

OPENMARU APM with OpenShift (Application Performance Management)

왜 OPENMARU APM 인가?

01

모니터링 보다는 장애 진단과 성능 최적화에 집중

- 실시간성 Dashboard 도 중요하지만
- Troubleshooting 에 필요한 전문화된 도구의 필요성
- 지속적인 성능관리와 운영 최적화 지원

02

Cloud Ready Architecture

- 시계열 DB, NoSQL 기반의 저장 구조
- HTML5 / Web-Socket 기반으로 상호운영성 확보
- 클라우드 환경을 위한 Auto-Scaling/Scale-out 지원

03

개방형 플랫폼에서 점유율 1위 제품

- UNIX-> Linux -> Virtualization -> Container
- Apache/Tomcat/JBoss 환경에서 주로 사용하는 상용 APM
- X86/Linux , 컨테이너, 클라우드 환경에서 점유율 1위

04

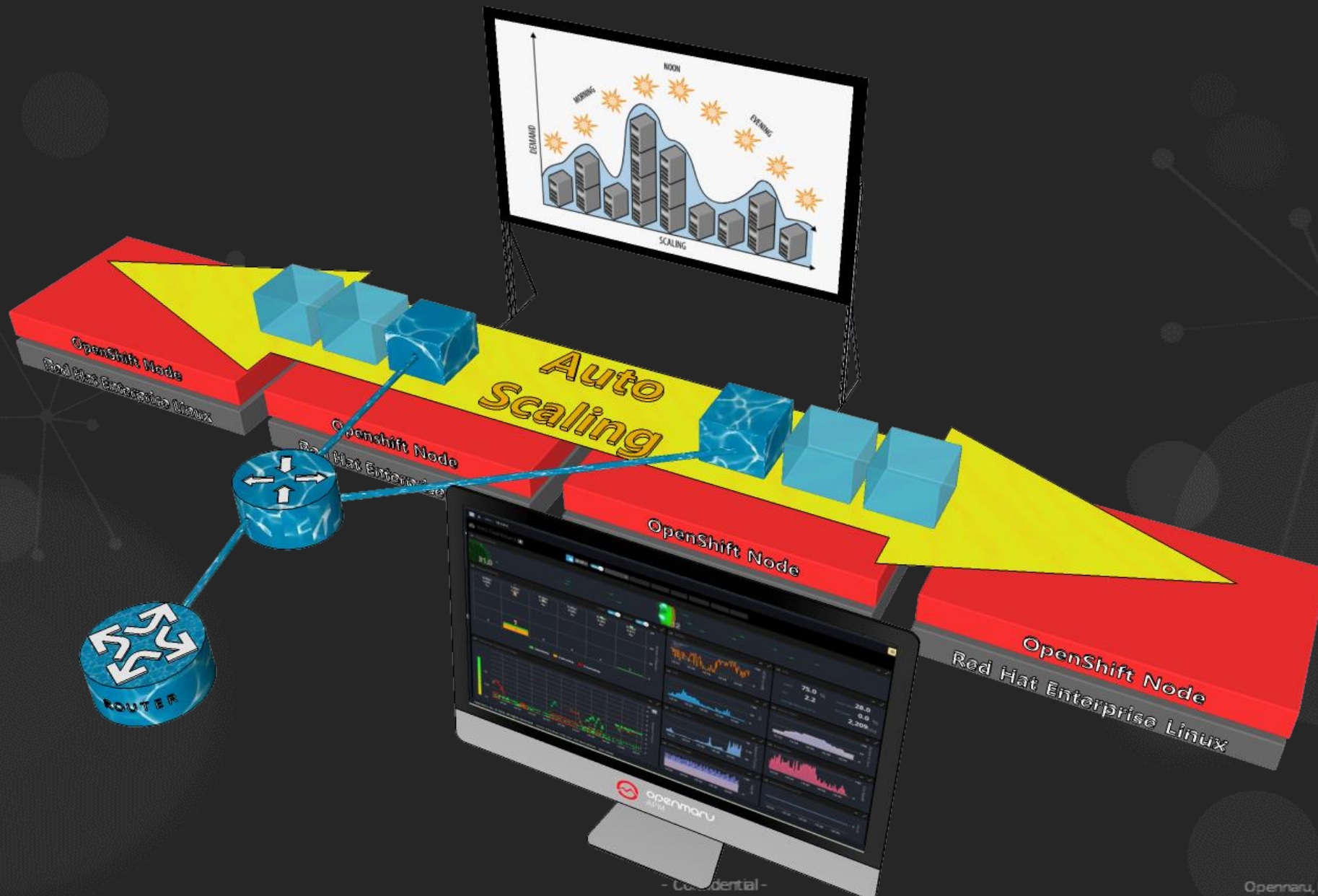
국내 최초 컨테이너 및 클라우드 환경 지원 APM

- Docker/Openshift/Openstack – immutable infrastructure 지원
- Amazon Cloud 지원
- Multicast 통신이 불가능한 환경 지원 / 유동IP 환경 지원

Application Performance Management

PaaS 환경에서 APM

OpenShift Auto Scaling 과 OPENMARU APM



PaaS 환경에서 OPENMARU APM이 필요한 이유는?



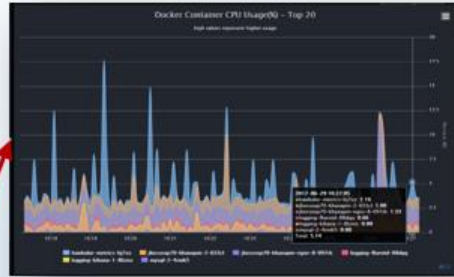
- Container 환경은 가상 OS 이며 유동 IP 이기 때문에 접속이나 로그 파일등의 확인이 어려움

가상 OS 환경에서 CPU/Memory/Disk/Network 정보 실시간 제공



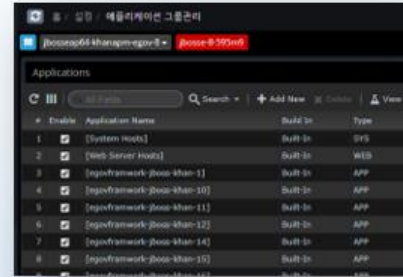
통계정보

/sys/fs/cgroup/cpu
/memory...



docker daemon

폐기된 POD 인스턴스에 대한 검색 및 상태 정보 제공



POD 검색



POD 상태정보 확인

오토스케일링 시에 부하 분산과 pod 별 처리 현황 파악

2개 Pod 컨테이너



오토스케일링 발생

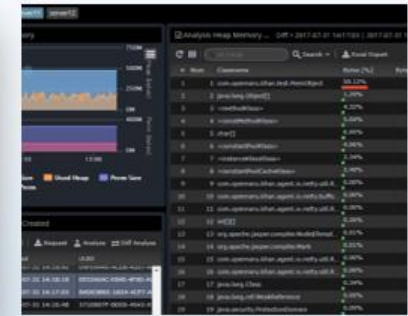
4개 Pod 컨테이너



WAS 장애 상황에서 필수적으로 필요한 분석 도구 제공



스레드덤프 분석



자바 메모리 객체 분석

PaaS 지원 APM 과 기존 APM 비교



구 분	세부 항목	PaaS 형 APM (OPENMARU APM)	기존 APM 제품
기존 온-나라 시스템	Java	<ul style="list-style-type: none"> 모니터링 지원 	<ul style="list-style-type: none"> 모니터링 지원
	WAS	<ul style="list-style-type: none"> 모니터링 지원 	<ul style="list-style-type: none"> 모니터링 지원
클라우드 온-나라 시스템	PaaS 환경	<ul style="list-style-type: none"> Docker Daemon 과 관련 이미지 정보 제공 POD 에 대한 리소스 정보 제공 (CPU/Memory/Disk/Network) 	<ul style="list-style-type: none"> 모니터링 지원 불가
	오토스케일링	<ul style="list-style-type: none"> 오토스케일링 관련 리소스 정보 제공 WAS 상태 정보 제공 	<ul style="list-style-type: none"> 모니터링 지원 불가
	PaaS 에서 Java 정보	<ul style="list-style-type: none"> POD 상에서 실행되는 Java 가상 머신 정보 제공 (Heap , Java 상태 정보 등) 	<ul style="list-style-type: none"> 모니터링 지원 불가
	PaaS 에서 WAS 정보	<ul style="list-style-type: none"> POD 상에서 실행된 WAS 에 대한 정보 	<ul style="list-style-type: none"> 모니터링 지원 불가
	POD 상태 정보 제공	<ul style="list-style-type: none"> 폐기된 POD 에 대한 검색 지원 과거 POD 에 대한 정보 제공 	<ul style="list-style-type: none"> 모니터링 지원 불가

컨테이너 자원 모니터링



- 컨테이너 환경에서는 전반적인 정보(분배된 Pod 위치, 컨테이너 리소스 등)를 모니터링 해야 한다.
- APM에서는 컨테이너 환경을 운영자가 편리하게 컨테이너(pod) 모니터링을 할 수 있다.

