DEVIEW 2021

1통의 재난 문자로 시작한 SRE 도전기

40배 이상 트래픽도 막아라!

한병수 NAVER / UGC&eXpert Dev
이정재 NAVER / UGC&eXpert Dev

CONTENTS

1. 들어가며

1. 들어가며

서비스를 운영하는 개발자에게
사이트 신뢰성 엔지니어링(Site Reliability Engineering)의 의미는?

오픈하는 순간 서비스는 레거시가 됨

오래된 서비스 뿐 아니라 신생 서비스도 높은 서비스 운영 품질이 필요함

SRE는 우수한 가동 시간뿐만 아니라 전반적으로 일관된 사용자 경험도 제공

고객이 느끼는 서비스 품질을 높게 유지할 수 있도록 개선하는 수단

결국 SRE는 소프트웨어 엔지니어링 신뢰성을 높이고
높은 신뢰 수준을 꾸준히 유지하고 개선하는 수단이라고 할 수 있음

1. 들어가며

DEVIEW 2021

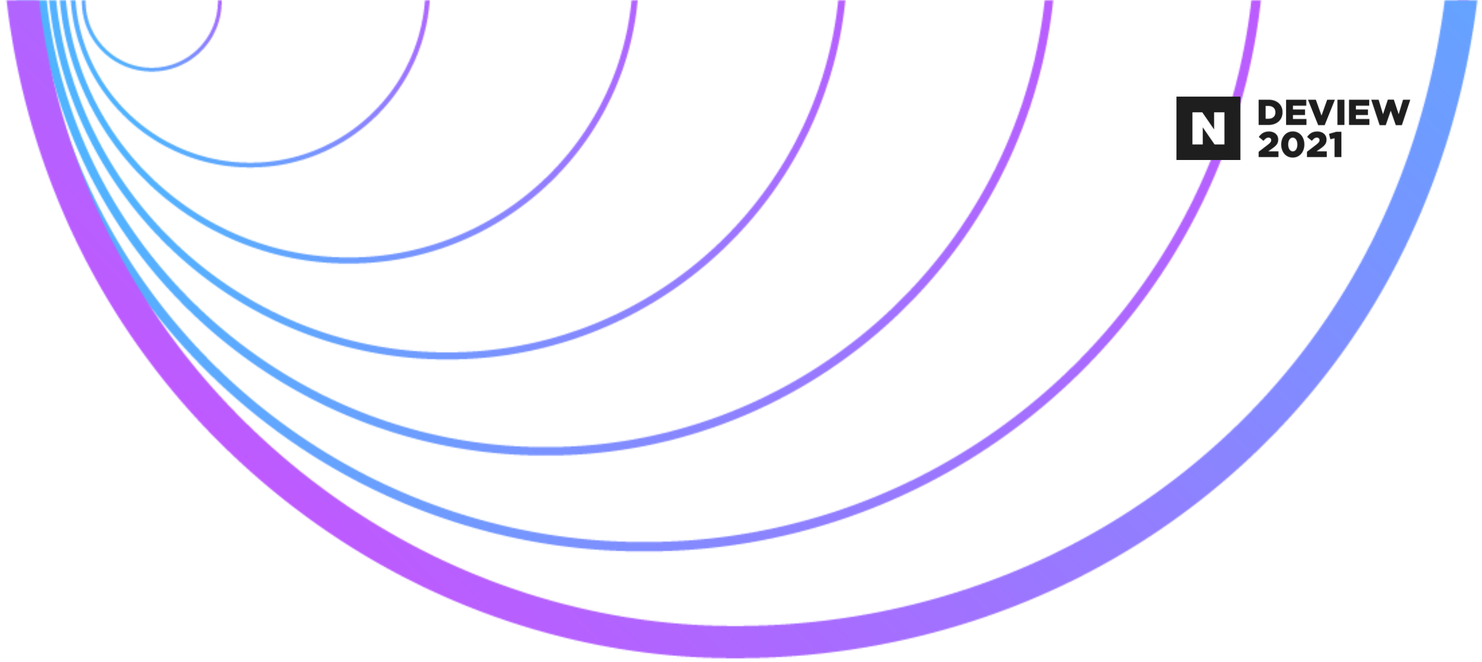
네이버 UGC 서비스가 SRE를 통해 서비스 품질을 높였던 경험 공유

네이버 UGC 서비스의 SRE 활동 소개

네이버 UGC 서비스도 구글의 SRE 실천법처럼 가용성 관점에서 SRE를 도입하고 실천을 함
가용량에 대한 정의, 가용성 목표, 장애 발생시 대응 등을 아우르는 전방위적인 활동을 함

블로그/지식iN의 SRE 실제 사례 및 개선 방법 소개

타 서비스도 응용할 수 있도록 블로그/지식iN 실제 사례를 공유



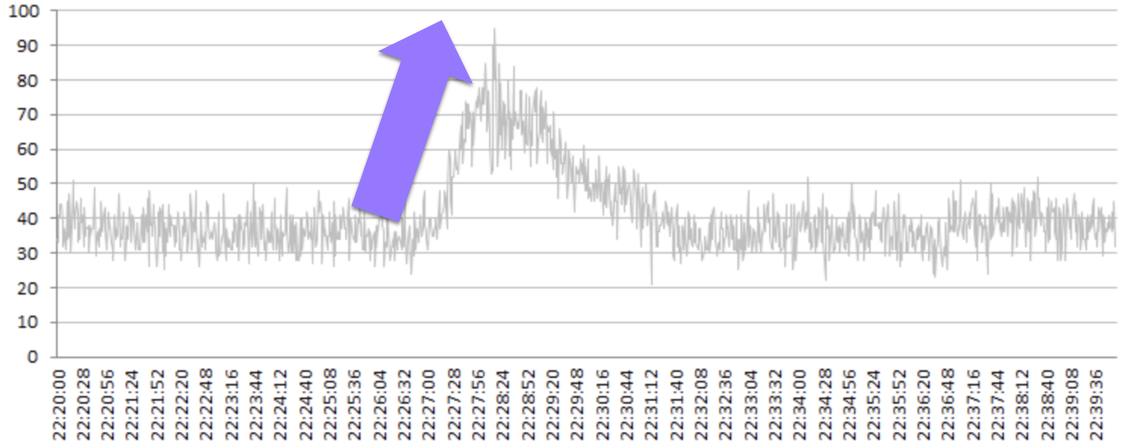
2. Accident

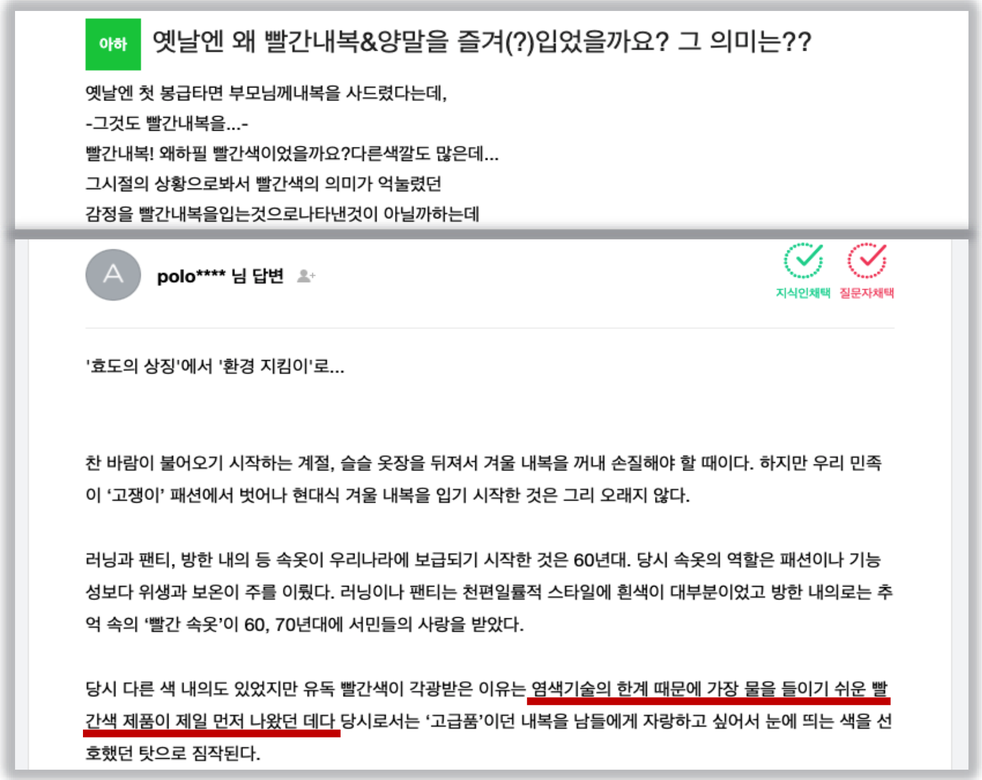
2.1 퀴즈쇼/재난 문자는 왜 장애를 유발했나?

DEVIEW 2021

전국민이 지식iN에서 퀴즈 정답을 찾으려 하다.

- 시청자들이 '1억 퀴즈쇼'의 질문인 '빨간 내복의 유래'를 검색
- 검색 결과 최상단에 지식iN 게시물 노출
- 순간적으로 평상시 대비 PC 2배, 모바일 8배 이상 트래픽 발생
- 특정 게시물로 조회가 집중되어 DB slow query도 급증

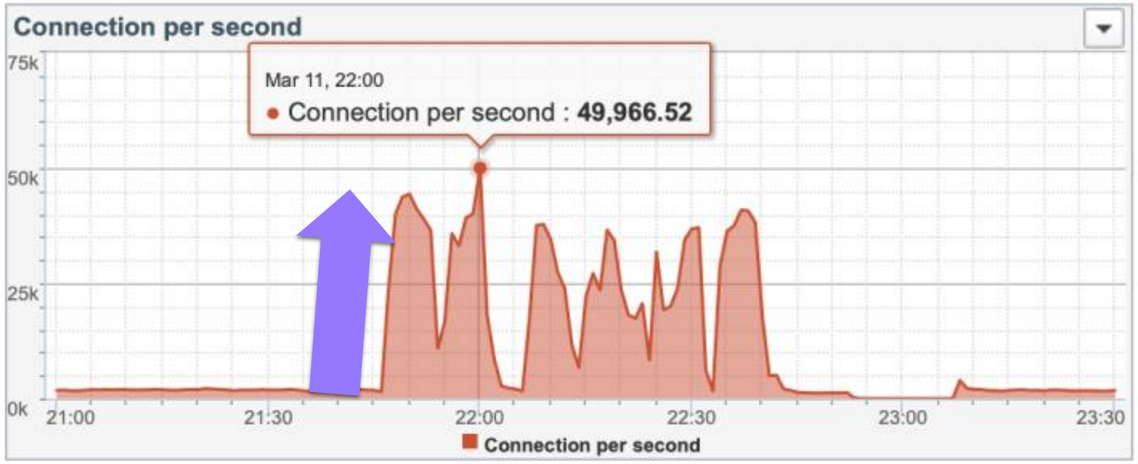




2.1 퀴즈쇼/재난 문자는 왜 장애를 유발했나?

블로그에 담긴 COVID-19 확진자 동선

- 지자체에서 재난 문자에 블로그 URL을 포함시켜 발송
- 코로나 확진자 동선에 대한 사회의 큰 관심으로 인해 사전 징후 없이 Spike성 트래픽 유입
- 소수 블로그로 집중된 트래픽 → Thread full → 전면 장애 발생
- 장애 임에도 동선을 확인하고자 하는 트래픽이 지속적으로 유입되어 긴 시간동안 장애 지속



평소 대비 28배 - 1시간 지속

2.2 SRE 관점에서 문제점/개선 포인트 도출

DEVIEW 2021

SRE 실천법 시각으로 앞으로 우리가 해야할 일들을 찾아보았습니다.

Metrics & monitoring

SLOs
Dashboards
Analytics

Capacity planning

Forecasting
Demand-driven
Performance

Change management

Release process
Consulting design
Automation

Incident response

Oncall
Analysis
Postmortems

<출처. 구글 넥스트 2018 발표 자료>

2.2 SRE 관점에서 문제점/개선 포인트 도출

Metrics & monitoring

SLOs

Dashboards

Analytics

서비스 관점의 Key Metric 부재, 기본 Metric의 빈약함

Infra
(nSight)

Application
(PINPOINT)

Error log
(NELO)

서비스 Key metric 정의/수집 및 통합할 수 있는 모니터링 대시보드의 필요성 부각

- 네이버에는 여러 모니터링 도구가 있지만 파편화되어 종합적인 서비스 상태 표현 불가

2.2 SRE 관점에서 문제점/개선 포인트 도출

Capacity planning

Forecasting

Demand-driven

Performance

Availability, Traffic Management 전략 부재

서비스 가용성 확대 전략 부재 → Spike 트래픽 발생시 대응 불가

- 서버 가용성 모니터링 및 증설 전략 필요
- 캐시 전략 업데이트 필요
- 서비스 병목 지점 확인 및 성능 튜닝 필요

트래픽 수용 전략 외 대응 방법 부재 → 몇 배의 트래픽이라도 무조건 수용?

