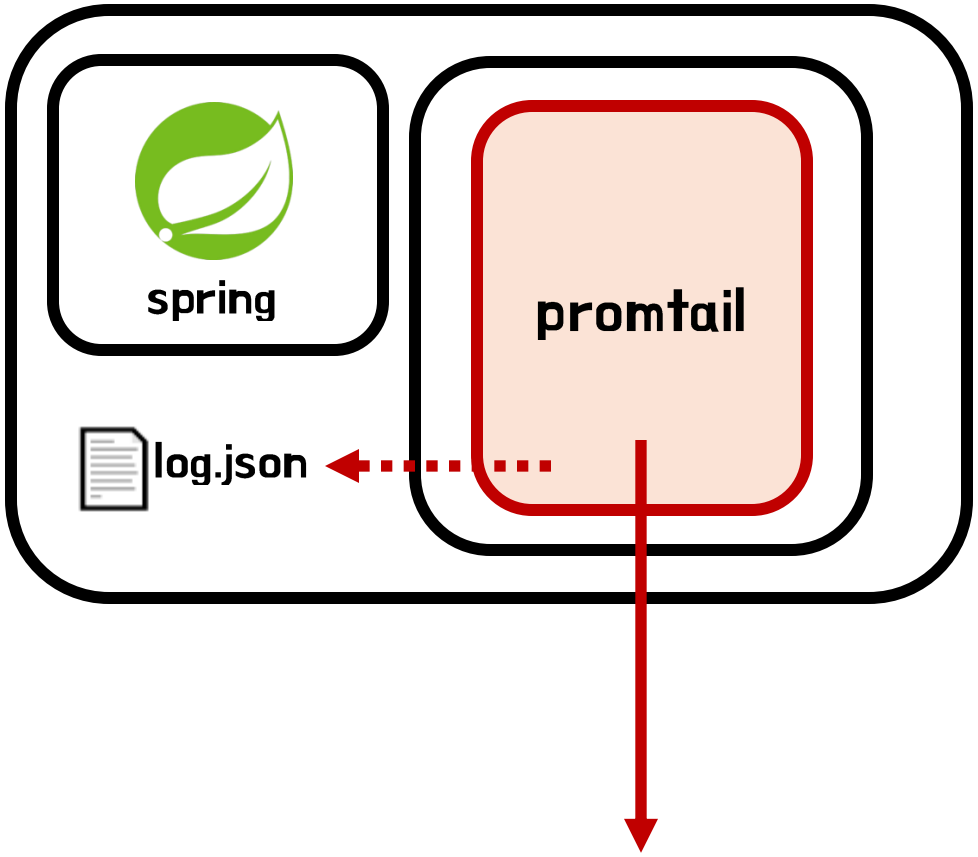
Grafana (Loki를 데이터소스로 등록 → Explore/대시보드에서 로그 확인)

로그

애플리케이션
서버 ec2

Amazon EC2

애플리케이션 서버
EC2



<promtail-config.yml>

```
server:
  http_listen_port: 9080
  grpc_listen_port: 0

positions:
  filename: /tmp/positions.yaml

clients:
  - url: http://<모니터링서버 ec2 IP>:3100/loki/api/v1/push

...
```

