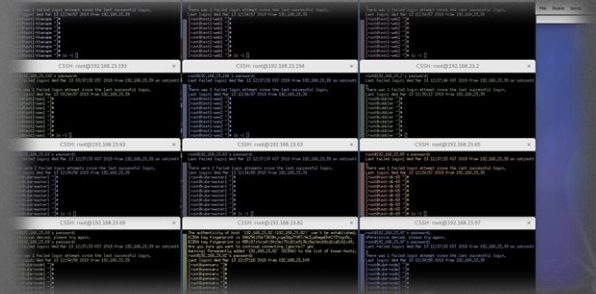


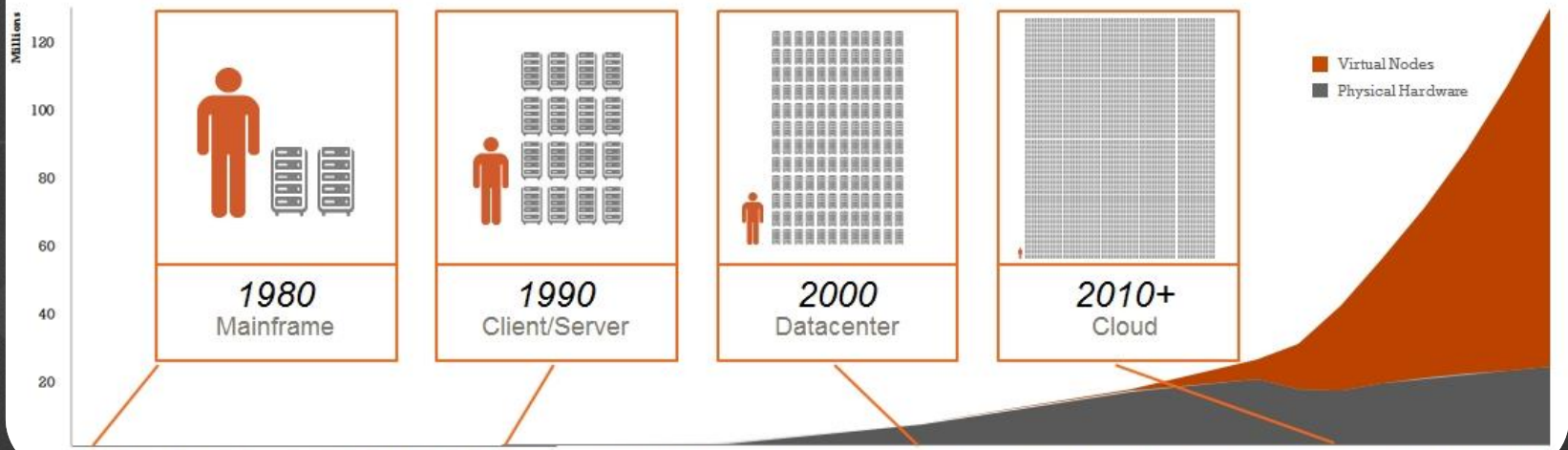
컨테이너 환경에서 모니터링 이슈와 해결 방안

Immutable Infrastructure인 컨테이너 환경에서의 모니터링 이슈 설명
Prometheus, Grafana, KHAN [apm]

- 직접 작업
- Script
- SSH Client Tool 활용



Scale x Complexity > Skills



직접 인프라 관리의 문제점

- 휴먼 에러
 - 오타, 누락
- 변경 이력 관리 : 누가, 언제, 무엇을?

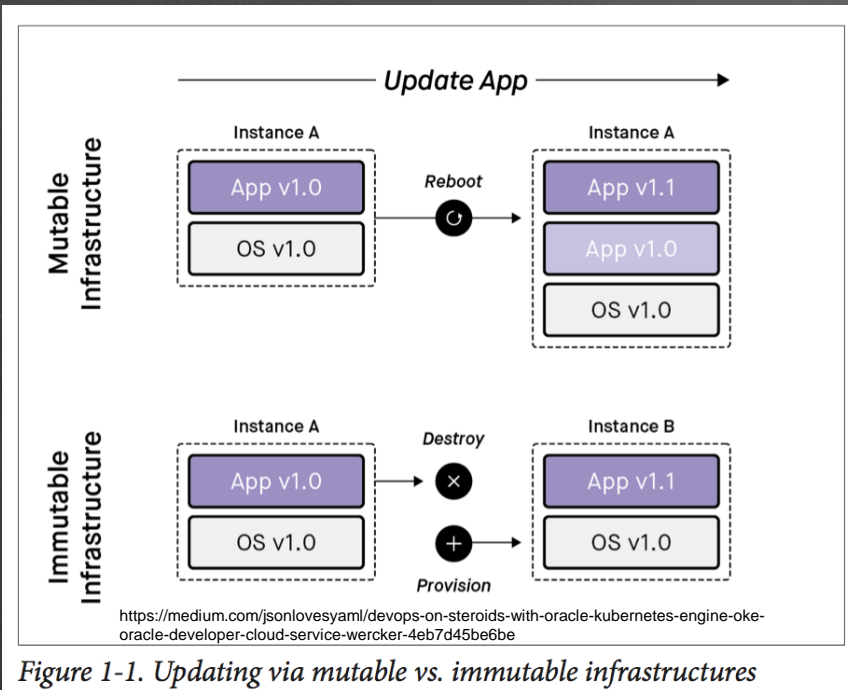


Figure 1-1. Updating via mutable vs. immutable infrastructures



opennaru

Application Performance Management

Orchestration

KHAN
[a p m]



인프라 관리 자동화



- 제품

- Ansible
- Chef
- Puppet



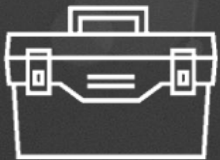
RED HAT[®]
ANSIBLE[®]
Automation



- 안전성 향상
 - 휴먼 에러 방지
 - 작업자에게 의존하지 않음 (인력 의존성 탈피)
 - 변경 이력 관리 : 누가, 언제, 무엇을?
 - 작업계획과 운영 환경의 차이 감소

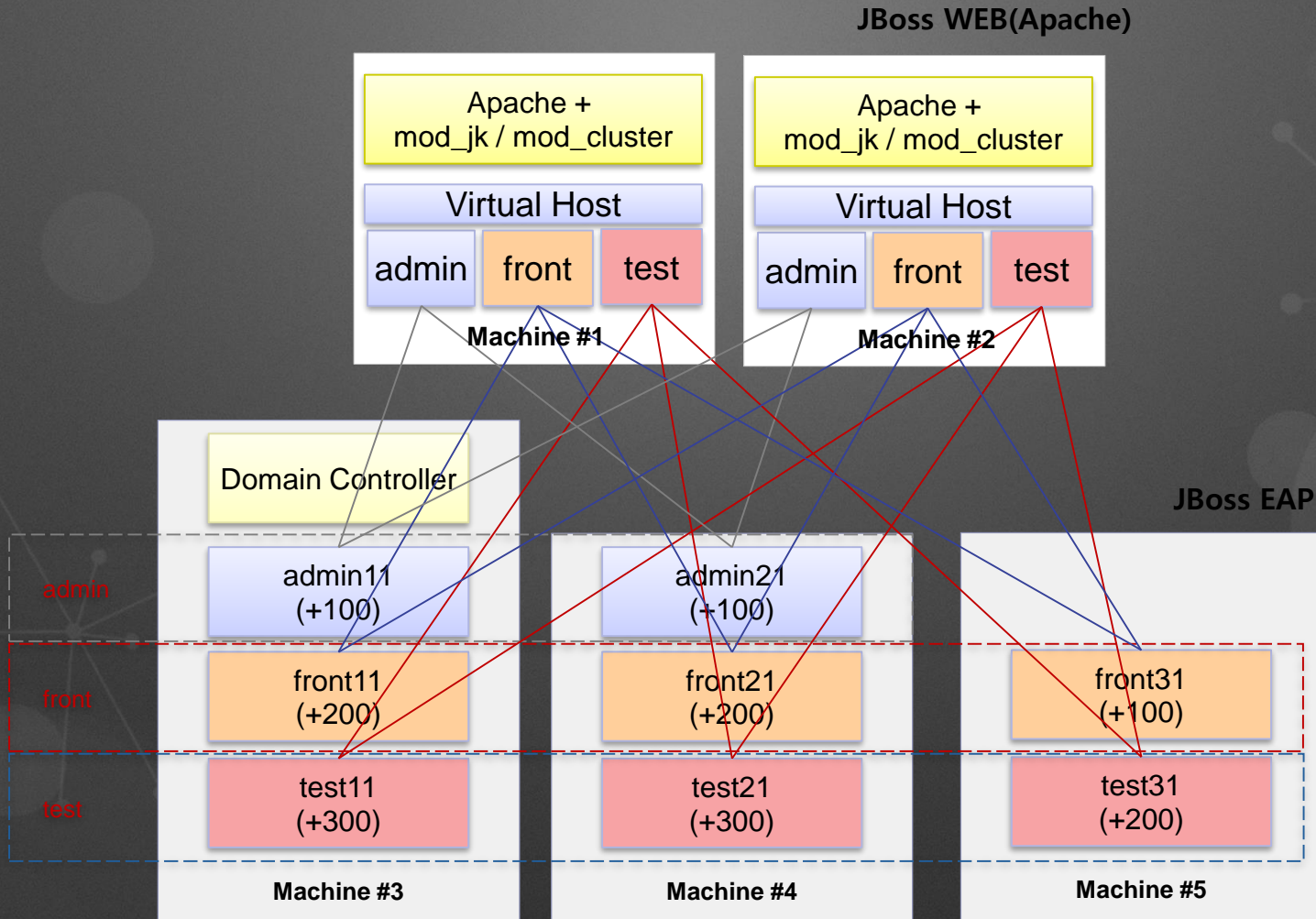


- 작업 효율 향상
 - 대상 서버 수와 상관없이 구축할 수 있으며, 병렬 실행
 - 장시간 작업이나 야간 작업에 대한 인력의존성 탈피
 - 신속한 릴리스 작업



- 다른 툴과 통합하여 자동화와 효율성 향상
 - 버전 관리툴(git, svn...)에 의한 순서/설정 관리
 - 자동 테스트 툴에 의한 환경 테스트(serverspec등)
 - 각종 CI툴과의 자동 연계 (jenkins등)
 - 모니터링 도구와 연계된 장애 대응 자동화(zabbix, nagios등)
 - Slack등과 연계해 채팅 베이스에서의 운용 작업 실행

실제 Web / WAS 구성 - 다양한 서비스



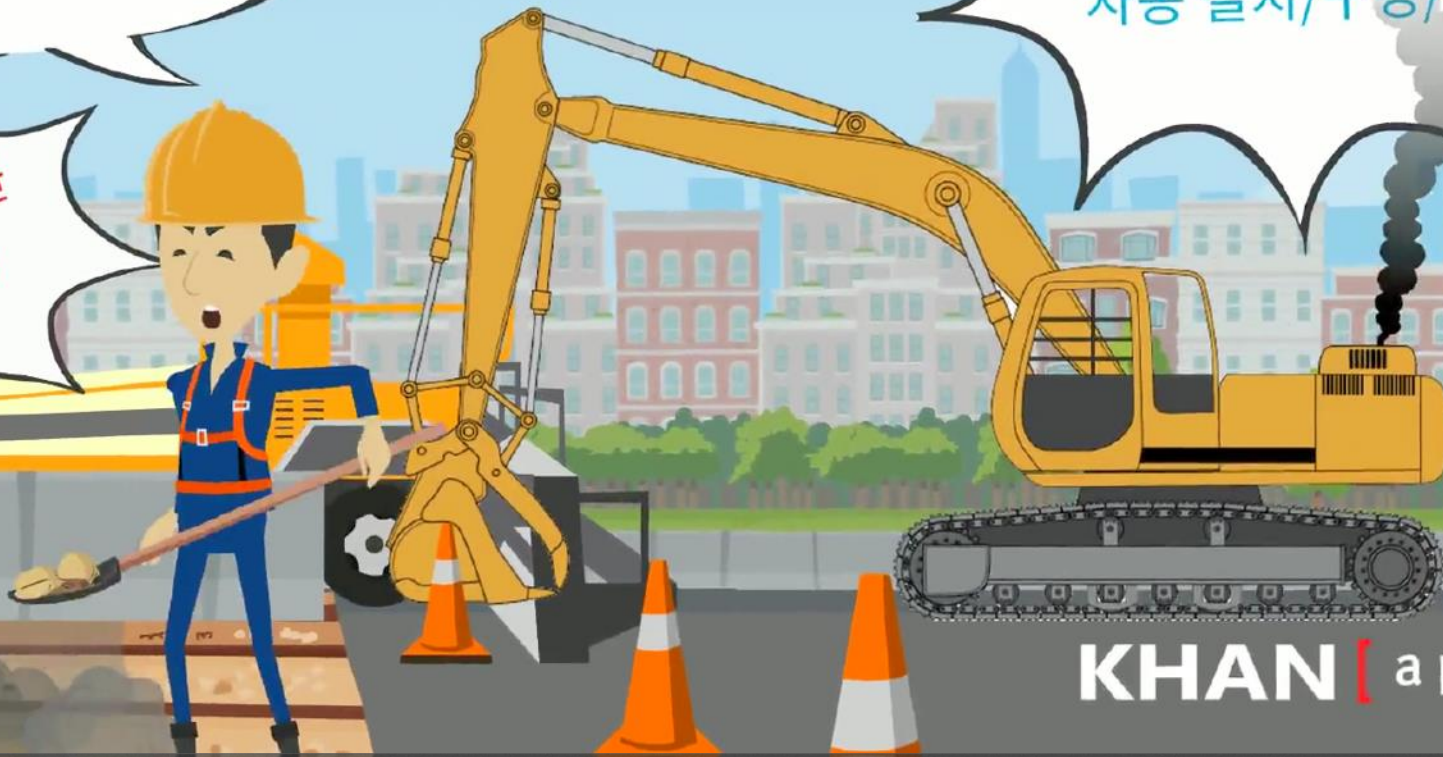
KHAN [APM] - 미들웨어 설치/구성/튜닝 자동화(Ansible)



자꾸 바꾸라고 그러지.... 째
하루 종일 삽질만 하네....