- 장애를 방지하면서 효과적으로 가용성도 확보할 수 있는 전략 필요
- 트래픽 제어 기술 도입 필요

2.2 SRE 관점에서 문제점/개선 포인트 도출

DEVIEW 2021

Incident response

Oncall

Analysis

Postmortems

장애 대응 방법의 체계 부재

경험에 의존한 모니터링 도구 확인 및 수동 대응 → Postmortems 를 통한 지속적 개선

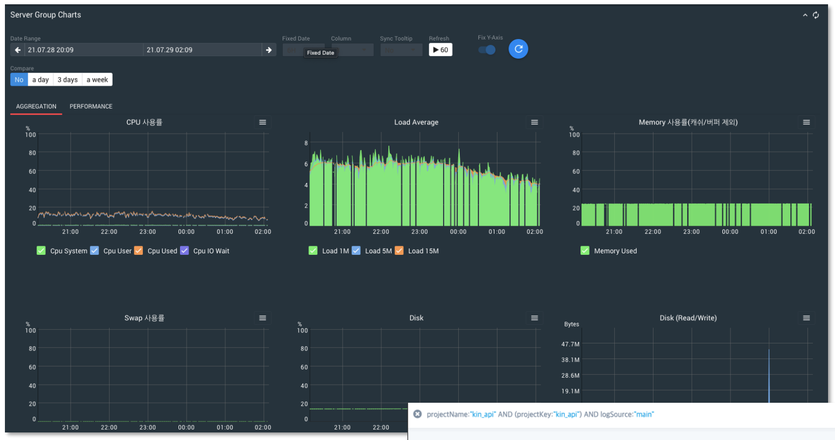
- 장애 대응 체계 리뉴얼 및 자동화 전환 필요
- 알람/배포 고도화, ChatOps 도입 필요

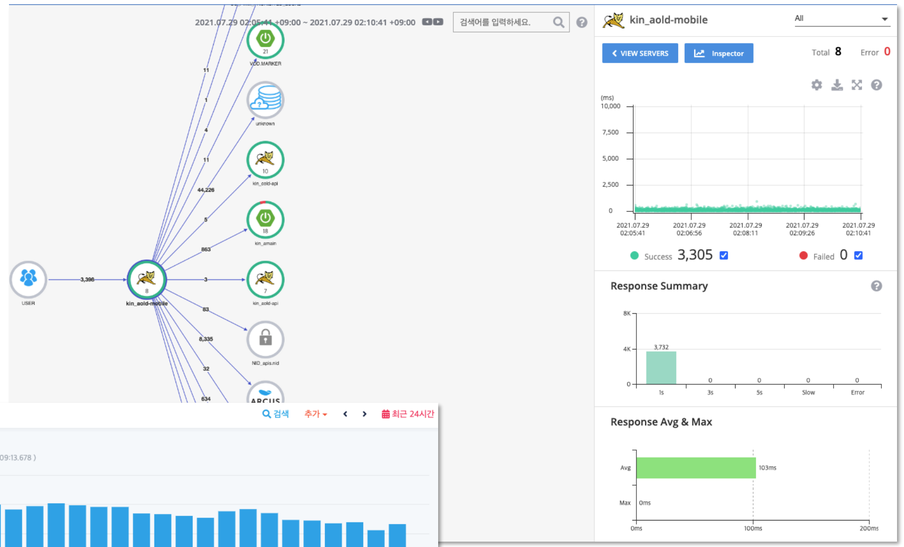
3. SRE - Monitoring

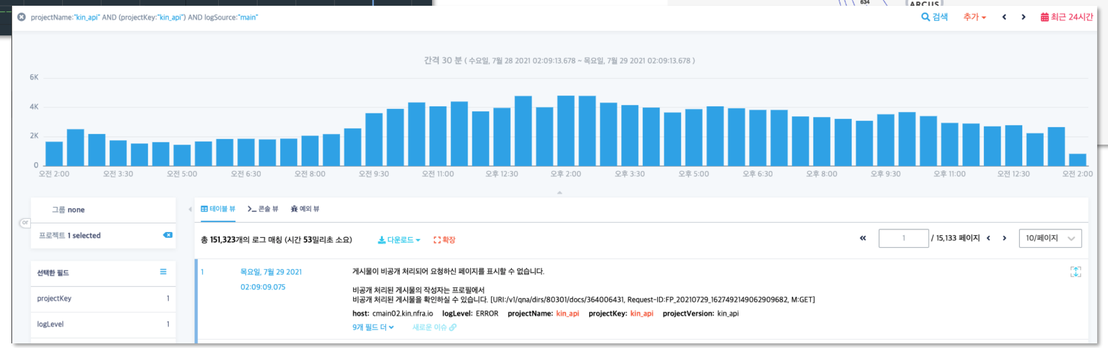
3.1 기존 서비스 모니터링의 한계

우리 서비스의 현재 수준이나 상태를 알 수 있는 Key Metric(핵심 지표) 부재

- 사내 모니터링 도구에 의존한 기본 Metric만으로는 서비스 현황을 종합적으로 파악하기 힘듦
- Key metric 부재로 인해 서비스 신뢰성을 높이기 위한 기준 설정 불가







3.1 기존 서비스 모니터링의 한계

서비스의 규모가 커지면서 파편화 된 Metric과 모니터링 도구로는 종합적인 인사이트를 얻기 어려워짐

수 많은 컴포넌트

NELO
PINPOINT
NPOT

다양한 모니터링 도구

APACHE kafka
A distributed streaming platform

신규 플랫폼 등장

2.6TB/day

대용량 로그

3.1 기존 서비스 모니터링의 한계

DEVIEW 2021

개선을 위해 서비스 모니터링의 방향과 목적 수립

어두운 곳을 비추고 기록하여 빠르고 점진적인 개선 유도

- 서비스 Key metric 정의 및 수집
- Grafana 기반의 통합 모니터링 포털 구축
- 통합 모니터링 환경을 통해 인사이트를 발굴하여 지속적인 개선 수행

측정할 수 없다면 개선할 수 없다.

3.2 SLI/SLO 정의 및 SLI Metric 수집하기

DEVIEW 2021

SRE의 빵과 버터 만들기 - SLI / SLO 란?

SLI
(Service Level Indicator)

서비스 수준 척도

- 서비스 수준을 판단할 수 있는 기준 몇 가지를 정량적으로 측정한 값

SLO
(Service Level Objectives)

서비스 수준 목표

- SLI에 의해 측정된 서비스 수준의 목표 값 혹은 일정 범위의 값

3.2 SLI/SLO 정의 및 SLI Metric 수집하기

DEVIEW 2021

블로그 / 지식iN 서비스의 SLI / SLO

서비스 가용성을 대표하는 Key Metric 정의 후 아래 Metric을 SLI로 선정하여 수집

- Latency
- Error rate
- Throughput(Access log count/Nginx RPS …)

SLO: SLI에 대한 목표 값/알림 조건 설정

SLO / *서버당*	max CPU (알림조건)	max MEM (알림조건)	max Network-In (kbps) (알림조건)	max Nginx Request/s (알림조건)	max Nginx Connection (알림조건)	max Traffic (알림조건)	응답코드 (알림조건)	응답시간 (ms) 알림조건	응답시간(ms) %
PC웹 (PM)	70% (> 10%)	90% (> 30%)	평촌 700,000 (> 90,000) 춘천 500,000 (> 90,000)	2,400 (> 310)	평촌 1700 (> 200) 춘천 1400 (> 180)	90,000,000 ~~(> 30,000,000)~~	502/504/503/~~499~~ (> 1)	분단위 3000 ms 이상 (> 100) 5000 ms 이상 (> 50)	0~1초 99% 1초이상 1% avg 200ms
모바일웹 (PM)	70% (> 15%)	90% (> 30%)	1,200,000 (> 150,000)	2,000 (> 210)	3,300 (> 620)	60,000,000 ~~(> 12,000,000)~~	502/504/503/~~499~~ (> 1)	분단위 3000 ms 이상 (> 100) 5000 ms 이상 (> 50)	0~1초 99% 1초이상 1% avg 200ms

Key Metric

System	• Cpu Usage(System / User) • Load Average • Memory Usage • L7 Check • Spare Tomcat port 8001~3 / 8011~3 • NetWork RX / TX
Nginx	• Request Per Sec • Connection Count • 500 Error Count Per Sec
Tomcat	• Current Thread Count • Current Thread Busy • JVM Heap Size
Access Log Based Metric(Service Metric)	• Access Log Count • PostView Count • PostView Latency(avg / p99 / p99.9) • Top URL List • PostList Count • PostList Latency(avg / p99 / p99.9)

3.2 SLI/SLO 정의 및 SLI Metric 수집하기

DEVIEW 2021

Access Log 기반 SLI Metric 수집

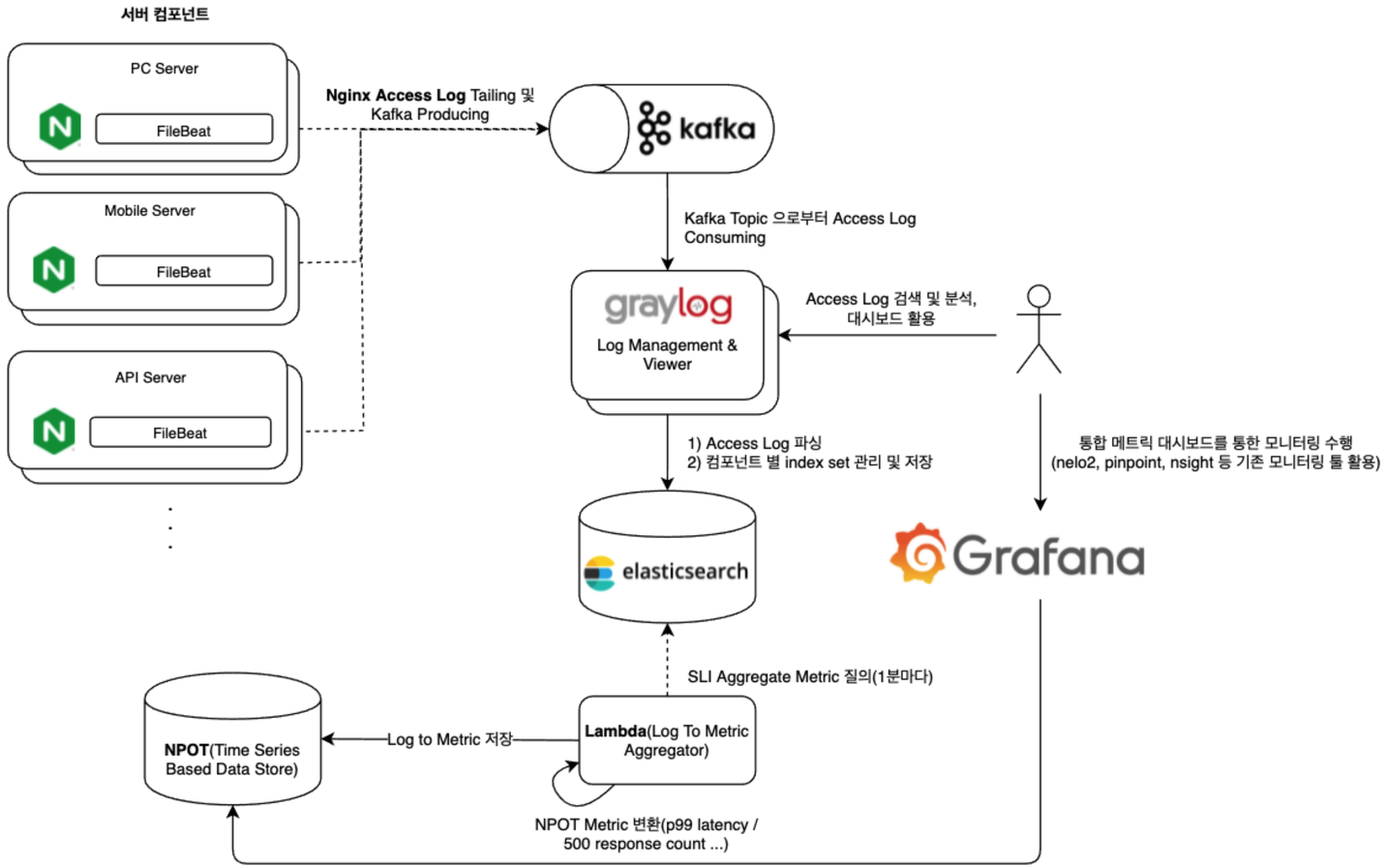
주요 수집 도구

- Graylog, Elasticsearch
- Lambda(Serverless computing)
- NPOT(Time series based data store)

기반 작업

- HTTP status code 정비

  (제대로 된 Error rate metric 수집 위해)