▶ 컨테이너(pod)에 대한 모니터링 (System > 호스트 > Docker > CPU)

Container

Container 환경 모니터링 메뉴

Docker

도커 정보

컨테이너

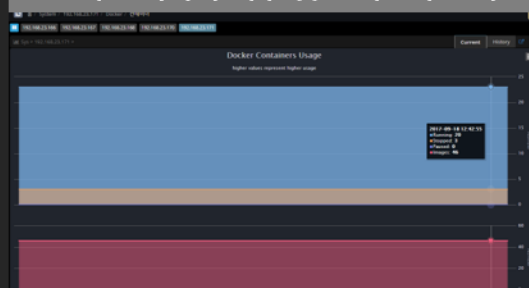
CPU

메모리

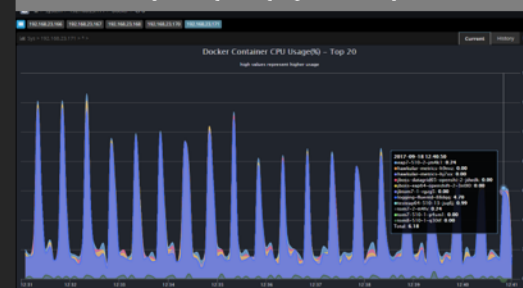


도커 모니터링 정보

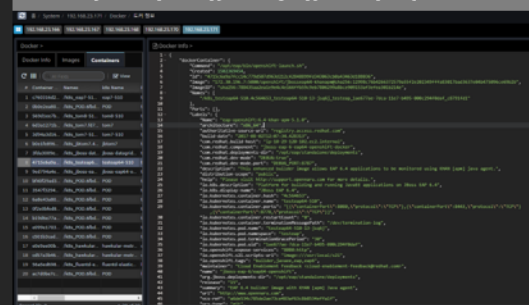
도커 이미지의 개수 및 컨테이너 상태



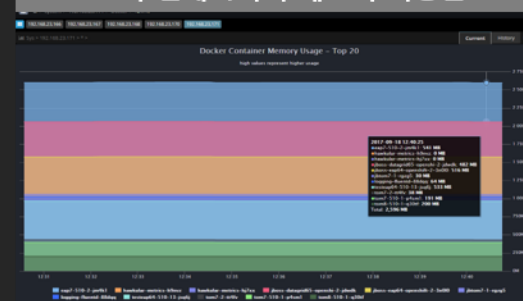
도커 컨테이너의 CPU 사용률



도커 컨테이너, 이미지, 도커 데몬 정보

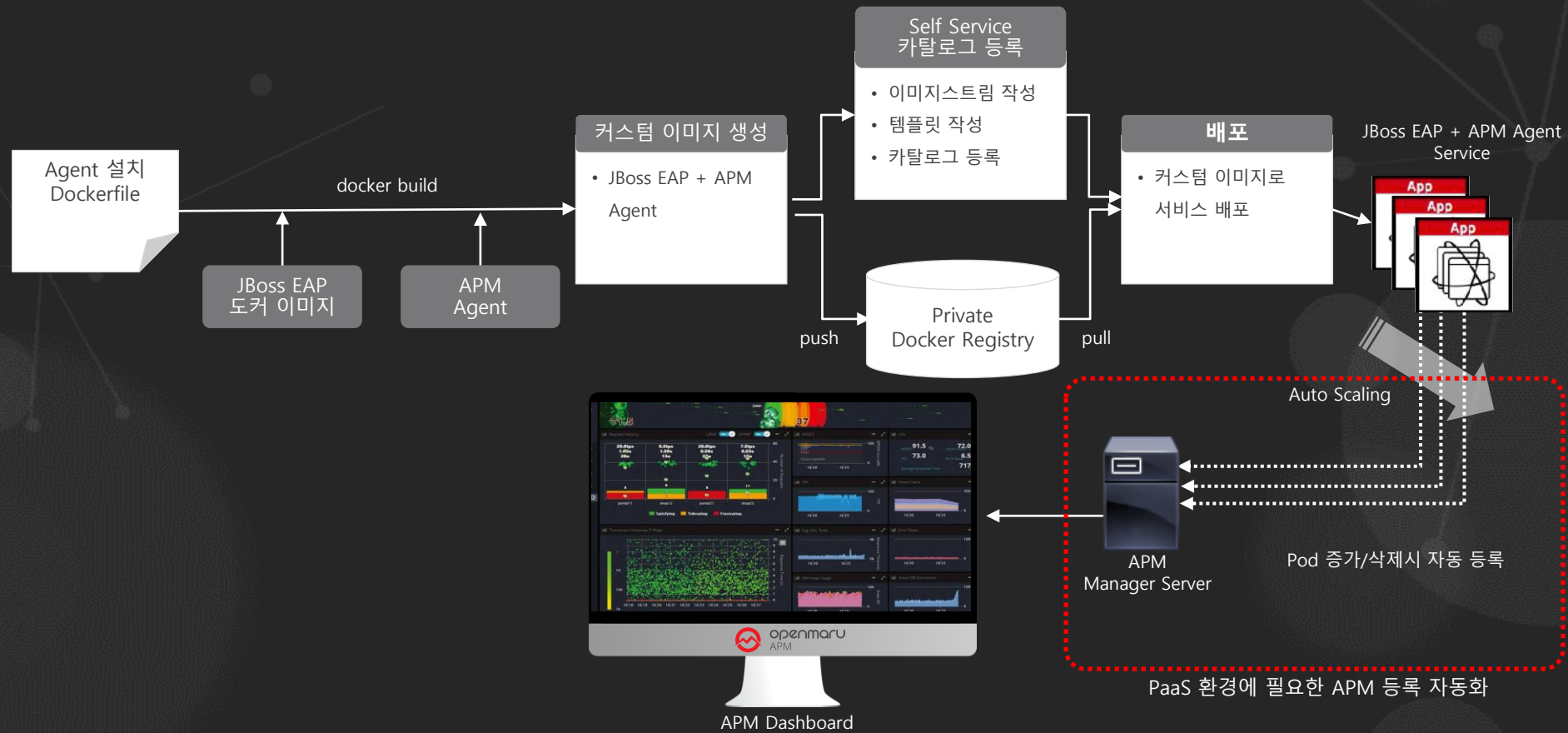


도커 컨테이너의 메모리 사용률



Kubernetes 환경에서 APM 구축 아키텍처

- PaaS 환경에서 OPENMARU APM 구축 아키텍처 제안



PaaS 환경에 필요한 APM 등록 자동화

PaaS 형 APM – 모니터링 범위

- OPENMARU APM은 기존 APM으로는 모니터링 할 수 없는 컨테이너와 컨테이너 상에서 운영되는 WAS 를 모니터링 할 수 있는 국내 최초의 제품이자 시장 점유율 1위 제품
- 구축 사례
 1. 삼성전자 OpenShift (PaaS 플랫폼)
 2. 두산정보통신 OpenShift (PaaS 플랫폼)
 3. 행정안전부 온-나라 클라우드 OpenShit (SaaS 플랫폼)
 4. 롯데카드 대고객 채널 서비스 OpenShift
 5. 코리아크레딧뷰로 채널 서비스 OpenShift
 6. 법무부 OpenShift