수작업에 의한
설치/구성/튜닝

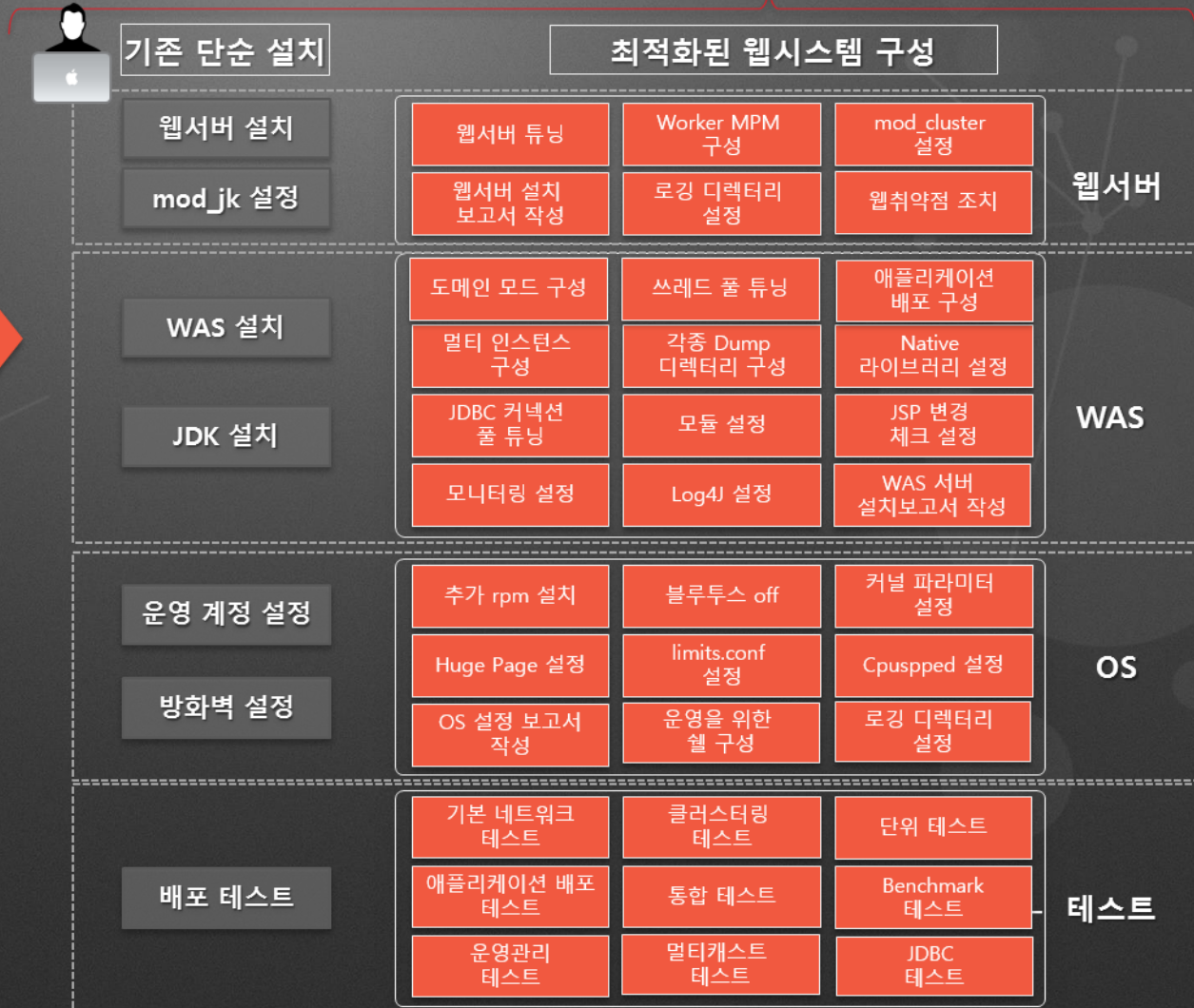
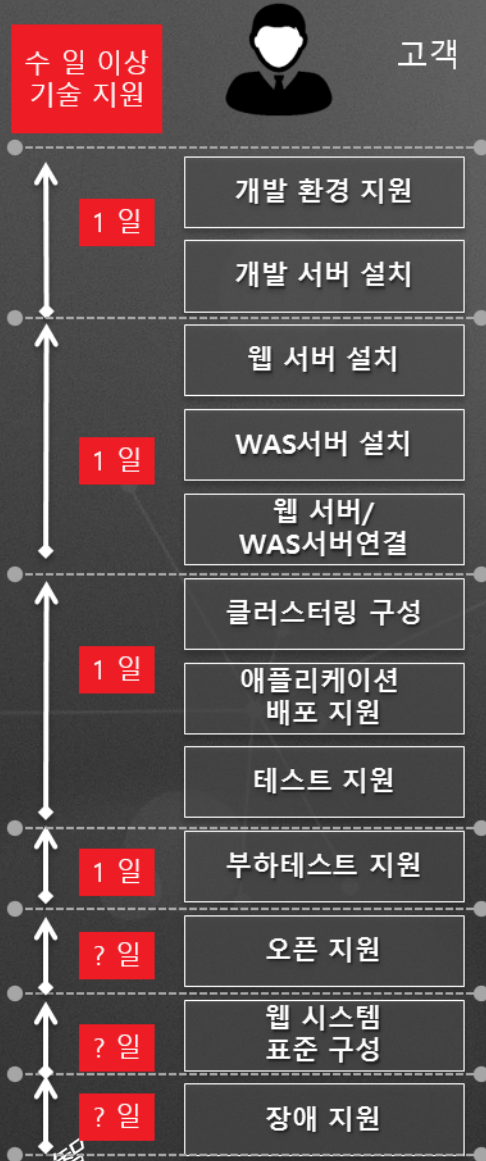
KHAN [apm]에 의한
자동 설치/구성/튜닝



KHAN [a p m]

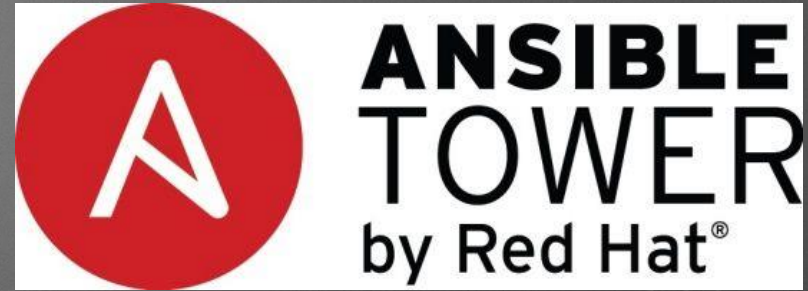
고객 요구사항에 따른 웹 시스템 작업

KHAN
[a p m]



인프라 관리의 자동화는 했는데...

- 휴먼 에러 – 그래도 사람이 하는 일
 - 작업 누락
- 변경 이력 관리에 대한 또 관리
 - 작업 내역 관리
 - Ansible Log 등



GUI, 스케줄링, 히스토리 관리, 인증환경

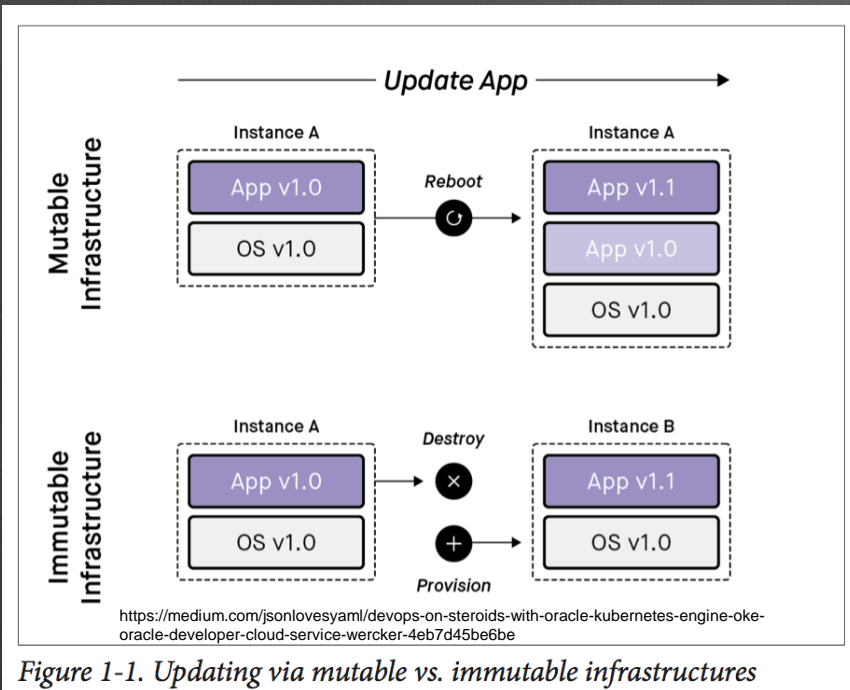


Figure 1-1. Updating via mutable vs. immutable infrastructures



Application Performance Management



Container

Docker by Google Trends

● Docker
검색어

● Virtual Machine
검색어

● Open Stack
검색어

+ 비교 추가

전 세계 ▼ 지난 5년 ▼ 모든 카테고리 ▼ 웹 검색 ▼

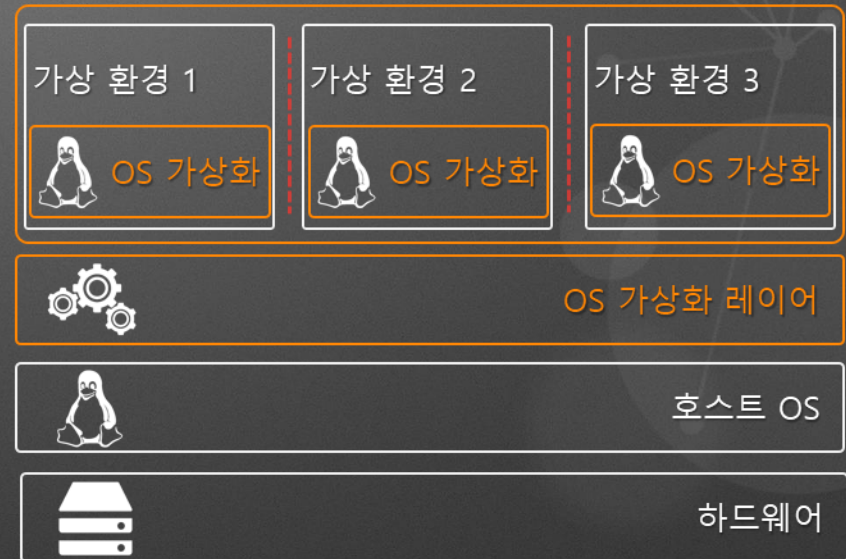
시간 흐름에 따른 관심도 변화 ?



하드웨어 가상화와 OS 가상화 비교

- 서버용 가상화 기술로 VMWare, RHV 나 KVM등의 가상머신모니터(VMM)/하이퍼바이저에 의한 "하드웨어 가상화"
- 하드웨어와 리소스를 가상머신에 할당하기 위해서는 하드웨어 전체가 가상화되어야 함

- 가상화라기보다는 격리된 가상적인 OS환경을 제공



하이퍼바이저에 의한 하드웨어 가상화
 호스트 OS와 게스트OS가 다를 경우도 가능
 디스크/메모리 소비 큼
 ↓
 구성 자유도가 높은 가상화 기술

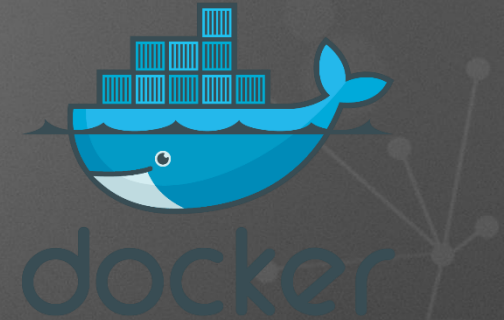
OS 가상화
 호스트OS와 게스트OS 동일
 디스크/메모리 소비 작음
 ↓
 가볍고 휴대 성이 높은 가상화 기술

Immutable Infrastructure

KHAN
[a p m]

Mutable vs Immutable

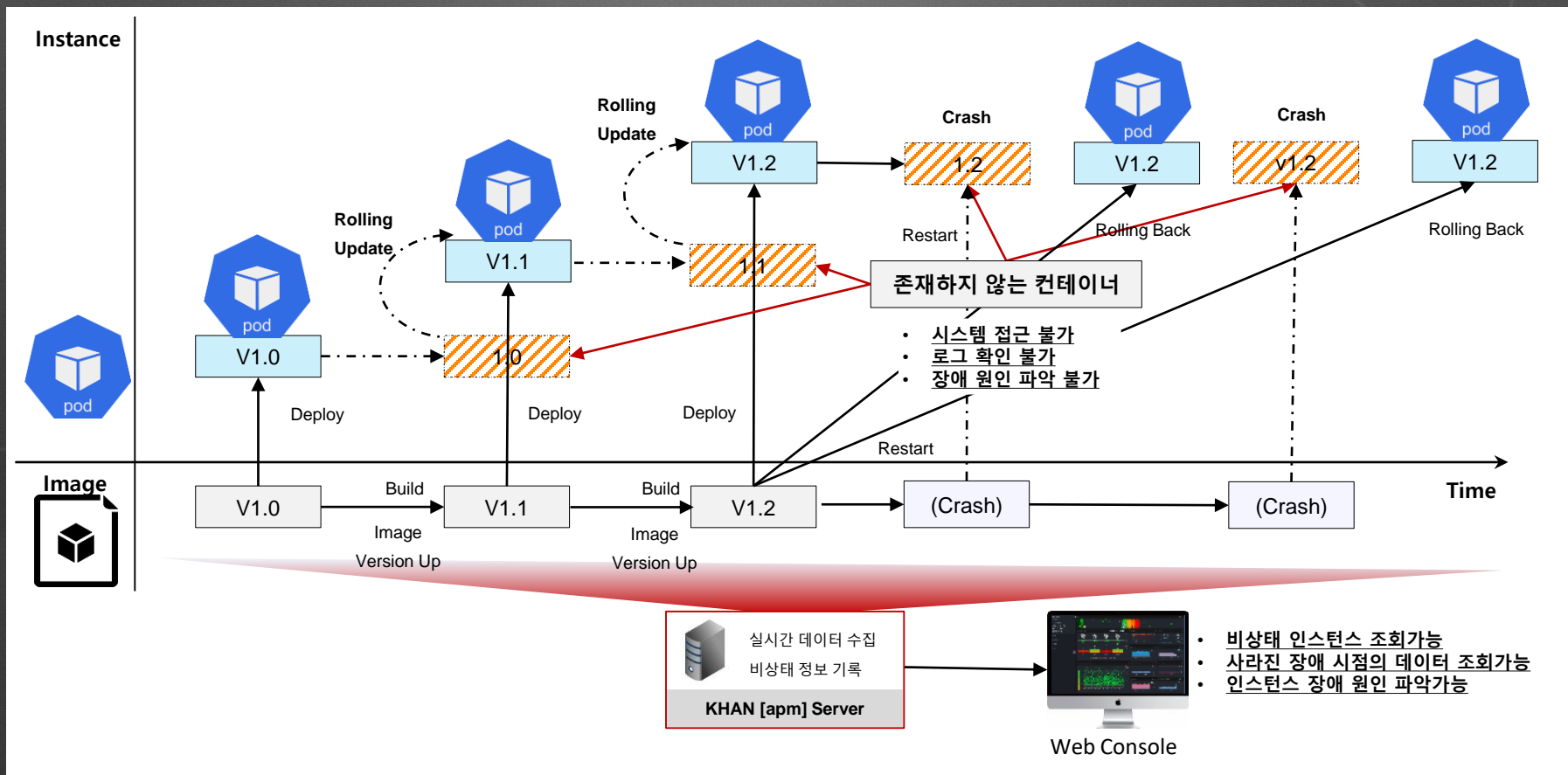
- Immutable 은 기존에 서버를 지속적으로 '관리'한다는 데서 벗어나 어떻게 하면 서버를 잘 쓰고 버리는 지를 다룹니다.
- Mutable (Snowflake Server)
 - 서버를 계속 유지하며 설치/업데이트
 - 모든 서버의 상태를 똑같이 유지하기 어려움
 - 수동 및 자동화 도구
 - 문서화, 담당자 변경
- Immutable (Phoenix Server)
 - 폐기 후 새롭게 셋팅
 - Code와 Data를 분리
 - Was는 Stateless
 - Log 및 Upload data 등은 별도로 관리 (NFS, Gluster)