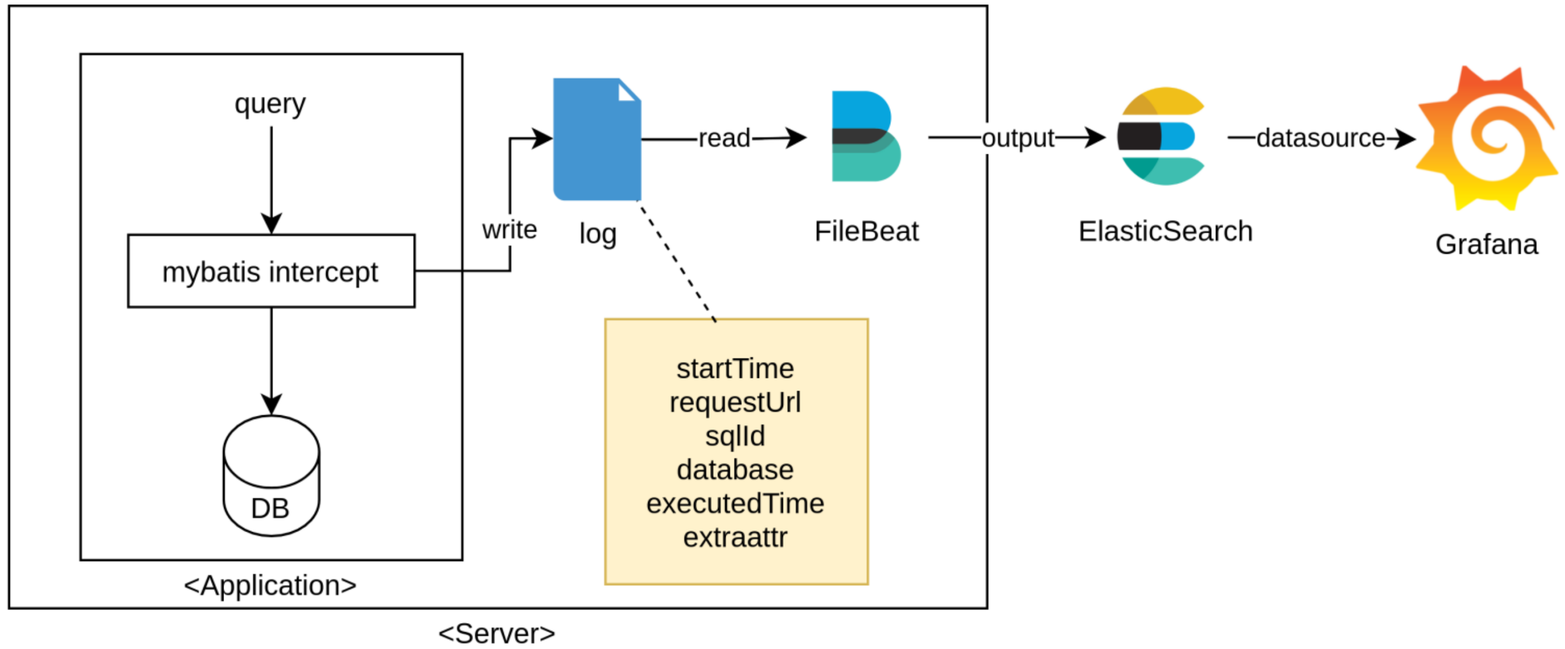
3.2 SLI/SLO 정의 및 SLI Metric 수집하기

DEVIEW 2021

DB Metric 수집

주요 수집 도구

- Mybatis interceptor
- Filebeat
- Elasticsearch



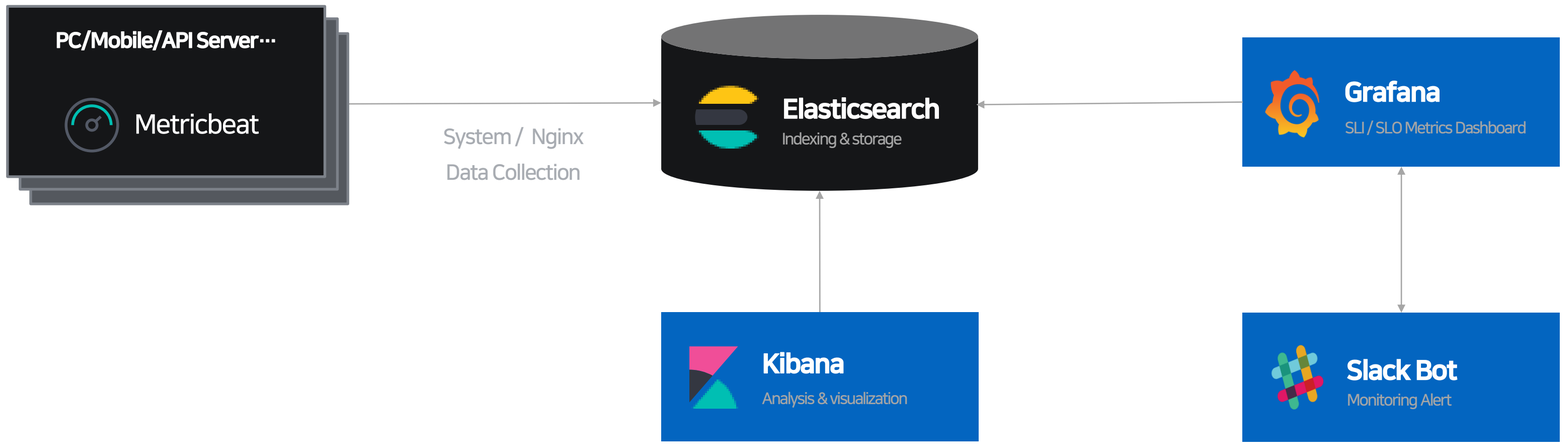
3.2 SLI/SLO 정의 및 SLI Metric 수집하기

DEVIEW 2021

Infra Metric 수집

주요 수집 도구

- Metricbeat, Elasticsearch



주요 Metric

- System: CPU, Memory, Network, Filesystem, Disk I/O…
- Nginx: Connections, Requests, Accepts, Handled, Dropped

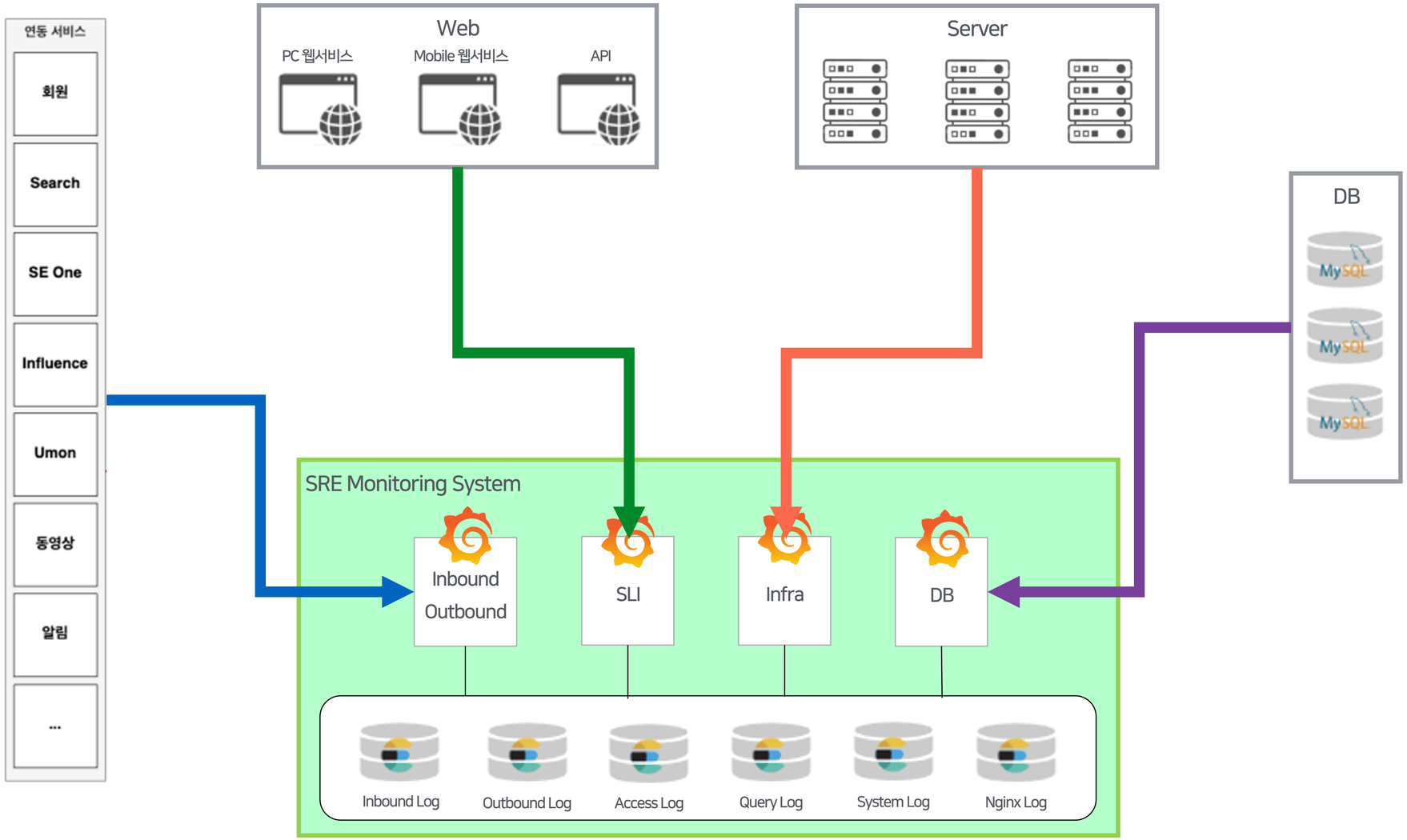
3.3 모니터링 대시보드

DEVIEW 2021

모든 것을 측정하고 실시간으로 인사이트 얻기

모든 서비스, 모든 구성 요소는
할 일이 있는 곳, 잘하고 있는 곳, 개선해야 될 곳을
실시간으로 볼 수 있어야 함.

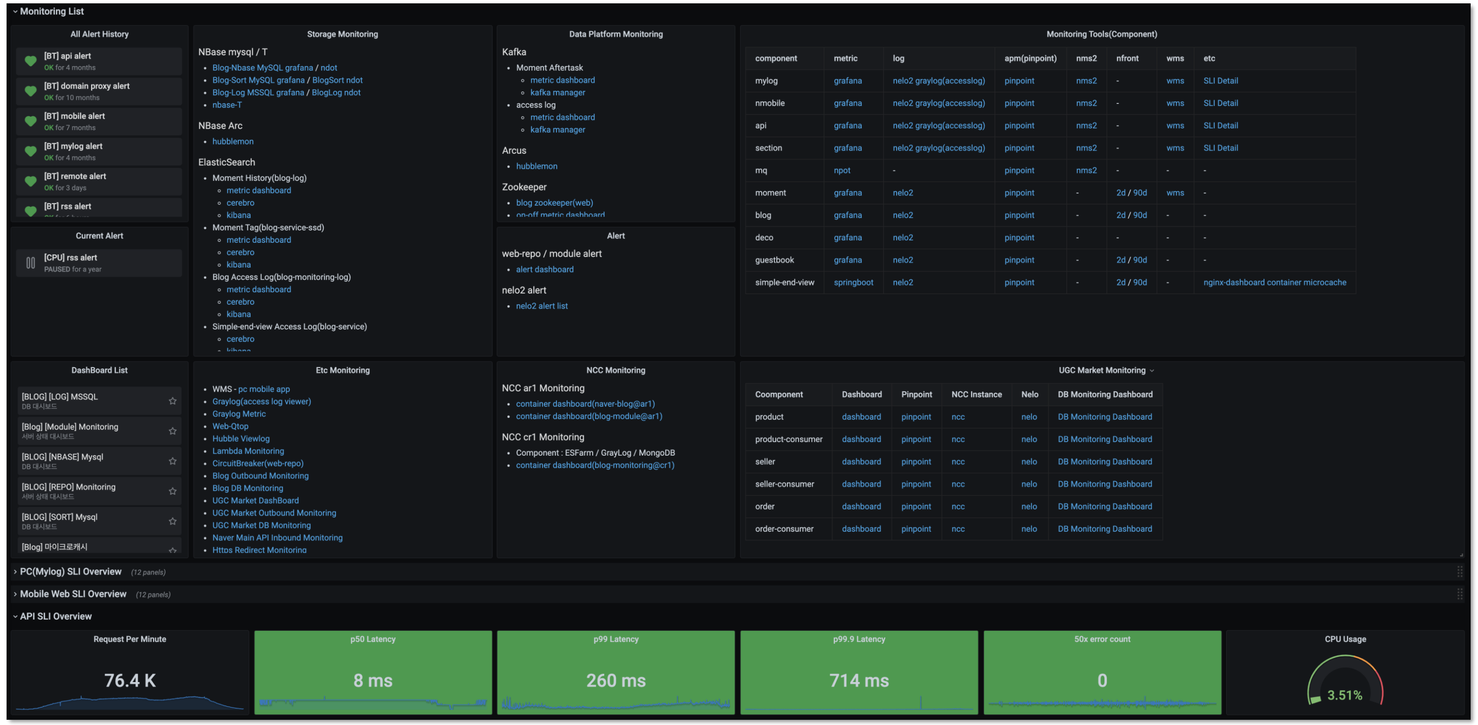
3.3 모니터링 대시보드


연동 서비스
회원
Search
SE One
Influence
Umon
동영상
알림
...
Web
PC 웹서비스
Mobile 웹서비스
API
Server
DB
MySQL
MySQL
MySQL
SRE Monitoring System
Inbound
Outbound
SLI
Infra
DB
Inbound Log
Outbound Log
Access Log
Query Log
System Log
Nginx Log