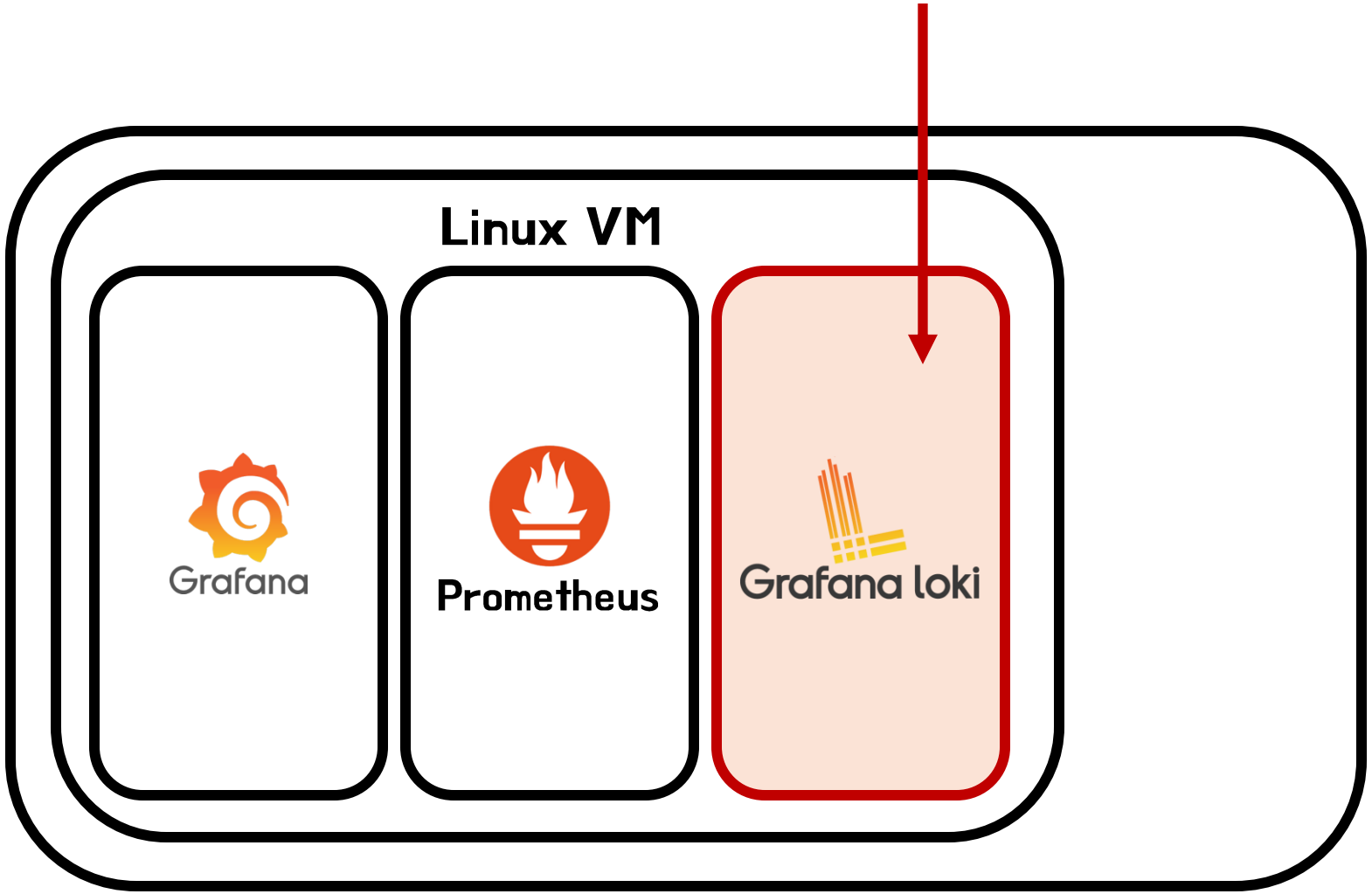
http://<모니터링서버 ec2 IP>:3100/loki/api/v1/push

로그

http://<모니터링서버 ec2 IP>:3100/loki/api/v1/push

모니터링 서버
ec2

Amazon EC2



모니터링 EC2

로그

Error Logs

```
> {"timestamp":"2025-10-04 17:01:23.704","level":"ERROR","message":"\n\n**********
> {"timestamp":"2025-10-04 17:01:08.143","level":"ERROR","message":"\n\n**********
> {"timestamp":"2025-10-04 16:59:59.452","level":"ERROR","message":"\n\n**********
> {"timestamp":"2025-10-04 16:48:07.007","level":"ERROR","message":"\n\n**********
> {"timestamp":"2025-10-04 16:45:41.608","level":"ERROR","message":"JDBC exception execut
> {"timestamp":"2025-10-04 16:45:41.483","level":"ERROR","message":"Table 'turip_dev.city
> {"timestamp":"2025-10-04 16:45:33.199","level":"ERROR","message":"JDBC exception execut
> {"timestamp":"2025-10-04 16:45:30.901","level":"ERROR","message":"Table 'turip_dev.city
```