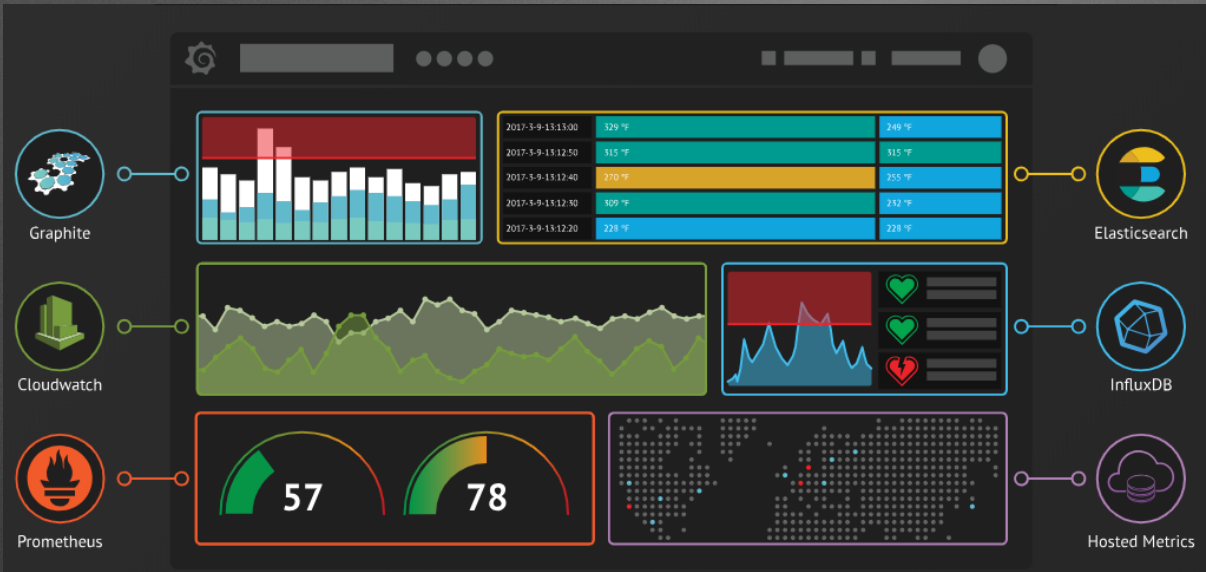


서버는 주기적으로 태워 버려라
- 마틴 파울러 -

Immutable

- Openshift (Kubernetes) 배포 프로세스





Grafana

오픈소스 모니터링

KHAN [a p m]

- SoundCloud 에서 제작 된 오픈 소스 시스템 모니터링 및 경고 툴 키트
- Prometheus는 Kubernetes 이후 2016 년에 두 번째 호스팅 프로젝트로 Cloud Native Computing Foundation 에 가입
- 특징
 - Key/Value 기반의 시계열 데이터
 - 유연한 Query PromQL
 - Pull 방식, Push 방식도 제공
 - 그래프 제공 및 대시보드 솔루션과 연계
- 구성요소
 - Exporters (100개 이상)
 - Push Gateway
 - Client Libraries
 - Alert Manager
 - Etc...



- Visualize

- Dashboard
- Panel

- Plugin (Apps)





- Kubernetes
- Zabbix
- etc...

- Datasource

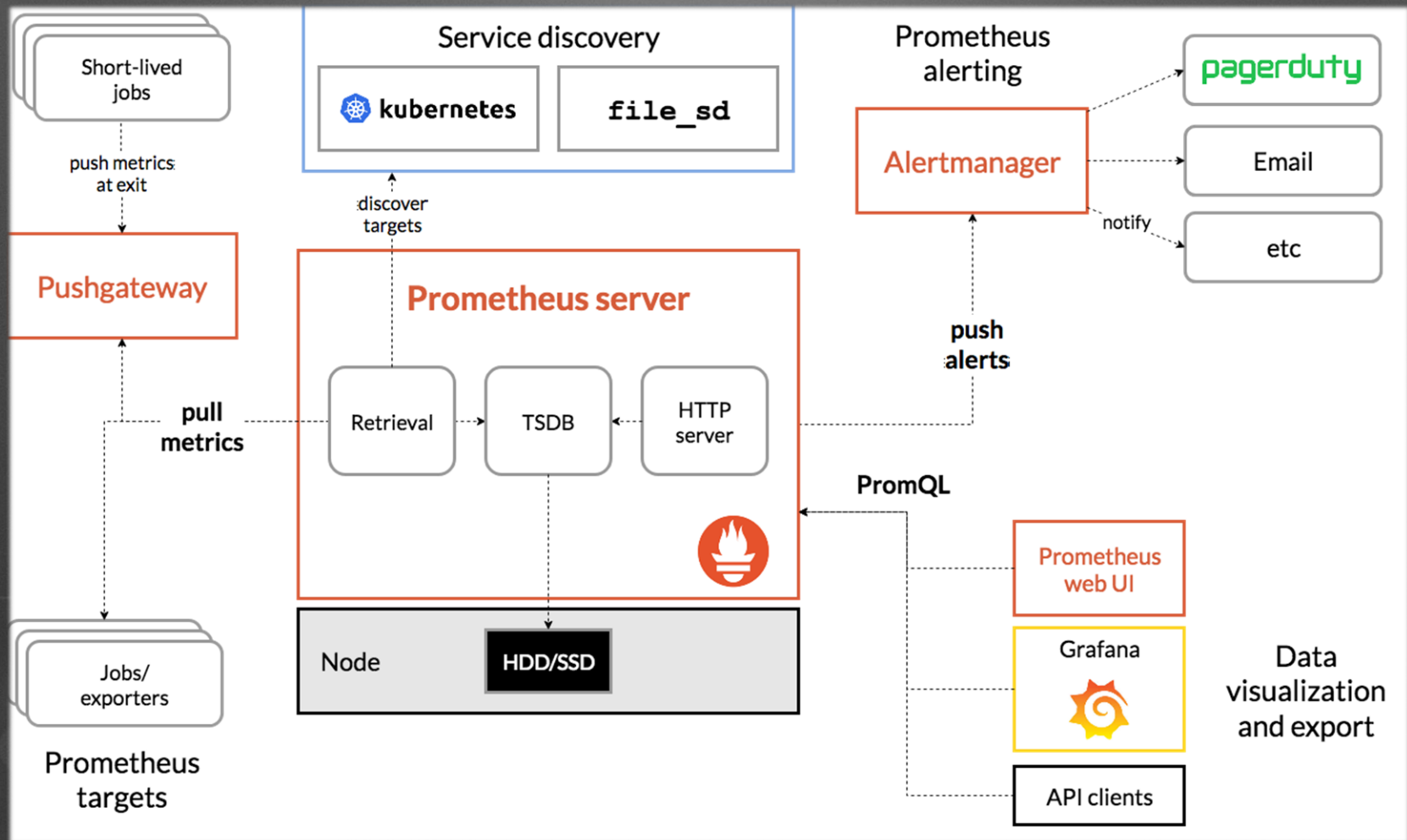
- Prometheus
- Influxdb
- Elasticsearch
- AWS CloudWatch
- etc...

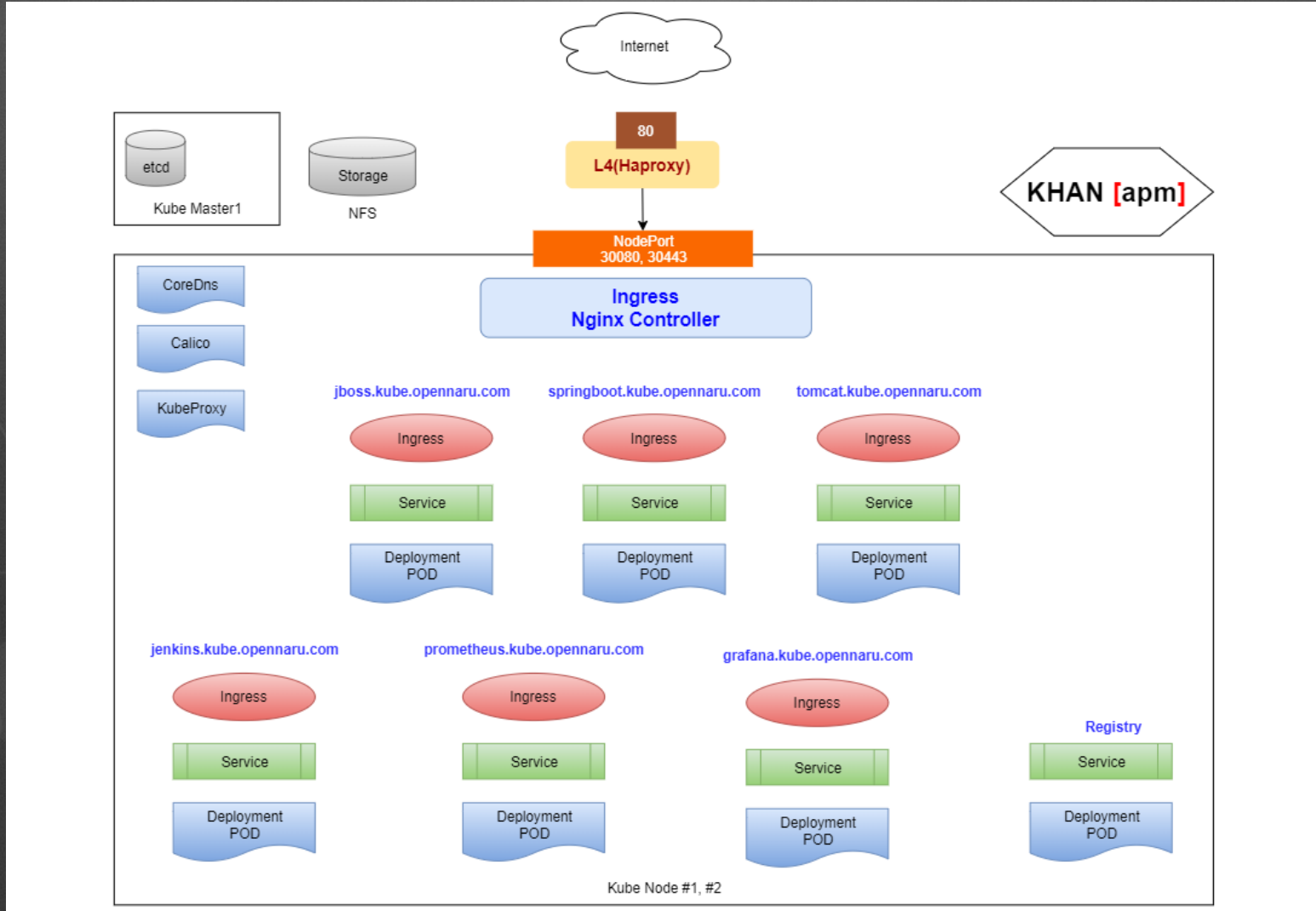
- Alert



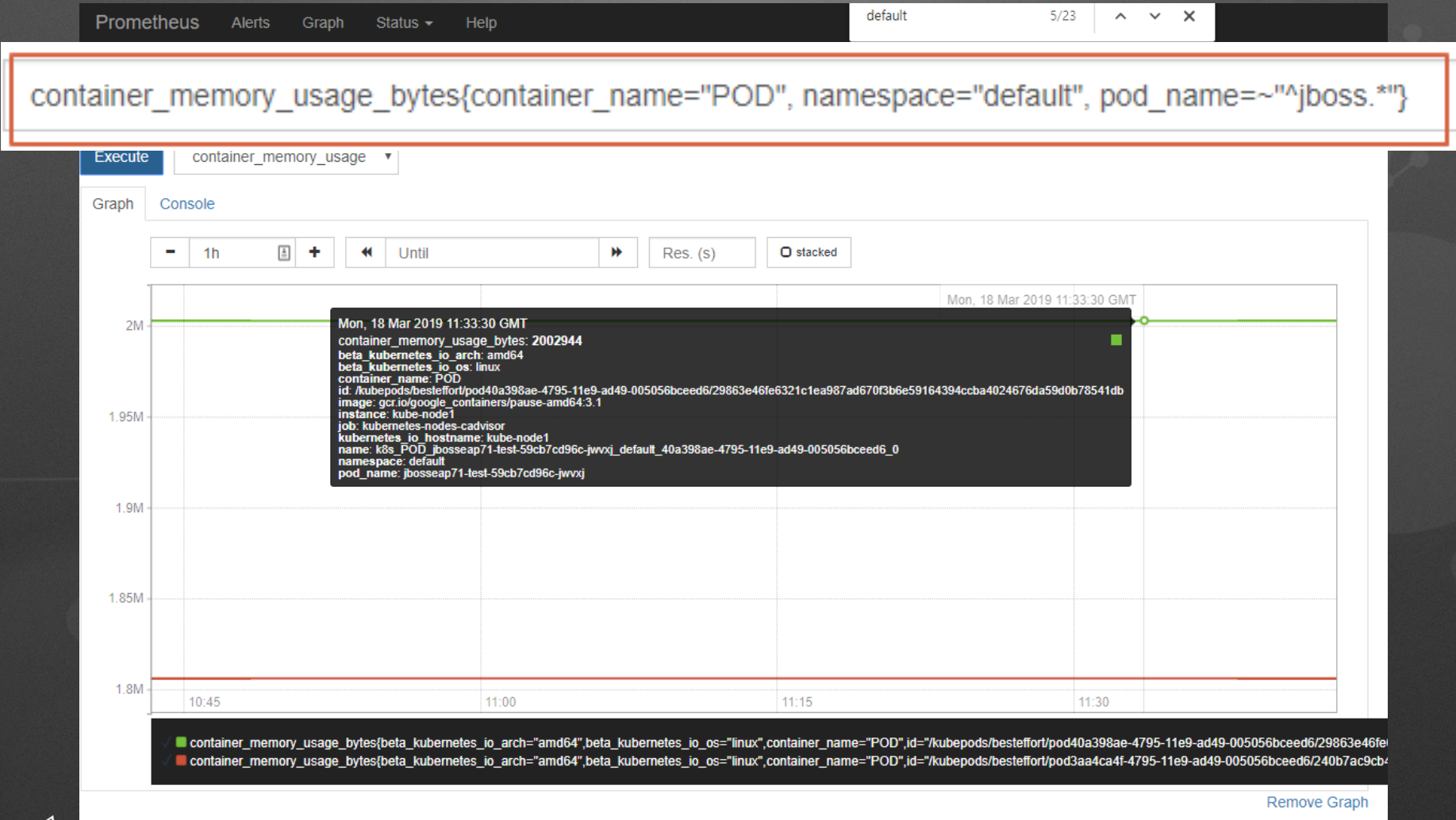
As of © right now, there are  55 data sources,  48 panels,  17 apps and  1646 dashboards available.

