NEObserver 프로젝트 Alpha 개발 결과 발표

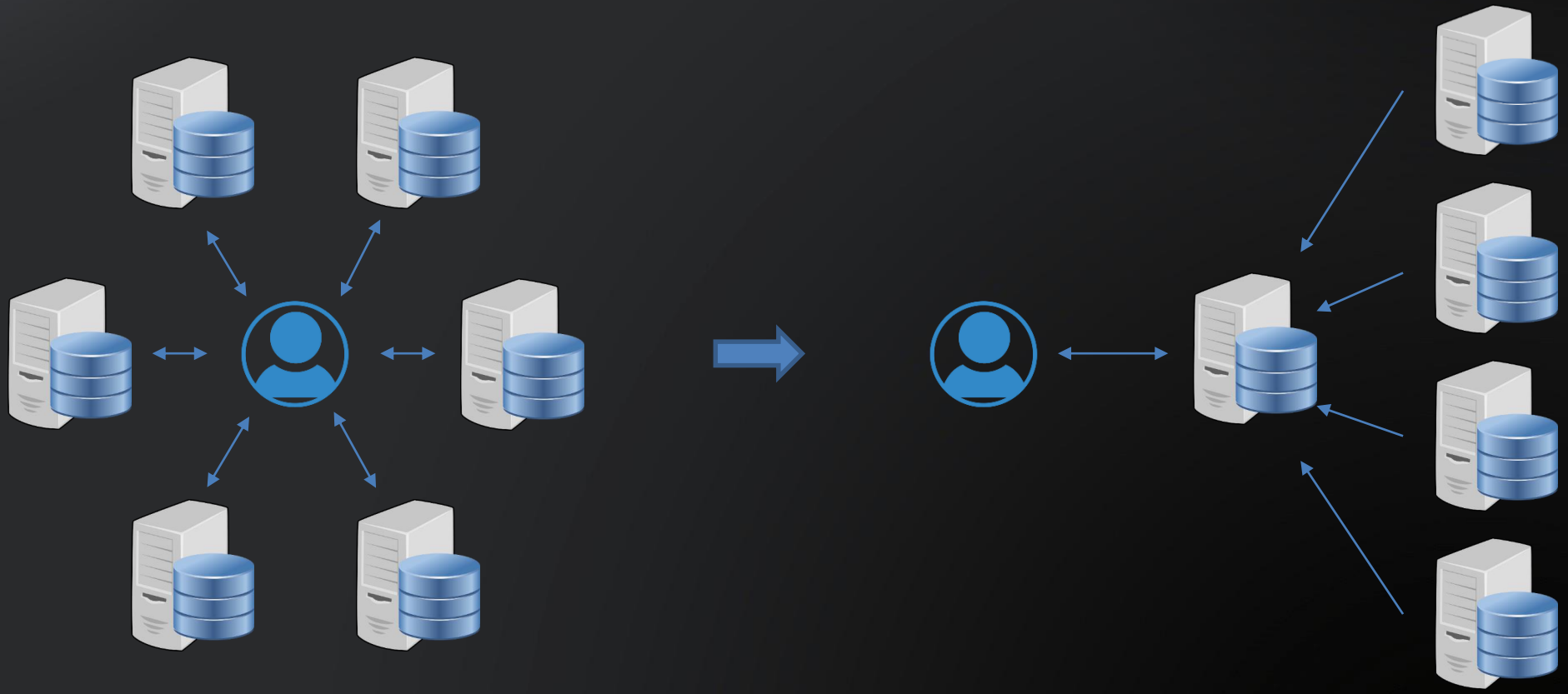
2018.02.21
작성 : 게임온기술팀 이동기
검수 : 김규현

발표순서

1. 개요
2. 기능
3. 접근방법
4. 구현결과

1. 개요

프로젝트 선정 배경

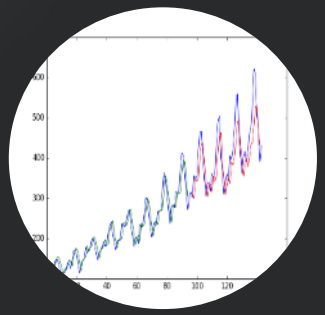


NEOWIZ ABLE STUDIO

NEObserver 란

NEObserver 는 모든 DB서버들의 OS와 DBMS상태를 하나의 View에서 확인하고 관리할 수 있게 도와줍니다.
또한 우수한 확장성으로 문제파악에 필요한 정보를 필요한 순간에 추가할 수 있습니다.

프로그램의 특징



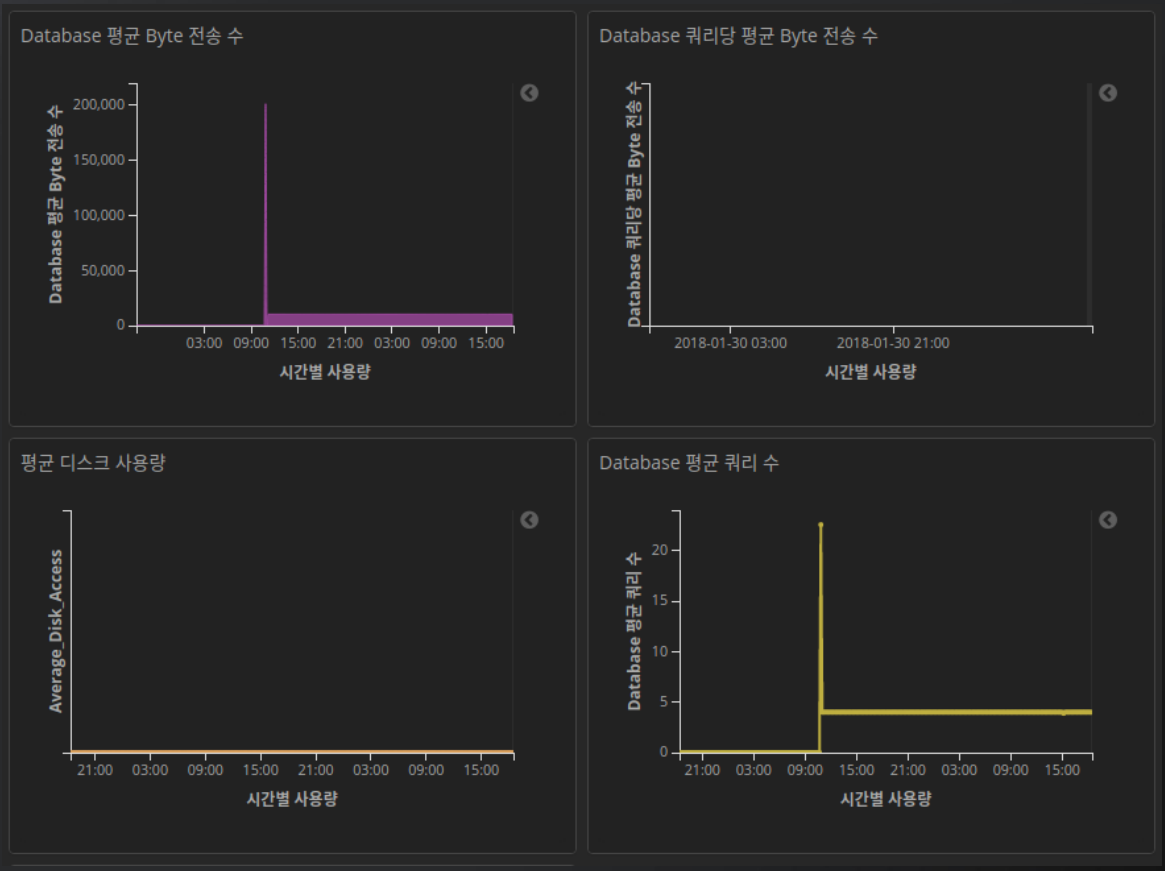
문제 예측 지원

편리한 성능 관리

효율적인 자원 활용

낮은 리소스 사용

문제 예측(Human Support)