Error 로그 히스토리

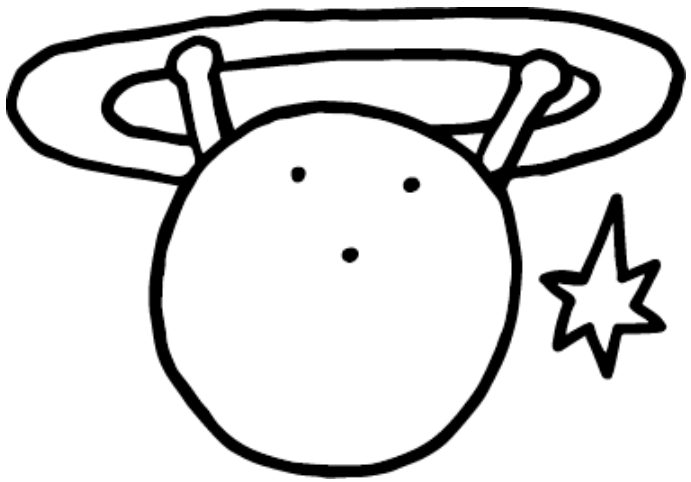
Warn Logs

```
> {"timestamp":"2025-10-05 00:52:49.351","level":"WARN","message":"No static resource ."
> {"timestamp":"2025-10-05 00:52:45.532","level":"WARN","message":"No static resource aa
> {"timestamp":"2025-10-05 00:52:41.612","level":"WARN","message":"No static resource aa
> {"timestamp":"2025-10-04 23:55:29.834","level":"WARN","message":"사용자를 찾을 수 없습니다."
> {"timestamp":"2025-10-04 17:13:59.165","level":"WARN","message":"spring.jpa.open-in-vi
> {"timestamp":"2025-10-04 17:01:23.425","level":"WARN","message":"Exception encountered
> {"timestamp":"2025-10-04 17:01:20.602","level":"WARN","message":"spring.jpa.open-in-vi
> {"timestamp":"2025-10-04 17:01:06.283","level":"WARN","message":"Exception encountered
> {"timestamp":"2025-10-04 17:01:00.262","level":"WARN","message":"spring.jpa.open-in-vi
> {"timestamp":"2025-10-04 16:59:55.970","level":"WARN","message":"Exception encountered
> {"timestamp":"2025-10-04 16:59:45.798","level":"WARN","message":"spring.jpa.open-in-vi
```

All Logs

```
> {"timestamp":"2025-10-05 01:12:24.299","level":"INFO","message":"API 응답","threadId":"
> {"timestamp":"2025-10-05 01:12:24.299","level":"INFO","message":"API 응답","threadId":"
> {"timestamp":"2025-10-05 01:12:24.295","level":"INFO","message":"API 요청","threadId":"
> {"timestamp":"2025-10-05 01:12:24.295","level":"INFO","message":"API 요청","threadId":"
> {"timestamp":"2025-10-05 01:12:23.255","level":"INFO","message":"API 응답","threadId":"
> {"timestamp":"2025-10-05 01:12:23.254","level":"INFO","message":"API 응답","threadId":"
> {"timestamp":"2025-10-05 01:12:23.254","level":"INFO","message":"API 응답","threadId":"
> {"timestamp":"2025-10-05 01:12:23.253","level":"INFO","message":"API 응답","threadId":"
> {"timestamp":"2025-10-05 01:12:23.251","level":"INFO","message":"API 응답","threadId":"
> {"timestamp":"2025-10-05 01:12:23.247","level":"INFO","message":"API 요청","threadId":"
> {"timestamp":"2025-10-05 01:12:23.246","level":"INFO","message":"API 요청","threadId":"
```