Prometheus + Grapana Architecture





- Query PromQL



The screenshot shows the Prometheus web interface. At the top, there is a navigation bar with 'Prometheus', 'Alerts', 'Graph', 'Status', and 'Help'. The current view is 'Graph', and the selected metric is 'container_memory_usage'. A red-bordered box highlights the PromQL query: `container_memory_usage_bytes{container_name="POD", namespace="default", pod_name=~"^jboss.*"}`. Below the query, there is a 'Console' tab and a graph. The graph shows memory usage over time, with a tooltip for a specific data point at 'Mon, 18 Mar 2019 11:33:30 GMT'. The tooltip contains the following details:

```
Mon, 18 Mar 2019 11:33:30 GMT
container_memory_usage_bytes: 2002944
beta_kubernetes_io_arch: amd64
beta_kubernetes_io_os: linux
container_name: POD
id: /kubepods/besteffort/pod40a398ae-4795-11e9-ad49-005056bceed6/29863e46fe6321c1ea987ad670f3b6e59164394ccba4024676da59d0b78541db
image: gcr.io/google_containers/pause-amd64:3.1
instance: kube-node1
job: kubernetes-nodes-cadvisor
kubernetes_io_hostname: kube-node1
name: k8s_POD_jbosseap71-test-59cb7cd96c-jvwxj_default_40a398ae-4795-11e9-ad49-005056bceed6_0
namespace: default
pod_name: jbosseap71-test-59cb7cd96c-jvwxj
```

At the bottom of the graph, there is a legend with two entries:

- container_memory_usage_bytes[beta_kubernetes_io_arch="amd64",beta_kubernetes_io_os="linux",container_name="POD",id="/kubepods/besteffort/pod40a398ae-4795-11e9-ad49-005056bceed6/29863e46fe6321c1ea987ad670f3b6e59164394ccba4024676da59d0b78541db"]
- container_memory_usage_bytes[beta_kubernetes_io_arch="amd64",beta_kubernetes_io_os="linux",container_name="POD",id="/kubepods/besteffort/pod3aa4ca4f-4795-11e9-ad49-005056bceed6/240b7ac9cb4..."]

A 'Remove Graph' button is located at the bottom right of the interface.

Prometheus Target



- Targets

Prometheus Alerts Graph Status ▾ Help

Targets

All Unhealthy

kubernetes-apiservers (1/1 up) [show less](#)

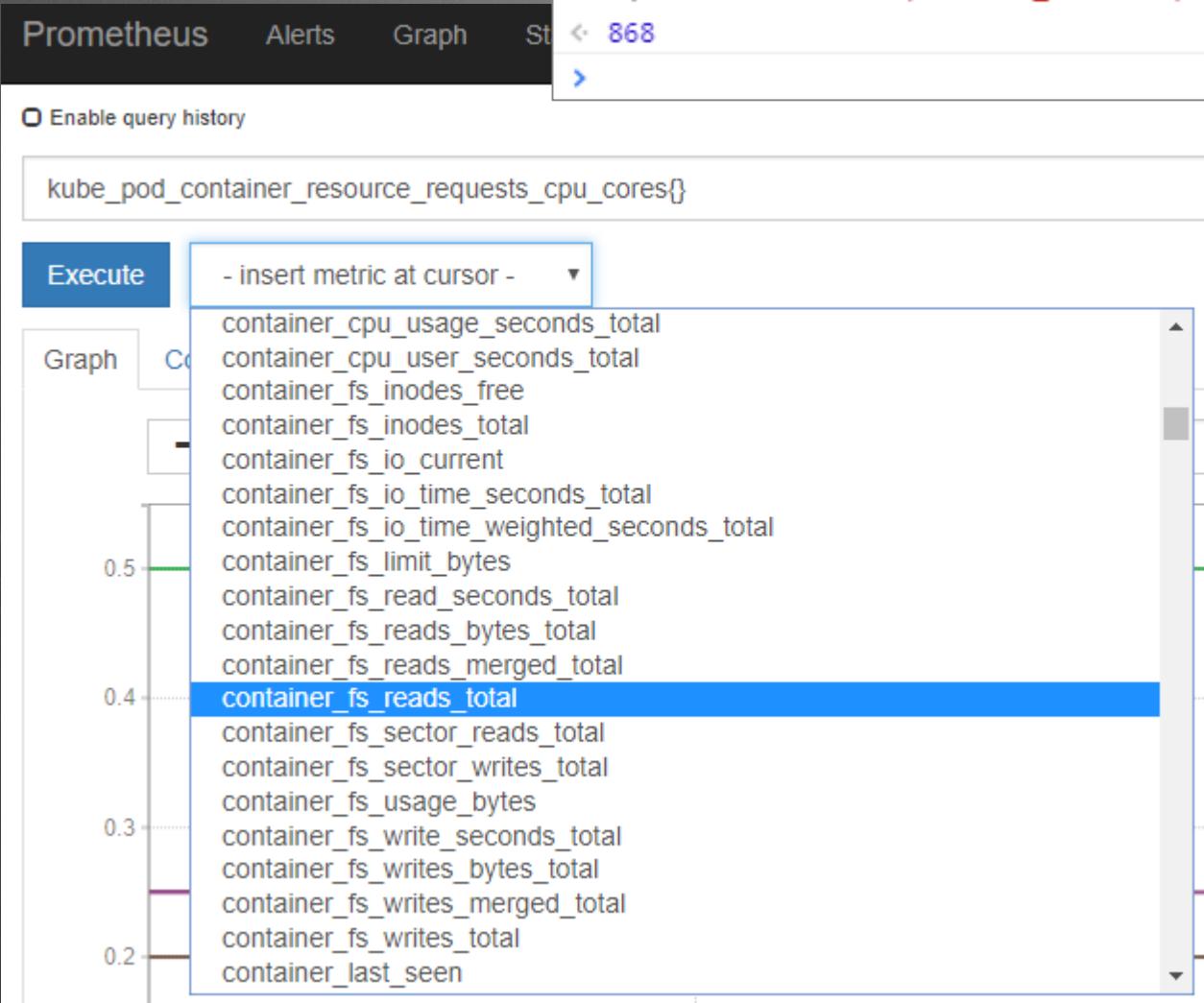
Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
https://192.168.23.63:6443/metrics	UP	instance="192.168.23.63:6443" job="kubernetes-apiservers"	23.196s ago	94.64ms	

kubernetes-nodes (3/3 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
https://kubernetes.default.svc:443/api/v1/nodes/kube-master1/proxy/metrics	UP	beta_kubernetes_io_arch="amd64" beta_kubernetes_io_os="linux" instance="kube-master1" job="kubernetes-nodes"	19.777s ago	17.27ms	

```
process_resident_memory_bytes 3.268608e+07
# TYPE process_resident_memory_bytes gauge
process_start_time_seconds 1.55237959997e+09
# HELP process_start_time_seconds Start time of the process since unix epoch in seconds.
# TYPE process_start_time_seconds gauge
process_virtual_memory_bytes 1.40193792e+08
# HELP process_virtual_memory_bytes Virtual memory size in bytes.
# TYPE process_virtual_memory_bytes gauge
process_virtual_memory_max_bytes -1
# HELP process_virtual_memory_max_bytes Maximum amount of virtual memory available in bytes.
# TYPE process_virtual_memory_max_bytes gauge
promhttp_metric_handler_requests_in_flight 1
# HELP promhttp_metric_handler_requests_in_flight Current number of scrapes being served.
# TYPE promhttp_metric_handler_requests_in_flight gauge
promhttp_metric_handler_requests_total{code="200"} 14588
# HELP promhttp_metric_handler_requests_total Total number of scrapes by HTTP status code.
# TYPE promhttp_metric_handler_requests_total counter
promhttp_metric_handler_requests_total{code="500"} 0
promhttp_metric_handler_requests_total{code="503"} 0
```

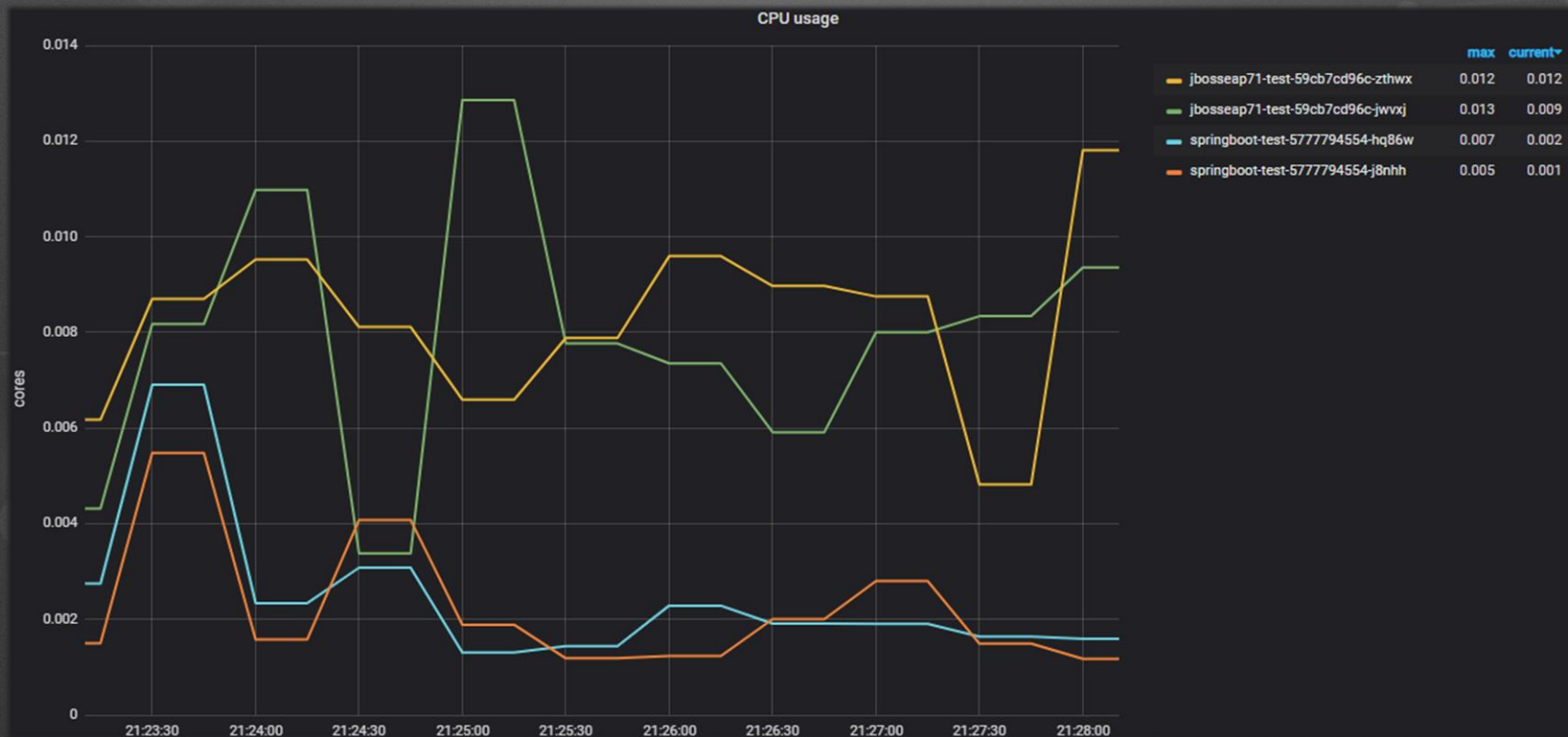
- Metrics



The screenshot shows the Prometheus web interface. At the top, there are tabs for 'Prometheus', 'Alerts', 'Graph', and 'Status'. Below the tabs, there is a search bar containing the query `kube_pod_container_resource_requests_cpu_cores{}`. To the right of the search bar, a status indicator shows '< 868'. A dropdown menu is open, displaying a list of metrics. The metric `container_fs_reads_total` is highlighted in blue. The dropdown menu also includes an 'Execute' button and a '- insert metric at cursor -' option. A code snippet is visible in the top right corner of the image: `> $('form-control.expression_select option').length`.