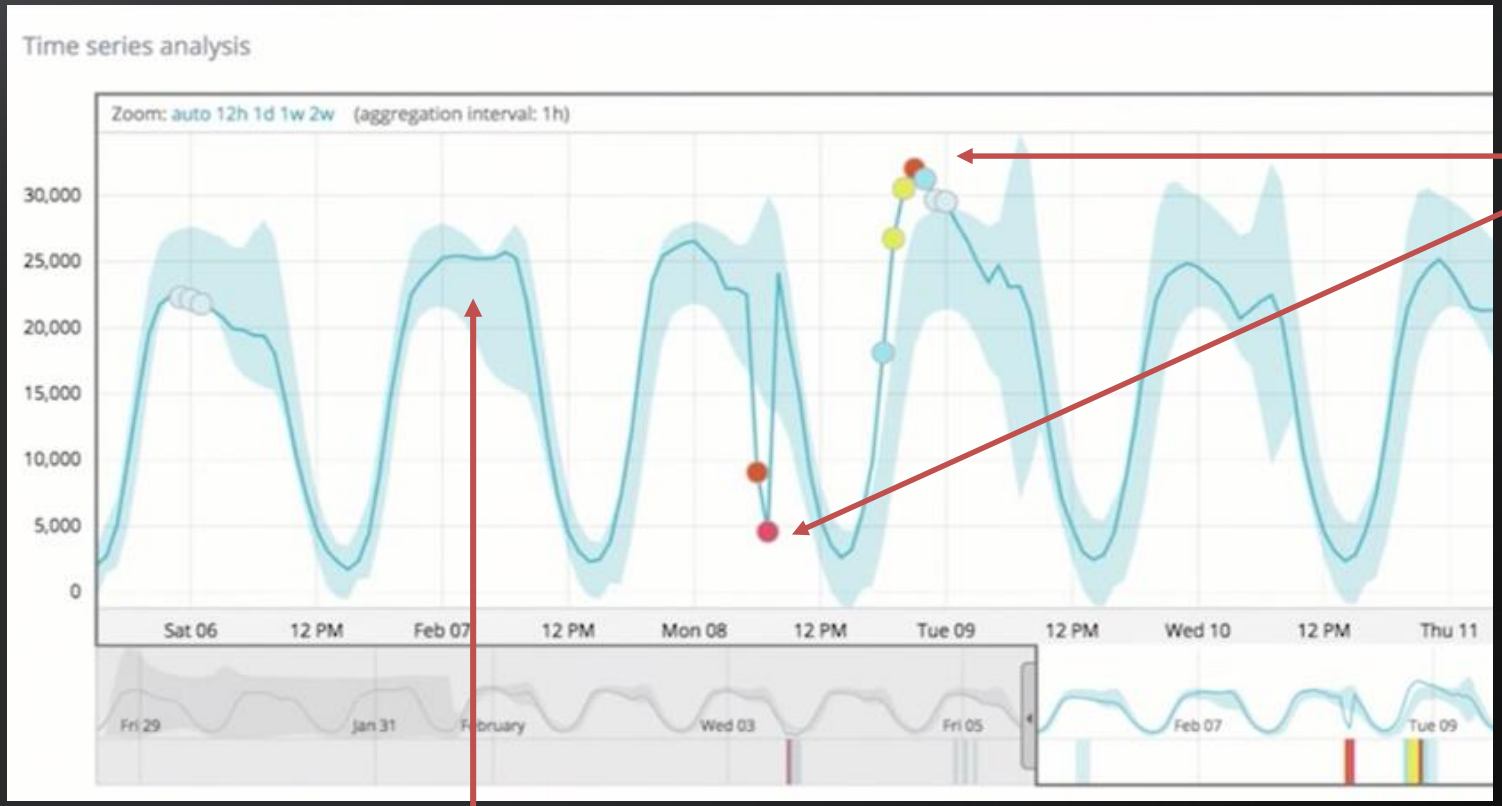
Monitoring 항목
이상징후 판별

문제 예측(Automatic)

Machine Learning

비지도 학습을 통한 이상징후 자동 판단

Single Matric 문제 예측

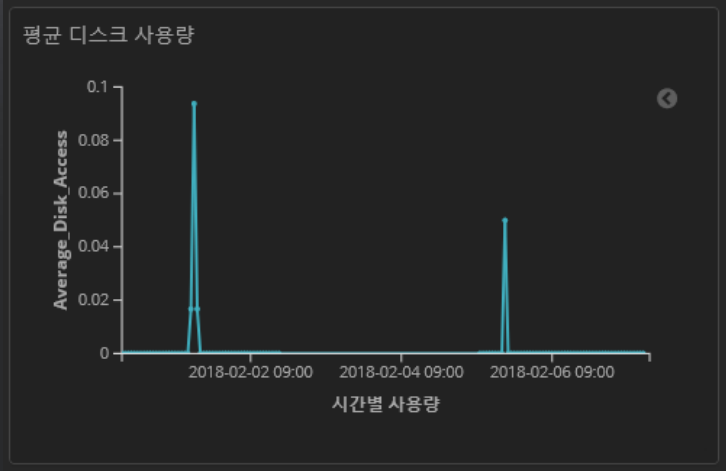


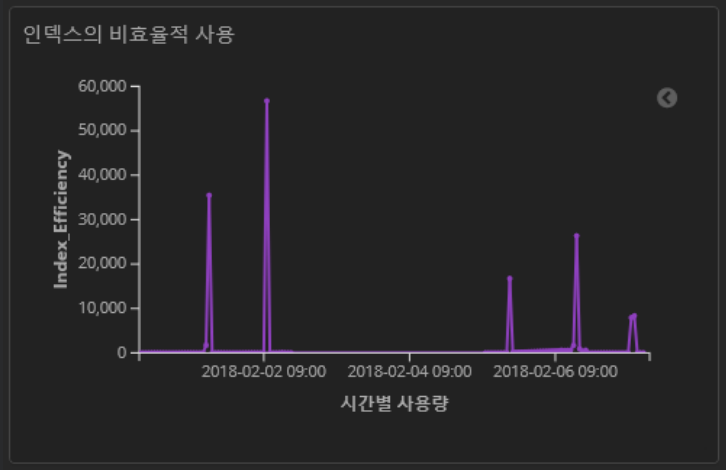
이상징후 판단

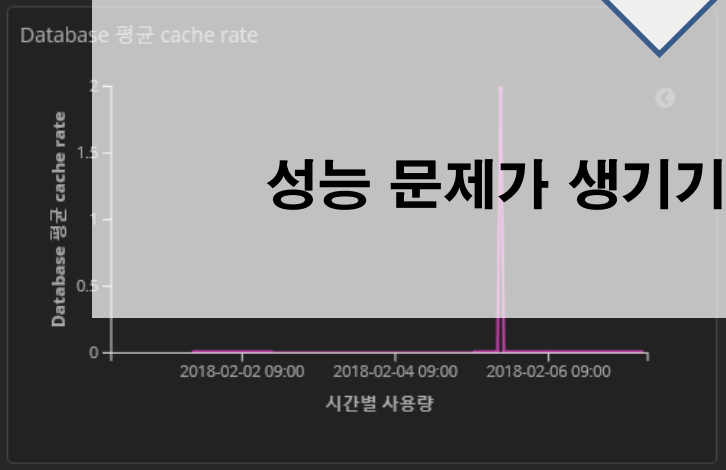
일상 패턴 학습(허용 범위의 학습)