K6 부하테스트 모니터링



부하테스트

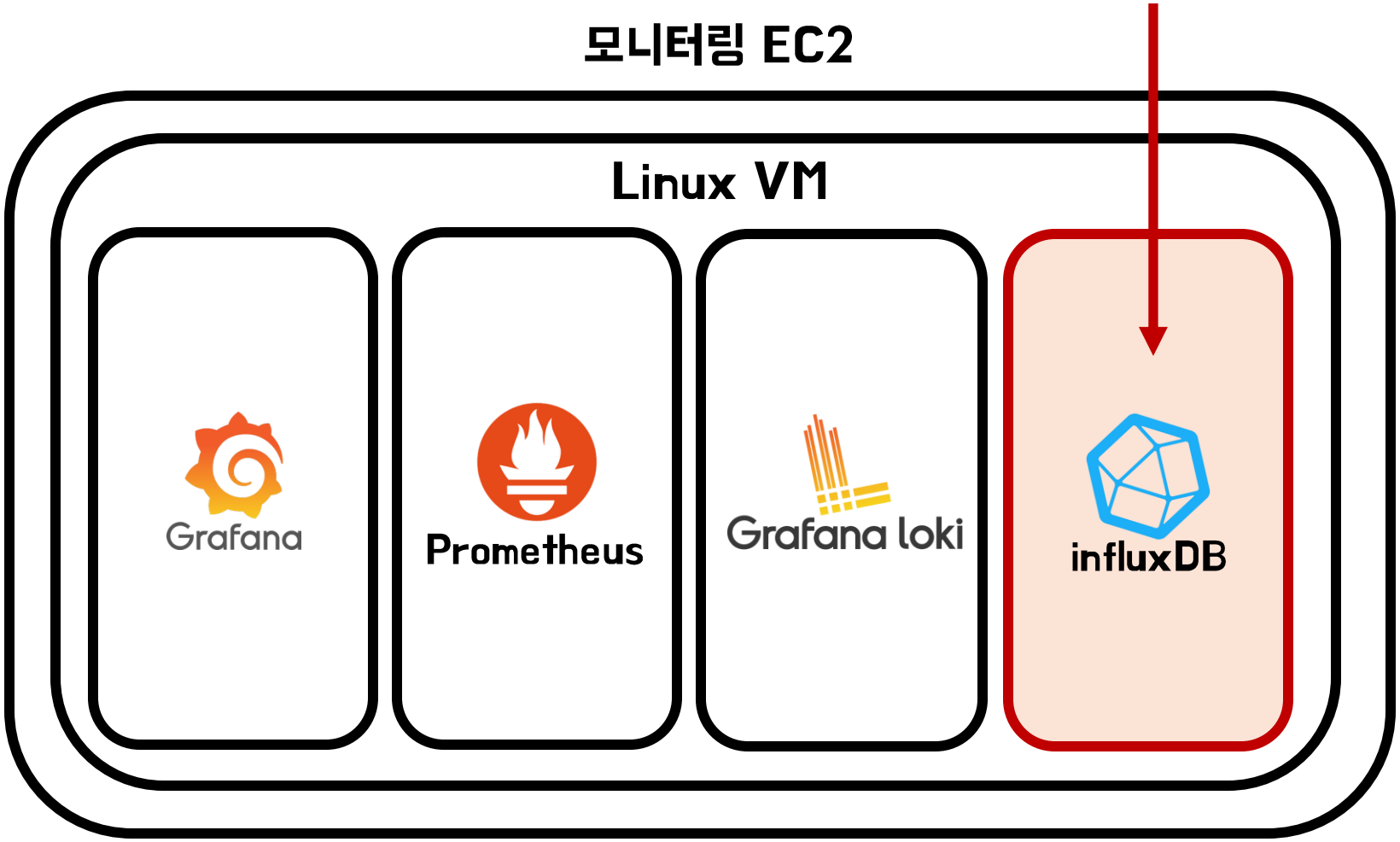
```
❯ k6 run test.js

         /\      Grafana   /‾‾/
    /\  /  \     |\  __   /  /
   /  \/    \    | |/ /  /   ‾‾\
  /          \   |   (  |  (‾)  |
 / __________ \  |_|\_\  \_____/

     execution: local
        script: test.js
        output: -

     scenarios: (100.00%) 1 scenario, 100 max VUs, 40s max duration (incl. graceful stop):
              * default: 100 iterations shared among 100 VUs (maxDuration: 10s, gracefulStop: 30s)


  TOTAL RESULTS

    HTTP
    http_req_duration..............: avg=1.13s min=147.94ms med=1.17s max=1.75s p(90)=1.68s p(95)=1.72s
      { expected_response:true }...: avg=1.13s min=147.94ms med=1.17s max=1.75s p(90)=1.68s p(95)=1.72s
    http_req_failed................: 0.00%  0 out of 100
    http_reqs......................: 100    35.973164/s

    EXECUTION
    iteration_duration.............: avg=2.15s min=1.16s    med=2.2s  max=2.77s p(90)=2.7s  p(95)=2.74s
    iterations.....................: 100    35.973164/s
    vus............................: 67     min=67       max=100
    vus_max........................: 100    min=100      max=100

    NETWORK
    data_received..................: 223 kB 80 kB/s
    data_sent......................: 10 kB  3.7 kB/s



running (02.8s), 000/100 VUs, 100 complete and 0 interrupted iterations
default ✓ [======================================] 100 VUs  02.8s/10s  100/100 shared iters
```