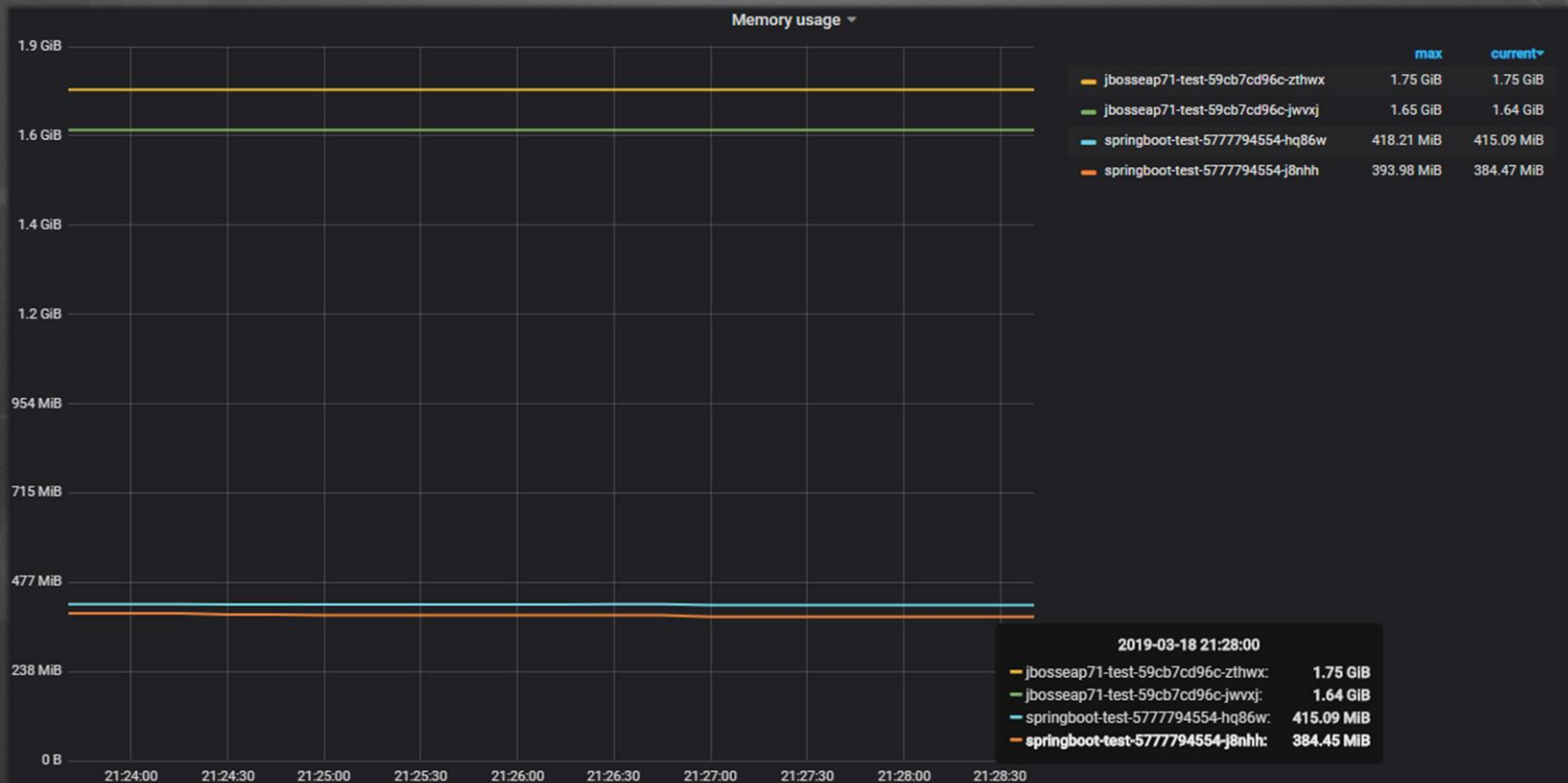
- Container CPU

```
sum (rate (container_cpu_usage_seconds_total{image!="",name=~"^k8s_.*",  
io_kubernetes_container_name!="POD",namespace=~"^$Namespace$",  
pod_name=~"^$Deployment$Statefulset$Daemonset.*$",  
kubernetes_io_hostname=~"^$Node$"}[1m])) by (pod_name,kubernetes_io_hostname)
```

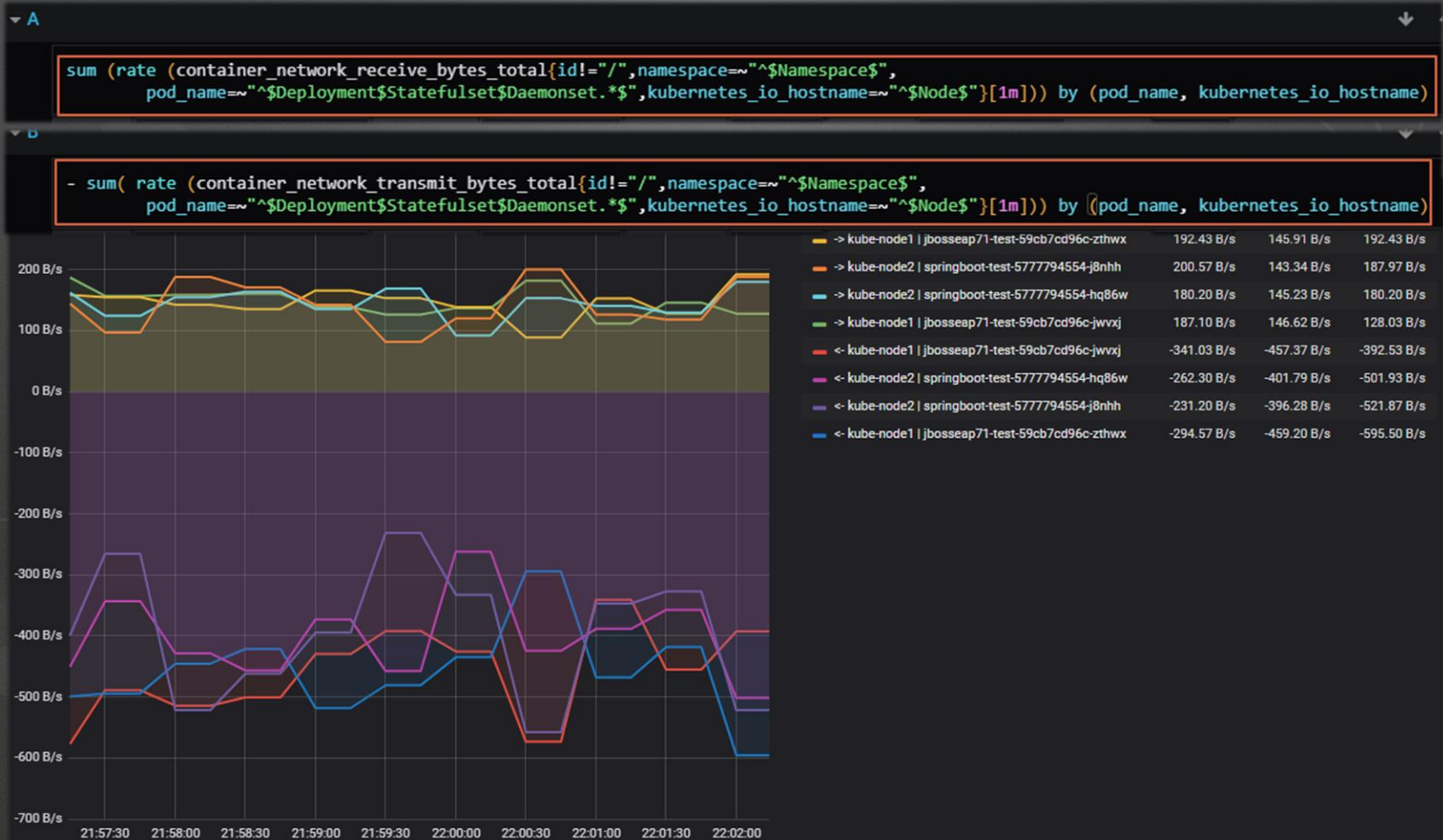


- Container Memory

```
sum (container_memory_working_set_bytes{id!="/",namespace=~"^$Namespace$",  
  pod_name=~"^$Deployment$Statefulset$Daemonset.*$",  
  kubernetes_io_hostname=~"^$Node$"}) by (pod_name,kubernetes_io_hostname)
```