각 항목으로 확장

성능 관리



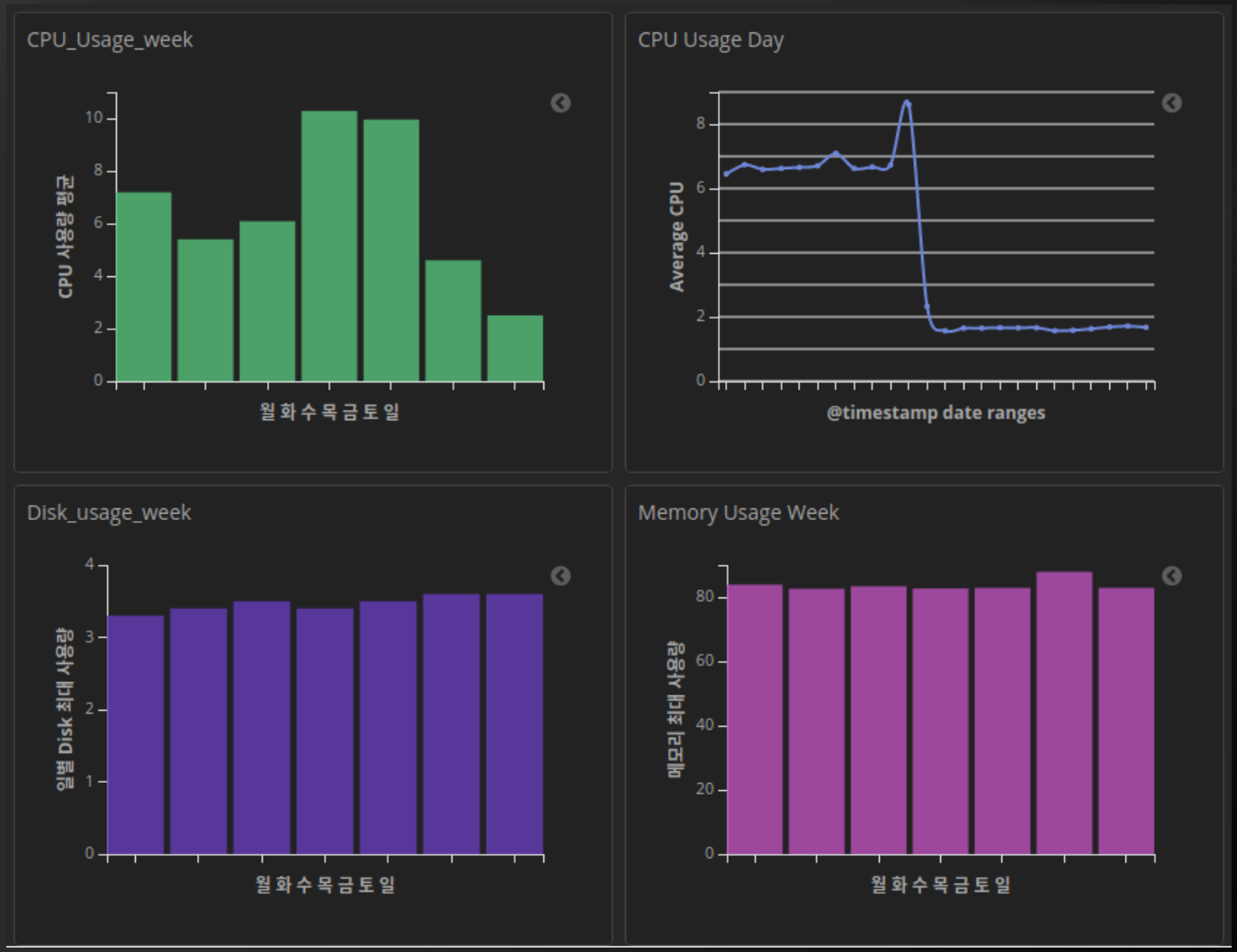




성능 저하의 문제를 각 항목을 통해 파악

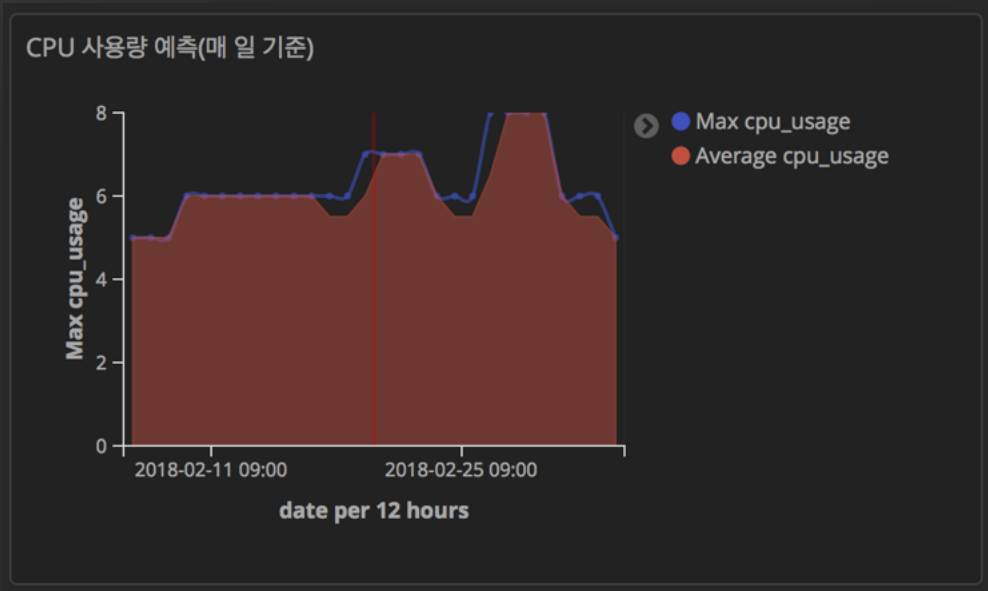
성능 문제가 생기기 전 조치 가능

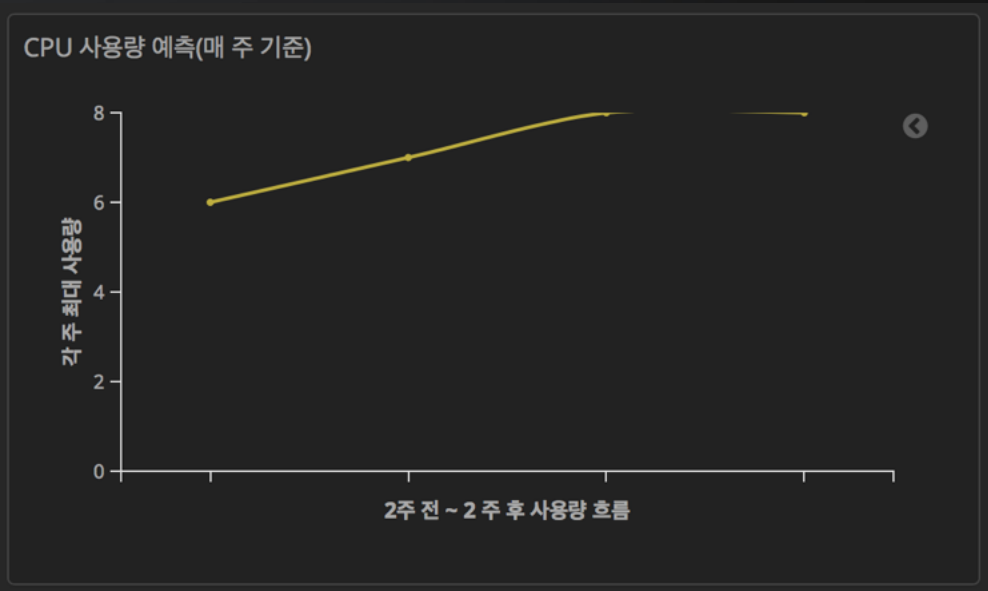
자원 활용



시간별 날짜별 사용량

자원 활용





미래 자원사용량 예측

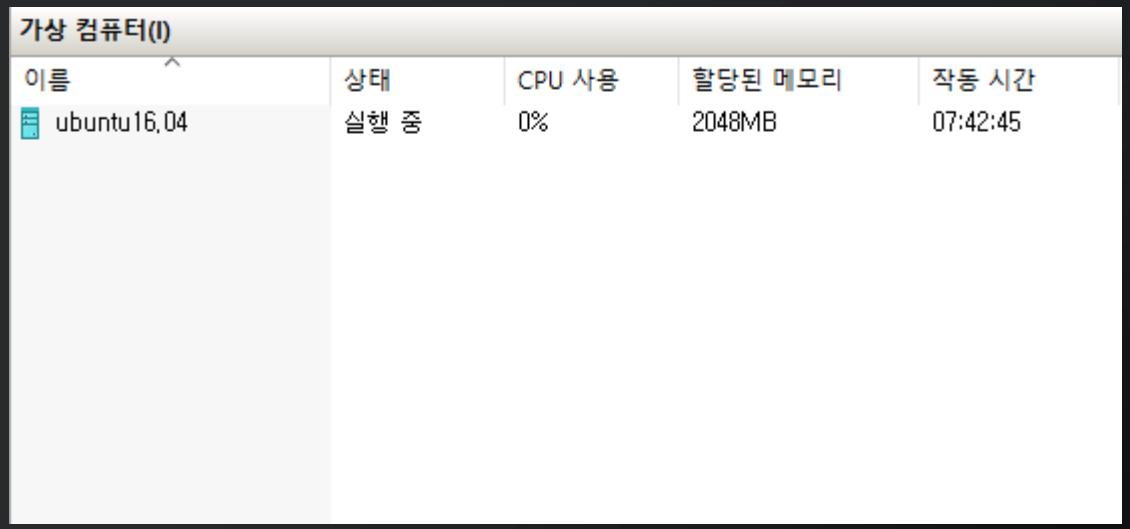
=>효과적인 자원 사용 가능

낮은 리소스 사용량

DB Server에 존재하는 Client는 정보 수집만을 담당

Resource 가 들어가는 작업은 Monitoring Server에서 담당

낮은 리소스 사용량



Client 가 실행 중일 때 0~1%의 리소스를 사용하는 것을 확인

2. 기능

제품의 기능

NEOWIZ ABLE STUDIO

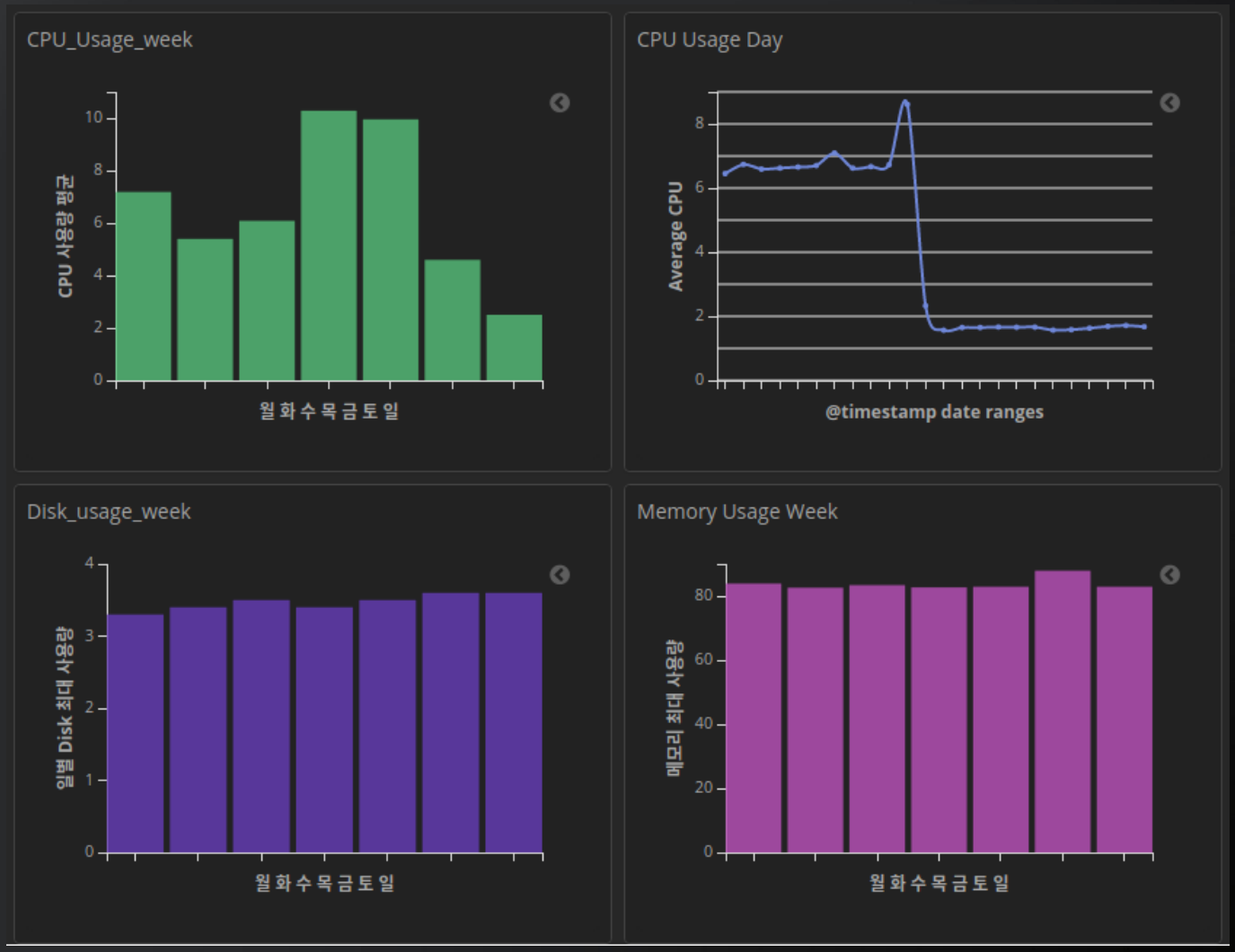
1. System Management System