부하테스트

모니터링 서버
ec2

Amazon EC2

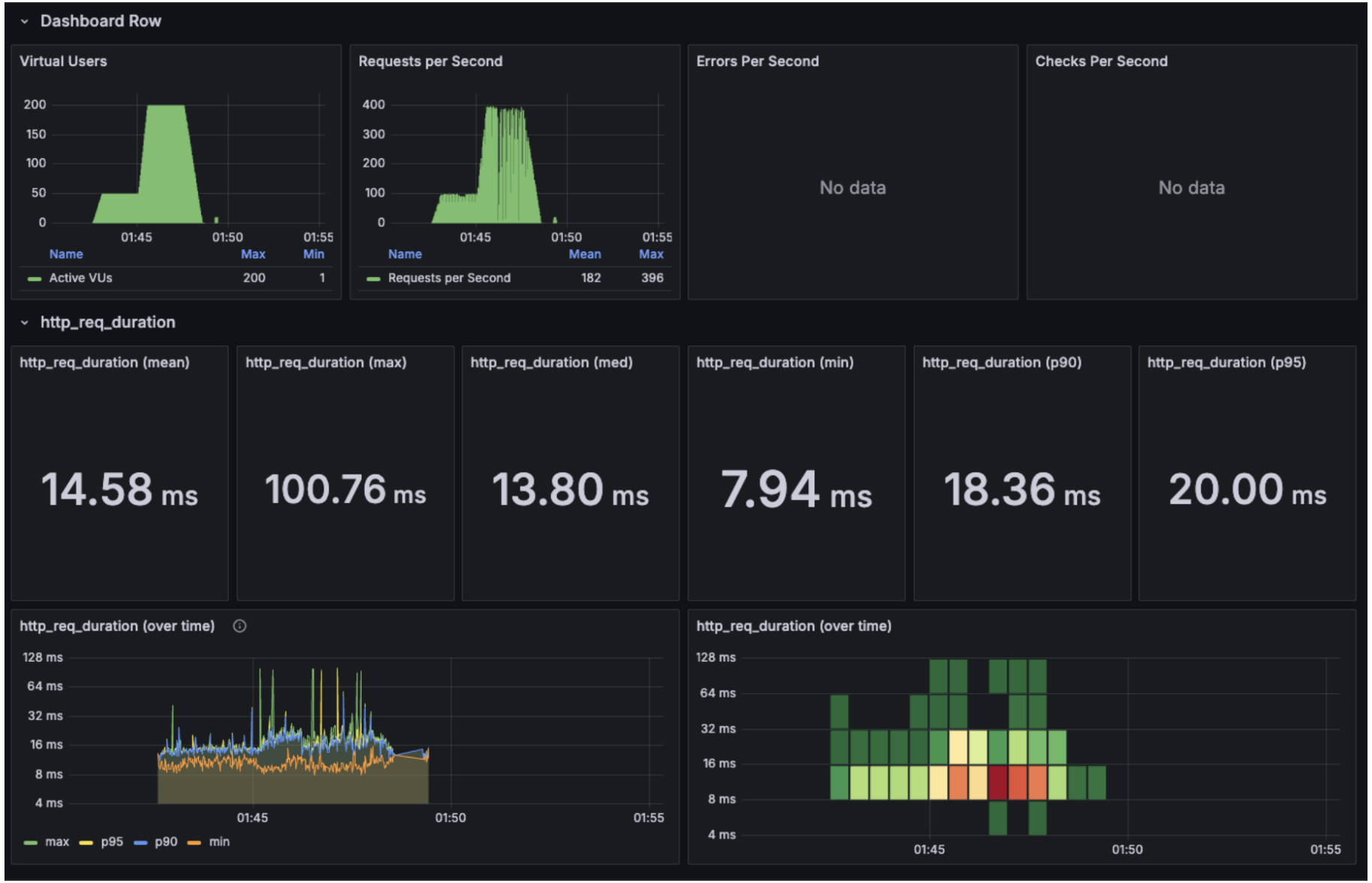
k6 run --out influxdb=http://<모니터링서버 ec2 IP>:8086/k6 test.js