3.3 모니터링 대시보드

Grafana 기반 통합 모니터링 포털 대시보드

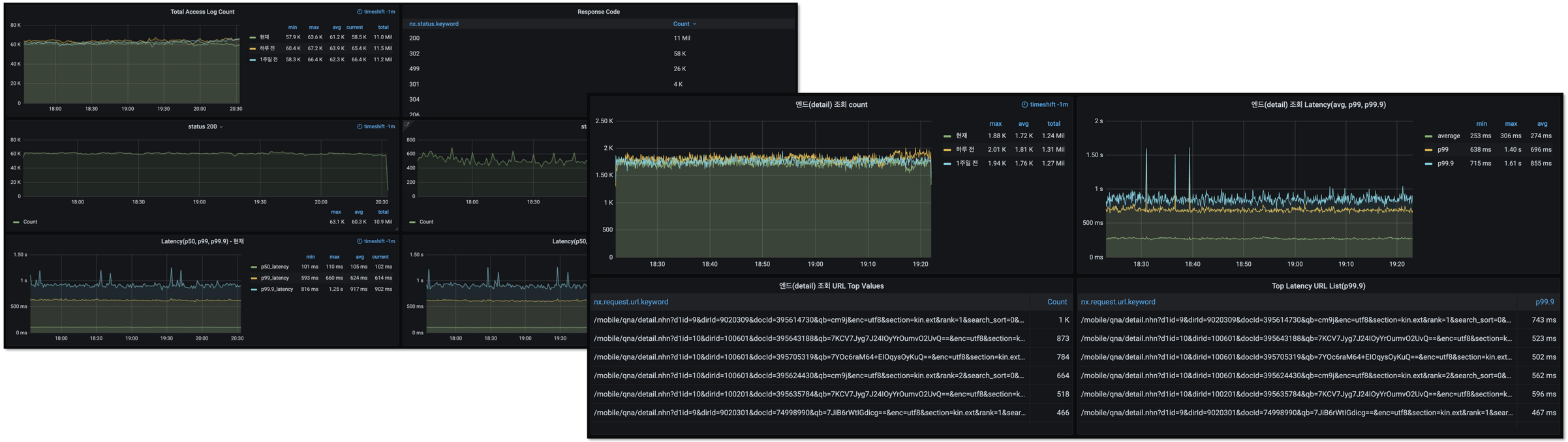
- 모든 모니터링 대시보드와 운영 도구의 관문 역할
- 서비스 전체 상태를 요약해서 파악하기 위한 통합 대시보드
- 컴포넌트 별 주요 Metric 소개 (Overview)



3.3 모니터링 대시보드

SLI 모니터링 대시보드

- 서비스 수준을 나타내는 Metric들로 현재 서비스의 신뢰성 수준 모니터링
- 주요 Metric: Access count, Latency, Response code
- 활용 사례: 주요 페이지 조회수/Latency 변화 추이 확인, 응답속도 개선



3.3 모니터링 대시보드

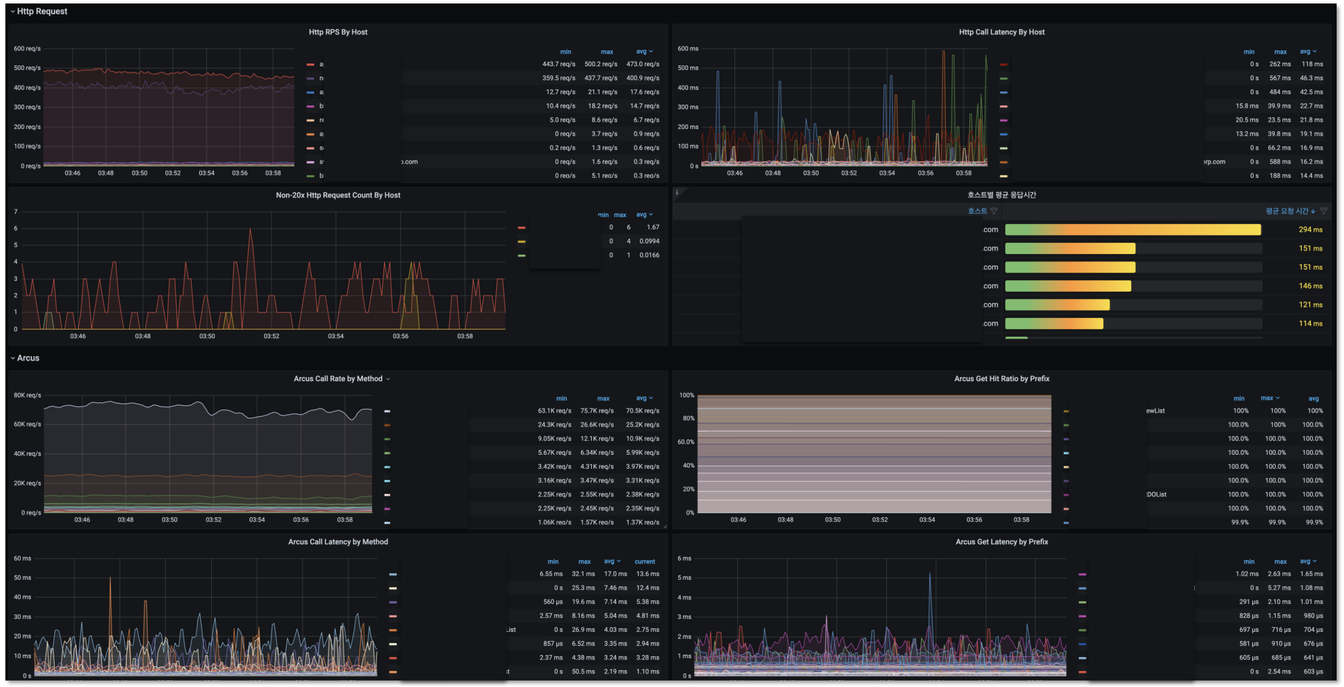
Inbound 모니터링 대시보드

- 서비스 외부로 제공하는 내부 주요 API 상태 모니터링
- 주요 Metric: API 호출 Count, Response code, Latency
- 활용 사례: API error 파악, API 응답속도 개선

3.3 모니터링 대시보드

Outbound 모니터링 대시보드

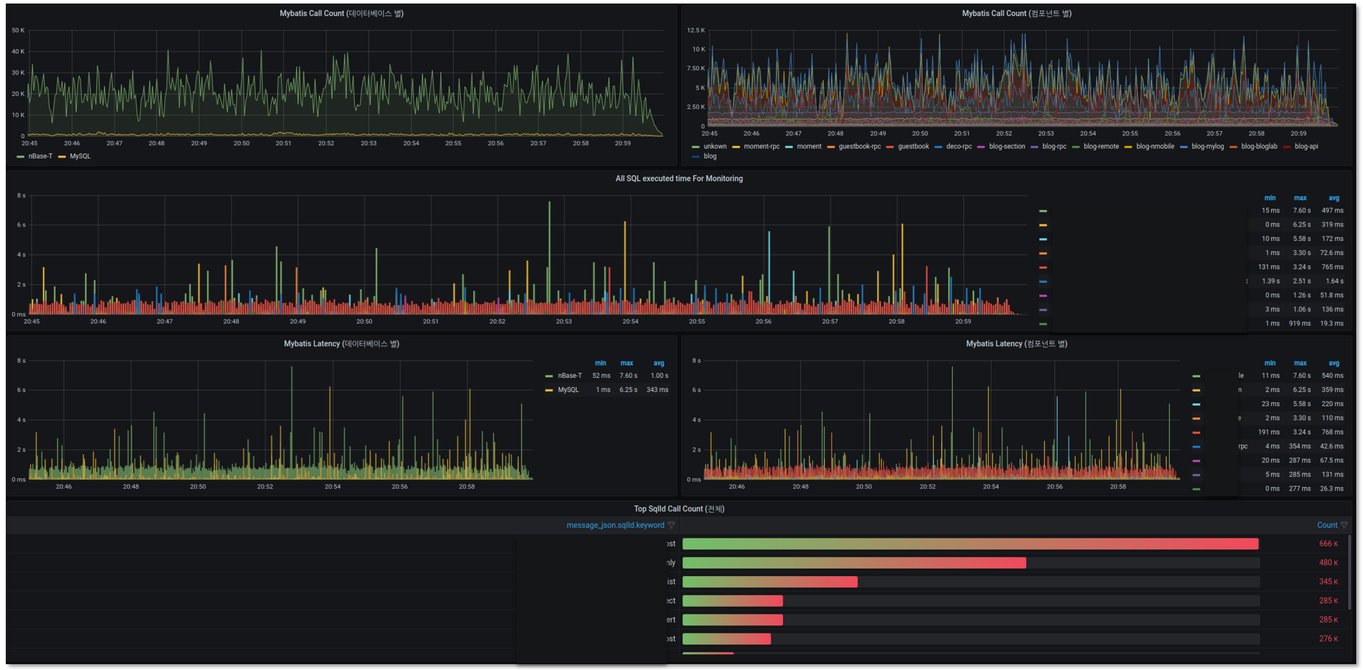
- Application에서 외부로 요청하는 구간 모니터링
- 주요 Metric: RPS, Latency, Response code, Cache hit ratio
- 활용 사례: 응답시간 개선 요청, 후처리 변경, Cache 최적화, 외부 장애 유입 지점 파악



3.3 모니터링 대시보드

DB 모니터링 대시보드

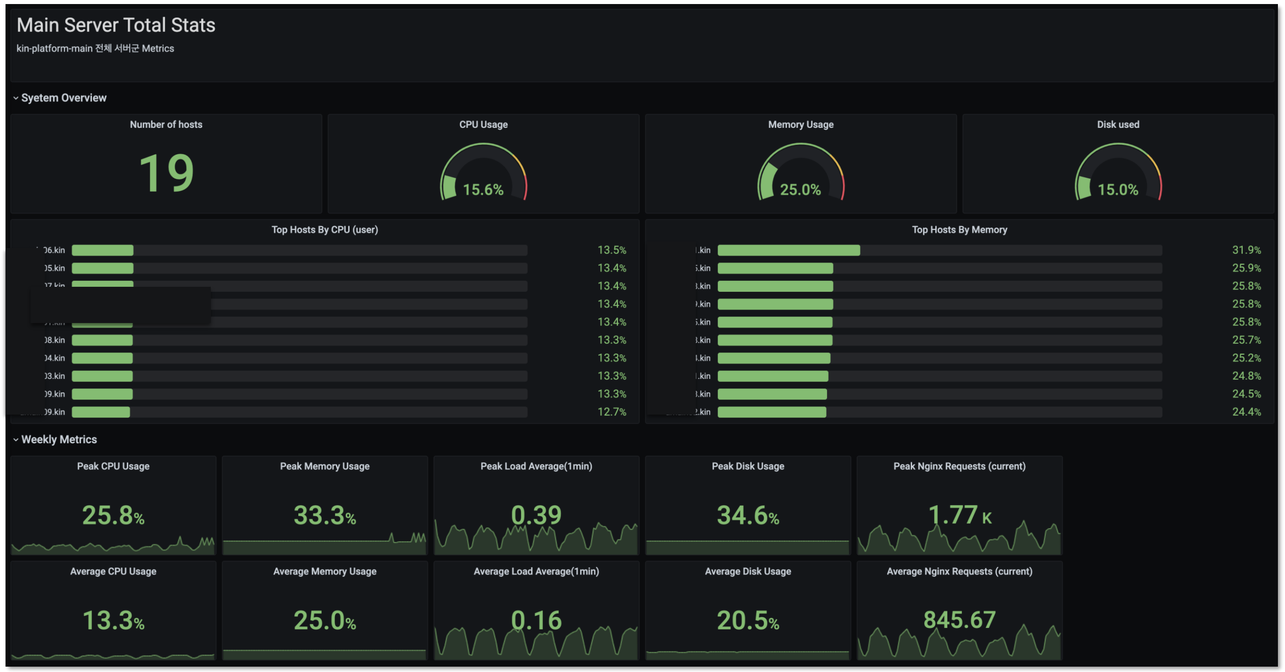
- Application에서 DB로 요청하는 Mybatis 로그 기반 Query 모니터링
- 주요 Metric: Sqlid별 Call count, RPS, Latency
- 활용 사례: Slow query 개선, DB 장애 구간/영향도 파악