2. Database Performance Management

3. Database and OS log Management

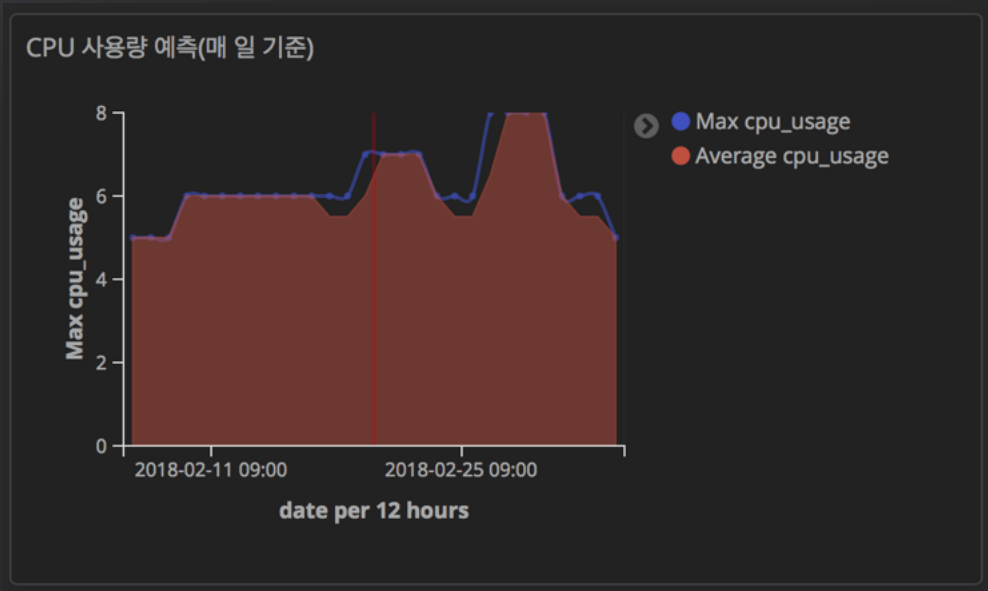
4. 실시간 문제 자동 탐지 시스템(추가 구현)

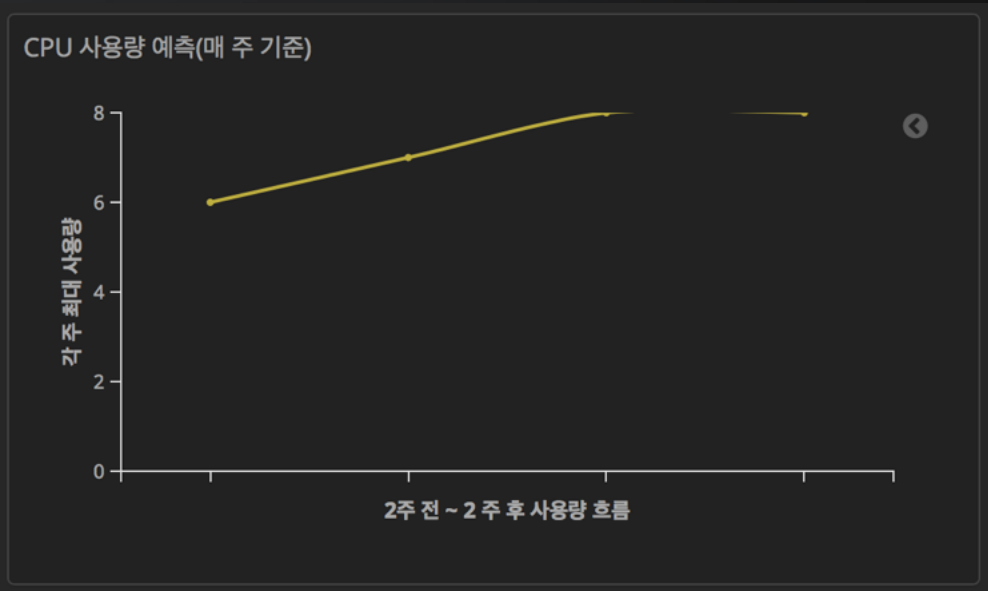
1. System Management System



각 항목에 대한 시간별, 날짜별 사용량 평균을 계산하여 표시

1. System Management System





평균값 간의 변화 량을 계산하여 미래 사용량 예측 모듈을 제공

제품의 기능

1. System Management System

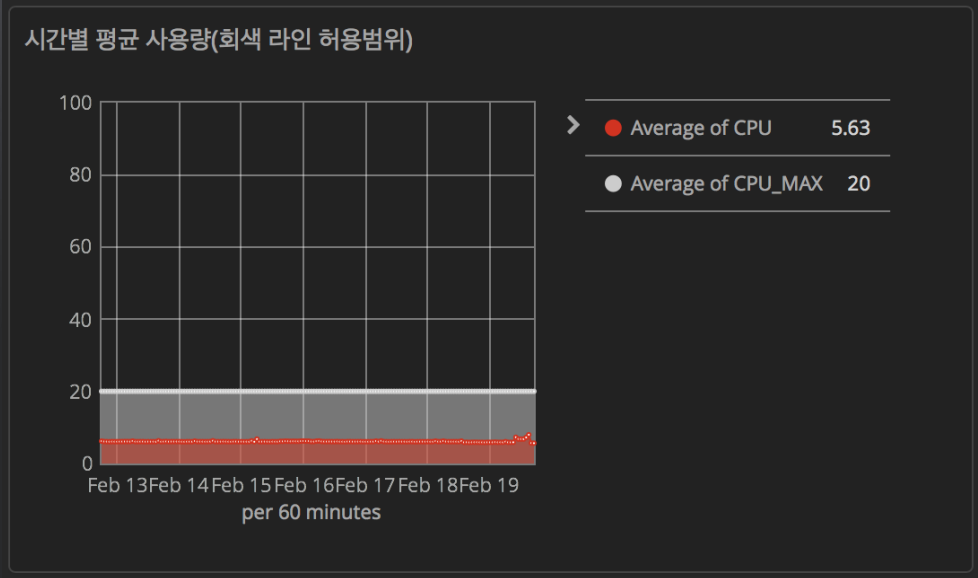
2. Database Performance Management

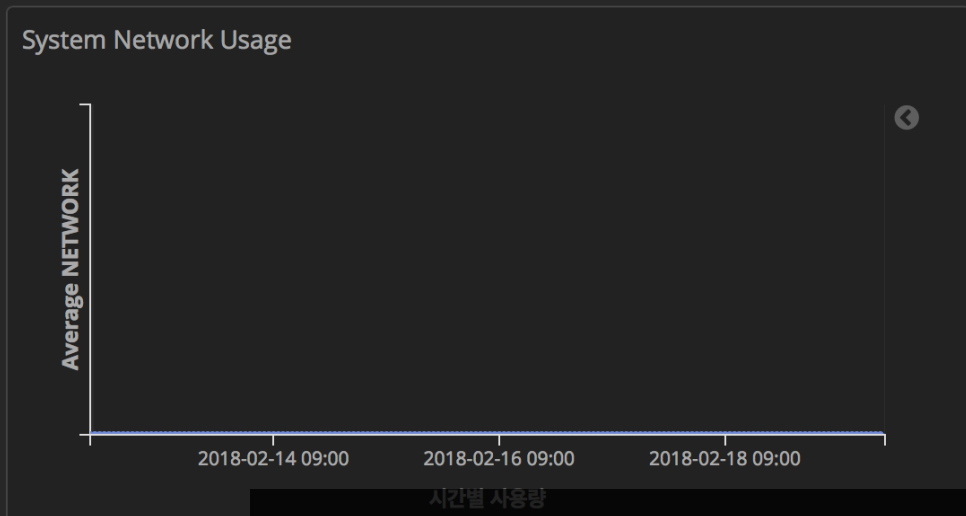
3. Database and OS log Management

4. 실시간 문제 자동 탐지 시스템(추가 구현)