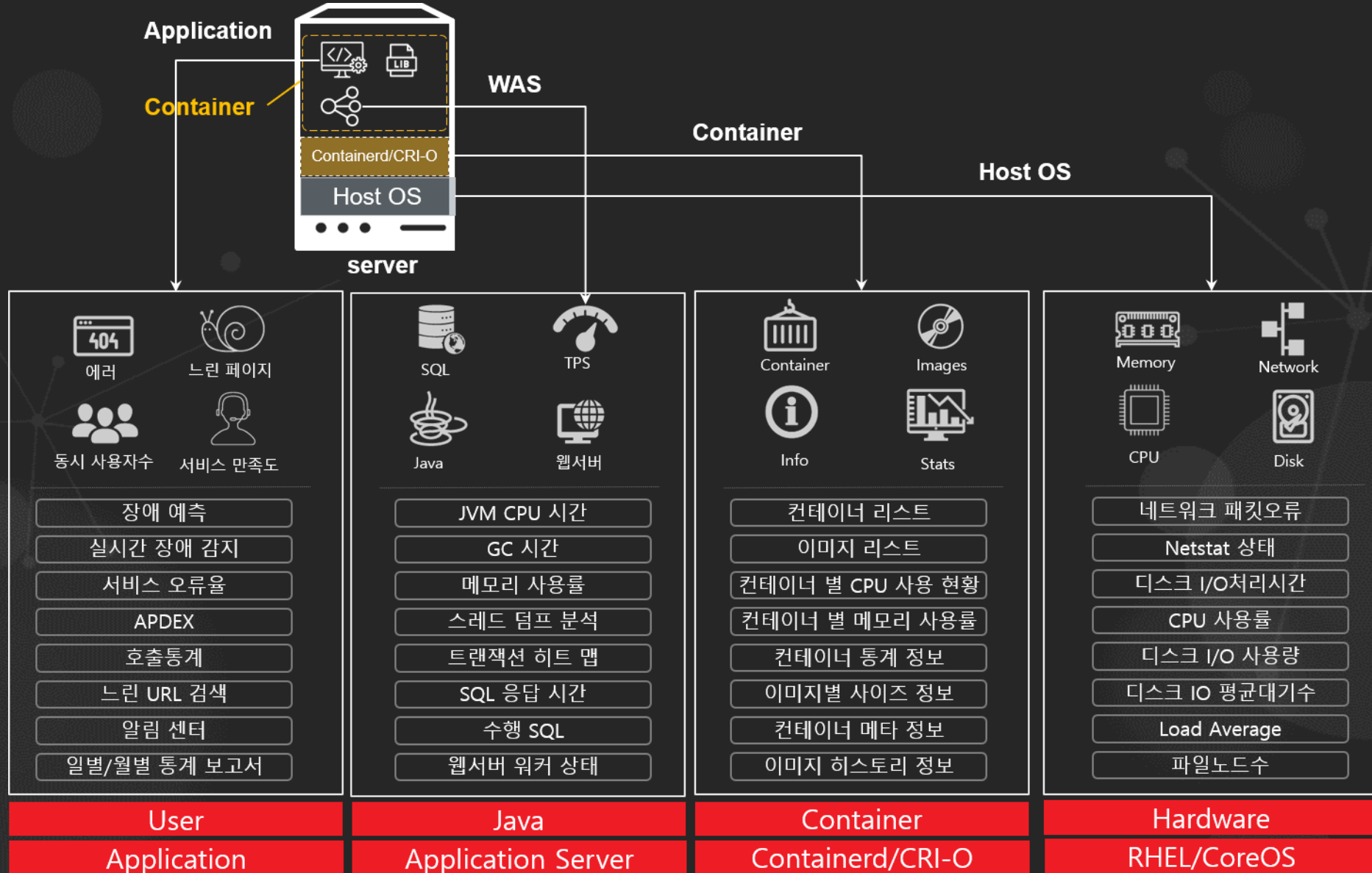


Application Performance Management



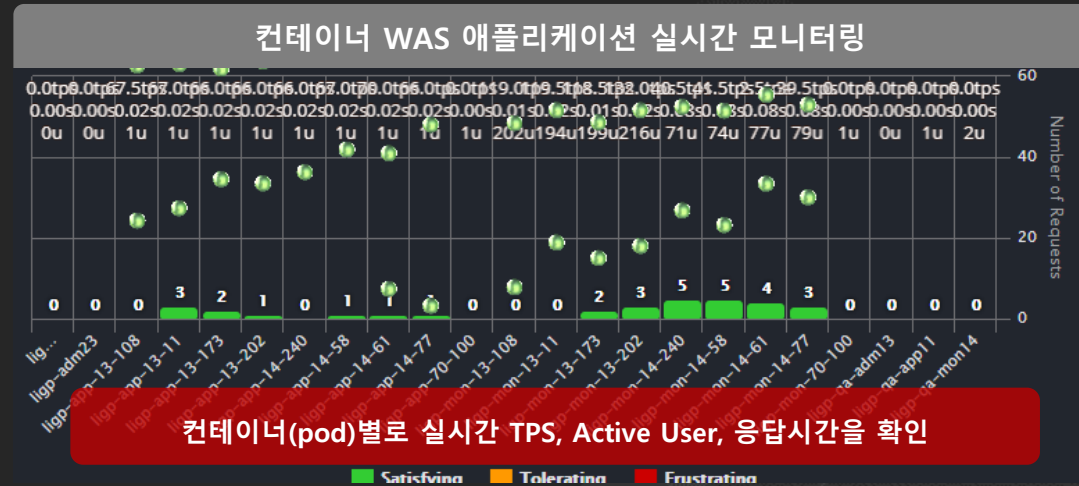
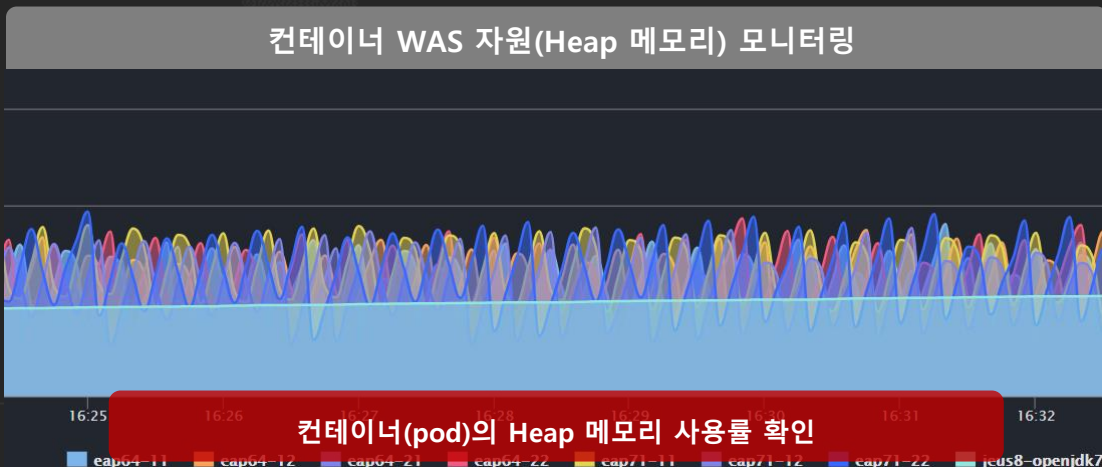
PaaS형 클라우드 지원 기능