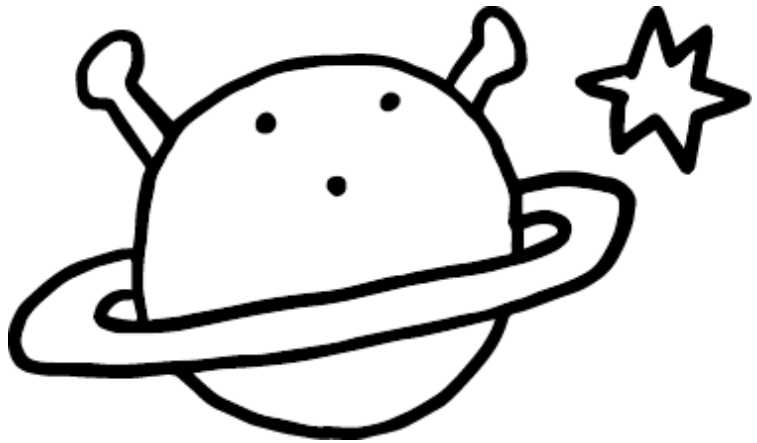


k6

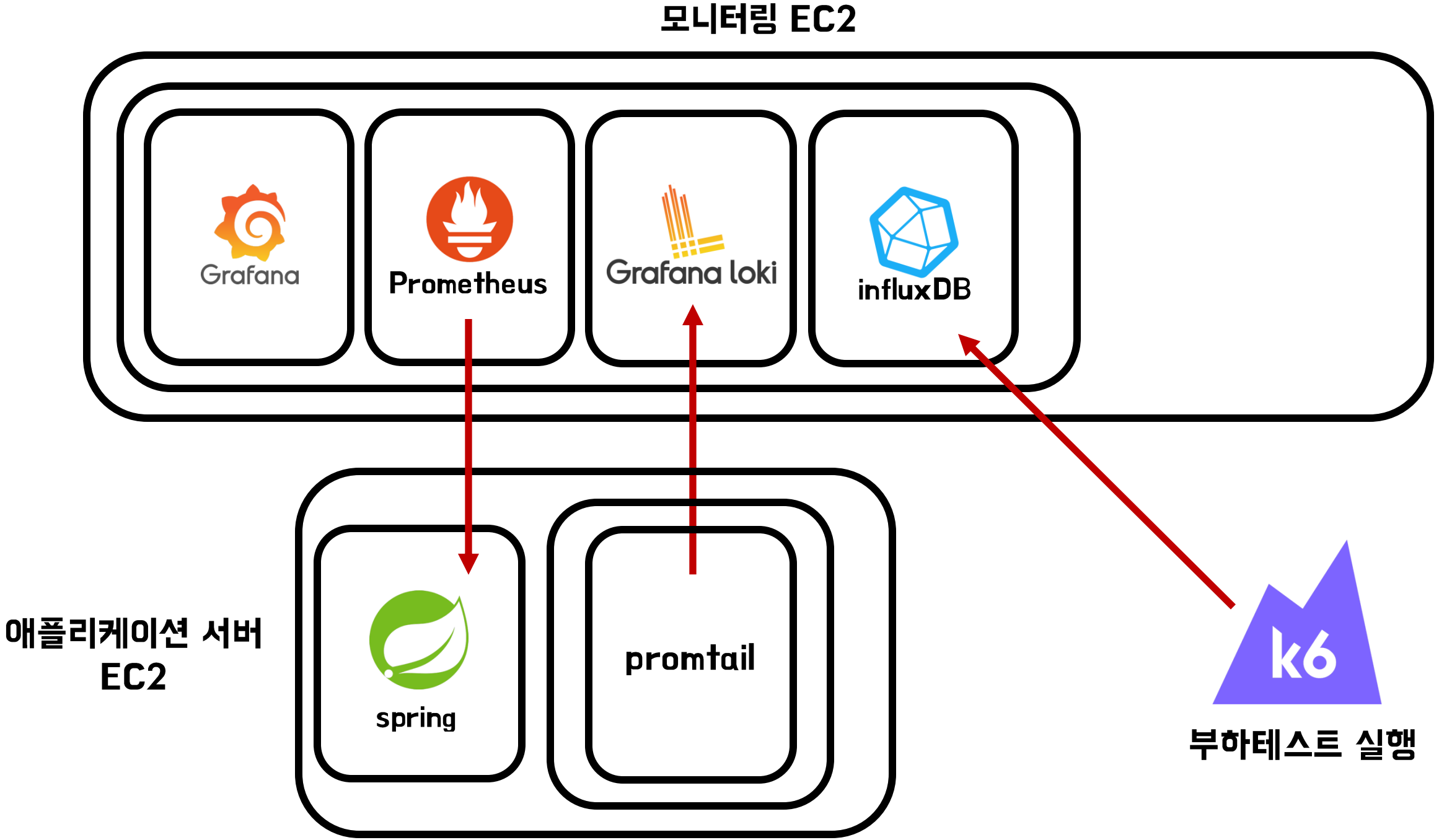
TPS, 응답시간, 실패율 ...