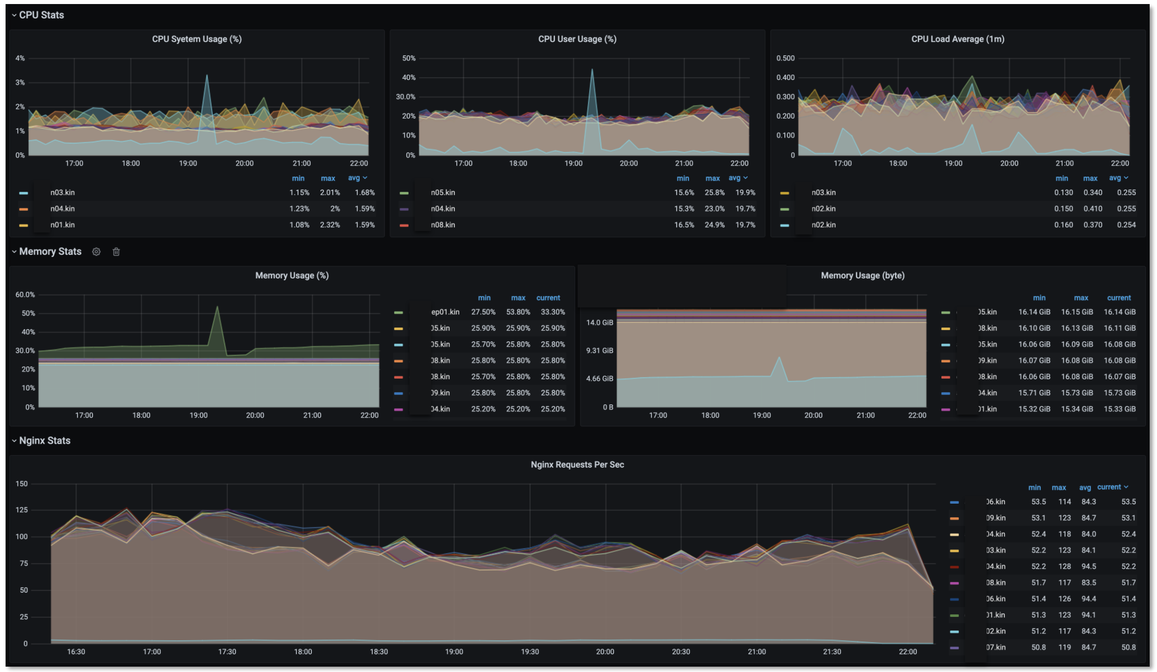


3.3 모니터링 대시보드

Infra 모니터링 대시보드

- Infra metric 수집을 통한 서버 상태 모니터링
- 주요 Metric: CPU, Memory, Disk, Network, Canary status, Tomcat busy thread, Nginx RPS
- 활용 사례: On-premise 환경 서버 증설 시점 파악, Infra 이슈 대응, 배포 모니터링





4. SRE - Availability, Traffic Management

4.1 가용성 확대 그리고 트래픽 제어

DEVIEW 2021

트래픽 제어

가용성 확대

Part 1.
대규모 트래픽 → 가용량 수준의 트래픽

Rate limiting	네트워크 트래픽 제한 전략 수용량 초과 트래픽 무시
Queue-based load leveling	네트워크 트래픽 수용 전략 Wait queue 활용

Part 2.
Cascading Failure 방지

Load shedding	문제 유발 트래픽 대상 제한 전략 문제 유발 트래픽만 무시
Bulkhead	Application 요소 서로 격리 일부의 실패가 전체로 전이 X
Circuit breaker	외부 시스템 감시 및 격리 외부 장애로부터 서비스 보호

Part 3.
가용량 확대 → 대규모 트래픽 수용

Scale out	서버 자원 증설 전략 On-premise ↔ Kubernetes
Performance improvement	Simple pages system Feature flag

Cache strategy	Cache-aside, Read-through… 빠른 데이터 조회 및 응답

4.2 서비스 가용성 확대를 위한 노력

서버 가용성 모니터링 및 증설 전략

On-premise : Key metric 기준으로 서버 가용성 판단, Manual scale-out

- 서비스 요청량 및 CPU 사용량 등을 고려하여 Scale-out 기준 수립
- On-premise 환경의 특성상 수용 목표를 정하고 미리 증설

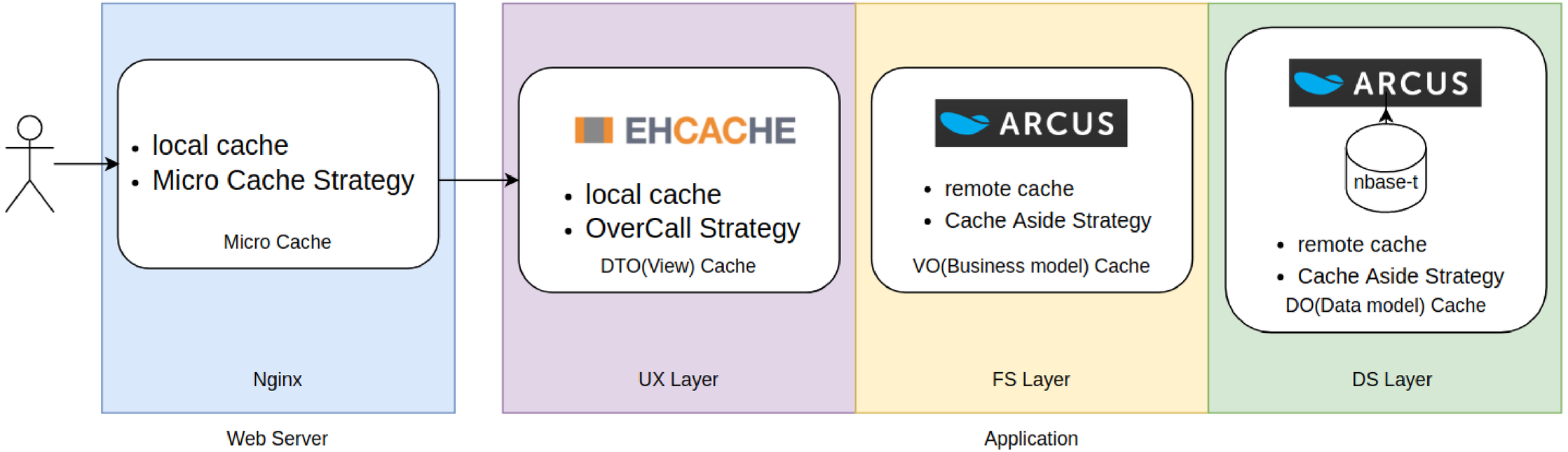
Kubernetes : Key metric 기준을 HPA 설정에 적용, Auto scale-out

- 서비스 요청량 및 CPU 사용량 등을 고려하여 Scale-out 기준 수립
- Kubernetes 환경상 HPA를 이용한 빠른 Auto scale-out 전략 사용

4.2 서비스 가용성 확대를 위한 노력

Layer Architecture 기반의 프로젝트에서 캐시 전략 적용

- 구간별 요청 특성과 데이터 활용도를 고려해 상황에 맞는 전략 적용

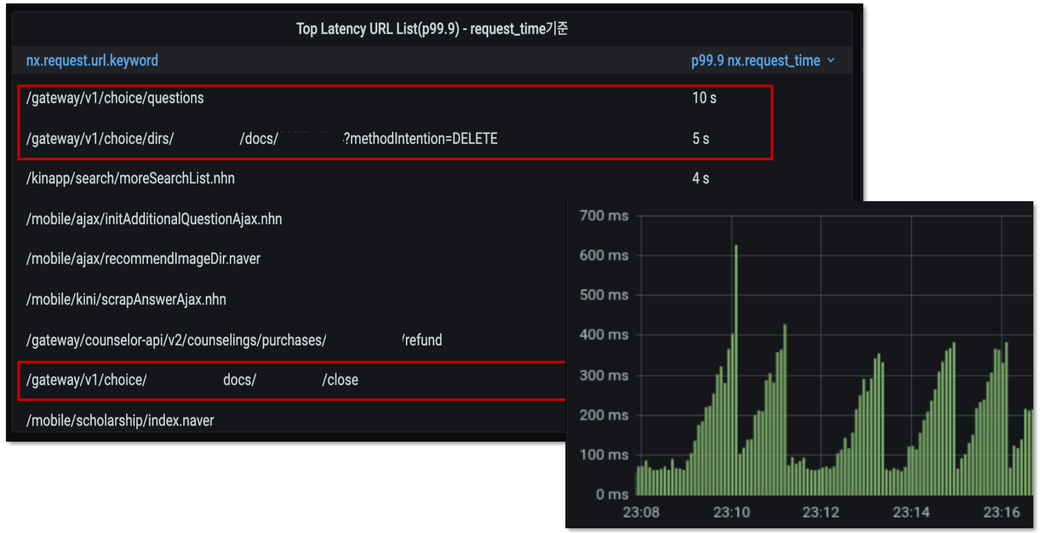


4.2 서비스 가용성 확대를 위한 노력

서비스 병목 지점 성능 튜닝 (성능테스트, SRE 모니터링)

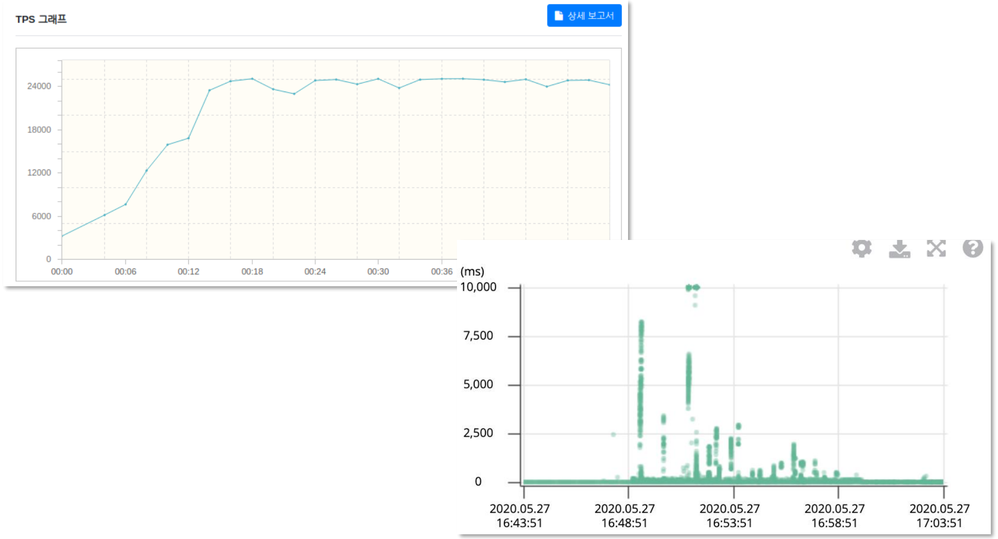
SRE 모니터링

실시간 모니터링 대시보드를 활용한
API 성능 튜닝 진행



성능 테스트

nGrinder를 활용한 성능테스트를 통해
병목 지점을 찾아 제거



4.3 주요 트래픽 제어 기술

Rate Limiting (with Nginx)

트래픽을 제어하는 기술

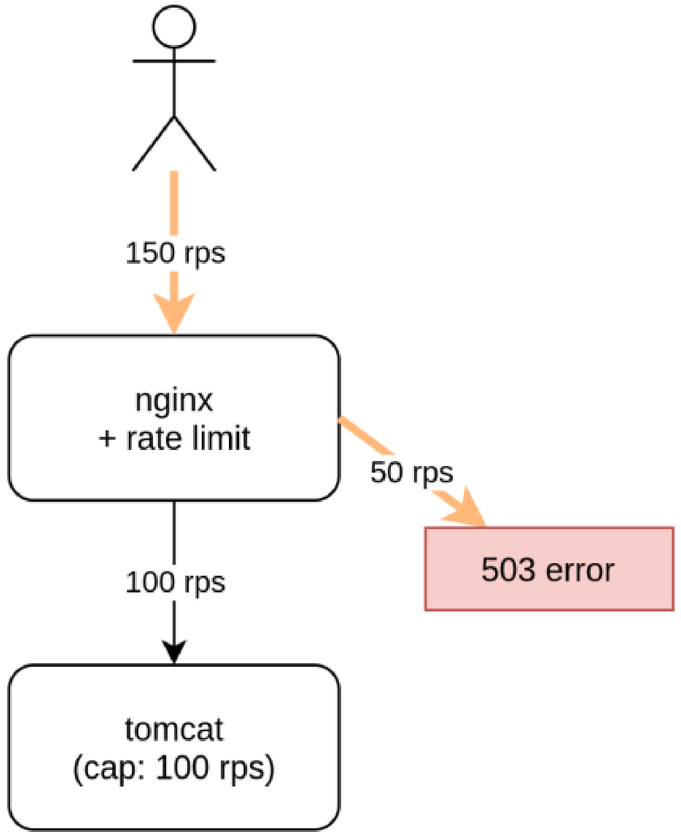
- 미리 정의한 서버 가용량에 해당하는 트래픽만 허용하는 기술

Nginx 활용

- Nginx 의 Rate Limit 기능 활용

적용 사례

- **지식인**: 홈/프로필 홈/엔드, 네이버메인 API 연동 영역 등에 적용
- **블로그**: PC/Mobile 페이지 중 다양한 곳에 적용