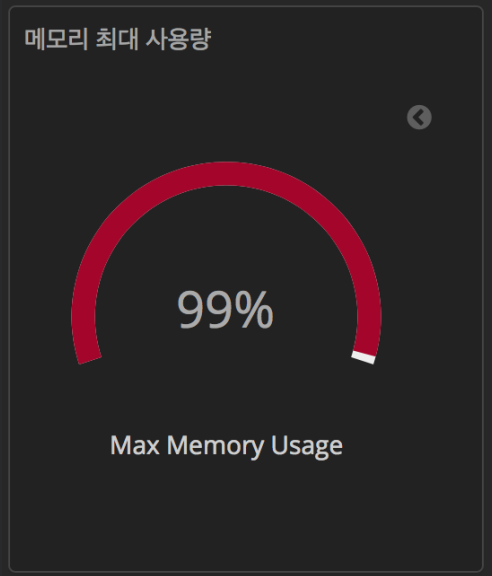
2. Database Performance Management

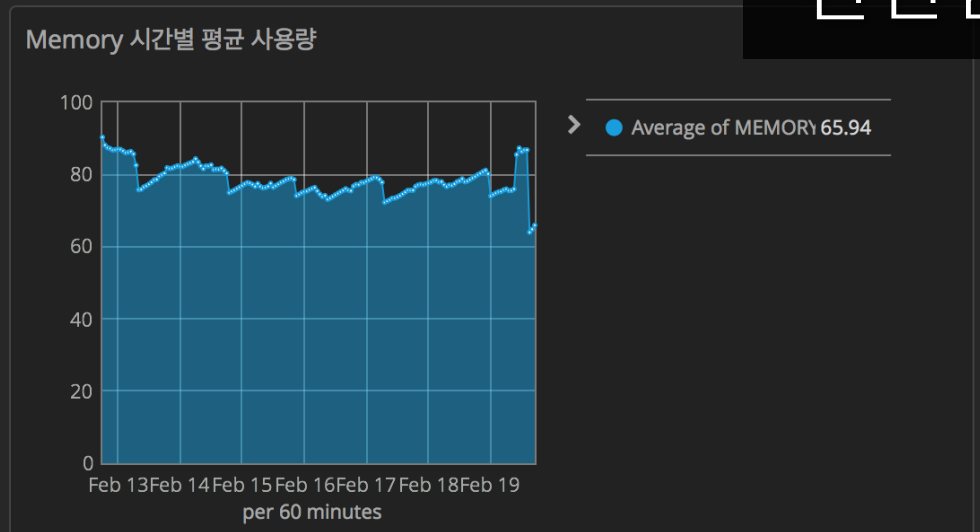
NEOWIZ ABLE STUDIO

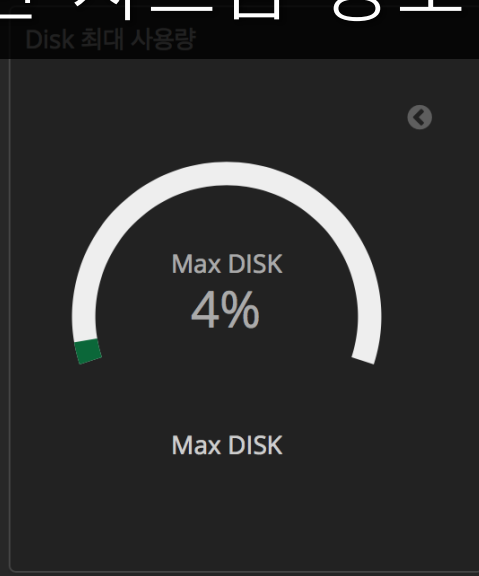




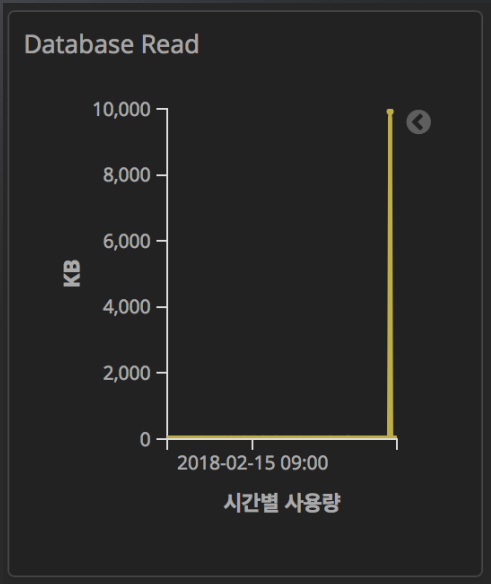
간단한 시스템 정보

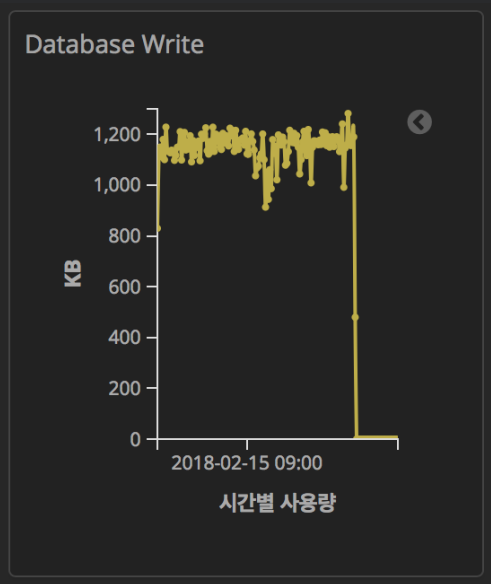


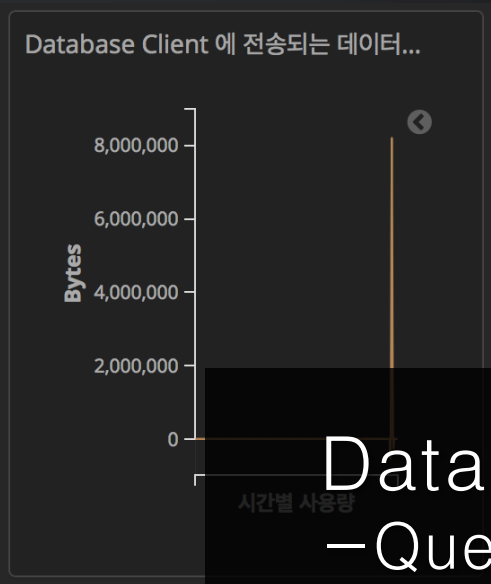


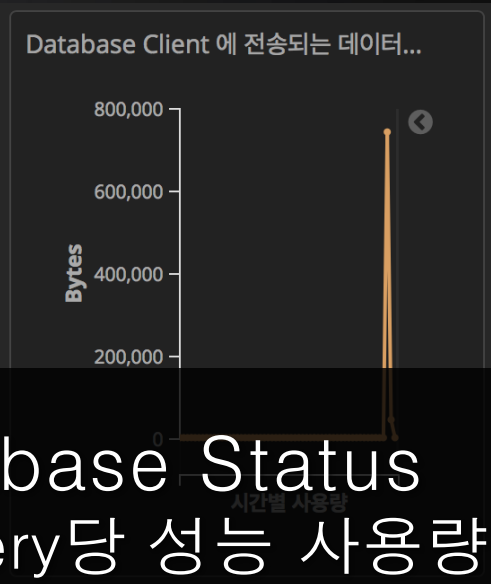


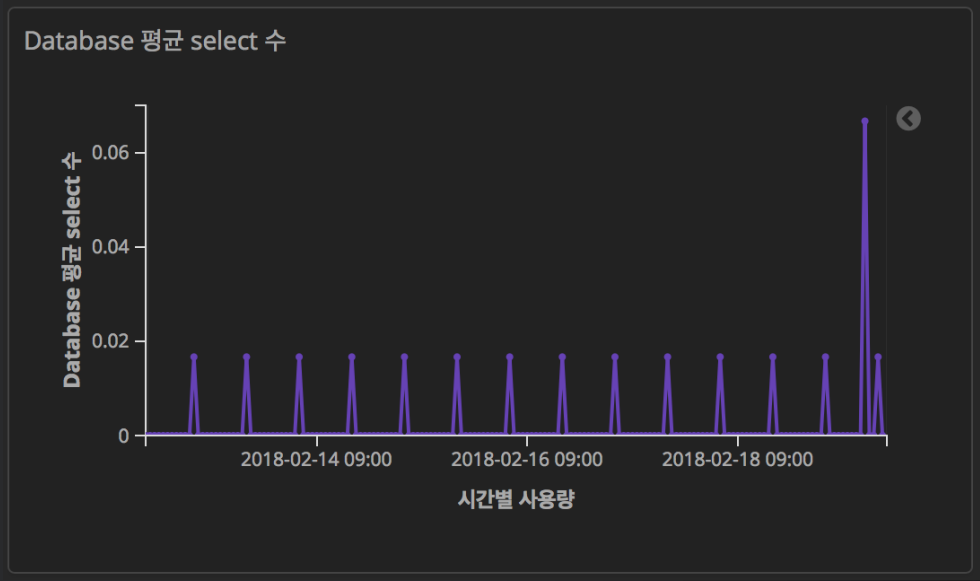
2. Database Performance Management







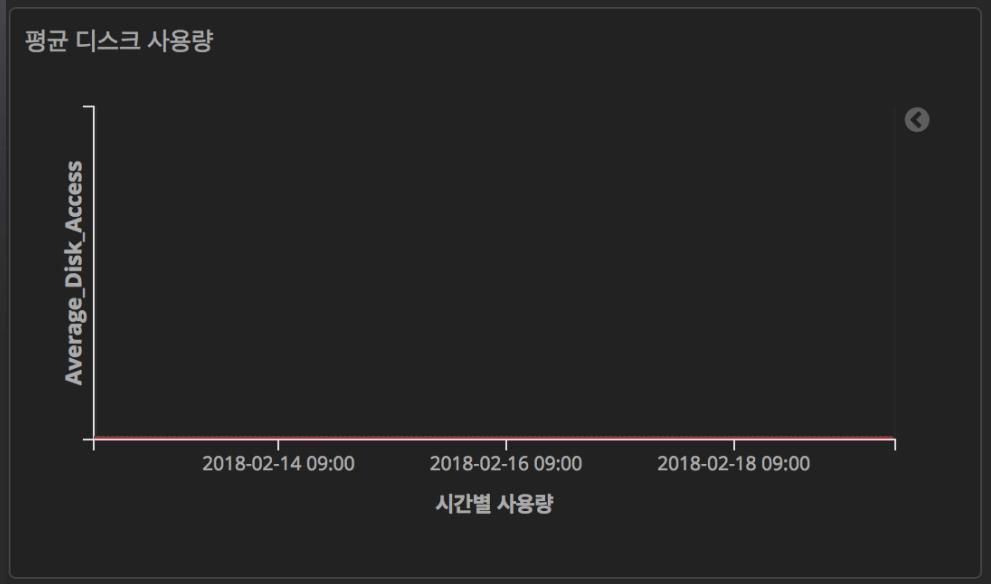


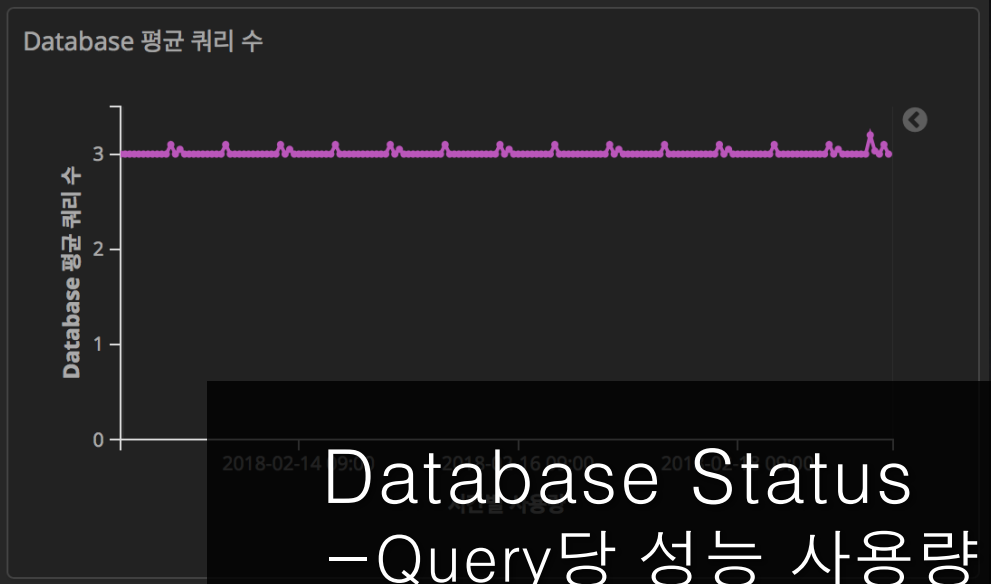


Database Status

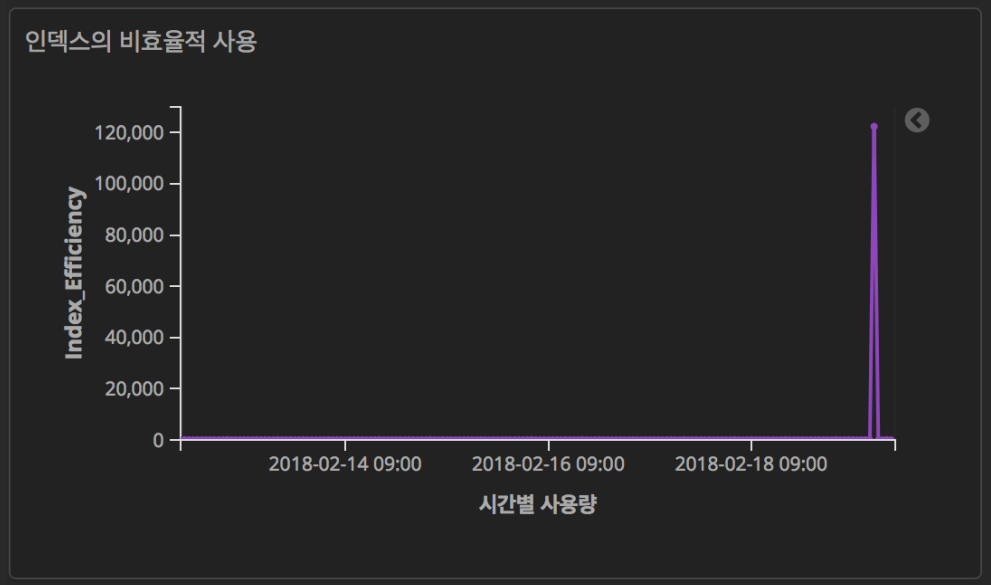
- Query당 성능 사용량
- Index 효율성
- 디스크 Read 양 등