성능 모니터링과 위한 모니터링 기능 제공



컨테이너에서 운영되는 WAS의 성능 모니터링 기능

- 컨테이너 내의 WAS 자원 및 애플리케이션 모니터링



장애 원인 분석 도구 제공



- 애플리케이션 성능 모니터링 및 진단, 장애 원인 분석을 통해 서비스를 최적의 상태로 운영할 수 있는 도구 제공

스레드 덤프 분석기

#	Lin...	Type	Name	State	URL	Durat...	CPU TL	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-CBD7FE713B48)
696 #005 "ajp-0.0.0.0-8009-275" daemon prio=10 tid=0x000077fc41828000 nid=0x5f6a runnable [0x000077fc41828000]
697   java.lang.Thread.State: RUNNABLE
698   at java.net.SocketInputStream.socketRead0(Native Method)
699   at java.net.SocketInputStream.read(SocketInputStream.java:130)
700   at org.apache.coyote ajp.AjpProcessor.read(AjpProcessor.java:1027)
701   at org.apache.coyote ajp.AjpProcessor.readAndUnpack(AjpProcessor.java:1110)
702   at org.apache.coyote ajp.AjpProcessor.readAndUnpack(AjpProcessor.java:1083)
703   at org.apache.coyote ajp.AjpProcessor.readAndUnpack(AjpProcessor.java:1178)
704   at org.apache.coyote ajp.AjpProcessor.readAndUnpack(AjpProcessor.java:1213)
705   at org.apache.coyote ajp.AjpProcessor.readAndUnpack(AjpProcessor.java:1213)
706   at org.apache.coyote ajp.AjpProcessor.readAndUnpack(AjpProcessor.java:1213)
707   at org.apache.catalina.connector.InputBuffer.read(ByteBuffer.java:333)
708   at org.apache.catalina.connector.CoyoteInputStream.read(CoyoteInputStream.java:183)
709   at sun.nio.cs.StreamDecoder.read(StreamDecoder.java:263)
710   at sun.nio.cs.StreamDecoder.implRead(StreamDecoder.java:306)
711   at sun.nio.cs.StreamDecoder.read(StreamDecoder.java:303)
712   Locked <@0000000770564f88> (a java.io.InputStreamReader)
713   at java.io.InputStreamReader.read(InputStreamReader.java:167)
714   at java.io.BufferedReader.read1(BufferedReader.java:136)
715   at java.io.BufferedReader.read(BufferedReader.java:200)
716   Locked <@0000000770564f88> (a java.io.InputStreamReader)
717   at java.io.BufferedReader.read(BufferedReader.java:167)
718   at org.apache.commons.lang.text.StrTokenizer.tokenize(StrTokenizer.java:17)
719   at org.apache.commons.lang.text.StrTokenizer.tokenize(StrTokenizer.java:21)
    
```

Lock을 추적가능, URL 정보표시

JVM 메모리 객체 분석기

히스토그램 생성일시

#	Created	UUID
1	2019-01-19 14:11:15	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	258,635
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	41,370

Java 메모리를 점유한 객체 분석/비교

네트워크 상태 분석기

네트워크 상태 생성일시

#	Created	UUID
1	2019-01-19 14:14:08	5749186D-7480-47D1-BF0F-2A1...

네트워크 상태 분석 > 2019-01-19 14:14:08 (2AC63786-36D8-4AF0-9BE3-0BBF28AC643E)

#	Proto	Recv-Q	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	UUID
1	2019-01-19 14:15:40	59C24E35-1AE5-442B-AFD4-9B...

오픈파일 목록 분석 > 2019-01-19 14:15:40 (59C24E35-1AE5-442B-AFD4-9B6EC2C57221)

#	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	rtd	DIR	253,28	261	1027	/
3	java	509	jboss	tst	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/eap/standalone/tmp/vfs/deploye...
5	java	509	jboss	mem	REG	253,28	24956	4372527	/opt/eap/standalone/tmp/vfs/deploye...
6	java	509	jboss	mem	REG	253,28	2666695	90124	/opt/eap/standalone/tmp/vfs/deploye...
7	java	509	jboss	mem	REG	253,28	220536	29437985	/opt/eap/standalone/tmp/vfs/deploye...

Java 프로세스가 오픈한 파일 분석

시스템 프로세스 분석기

프로세스 스냅샷 목록

#	Created	UUID
1	2019-01-18 18:41:34	505212AA-0DF0-43DB-8100-BF...

시스템 프로세스 상태 분석 > 2019-01-18 18:41:34

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

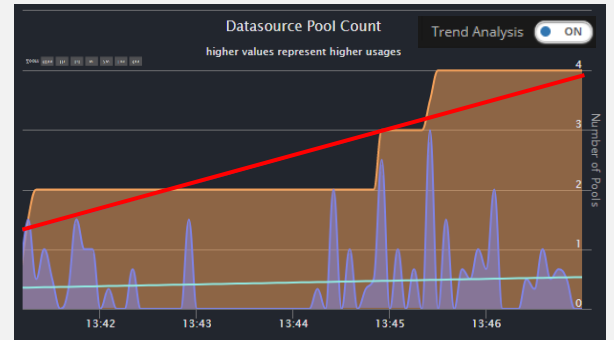
Processes: 750 Total / 749 Sleeping, 1 Running, 0 Zombie, 0 Stopped

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

Application Logging 메시지를 T-Map Trace 에서 수집



Transactions > 2019-02-22 16:36:31 ~ 2019-02-22 16:36:58 > 558ms ~ 2626ms Export

Agent Type	Server IP	Client IP	Instance ID	URL	Status	Warn#	Duration(ms)	SQL Time(ms)	Fetch Gal
WAS	192.168.23.65	0.0.0.0	jws11-bGUg	/log4jtest/index.jsp	200	4	1,354	0	

애플리케이션에서 WARN, ERROR 로 출력된
메시지 Trace 추적
(log4j, log4j2, logback)

Transaction Detail | Trace | StackTrace

Agent type: WAS Transaction ID: 1691422badc-d762fe543057f0

Warning Message

- warn message 1
- warn message 2
- error message 1
- error message 2

Method Traces