4.3 주요 트래픽 제어 기술

Rate Limiting (with Nginx)

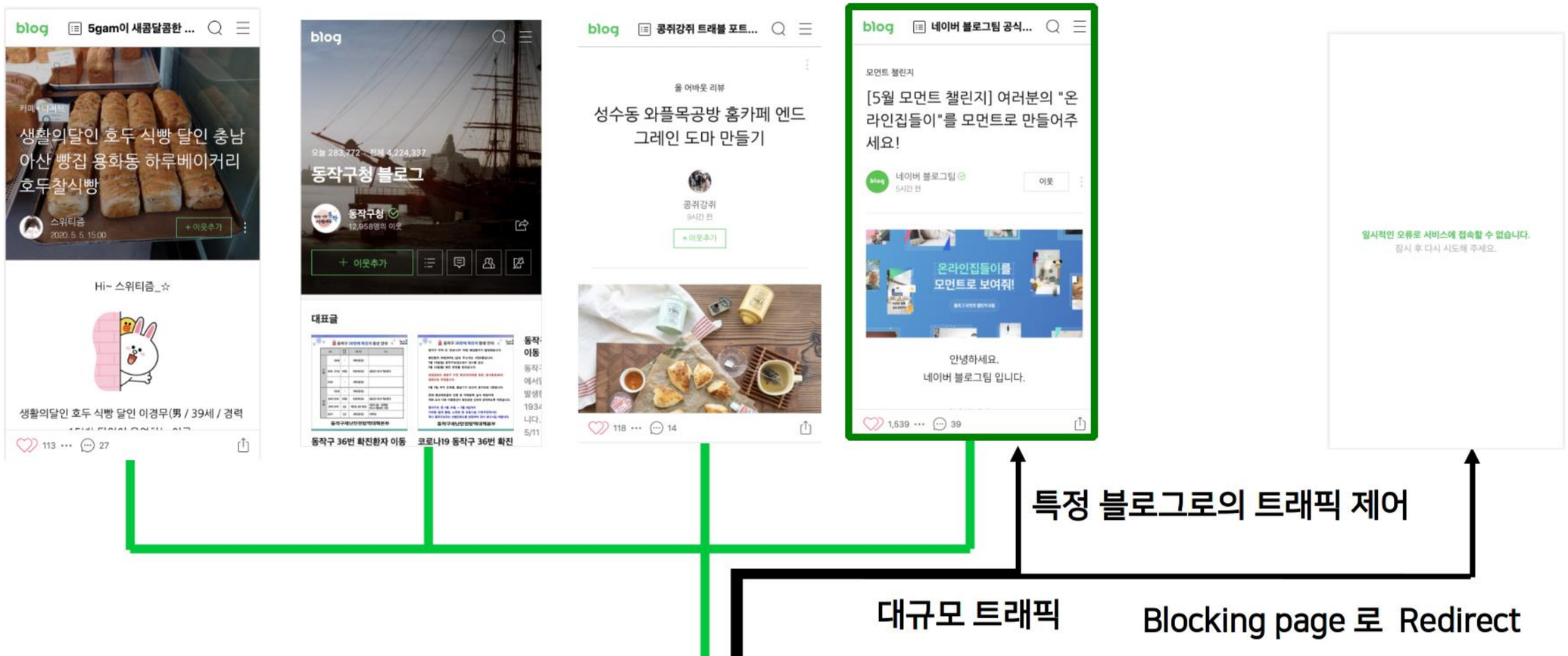
- Nginx 버전 관계 없이 기본 설정으로 사용 가능. (추가 모듈 설치 X)

```
http {
    ...
    limit_req_zone $server_name zone=mylimit:10m rate=1000r/s;
    ...
    location / {
        ...
        limit_req zone=mylimit burst=1000 nodelay;
        ...
    }
}
```

4.3 주요 트래픽 제어 기술

Load Shedding

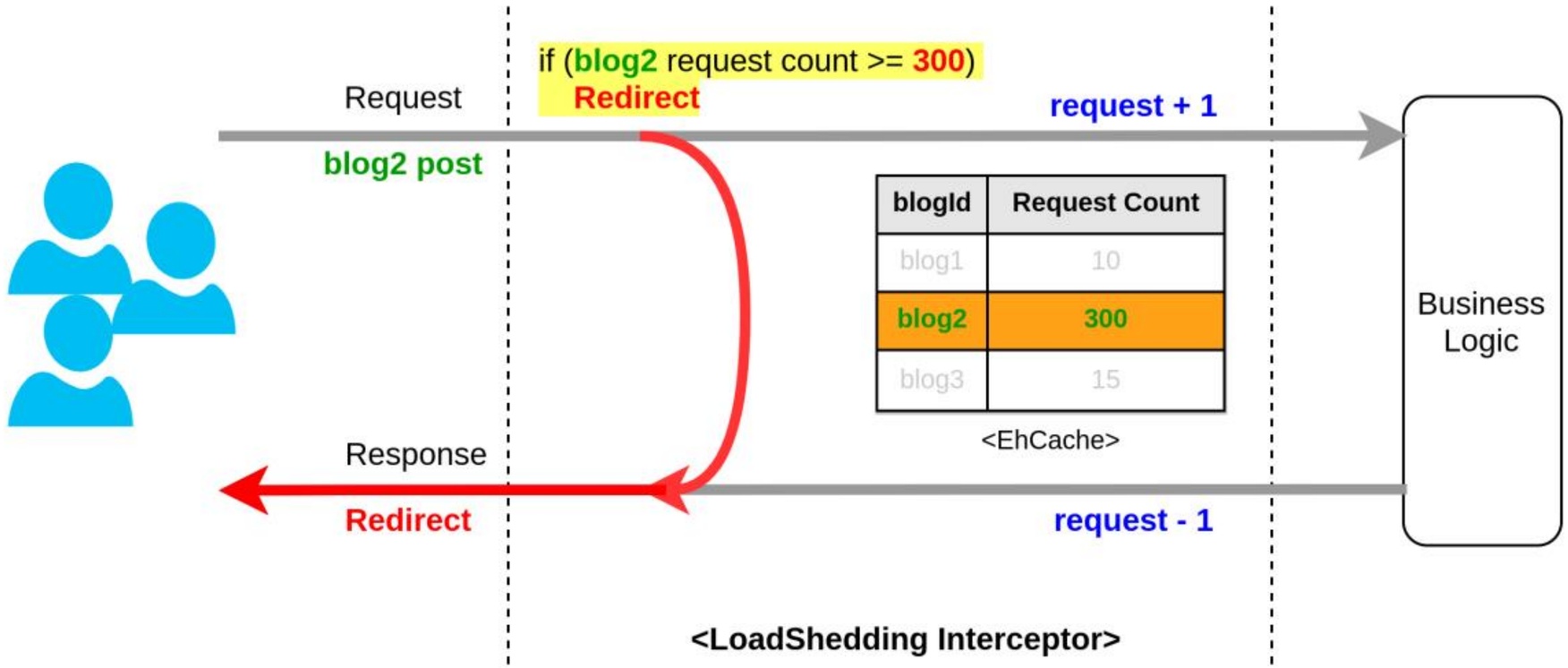
- 전체의 공급 불가를 막기 위해 일부의 공급을 제어하는 방식



4.3 주요 트래픽 제어 기술

Load Shedding

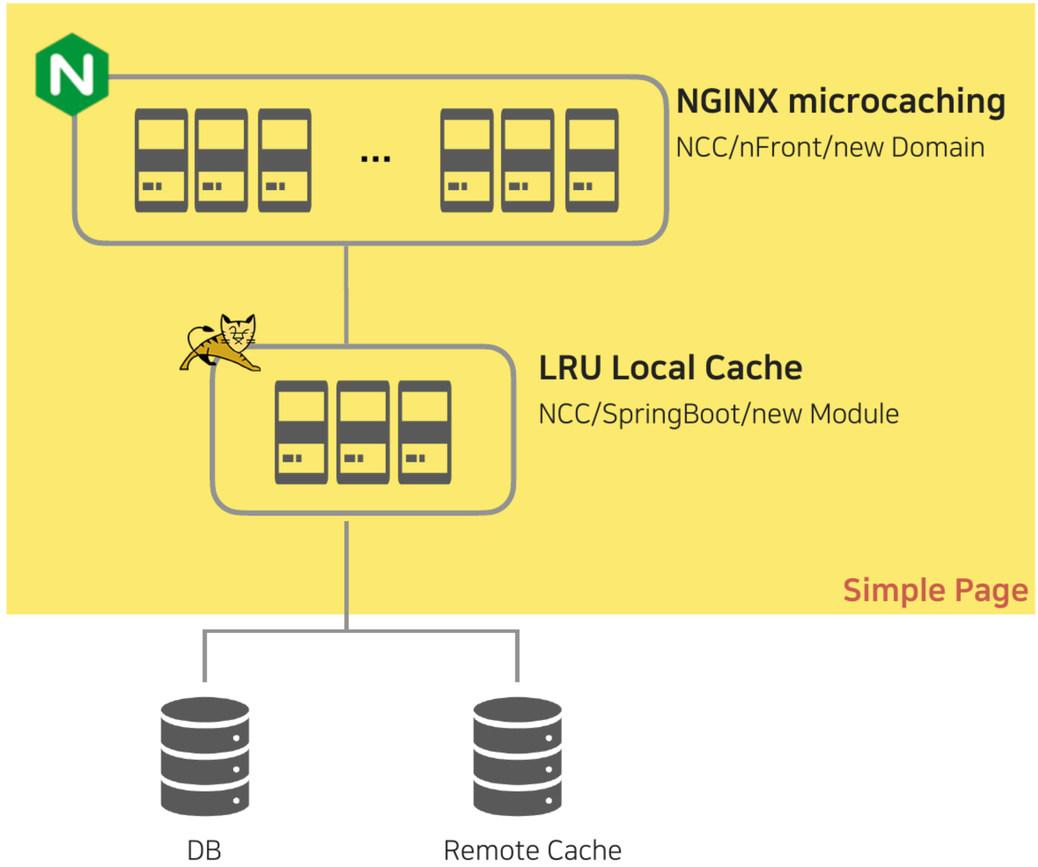
- 특정 페이지 및 블로그 ID를 기반으로 요청 카운트를 통해 제어



4.3 주요 트래픽 제어 기술

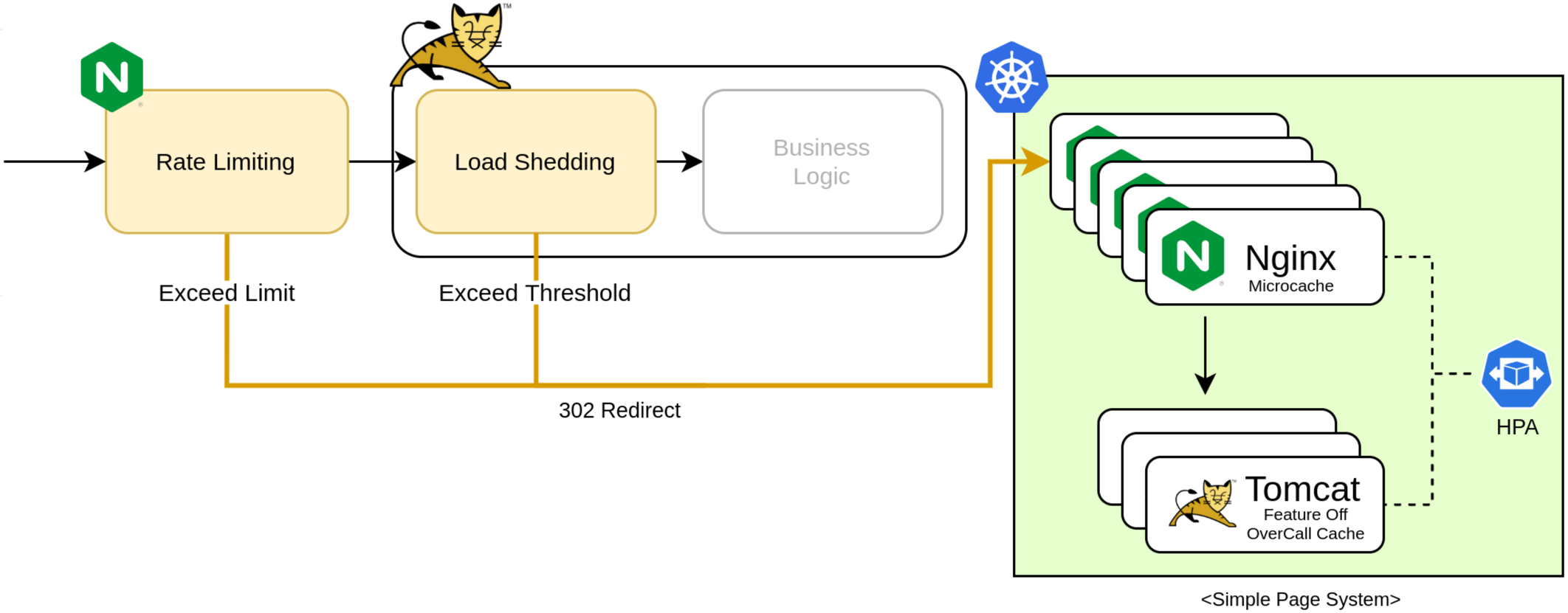
Simple Page System

- 트래픽 수용 목적의 엔드 페이지 전용 시스템
- 대규모 Spike 트래픽에 초점을 맞춘 셋팅
- 읽기전용, 긴 캐시 Expire, Tomcat 트래픽 방어
- Auto scale-out: 36만 TPS -> 72만 TPS
- 평소대비 28배 트래픽 대응, 최대 40배