부하테스트

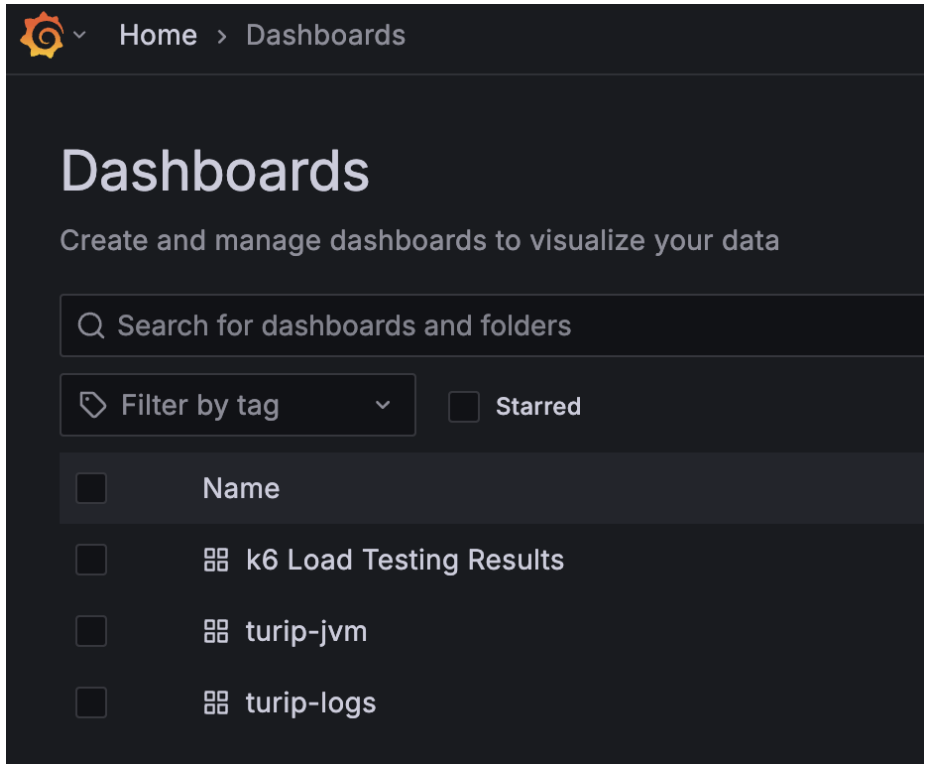

Dashboard Row
Virtual Users
200
150
100
50
0
01:45
01:50
01:55
Name
Max
Min
Active VUs
200
1
Requests per Second
400
300
200
100
0
01:45
01:50
01:55
Name
Mean
Max
Requests per Second
182
396
Errors Per Second
No data
Checks Per Second
No data
http_req_duration
http_req_duration (mean)
14.58 ms
http_req_duration (max)
100.76 ms
http_req_duration (med)
13.80 ms
http_req_duration (min)
7.94 ms
http_req_duration (p90)
18.36 ms