```
=====
[Num.][ Start Time | Elapsed| % | Exclusi| A-Gab| CPUTime]          Method Call
-----
[ 1][16:36:54.237| 1,354| 100| 74| 0| 625.1]+ org.apache.catalina.connector.CoyoteAdapter.service()
[ 2][16:36:54.292| 1,280| 94| 937| 343| 581.5] + org.apache.jasper.servlet.JspServlet.service()
[ 3][16:36:55.229| 343| 25| 333| 0| 96.2] + org.apache.jsp.index_jsp._jspService()
[ 4][16:36:55.560| 3| 0| 3| 0| 2.9] + org.apache.log4j.Category.forcedLog()
> Logger: WARN [test log4j] warn message 1
[ 5][16:36:55.563| 7| 0| 7| 0| 2.6] + org.apache.log4j.Category.forcedLog()
> Logger: WARN [test log4j] warn message 2
> Err.At:
at org.apache.jsp.index_jsp._jspService(index_jsp.java:133)
=====
```

헬스체크 TCP/프로세스 체크 기능 추가

Name * script

Command Execute * /tmp/check.sh

Method * Command Execute

HTTP(GET, POST, PUT, DELETE): 확인할 URL(http, https)을 입력하십시오.
ex) http://www.openmaru.com/index.jsp

TCP Socket: 확인할 IP 및 PORT(local or remote)를 입력하십시오.
ex) 192.168.23.190:8080

Command Execute: OS 내에서 실행할 명령을 입력하십시오.
종료 코드가 0이면 명령이 성공한 것으로 간주됩니다.
ex) /tmp/check.sh or d:\check.bat

Error Next * True

Retry Counts * 2

Enable * Enable

저장 취소

스크립트 실행 후 0 이 리턴되면 성공

로컬 및 원격 포트 활성화 여부 체크

Name * PORT 확인

TCP Socket(IP:PORT) * 192.168.23.190:80

Method * TCP Socket(IP:PORT)

HTTP(GET, POST, PUT, DELETE): 확인할 URL(http, https)을 입력하십시오.
ex) http://www.openmaru.com/index.jsp

TCP Socket: 확인할 IP 및 PORT(local or remote)를 입력하십시오.
ex) 192.168.23.190:8080

Command Execute: OS 내에서 실행할 명령을 입력하십시오.
종료 코드가 0이면 명령이 성공한 것으로 간주됩니다.
ex) /tmp/check.sh or d:\check.bat

Timeout * 30 sec

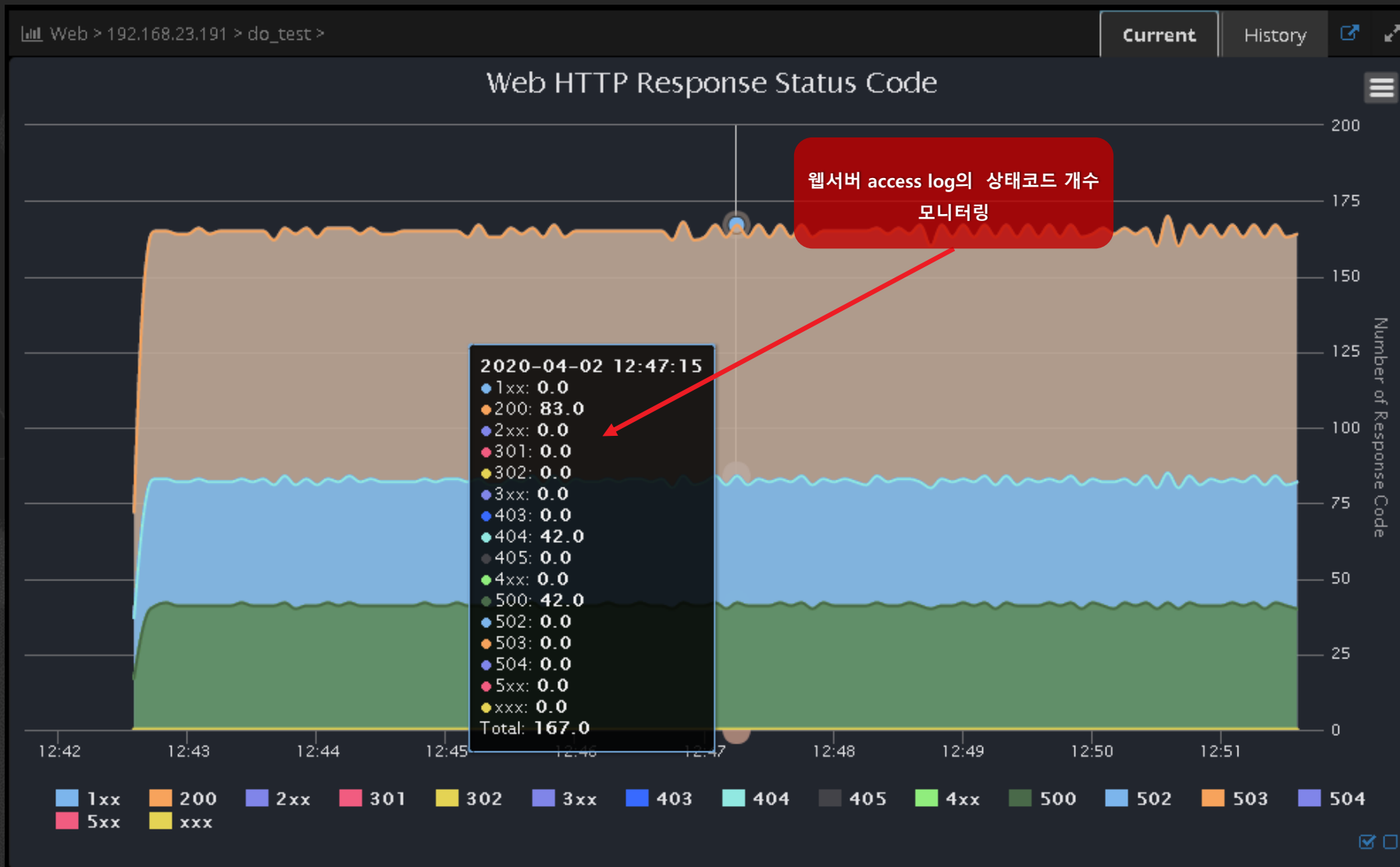
Error Next * True

Retry Counts * 2

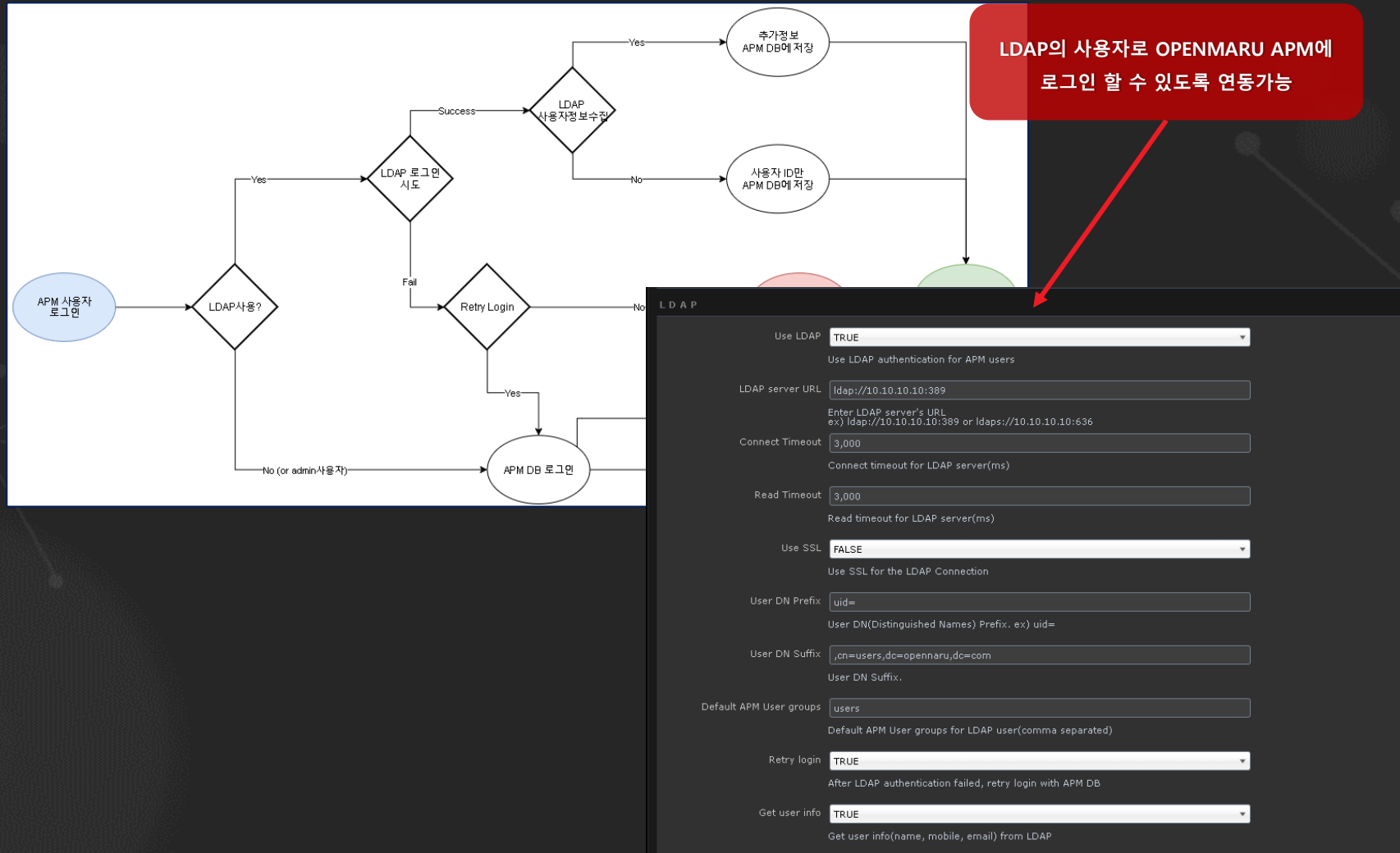
Enable * Enable

저장 취소

Apache HTTPD Access LOG 모니터링



LDAP 연동



Application Performance Management

OpenShift 4 - 최신 컨테이너 표준 지원 (CRI-O , CoreOS)

MIRANTIS 가 DOCKER ENTERPRISE 사업을 인수

- 미국 시간 2019년 11월 13일 Mirantis 는 Docker 의 Docker Enterprise 사업을 인수 발표
- 컨테이너 기술인 Docker 와 그것의 기업용 상용 버전을 제공하는 미국의 Docker 사는 Docker Enterprise Platform 사업을 Mirantis 에 매각하기로 합의



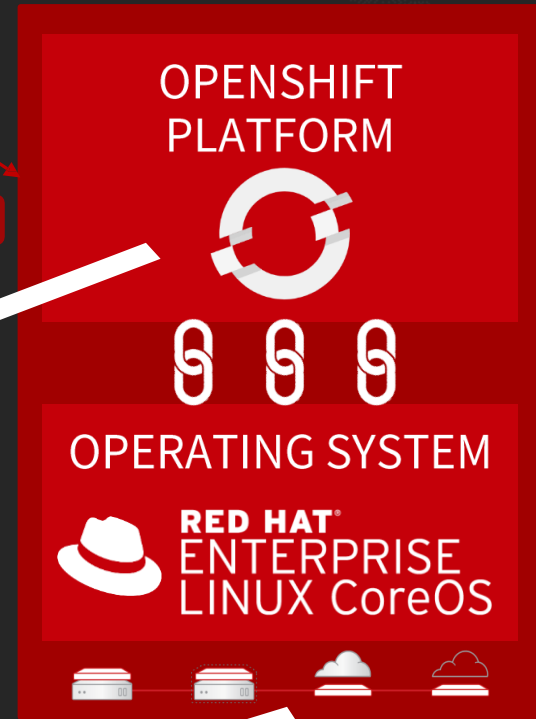
Image Source : <https://www.docker.com/blog/au-revoir/>