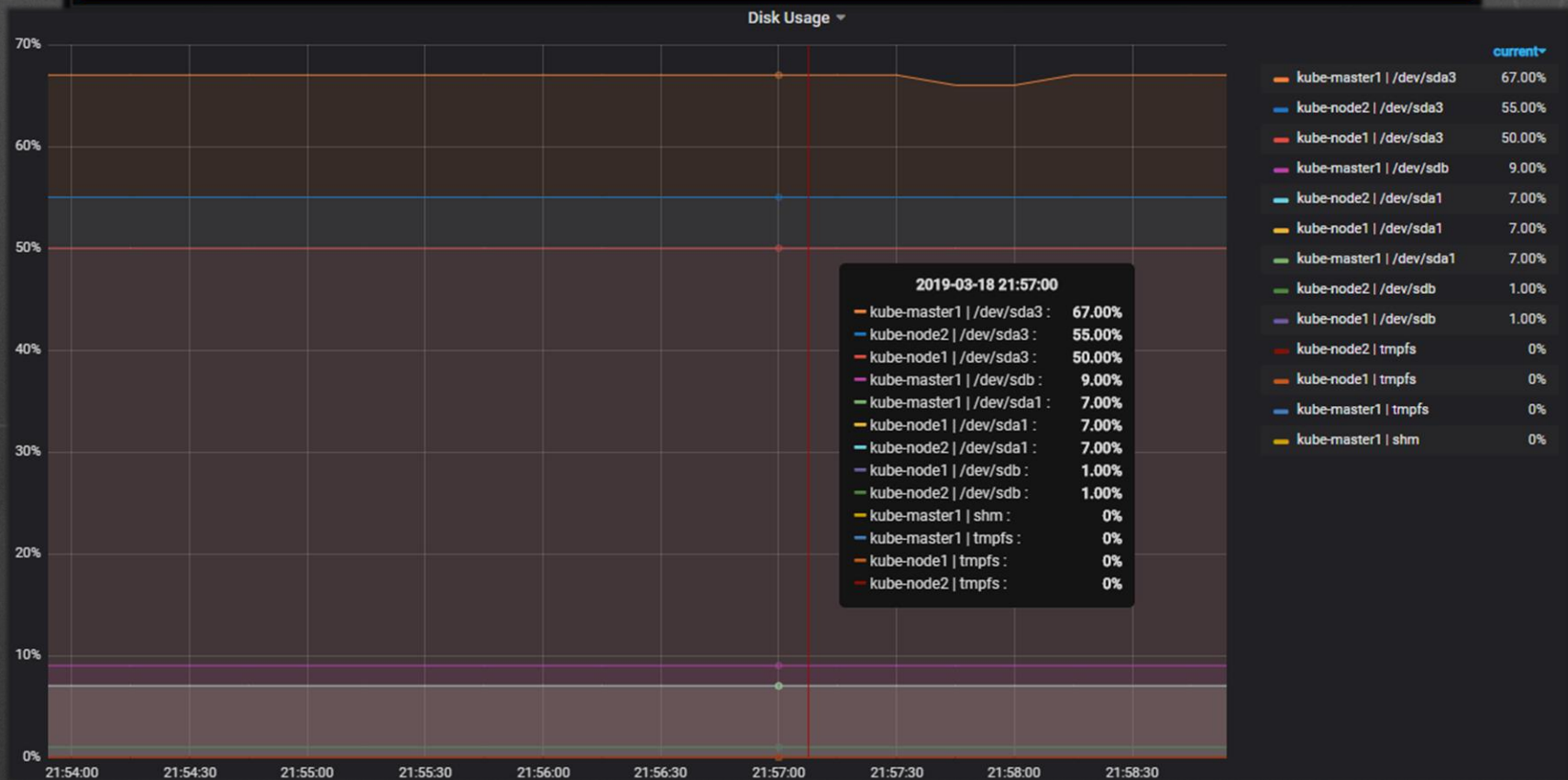


- Container processes Network I/O

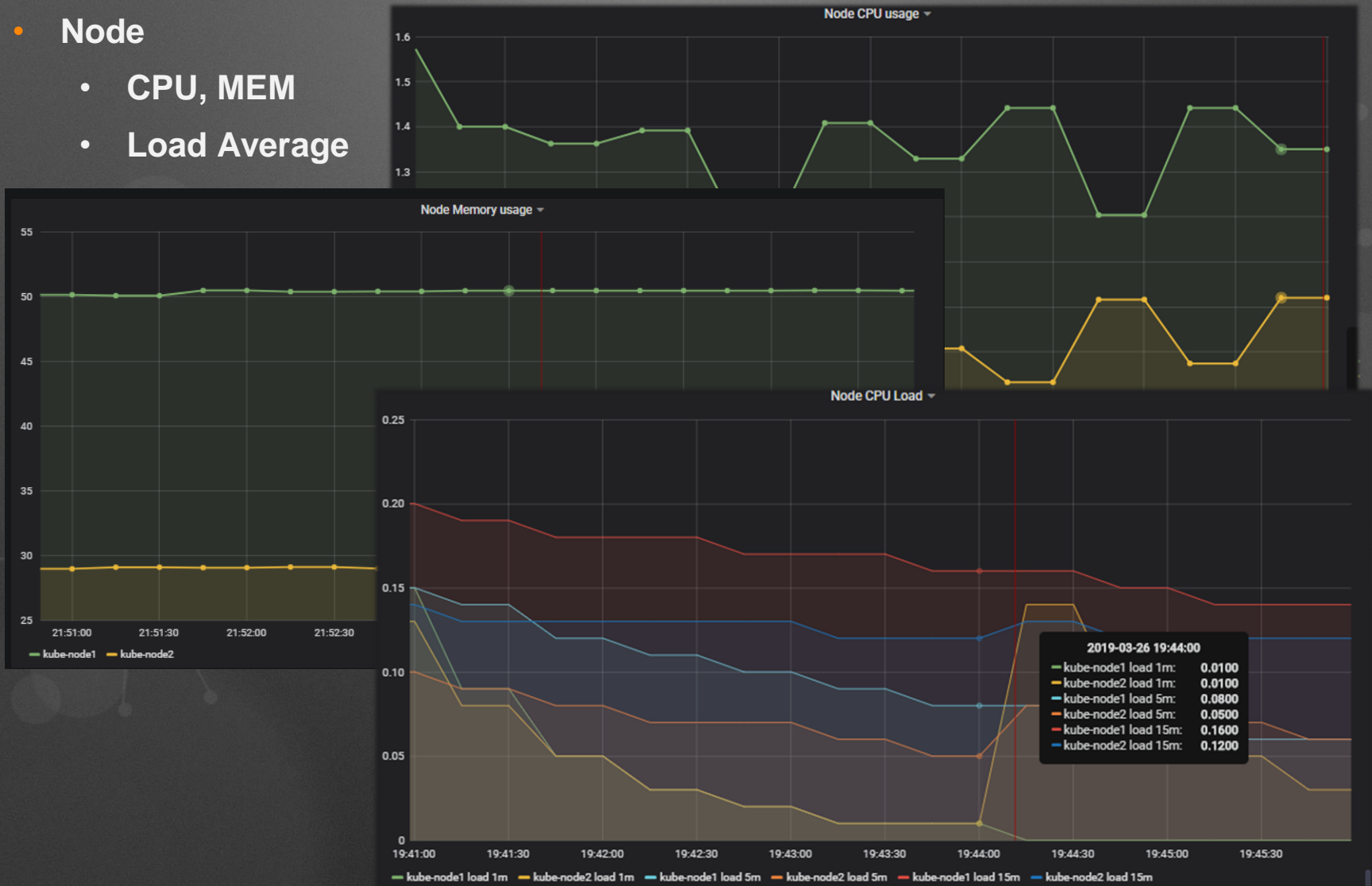


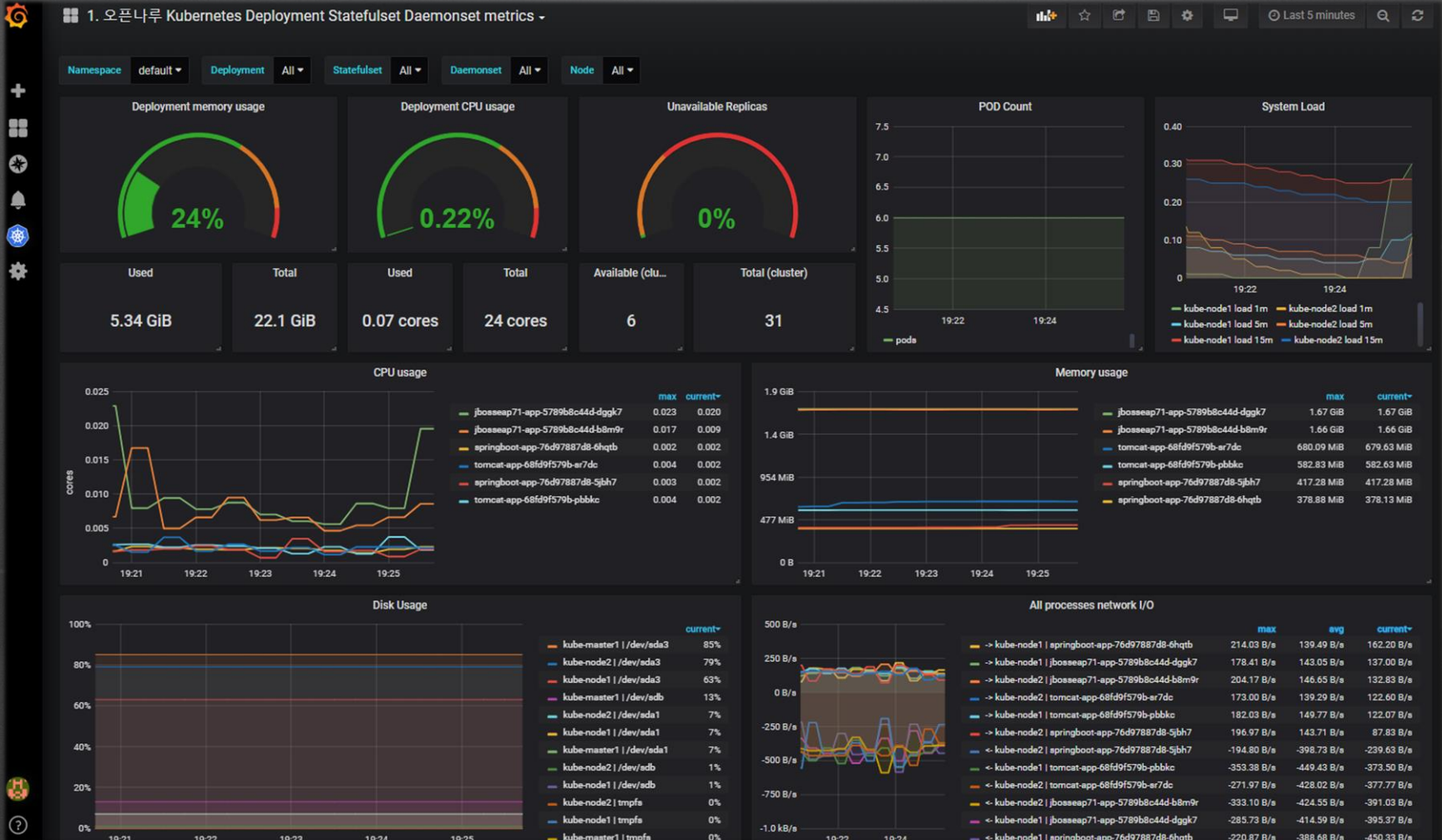
- Node DISK

```
sum(ceil(container_fs_usage_bytes{container_name=""}/
container_fs_limit_bytes{container_name=""}*100)) by (instance, device)
```



- Node
 - CPU, MEM
 - Load Average





하지만

- OS 및 OCP, K8S 관점의 모니터링
- 애플리케이션 관점의 모니터링 항목 부재
 - 응답 분포도
 - 지연 트랜잭션
 - Stack Trace
 - 느린 Query
- Metric 내용을 학습
- 커스텀 대시보드 작성
 - Prometheus Query(PromQL) 학습
- 빠른 데이터 추적의 어려움



HTML5 User Interface

KHAN [a p m]

Responsive Web
Cross Browsing
Multi



KHAN [monitoring]

KHAN
[a p m]

KHAN [apm] – 분포도 및 Stack Trace 분석

트랜잭션 맵(T-Map)

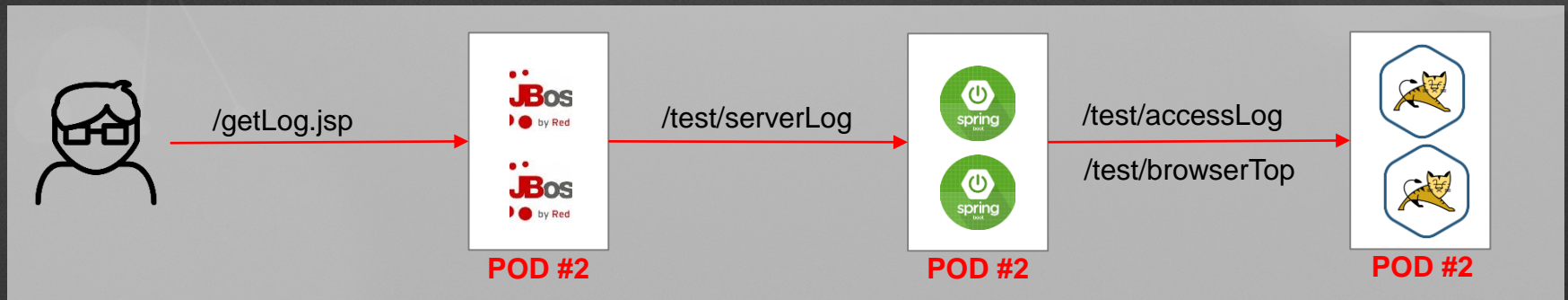
```
0.0] + org.mariadb.jdbc.MySQLConnection.prepareStatement()
0.0] + org.jboss.resource.adapter.jdbc.WrappedPreparedStatement
0.0] + org.mariadb.jdbc.MySQLPreparedStatement.execute()
> DB : URL: jdbc:mysql://192.168.10.89:3303/kisa_ehrd3
> Query : /* finish.updateStudentScore */ UPDATE TB_LMS_STUDENT
STUDYTIME=?, ISGRADEABLE=?, LDATE=F_DATE() WHERE SUBJ= ? AND YEAR=? AND S
> Params: 0.0, 15.3846, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
> Mapped: /* finish.updateStudentScore */ UPDATE TB_LMS_STUDENT
0.0, AVREPORT=0.0, AVACT= 0.0, STUDYTIME=63, ISGRADEABLE='N', LDATE=F_D
> SQLERR: Query timed out
> CallAt:
```

대시보드





마이크로 서비스 추적



KHAN [apm] - 분석 기능



스레드 덤프 분석기

#	Lin.	Type	Name	State	URL	Durati...	CPU Ti...	Class	At
1	696	Active Thread	ajp-0.0.0.0-8009...	RUNNABLE			1,309	0.0	java...
2	871	Active Thread	ajp-0.0.0.0-8009...	RUNNABLE			123,355	0.0	java...
3	936	Active Thread	ajp-0.0.0.0-8009...	RUNNABLE			156,825	10.0	java...
4	1,556	Active Thread	ajp-0.0.0.0-8009...	RUNNABLE			183,887	0.0	java...

Thread Dump Text > 2017-07-10 15:24:56 (616D1A83-27FC-4F29-81B3-C8D7F13848)

```

694 # java.lang.Thread.State: RUNNABLE
695 # java.net.SocketInputStream: socketRead(CharLive Method)
696 # org.apache.coyote.ajp.AjpProcessor: read(AjpProcessor:Java:1077)
697 # org.apache.coyote.ajp.AjpProcessor: read(AjpProcessor:Java:1110)
698 # org.apache.coyote.ajp.AjpProcessor: read(AjpProcessor:Java:1085)
699 # org.apache.coyote.ajp.AjpProcessor: read(Buffer:Java:1370)
700 # org.apache.coyote.Request: read(Buffer:Java:423)
701 # org.apache.catalina.connector.InputBuffer: read(Bytes:Java:337)
702 # org.apache.catalina.connector.InputBuffer: read(Bytes:Java:420)
703 # org.apache.catalina.connector.InputBuffer: read(Bytes:Java:303)
704 # org.apache.catalina.connector.CoyoteInputStream: read(CoyoteInputStream:Java:193)
705 # sun.nio.cs.StreamDecoder: read(StreamDecoder:Java:205)
706 # sun.nio.cs.StreamDecoder: read(StreamDecoder:Java:158)
707 # java.io.InputStream: read(InputStreamHeader:Java:167)
708 # java.io.InputStream: read(InputStream:Java:103)
709 # java.io.BufferedReader: read(BufferedReader:Java:209)
710 # java.io.BufferedReader: read(BufferedReader:Java:105)
711 # org.apache.commons.io.IOUtils: read(BufferedReader:Java:67)
712 # org.apache.commons.io.IOUtils: read(BufferedReader:Java:113)
713 # org.apache.commons.io.IOUtils: read(BufferedReader:Java:113)
714 # org.apache.commons.io.IOUtils: read(BufferedReader:Java:113)
715 # org.apache.commons.io.IOUtils: read(BufferedReader:Java:113)
716 # org.apache.commons.io.IOUtils: read(BufferedReader:Java:113)
717 # org.apache.commons.io.IOUtils: read(BufferedReader:Java:113)
718 # org.apache.commons.io.IOUtils: read(BufferedReader:Java:113)
719 # org.apache.commons.io.IOUtils: read(BufferedReader:Java:113)
    
```

Lock을 추적가능, URL 정보표시

JVM 메모리 객체 분석기

히스토그램 생성일시

Created: 2019-01-19 14:11:15 | UUID: 81BF1078-768D-46F1-B5CD-B9...

JVM 메모리 객체 분석 > 2019-01-19 14:11:15 (Classes: 8,235, Instances: 1,570,240, Bytes: 99,720,112)

#	Num	Classname	Bytes [%]	Bytes	Instances
1	1	char[]	35.25%	35.2 MB	258,536
2	2	byte[]	8.62%	8.6 MB	10,388
3	3	java.lang.String	6.11%	6.1 MB	253,859
4	4	java.util.HashMap\$Node	5.55%	5.5 MB	173,073
5	5	java.util.jar.JarFile\$JarFileEntry	5.06%	5.0 MB	52,595
6	6	java.lang.Object[]	3.86%	3.8 MB	41,720

Java 메모리를 점유한 객체 분석/비교

네트워크 상태 분석기

네트워크 상태 생성일시

Created: 2019-01-19 14:14:08 | UUID: 5749186D-7480-47D1-BF0F-2A1...

네트워크 상태 분석 > 2019-01-19 14:14:38 (2AC63786-36D8-4AF0-9BE3-08BF28AC643E)

#	Proto	Recv.	Send-Q	Local Address	Foreign Address	State	PID/Program name
1	tcp	0	0	127.0.0.1:9999	0.0.0.0:*	LISTEN	509/java
2	tcp	0	0	10.6.3.224:5455	0.0.0.0:*	LISTEN	509/java
3	tcp	0	0	10.6.3.224:7600	0.0.0.0:*	LISTEN	509/java
4	tcp	0	0	0.0.0.0:8080	0.0.0.0:*	LISTEN	509/java

Java 프로세스,시스템이 사용중인 네트워크 분석

오픈파일 분석기

오픈파일 분석 생성일시

Created: 2019-01-19 14:15:40 | UUID: 59C24E35-1AE5-442B-AFD4-9B...

오픈파일 목록 분석 > 2019-01-19 14:15:40 (59C24E35-1AE5-442B-AFD4-9B66EC2C5721)

#	Com.	PID	User	FD	Type	Device	Size/Off	Node	Name
1	java	509	jboss	cwd	DIR	253,28	85	4199509	/home/jboss
2	java	509	jboss	rtld	DIR	253,28	261	1027	/
3	java	509	jboss	txt	REG	253,28	7376	6295581	/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.1...
4	java	509	jboss	mem	REG	253,28	411090	8420366	/opt/esp/standalone/tmp/vfs/deployme...
5	java	509	jboss	mem	REG	253,28	24956	4372527	/opt/esp/standalone/tmp/vfs/deployme...
6	java	509	jboss	mem	REG	253,28	2666695	90124	/opt/esp/standalone/tmp/vfs/deployme...
7	java	509	jboss	mem	REG	253,28	220536	28437985	/opt/esp/standalone/tmp/vfs/deployme...

Java 프로세스가 오픈한 파일 분석

시스템 프로세스 분석기

프로세스 스냅샷 목록

Created: 2019-01-18 18:41:34 | UUID: 505212AA-0DF0-43DB-8100-BF...