4.3 주요 트래픽 제어 기술

Simple Page System

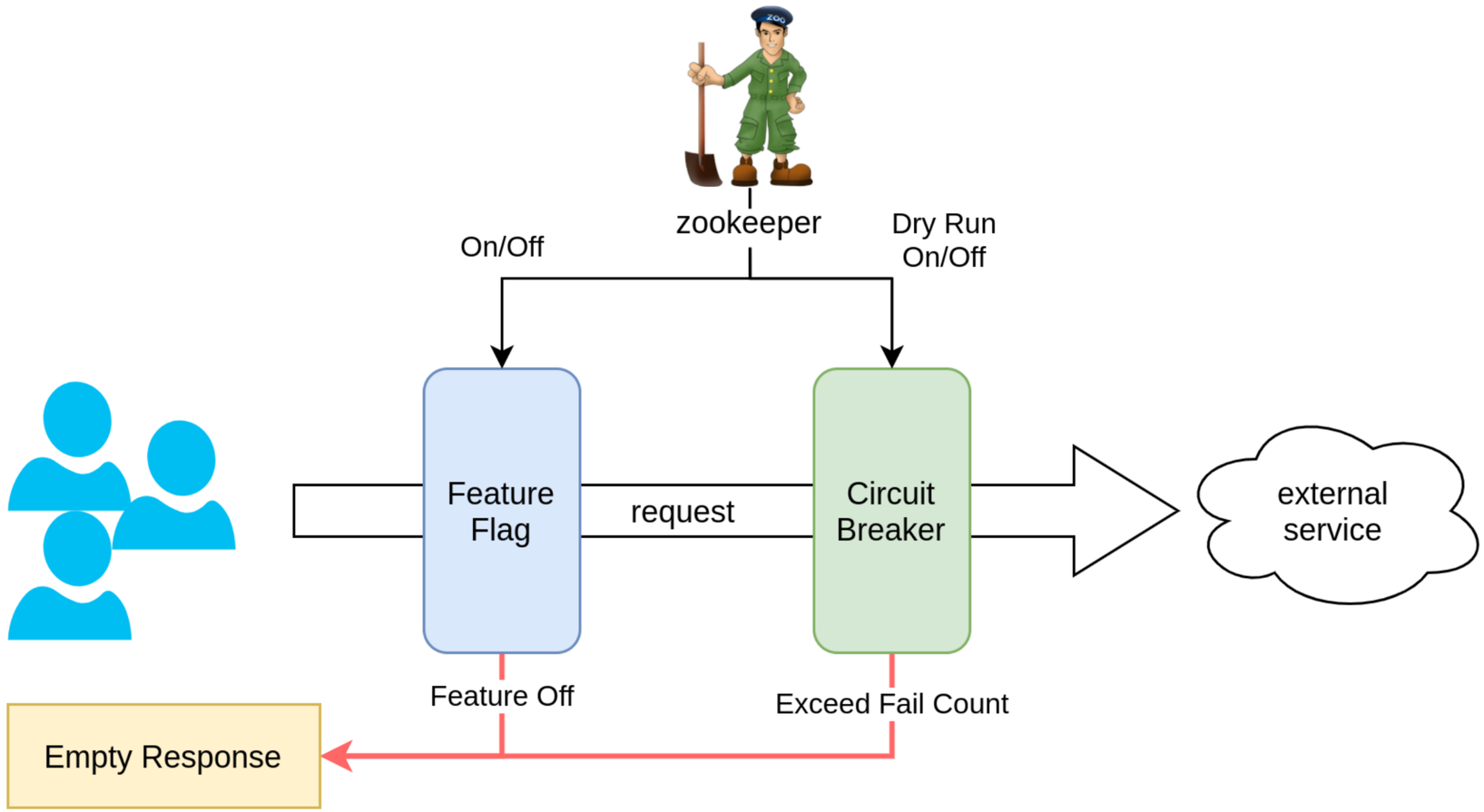


<Simple Page System>

서비스 핵심 페이지의 트래픽 제어로 트래픽 우회 시킨 후 Simple Page System에서 최대한 트래픽 수용

4.3 주요 트래픽 제어 기술

Feature Flag + Circuit Breaker



DEVIEW 2021

5. SRE - Incident Response

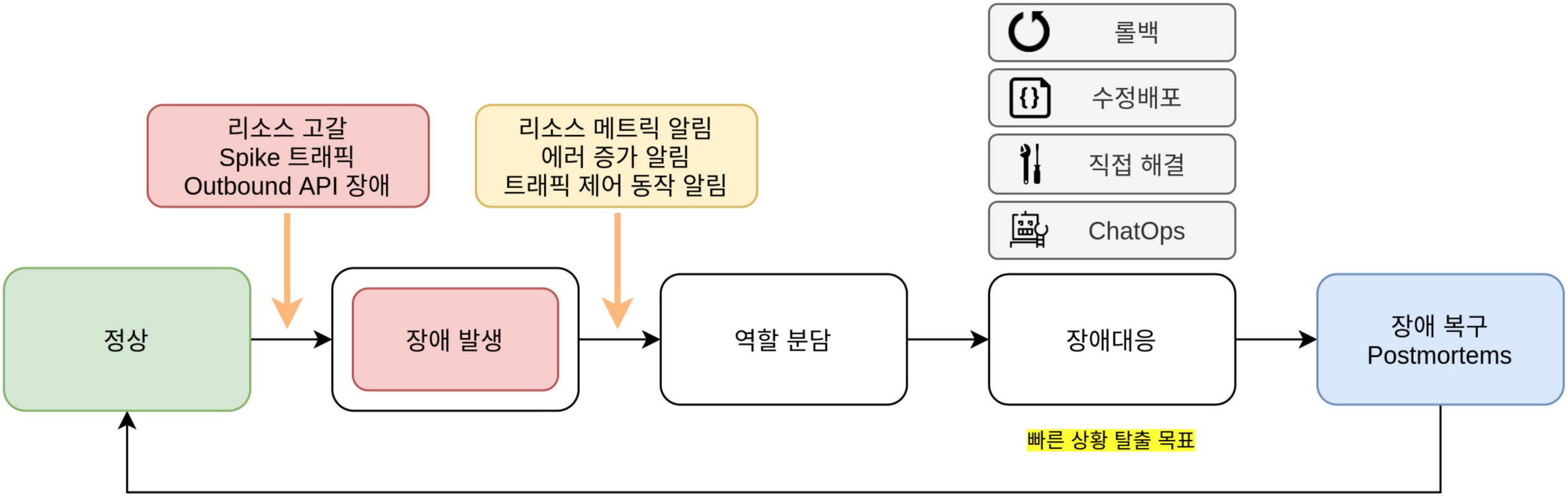
5.1 Incident Response의 의미

높은 서비스 신뢰성 유지에 필요한

사고 대응 프로세스 체계를 세우고

빠른 복구를 위한 기반 환경 조성

5.2 장애대응체계


롤백
수정배포
직접 해결
ChatOps
리소스 고갈
Spike 트래픽
Outbound API 장애
리소스 메트릭 알림
에러 증가 알림
트래픽 제어 동작 알림
정상
장애 발생
역할 분담
장애대응
장애 복구
Postmortems
빠른 상황 탈출 목표

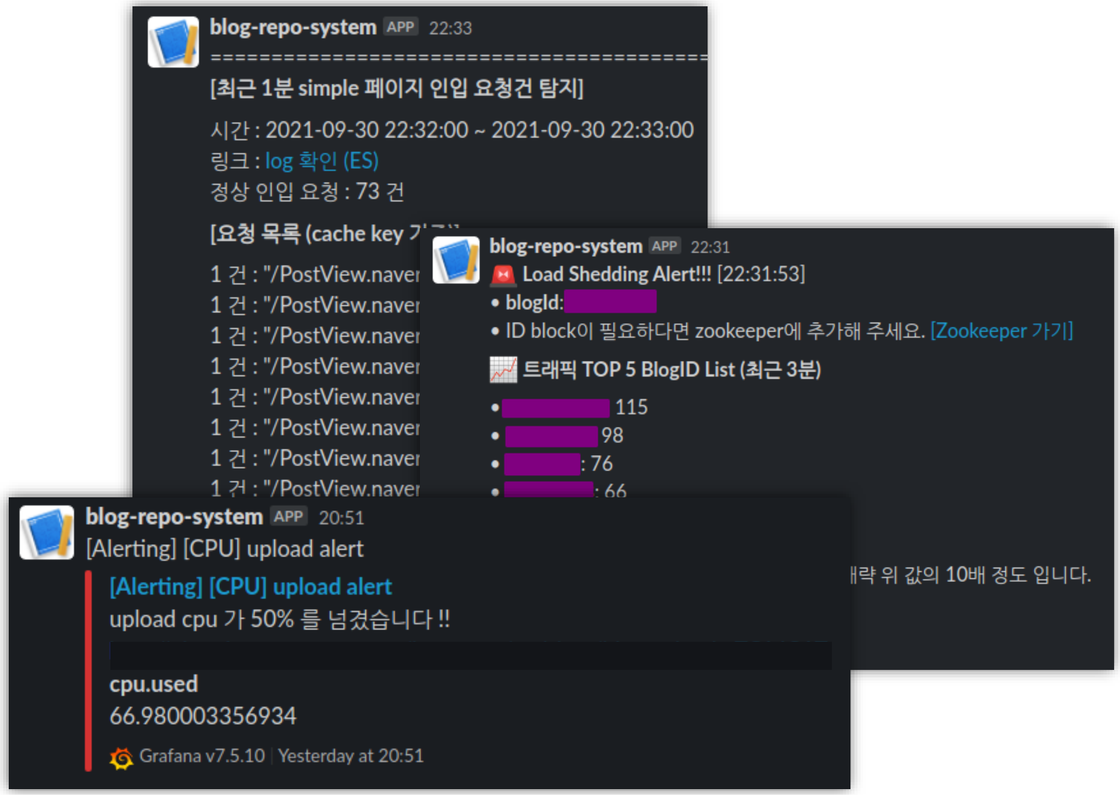
5.3 알람 고도화

서버 이슈 알람 (즉각적인 대응)

- CPU, Memory, Network
- Nginx active connections
- Healthcheck

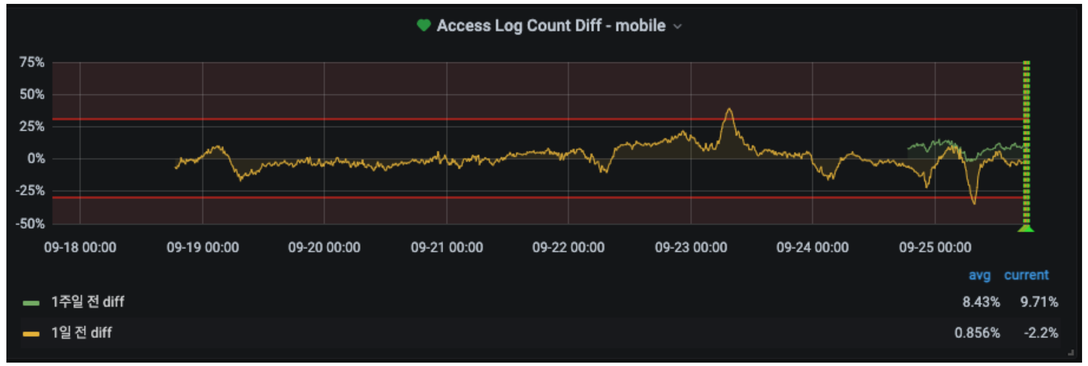
트래픽 제어 동작 알람

- Load shedding 발동
- Simple page system 인입

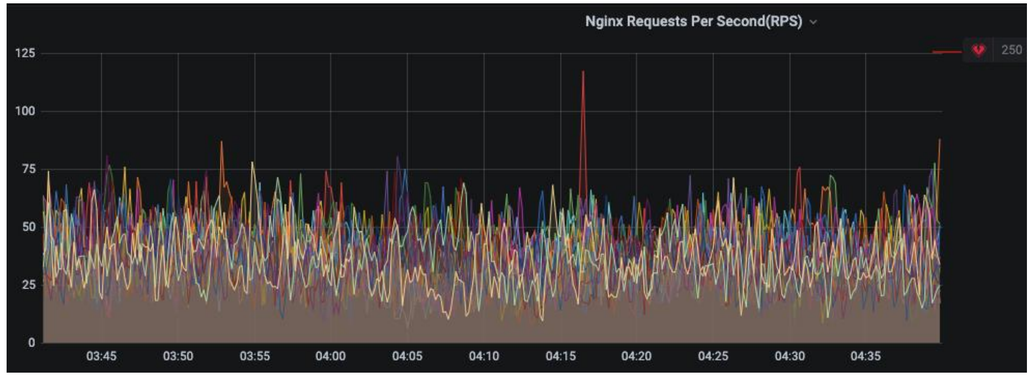


5.3 알람 고도화

Traffic 이상 탐지



Access log count diff (1d / 1w)