- Mirantis 의 직원은 450명 정도이며, 이번 인수를 통해 Docker 지원 300명 정도가 조직에 새롭게 통합
- Docker 의 공동창업자 중 한명인 솔로몬 하이쿠스 최고기술책임자 (CTO) 는 2019년 11월 28일 회사 블로그를 통해 퇴사를 밝혔습니다.

최신 Container 기술 – CRI-O, CoreOS 지원

Immutable OS
사용자 패키지 설치 불가

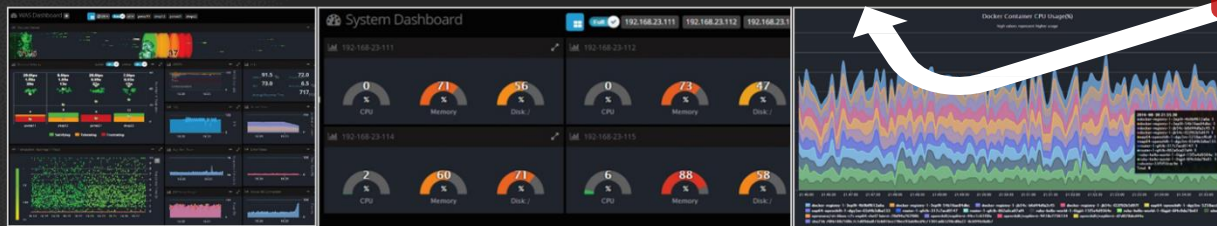
OPENSIFT 4



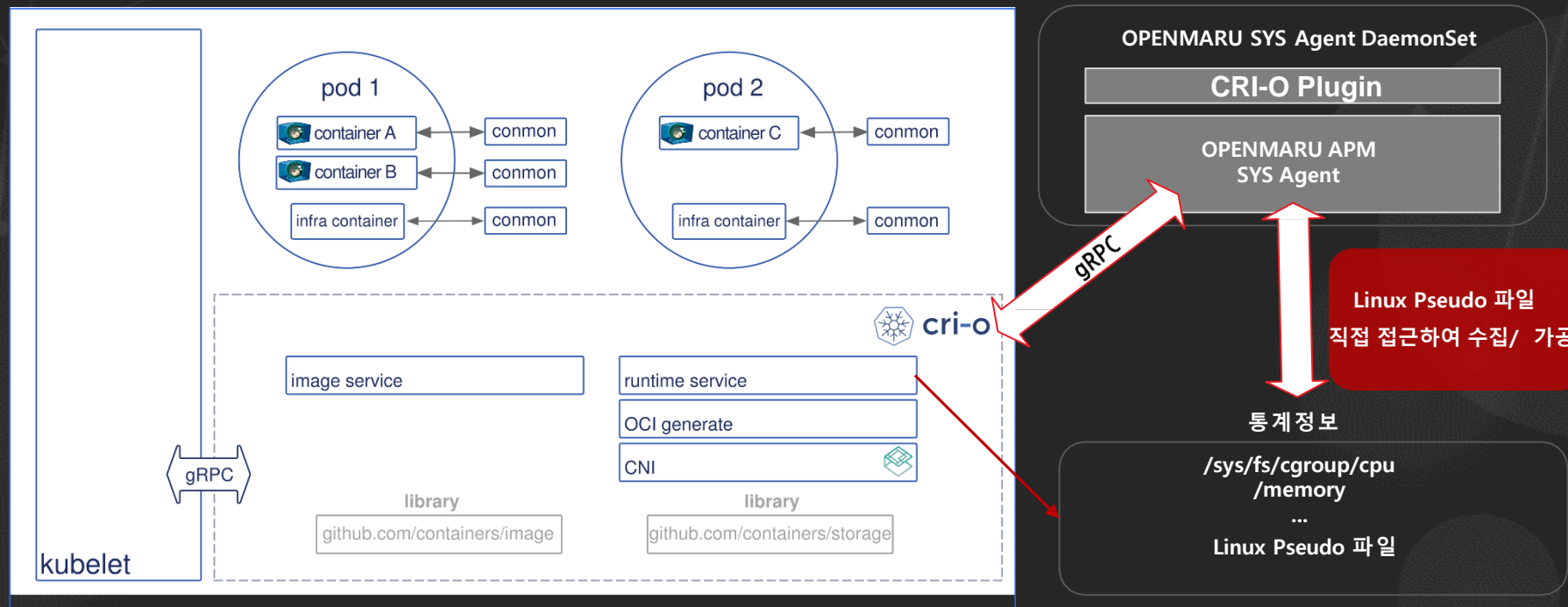
컨테이너 모니터링

User Application	Java Application Server	Hardware OS	Container Docker/CRI-O
<ul style="list-style-type: none"> 에러 느린 페이지 동시 사용자수 서비스 만족도 장애 예측 실시간 장애 감지 서비스 오류율 APDEX 호출통계 느린 URL 검색 알림 센터 일별/월별 통계 보고서 	<ul style="list-style-type: none"> SQL TPS Java 웹서버 JVM CPU 시간 GC 시간 메모리 사용률 스레드 덤프 분석 트랜잭션 히트 맵 SQL 응답 시간 수행 SQL 웹서버 워커 상태 	<ul style="list-style-type: none"> Memory Network CPU Disk 네트워크 패킷 오류 Netstat 상태 디스크 I/O 처리 시간 CPU 사용률 디스크 I/O 사용량 디스크 IO 평균 대기 수 Load Average 파일 노드 수 	<ul style="list-style-type: none"> Container Images Docker Info Docker Stats 도커 컨테이너 리스트 도커 이미지 리스트 컨테이너 별 CPU 사용 현황 컨테이너 별 메모리 사용률 컨테이너 통계 정보 이미지별 사이즈 정보 도커 메타 정보 이미지 히스토리 정보

CoreOS Host 자원 모니터링



OPENMARU APM – CoreOS 및 컨테이너 모니터링



설치 - SYS Agent 배포방법



Red Hat OpenShift Container Platform

You are logged in as a temporary administrative user. Update the [cluster OAuth configuration](#) to allow others to log in.

Project: openmaru-apm

Daemon Sets > Daemon Set Details

openmaru-sys-agent

Actions

Overview | YAML | Pods | Environment | Events

Filter by name...

2 Running | 0 Pending | 0 Terminating | 0 CrashLoopBackOff | 0 Completed | 0 Failed | 0 Unknown

Select All Filters 2 Items

Name	Namespace	Owner	Node	Status	Readiness
openmaru-sys-agent-fmrr	openmaru-apm	openmaru-sys-agent	worker2.ocp43.example.com	Running	Ready
openmaru-sys-agent-gfbw	openmaru-apm	openmaru-sys-agent	worker1.ocp43.example.com	Running	Ready

openmaru-sys-agent

Actions

Overview | YAML | Pods | Environment | Events

Container: openmaru-sys-agent

Single values (env)

NAME	VALUE
KHAN_HOST	192.168.23.230
KHAN_PORT	80

+ Add Value + Add from Config Map or Secret

OPENMARU APM – 컨테이너 정보 수집



Docker Info | Images | Containers

모든 탭들

#	Name	Value
1	"engine"	→
2	"EngineVersion"	1.16.2-15.dev.rha
3	"Node OS"	Linux
4	"Node IP"	192.168.23.213
5	"Node CPU"	4

Record ID: 4 1-5 of 5

Docker Info >