2. Database Performance Management





Database Status
- Query당 성능 사용량
- Index 효율성
- 디스크 Read 양 등



제품의 기능

NEOWIZ ABLE STUDIO

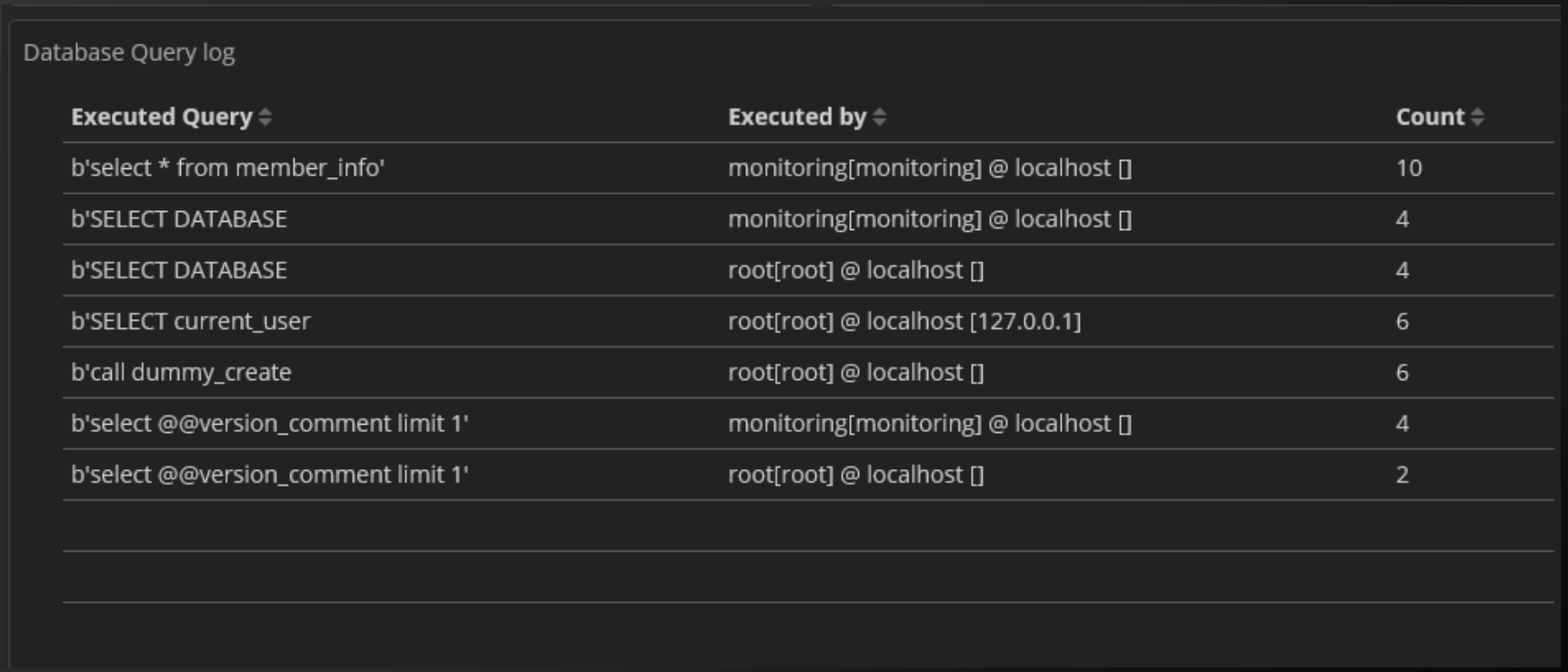
1. System Management System

2. Database Performance Management

3. Database and OS log Management

4. 실시간 문제 자동 탐지 시스템(추가 구현)

3. Database and OS log Management



Database Query log

Executed Query	Executed by	Count
b'select * from member_info'	monitoring[monitoring] @ localhost []	10
b'SELECT DATABASE	monitoring[monitoring] @ localhost []	4
b'SELECT DATABASE	root[root] @ localhost []	4
b'SELECT current_user	root[root] @ localhost [127.0.0.1]	6
b'call dummy_create	root[root] @ localhost []	6
b'select @@version_comment limit 1'	monitoring[monitoring] @ localhost []	4
b'select @@version_comment limit 1'	root[root] @ localhost []	2

Query log

3. Database and OS log Management

NEOWIZ ABLE STUDIO

문제 파악에 필요한 로그를 한곳에 저장
(System log, DBMS log)

하나의 컴퓨터에서 모든 서버의 로그 파악

빠른 문제 해결

제품의 기능

1. System Management System

2. Database Performance Management

3. Database and OS log Management

4. 실시간 문제 자동 탐지 시스템(추가 구현)

4. 실시간 문제 자동 탐지 시스템(추가구현)