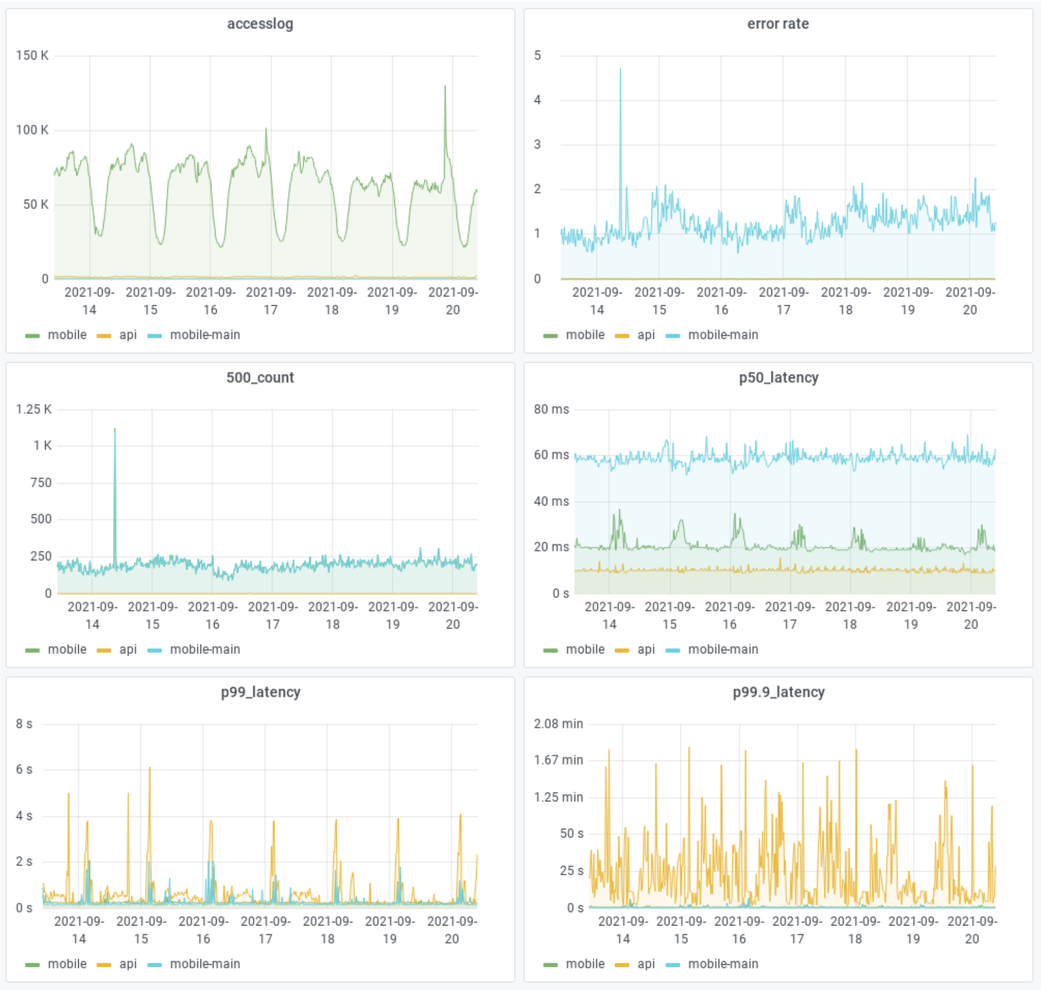
Nginx RPS 급증

5.3 알람 고도화

트래픽 동향 알람 (서비스 트렌드 대응)

- 장기적인 변화 확인
- 주요 서비스 이벤트의 반응 추이 확인

장기적인 트래픽 변화 추이에 대응



5.4 배포 고도화

Blue/Green, Canary Deploy

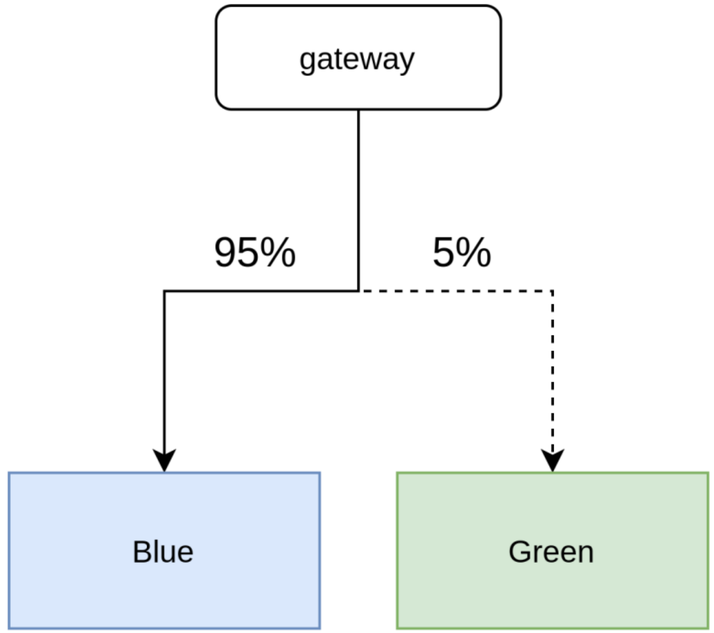
On-premise/Kubernetes 환경 모두 사용 가능한 빠른 배포 및 롤백 전략

On-premise

- Nginx split_client 기반 제공
- 빠른 롤백을 위한 Spare 상시 대기

Kubernetes

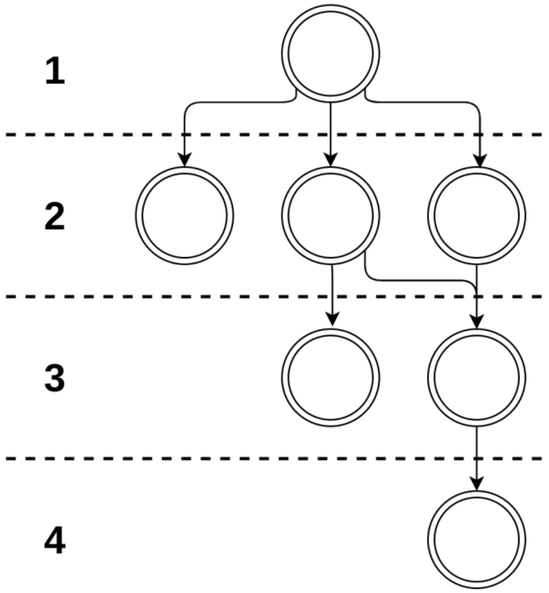
- 사내 플랫폼 n2c를 통해 제공되는 Blue/Green

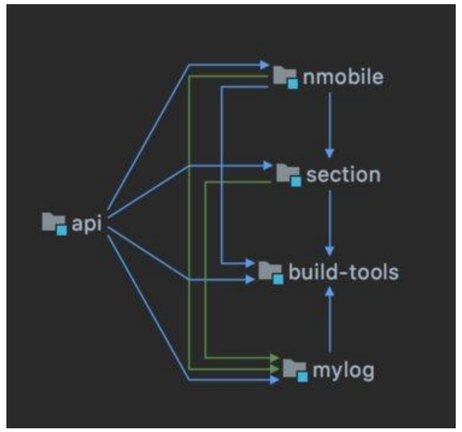


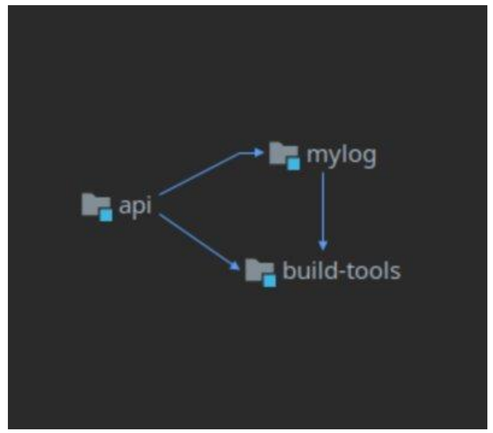
5.4 배포 고도화

빌드 구조 개선으로 얻었던 빠른 코드 수정 대응 환경

- 빌드 과정 병렬화 (n2c 병렬 빌드, Build tool 병렬기능 지원)
- 패키지 의존성 단순화 → 병렬성 향상
- 배포 영향범위 국지화 (통합 빌드 → 개별 빌드 추가)







5.5 Slack 기반 ChatOps 통합환경

모니터링 자동화 및 장애 대응

- Slack으로 인입된 알람 확인에 필요한 Metric 자동 확인
- Zookeeper를 통한 Application 설정 변경 기능 구현

<자동 Metric 조회>

<Zookeeper 설정 변경>

6. 마무리 - 신뢰성 있는 서비스 만들기 위한 추가적인 노력들

6.1 테스트 강화 - 선제적 장애 예방

DEVIEW 2021

Cucumber - End-To-End 인수 테스트 검증

- 방대한 스펙으로 인해 수정시 발생되는 사이드 이펙트 테스트

cucumber

```
@nbaseCsvLoad
@ckey=nmobile_id01
@arcus=blog:700000001,blogNo:nmobile_id01,buddyCollection:700000001
Feature: [ckey=nmobile_id01] 공감 목록을 조회한다.
  Background:
    Given 블로그 아이디 파라미터는 "nmob
    Given 페이지는 1 이고,

  Scenario: 블로그 주인이 게시글 공감목록
    Given 게시글 아이디 파라미터는 70000
    Given 사용자는 "nmobile_id01" 이고,
    Given 공감 카테고리 ID는 "POST" 이고
    Given 공감 유저 리스트 api가 응답되고
    When 공감 유저를 조회하면,
    Then 공감 유저가 조회된다.
```

	Steps						Scenarios			Features	
Feature	Passed	Failed	Skipped	Pending	Undefined	Total	Passed	Failed	Total	Duration	Status
쿠폰 발급 여부를 조회한다.	61	0	0	0	0	61	9	0	9	0.072	Passed
쿠폰을 발급한다.	80	0	0	0	0	80	12	0	12	0.926	Passed
주문을 수행한다.	176	0	0	0	0	176	15	0	15	0.807	Passed
상품정보를 조회한다.	42	0	0	0	0	42	4	0	4	0.244	Passed
주문을 수행한다	154	0	0	0	0	154	15	0	15	0.634	Passed
상품정보를 조회한다	36	0	0	0	0	36	4	0	4	0.302	Passed
	1306	**0**	**0**	**0**	**0**	**1306**	**149**	**0**	**149**	**4.727**	**11**
	100.00%	**0.00%**	**0.00%**	**0.00%**	**0.00%**		**100.00%**	**0.00%**			**100.00%**

6.1 테스트 강화 - 선제적 장애 예방

DEVIEW 2021

goss

- Server testing/Validation 툴
- 서버 환경 변경시 사이드 이펙트 검증
- CI 및 설정 배포 상황에서 감지
- 단일 바이너리 파일
- Yaml 기반 테스트 환경 정의

Supported resources

- package - add new package
- file - add new file
- addr - add new remote address:port - ex: google.com:80
- port - add new listening [protocol]:port - ex: 80 or udp:123
- service - add new service
- user - add new user
- group - add new group
- command - add new command
- dns - add new dns
- process - add new process name
- kernel-param - add new kernel-param
- mount - add new mount
- interface - add new network interface
- http - add new network http url with proxy support
- goss - add new goss file, it will be imported from this one
- matching - test for matches in supplied content

6.2 마무리

DEVIEW 2021

SRE는 서비스의 신뢰성을 어떻게 높이는가

Metrics & monitoring

시스템 가시성 확보, 이슈 감지, 사후 분석에 필요한 모니터링 환경 제공

Capacity planning

서비스 가용성 및 이를 오버하는 트래픽을 전략적으로 핸들링

Incident response

장애를 거칠수록 보완되는 장애대응 프로세스

Thank You