```

1 {
2   "engine": {
3     "versions": {
4       "version": "0.1.0",
5       "runtimeName": "cri-o",
6       "runtimeVersion": "1.16.2-15.dev.rhaos4.3.gita83f883.e18",
7       "runtimeApiVersion": "v1alpha1"
8     },
9     "node": {
10      "hostname": "worker1.ocp43-test1.example.com",
11      "address": "",
12      "defaultAddress": "192.168.23.213",
13      "availableProcessors": "1",
14      "os": {
15        "desc": "Red Hat Enterprise Linux 4",
16        "name": "linux",
17        "version": "4.18.0-147.3.1.el8_1.x86_64",
18        "arch": "x86_64",
19        "vendor": "Red Hat"
20      },
21      "network": {
22        "address": "192.168.23.213",
23        "hwaddr": "08:50:56:98:2E:FA",
24        "netmask": "255.255.255.0",
25        "hostname": "worker1.ocp43-test1.example.com",
26        "domainName": "(none)",
27        "gateway": "192.168.23.1",
28        "dns1": "192.168.23.2",
29        "dns2": ""
30      },
31      "cpu": {
32        "vendor": "Intel",
33        "model": "Xeon",
34        "mhz": "2693",
35        "totalCores": "4",
36        "totalSockets": "4",
37        "coresPerSocket": "1",
38        "cacheSize": "30720"
39      },
40      "mem": {
41        "total": "16815345664"
42      },
43      "swap": {
44        "total": "0"
45      }
46    }
47  }

```

Docker Info | Images | **Containers**

모든 탭들

#	Container ...	Names	k8s Name
22	236880582...	alertmanager...	alertmana
23	7fce47727...	config-reloader	config-relo
24	39a151dfc...	prom-label-pro...	prom-label
25	b7dba03e4...	kube-rbac-proxy	kube-rbac-
26	99364df8a...	alertmanager	alertmana
27	ce3fe413a8...	oauth-proxy	oauth-prox
28	a2a225727...	thanos-querier	thanos-que
29	82c286281...	kube-rbac-pro...	kube-rbac-
30	3079e3d80...	kube-rbac-pro...	kube-rbac-
31	1eeed2cb2...	registry	registry
32	e96edf5f15...	kube-rbac-pro...	kube-rbac-
33	7870f233b...	kube-rbac-pro...	kube-rbac-
34	581b2c5d2...	dns-node-resol...	dns-node-t
35	d8b9b7b3...	dns	dns
36	2828cc16d...	node-ca	node-ca
37	429f643f44...	router	router
38	95de99174...	kube-multus	kube-mult
39	dbbd82588...	kube-rbac-proxy	kube-rbac-
40	bed0b2541...	node-exporter	node-expo
41	726b73cc6...	sdn	sdn
42	ba6b786bf...	openvswitch	openvswit
43	40c4fb01dc...	machine-confi...	machine-c

Record ID: 31 22-43 of 43

Docker Info >

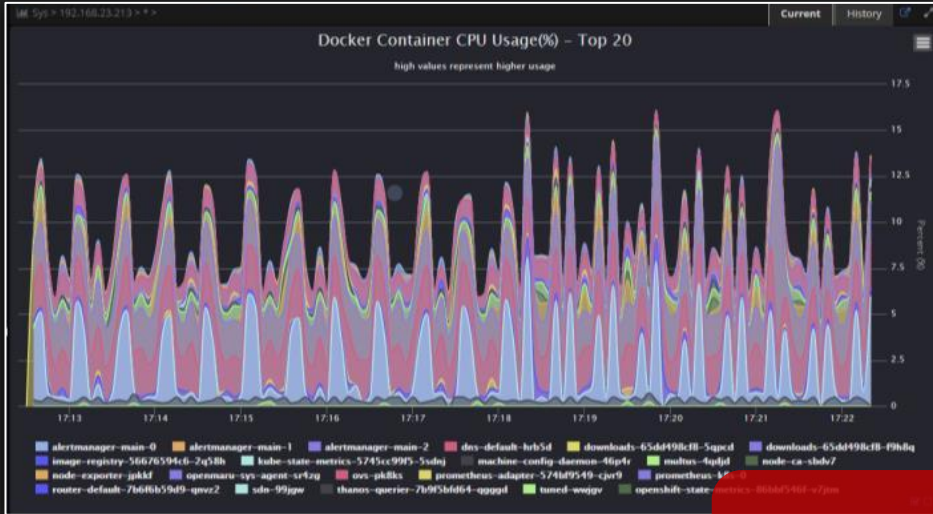
```

1 {
2   "container": {
3     "ociVersion": "1.0.1-dev",
4     "process": {
5       "user": {
6         "uid": 1000410000,
7         "gid": 0,
8         "additionalGids": [
9           1000410000
10        ]
11      },
12      "args": [
13        "sh",
14        "-c",
15        "update-ca-trust && exec \"$@\"",
16        "arg0",
17        "/usr/bin/dockerregistry"
18      ],
19      "env": [
20        "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
21        "TERM=xterm",
22        "HOSTNAME=image-registry-56676594c6-2q58h",
23        "REGISTRY_HTTP_SECRET",
24        "=d248ba9971fb8a15669128abb7a86c7896811d977eaf8386f86a7532c1d3209ce04dd808c12af8d0d15409be4b2ef96f38",
25        "REGISTRY_STORAGE_FILESYSTEM_ROOTDIRECTORY=/registry",
26        "REGISTRY_HTTP_ADDR=5000",
27        "REGISTRY_HTTP_NET=tcp",
28        "REGISTRY_STORAGE_CACHE_BLOBDESCRIPTOR=inmemory",
29        "REGISTRY_STORAGE_DELETE_ENABLED=true",
30        "REGISTRY_OPENSIFT_METRICS_ENABLED=true",
31        "REGISTRY_OPENSIFT_SERVER_ADDR=image-registry.openshift-image-registry.svc:5000",
32        "REGISTRY_HTTP_TLS_CERTIFICATE=/etc/secrets/tls.crt",
33        "REGISTRY_STORAGE=filesystem",
34        "REGISTRY_LOG_LEVEL=info",
35        "REGISTRY_OPENSIFT_QUOTA_ENABLED=true",
36        "REGISTRY_HTTP_TLS_KEY=/etc/secrets/tls.key",
37        "IMAGE_REGISTRY_SERVICE_HOST=172.30.162.0",
38        "IMAGE_REGISTRY_SERVICE_PORT=5000",
39        "IMAGE_REGISTRY_SERVICE_PORT_5000_TCP=5000",
40        "KUBERNETES_SERVICE_HOST=172.30.0.1",
41        "KUBERNETES_PORT_TCP=/172.30.0.1:443",
42        "KUBERNETES_PORT_443_TCP=/tcp://172.30.0.1:443",
43        "KUBERNETES_PORT_443_TCP_PROTO=tcp",
44        "KUBERNETES_PORT_443_TCP_ADDR=172.30.0.1",
45        "IMAGE_REGISTRY_PORT_5000_TCP_PROTO=tcp"
46      ]
47    }
48  }