http_req_duration (p95)
20.00 ms
http_req_duration (over time)
128 ms
64 ms
32 ms
16 ms
8 ms
4 ms
01:45
01:50
01:55
max
p95
p90
min
http_req_duration (over time)
128 ms
64 ms
32 ms
16 ms
8 ms
4 ms
01:45
01:50
01:55



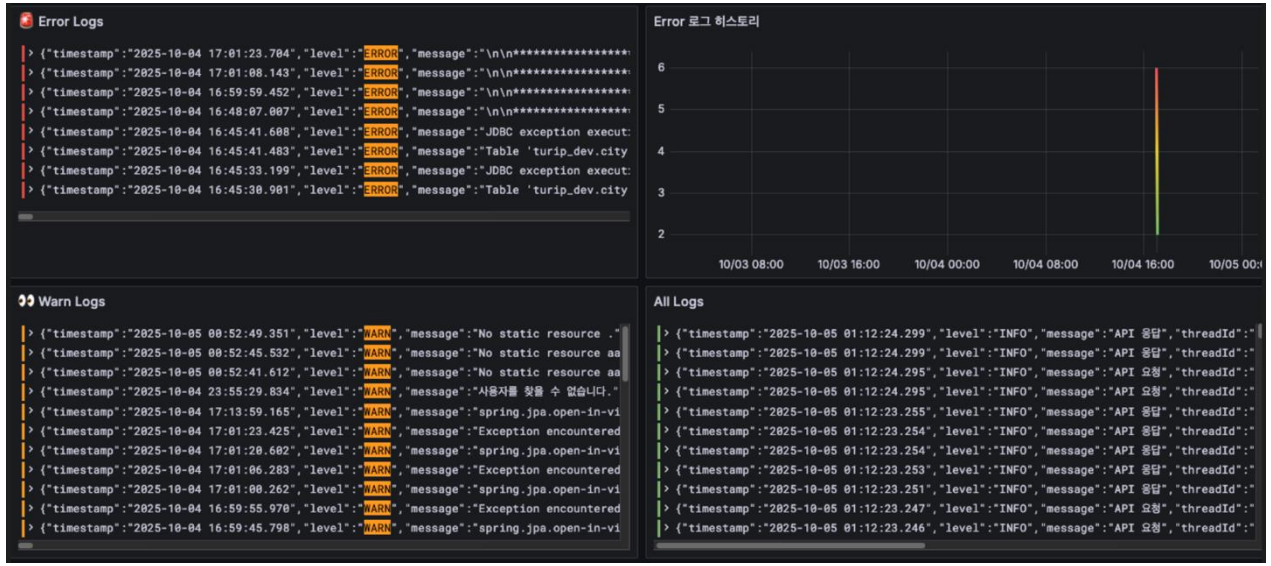
결론


모니터링 EC2
Grafana
Prometheus
Grafana loki
influxDB
애플리케이션 서버
EC2
spring
promtail
k6
부하테스트 실행

Grafana로 이사하니까 좋았던 것



통합적 모니터링



대시보드 커스텀

추가로 고려해야 할 부분

- 트레이스 모니터링
- 디스코드 웹훅 설정
- 서버별 모니터링 분리
- 모니터링 서버 볼륨

Q&A