시스템 프로세스 상태 분석 > 2019-01-18 18:41:34

Created: 2019-01-18 18:41:33 | Uptime: 4 days 02:49:11 | Load Avg: 3.4, 3.9, 3.6 (1m, 5m, 15m)

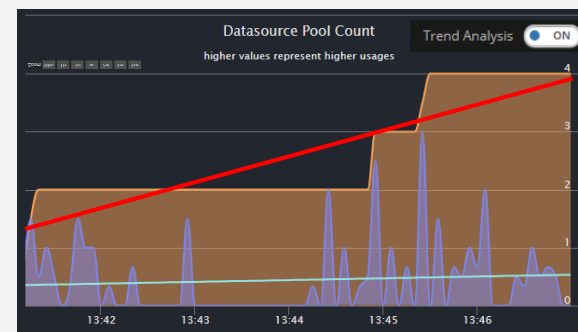
Processes: 750 Total / 749 Sleeping, 1 Running, 0 Zombie, 0 Stopped

CPU Usage: User: 10.0%, System: 0.0%, Nice: 0.0%, Wait: 0.0%, Idle: 90.0%

#	PID	User	Start Time	Mem	RSS Mem	Shared	State	CPU Time	CPU...	MEI
1	115,263	root	2019-01-14 17:53...	4.2 GB	254.5 MB	10.5 MB	Sleeping	05:37:11.2	23.1%	1
2	78,365	???	2019-01-18 18:32...	6.3 GB	404.8 MB	20.1 MB	Sleeping	00:02:20.0	21.5%	2

시스템의 프로세스 CPU, 메모리 사용량 분석/비교

데이터 추세 분석



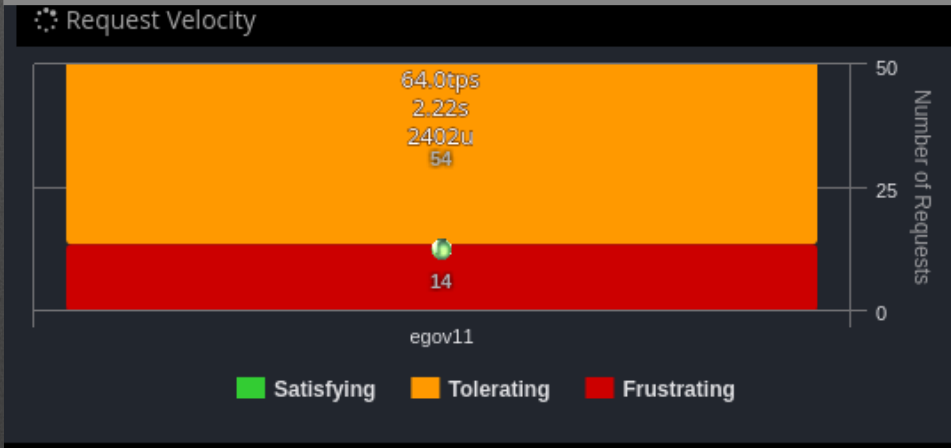
과거 데이터의 증감 추세를 분석하는 기능



KHAN [apm] - 스레드 덤프 분석



스레드 덤프 분석기



Thread Dump List

- 스레드 덤프 요청
- 스레드 덤프 분석

#	Created	Host	Instance
1	2017-05-04 17:21:48	192.168.122.1	egov11

Record ID: 1

- 스레드 상세 분석
- Lock 번호 클릭 시 Lock Owner 번호로 이동

Thread Dump Analy...

스레드 상태 분석표

#	Line#	Type	Name	State	URL	Durat...
1	68	Active Thread	http-thread-poo...	RUNNABLE	/oe1/cms/com/action...	5...
2	206	Active Thread	http-thread-poo...	BLOCKED	/oe1/cms/com/action...	4...
3	322	Active Thread	http-thread-poo...	TIMED_WAITING	/oe1/cms/com/action...	6...
4	454	Active Thread	http-thread-poo...	WAITING	/oe1/cms/sys/EgovO...	...

Thread Dump Te...

```

66 - None
67
68 "http-thread-pool-threads_249" #448 prio=5 os_prio=0 ti
java.lang.InFeasibleState: BLOCKED (on object monito
at org.apache.log4j.AppenderSkeleton.doAppend(Append
- waiting to lock <0x00000006cf8ca588> (a org.ap
at org.apache.log4j.helpers.AppenderAttachableImpl
/4
at java.lang.Exception.<init>(Exception.java:66)
at org.apache.log4j.AppenderSkeleton.doAppend(Refl
locked <0x00000006cf8ca588> (a org.ap
a:2070)
a:2070)

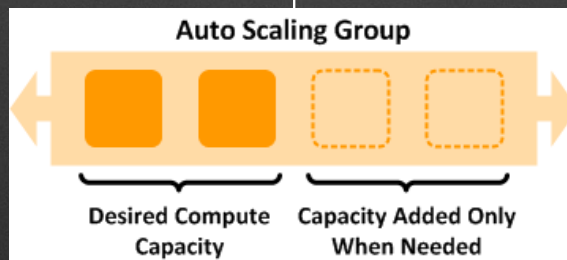
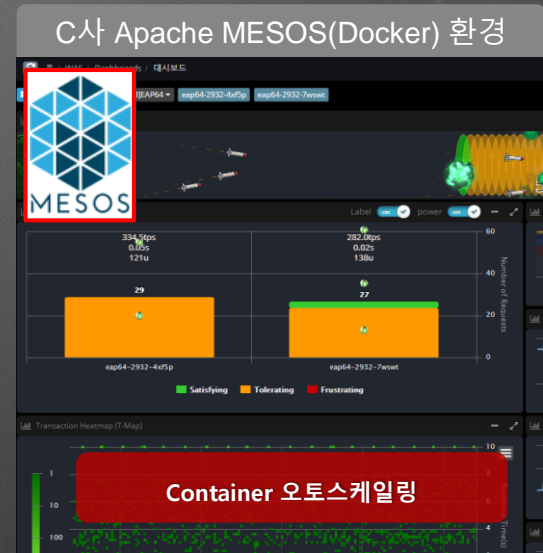
```