NEOWIZ ABLE STUDIO

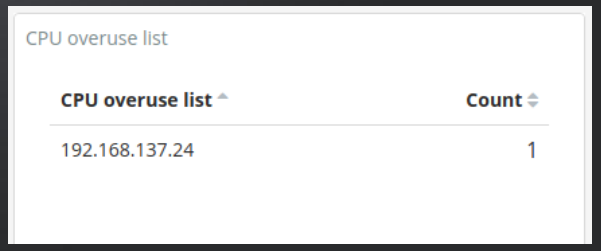
일상적인 패턴에 대해 학습

새로운 문제가 일어나기 전 자동 탐지

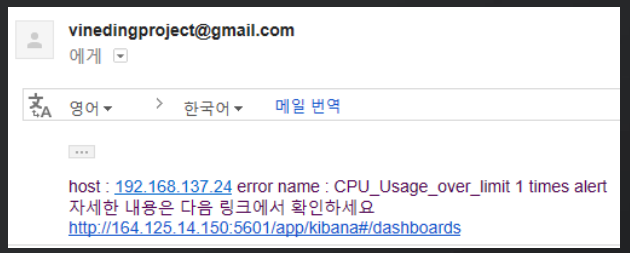
제품의 사용 흐름

NEOWIZ ABLE STUDIO

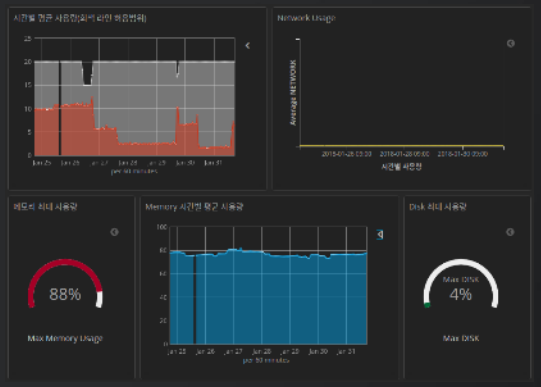
Error Dashboard



Email Alarm



문제가 있는 PC를 한번에 파악



해당 PC의 상세정보 제공

하나의 View로 전체 PC 관리가 가능해짐

3. 접근방법

사용 기술

NEOWIZ ABLE STUDIO

데이터 저장 흐름

Server Side

mongo DB

elastic

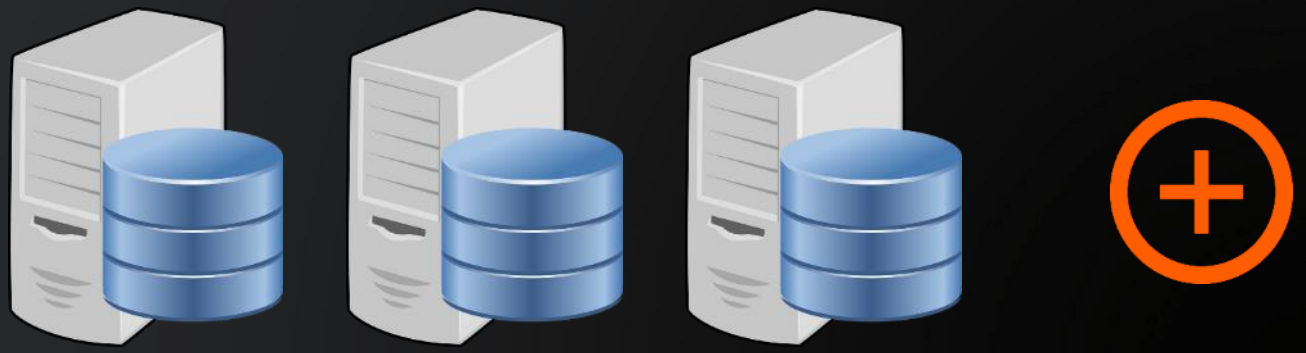
시스템 정보 처리 모듈

DB 정보 처리 모듈

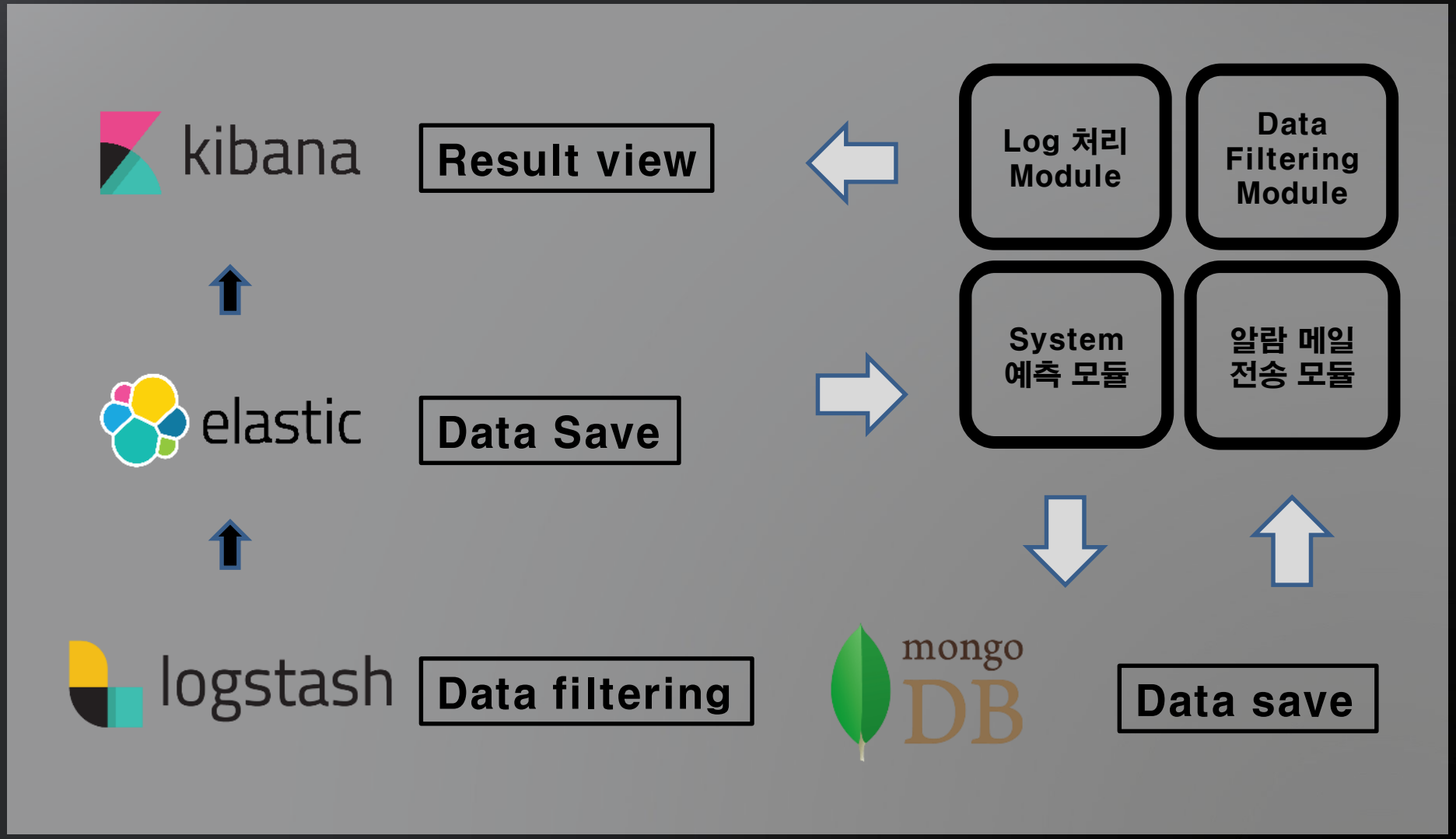
Log 정보 처리 모듈

Error 정보 처리 모듈

Client Side

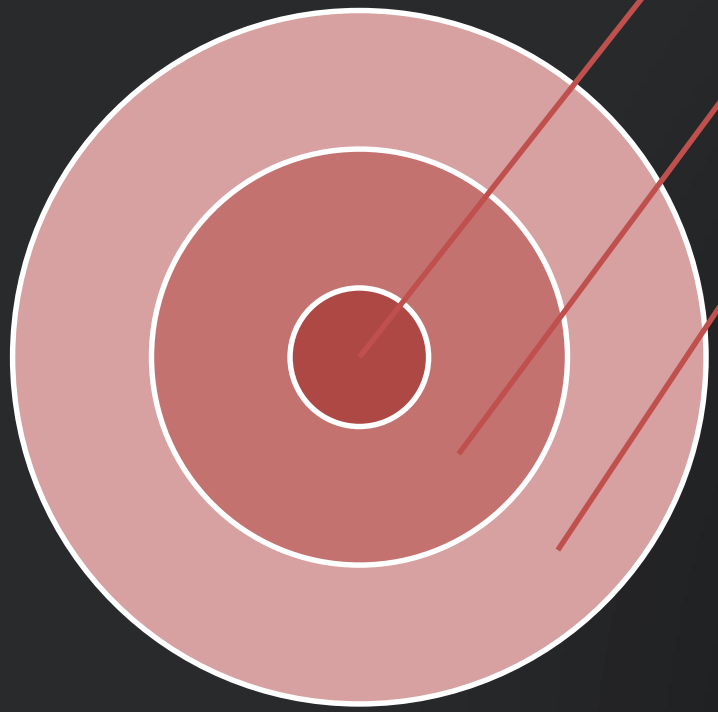


서버 내부 데이터 처리도



4. 구현결과

구현 목표



Alpha(6 week)

- 일부기능 구현
- 시제품 발표

Beta(7 week)

- 전체기능 구현
- 모든 항목 추가

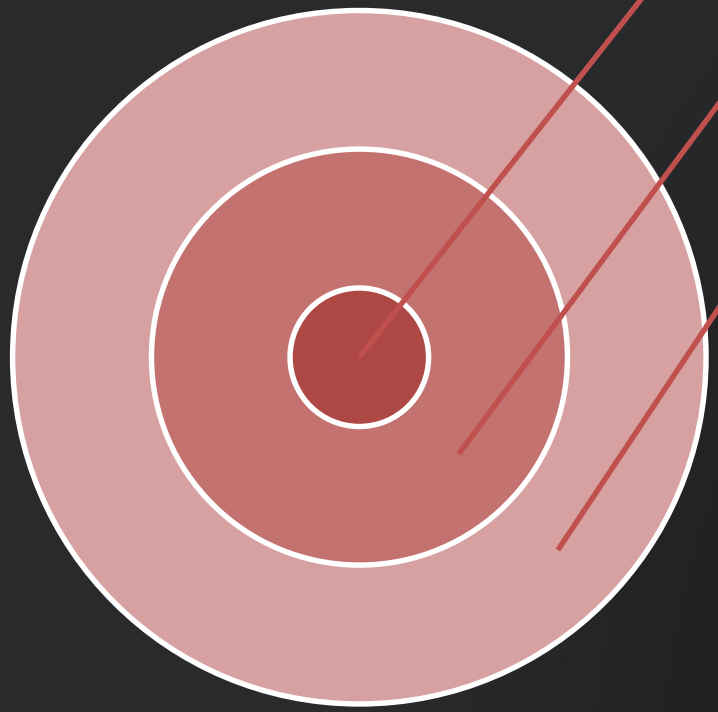
Release(13 week)

- 예측모듈 구현
- 제품화

총 개발기간

26주

구현 목표



Alpha(6 week)

- 일부기능 구현
- 시제품 발표

Beta(7 week)

- 전체기능 구현
- 모든 항목 추가

Release(13 week)

- 예측모듈 구현
- 제품화

Alpha 단계 구현 완료

(6주)

Alpha 목표 : 각 주차 별 구현 내용

1 주차 1/1 ~ 1/5	ELK 환경 구축	이동기	3h	1
	System 정보 수집방법 확인	이동기	4h	1
	System 정보 수집 제작 1(client side)	이동기	3h	1
	System 정보 수집 제작 2(server side)	이동기	3h	1
	System 정보 server DB 에 연결	이동기	1h	1
	System 정보 서버 전송 모듈 제작	이동기	3h	1
	DB status query 정리	이동기	3h	1
	DB status 서버 전송 모듈 제작	이동기	4h	1

1주차

환경구축

System 정보 전송 모듈 제작

Alpha 목표 : 각 주차 별 구현 내용