스레드 덤프 분석

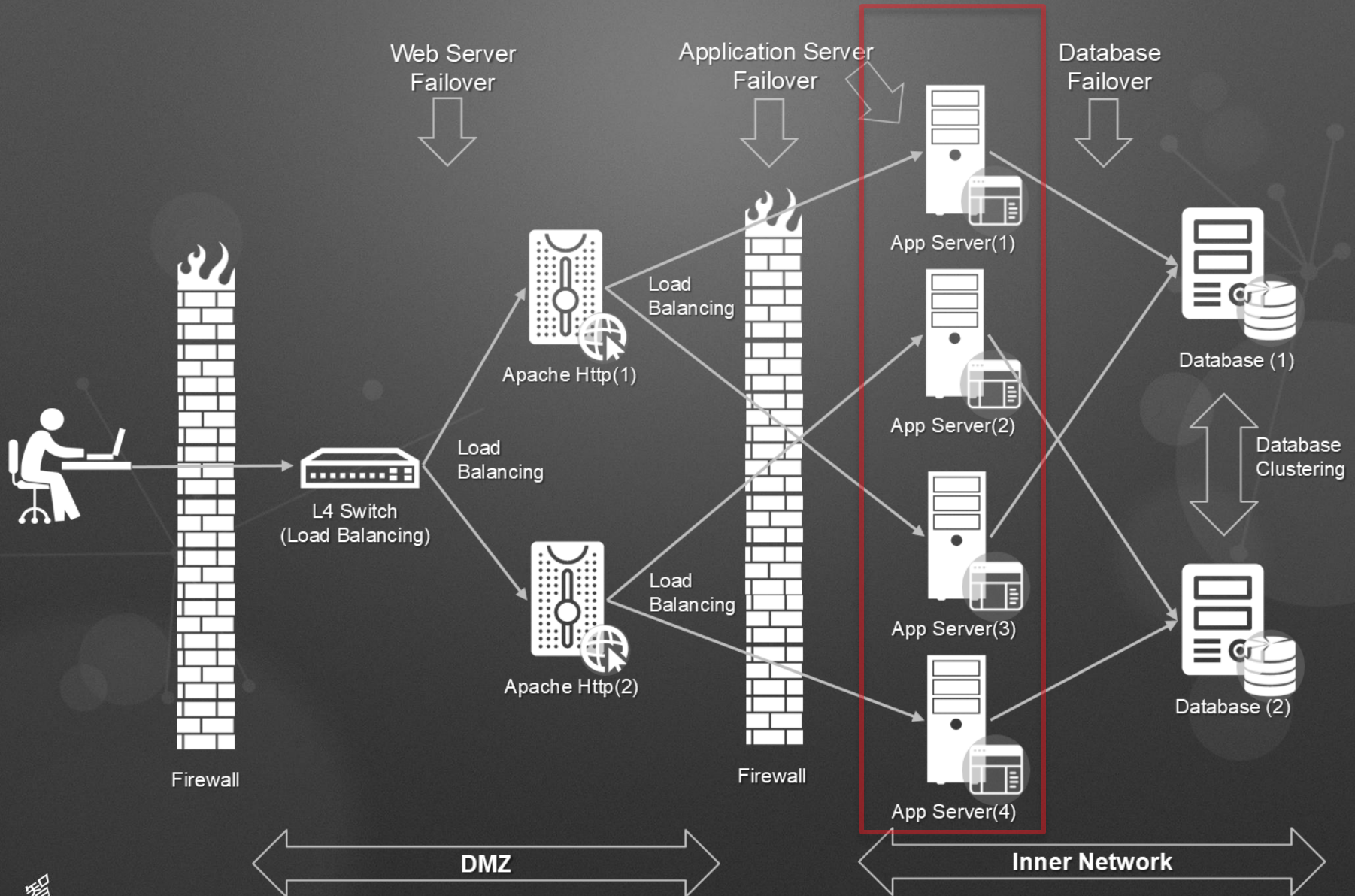
KHAN [apm] – Auto Scaling



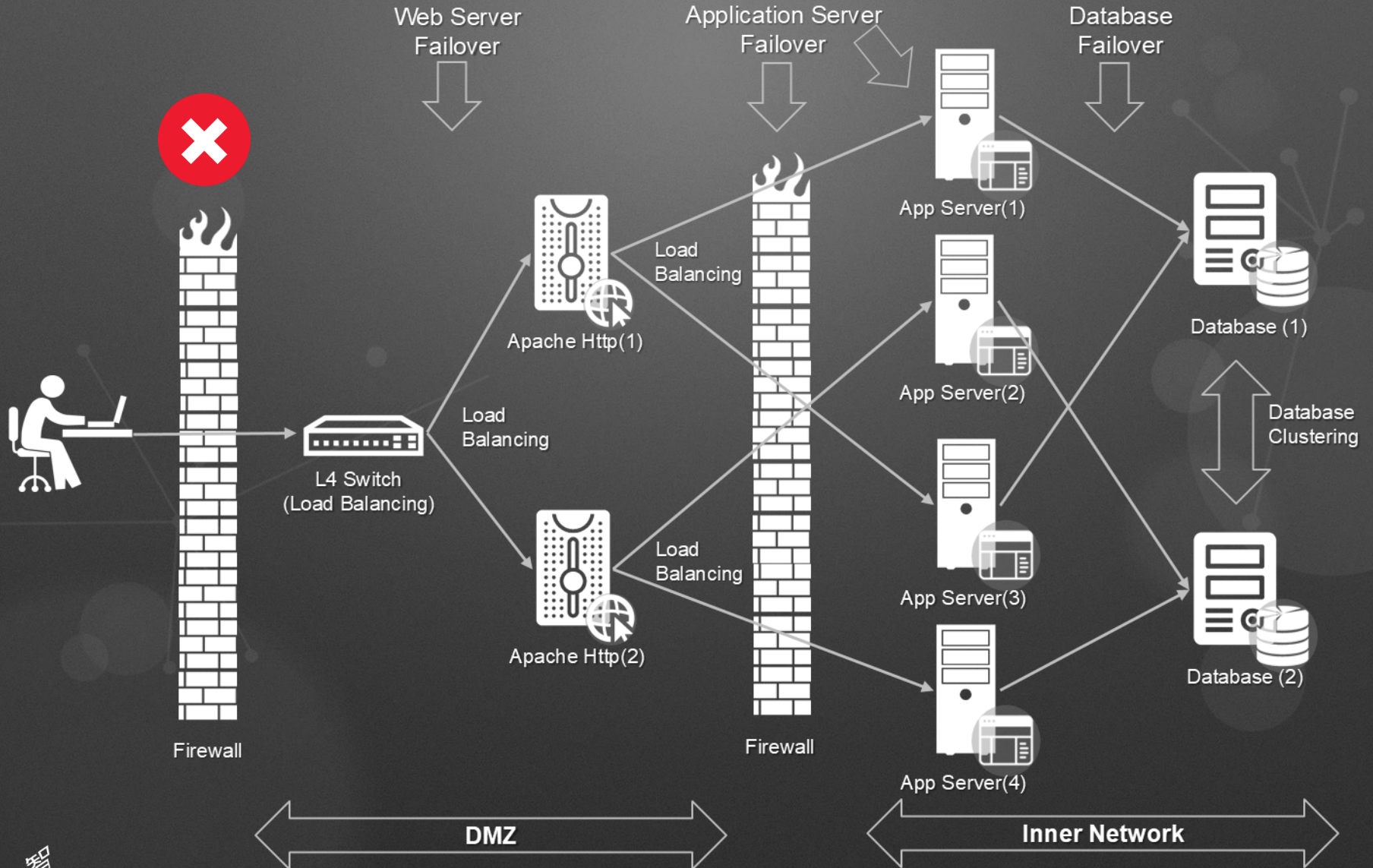


오토 스케일링

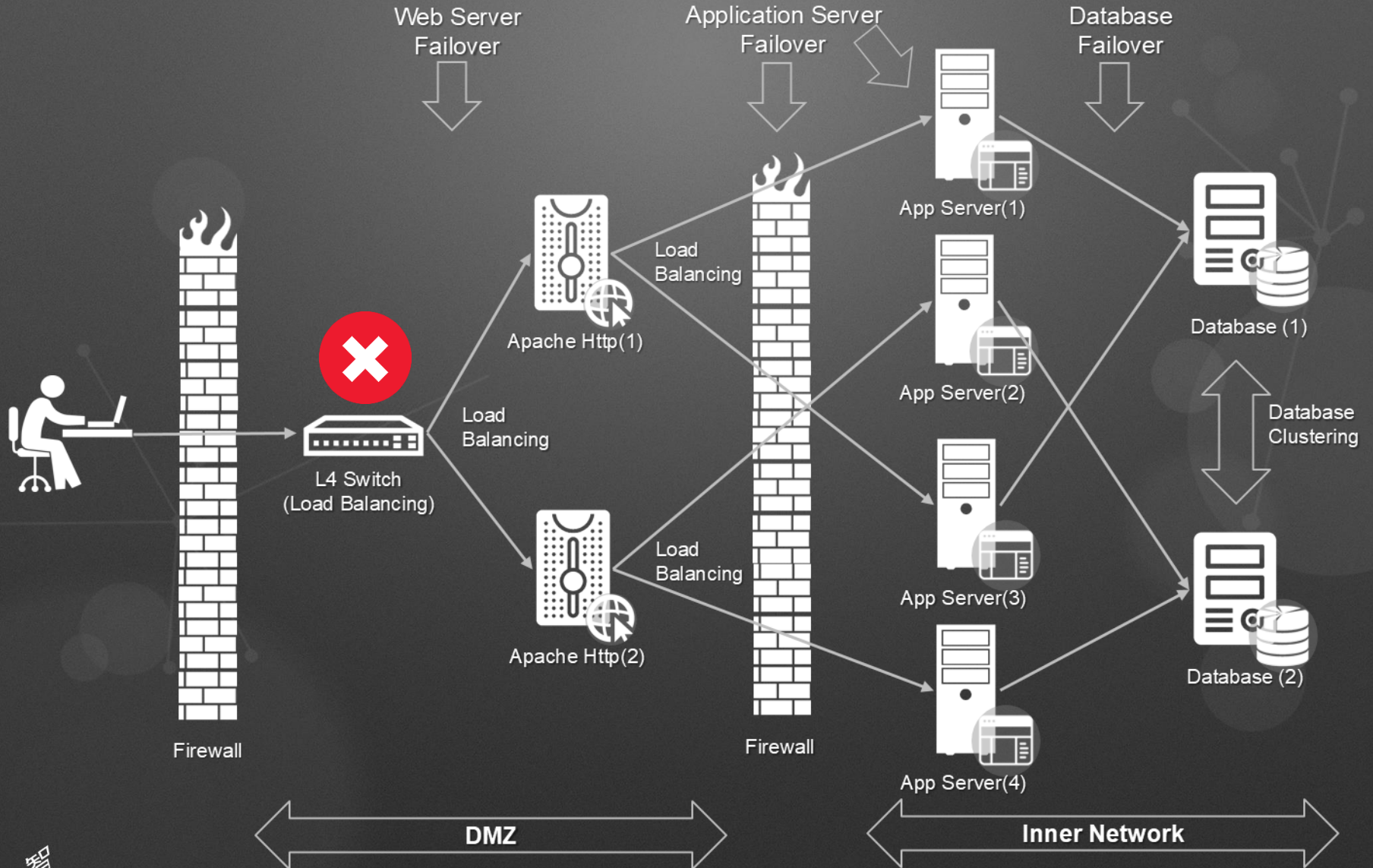
웹시스템 클러스터링 아키텍처



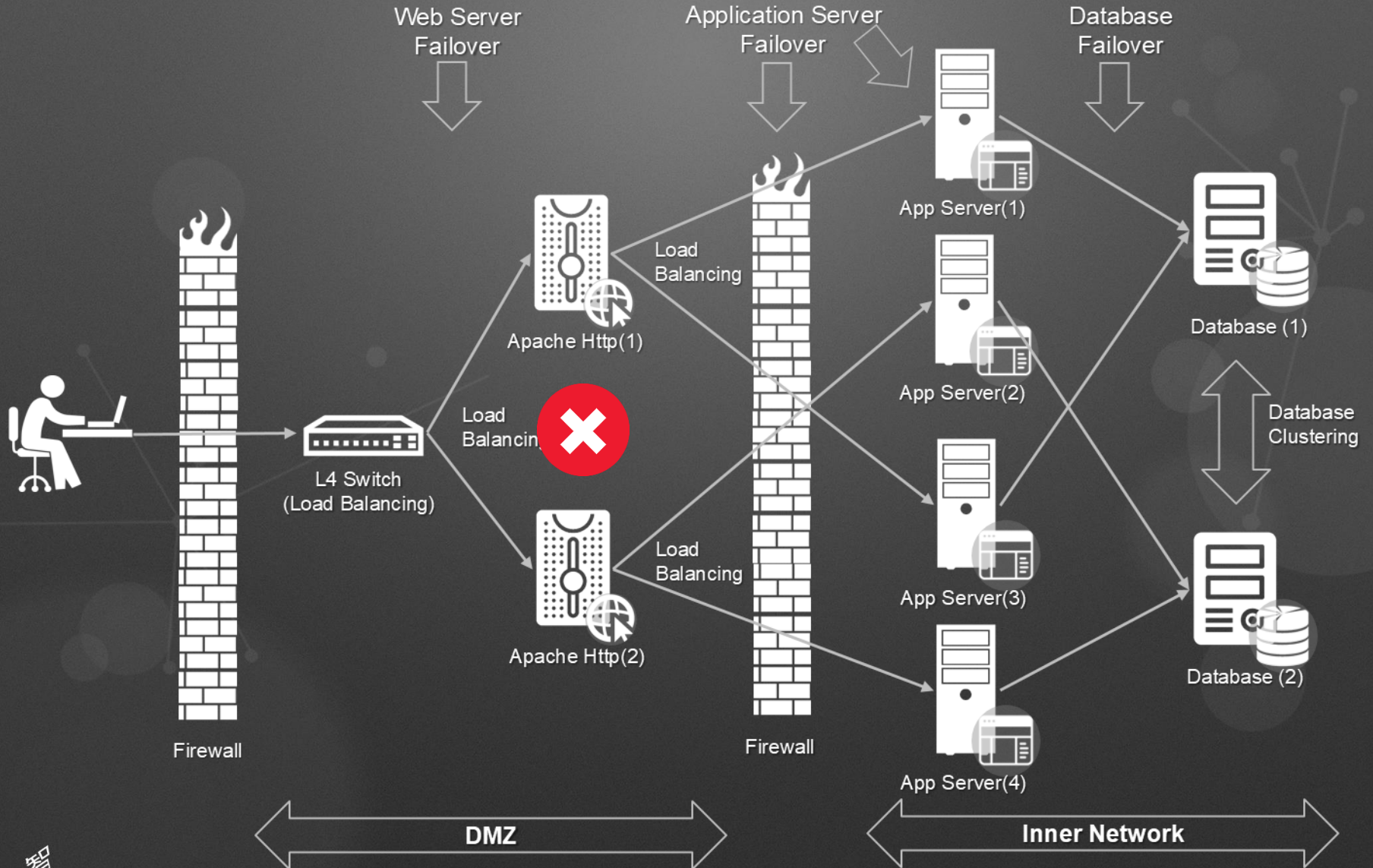
웹시스템 클러스터링 아키텍처



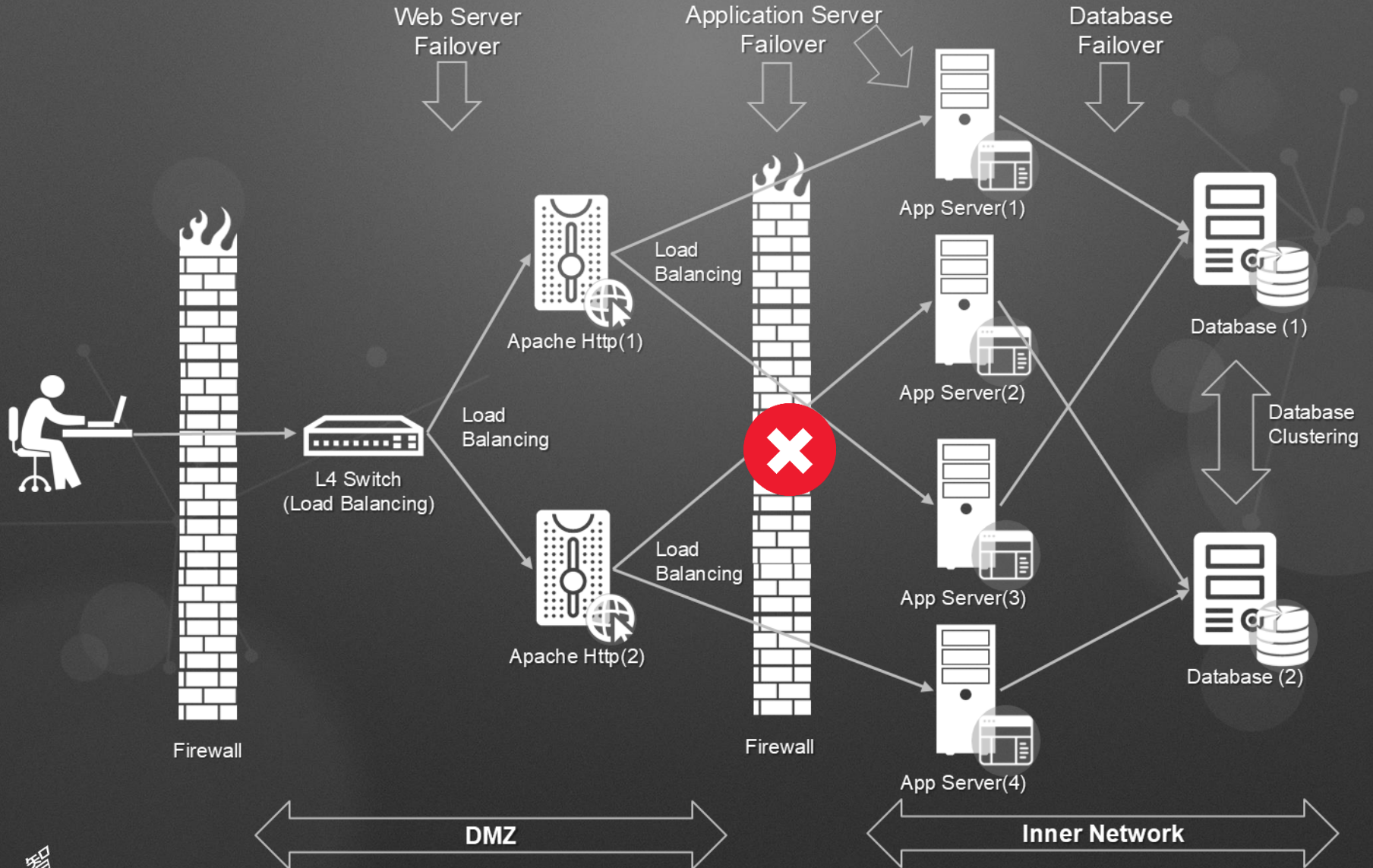
웹시스템 클러스터링 아키텍처



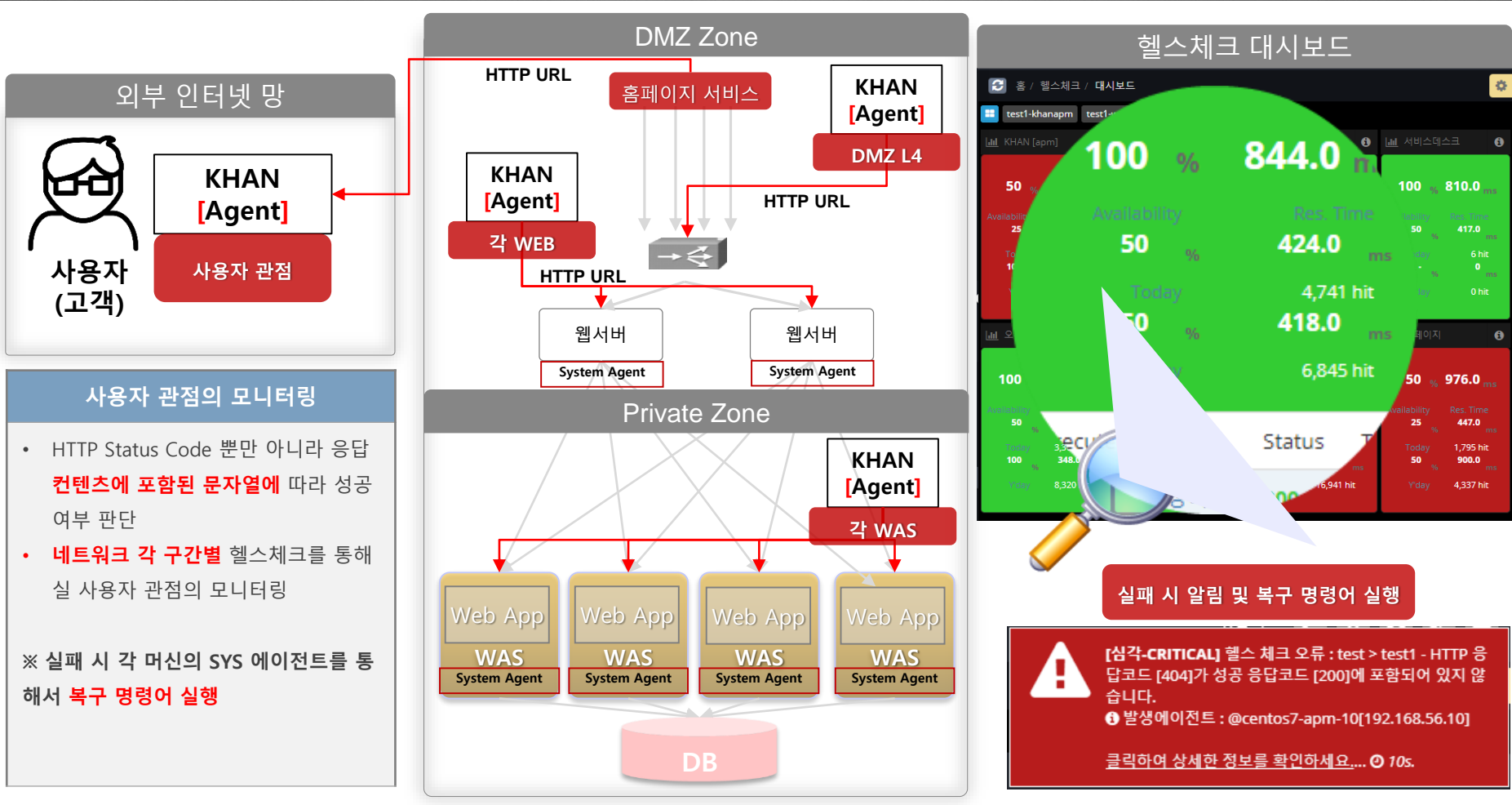
웹시스템 클러스터링 아키텍처



웹시스템 클러스터링 아키텍처



KHAN [apm] - 헬스 체크





서비스 헬스 체크

KHAN [apm] - Quick Service ON



퀵서비스 전송

이슈 내용을 간략하게 남겨주세요.

Company: 오픈나루 Username: 나오픈

Phone: 000-0000-0000 Email: customer@opennaru.com

Summary: 장애가 발생하였습니다. 분석정합니다.

Description: 화면과 함께 스레드덤프 함께 보냅니다. 분석 요청합니다.

해당 Thread Dump 를 퀵서비스 버튼을 이용하면 화면 및 덤프 파일이 오픈나루 서비스 데스크에 접수

1. Thread Dump Table

Host	Instance	Level	UUID
10.6.5.73	eap64-1419-vz4gd	INFO	A54117A3-A351-48D

2. Quick Service ON Button

퀵서비스 ON

QuickService ON

1. Issue Report

2. Thread Dump

3. Thread Dump File

Quick Service ON





퀵 서비스 ON

“살아 남는 종(種)은 강한 종이 아니고,
또 우수한 종도 아니다.
변화에 적응하는 종이다.”

- *Charles Darwin, 1809*



감사합니다.



제품이나 서비스에 관한 문의

콜 센터 : 02-469-5426 (휴대폰 : 010-2243-3394)

전자 메일 : sales@opennaru.com