2 주차 1/8 ~ 1/12	Status 용 logstash filter 제작	이동기	4h	1
	DB Monitoring Query 정리	이동기	5h	1
	DB Monitoring Query 테스트	이동기	1h	1
	Query 결과값 전송용 client module 제작	이동기	4h	1
	DB Monitoring 용 logstash filter 제작	이동기	4h	1
	예시 데이터 – Kibana View 제작	이동기	4h	1
	DB status filter 확장	이동기	5h	1

2주차

DB 정보 전송 모듈 제작

Alpha 목표 : 각 주차 별 구현 내용

주차	항목	담당자	예상 개발 시간	개발 우선순위
3 주차 1/15 ~ 1/19	일부 데이터 Kibana 연동	이동기	4h	1
	문제 판단 모듈 설계	이동기	4h	1
	문제 판단 모듈 제작	이동기	3h	1
	실제 데이터 Kibana View 설계(DB)	이동기	5h	1
	문제 판단 모듈 데이터를 Kibana 에 연결	이동기	5h	1

3주차

Error 처리 모듈 제작

Alpha 목표 : 각 주차 별 구현 내용

4 주차 1/22 ~ 1/26	System 정보 예측 모듈 설계	이동기	3h	2
	System 정보 예측 모듈 제작	이동기	3h	2
	System 예측 정보 View 에 연결	이동기	3h	2
	실제 데이터 Kibana View 설계(System)	이동기	5h	1
	System 항목들 Data 확인 및 수정	이동기	5h	1

4주차

자원 예측 정보 모듈 완료

Alpha 목표 : 각 주차 별 구현 내용

5 주차 1/29 ~ 2/2	설계에 따른 Kibana 연동(System side)	이동기	4h	1
	설계에 따른 Kibana 연동(DB side)	이동기	4h	1
	메일 전송용 정보 설계	이동기	4h	2
	메일 전송 모듈 제작(SNMP 사용)	이동기	4h	2
	메일 전송 모듈 제작(전송, 내용부분)	이동기	3h	2

5주차

View 완성

Alarm 전송 모듈 제작

Alpha 목표 : 각 주차 별 구현 내용

6 주차 2/5 ~ 2/9	데이터 연동 확대(각 작업 별 세부분류)	이동기	15h	1
	데이터 추가 항목 Kibana 연동	이동기	5h	1
	전 주차 미 구현 부분 마무리	이동기	5h	1

6주차

디버깅, 최적화

시연

이후 목표

1. 일부 항목을 전체 항목으로 확장
2. 사용량 예측 알고리즘을 개선
3. 프로그램 안정성 향상 및 최적화
4. OS, DBMS 호환성 확장
5. 문제상황 자동 예측 모듈 제작

Q&A