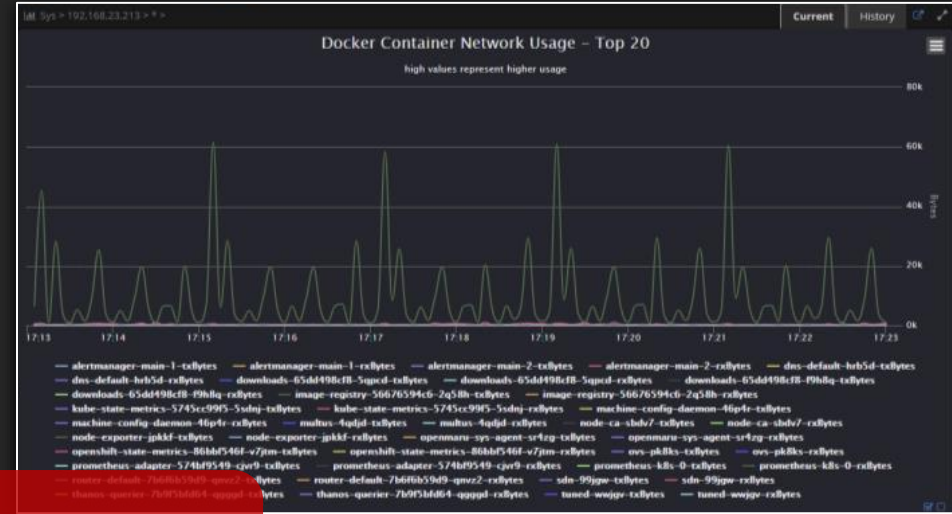
```

Container 정보

OPENMARU APM – CRI-O 컨테이너 사용량 모니터링

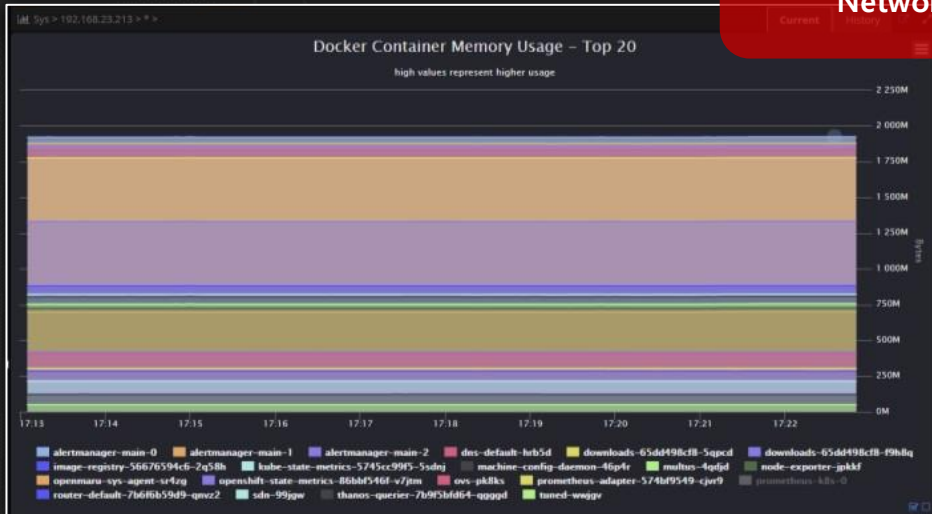


Container CPU Usage

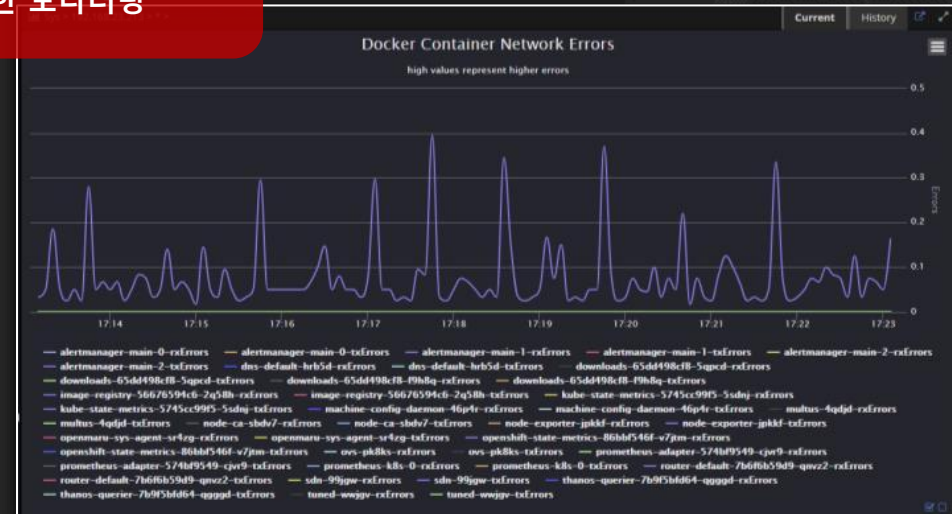


OpenShift 4 버전의 컨테이너 별 CPU, Memory, Container Network Usage

Network 사용량에 대한 모니터링



Container Memory Usage



Container Network Errors

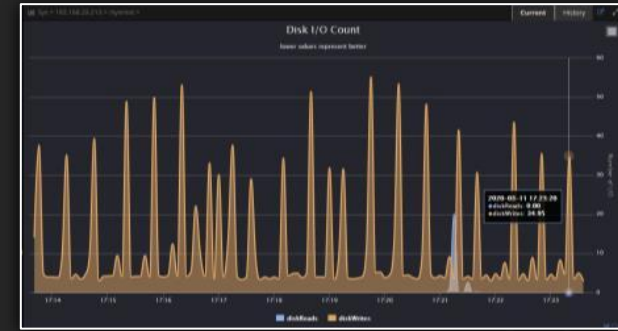
OPENMARU APM – CoreOS 호스트 모니터링



CoreOS 호스트 CPU, Memory, Disk

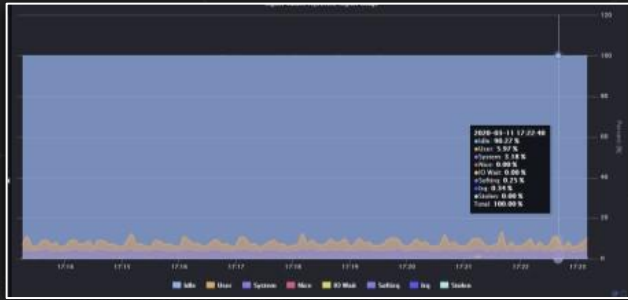


CoreOS CPU 평균 로드



Disk I/O Count

OpenShift 4 버전의 CoreOS 및 RHEL Worker의 CPU, Memory, Disk, Network 등 호스트 모니터링



CPU Usage



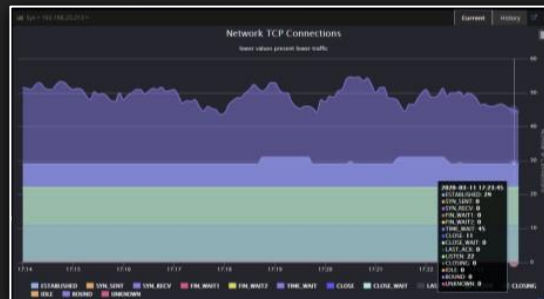
CoreOS Disk Usage



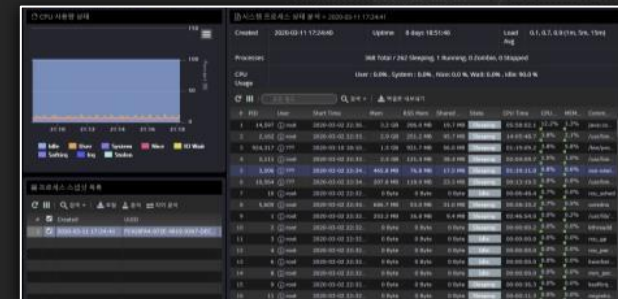
CoreOS 네트워크 상태분석



Memory Usage



Network TCP Connections



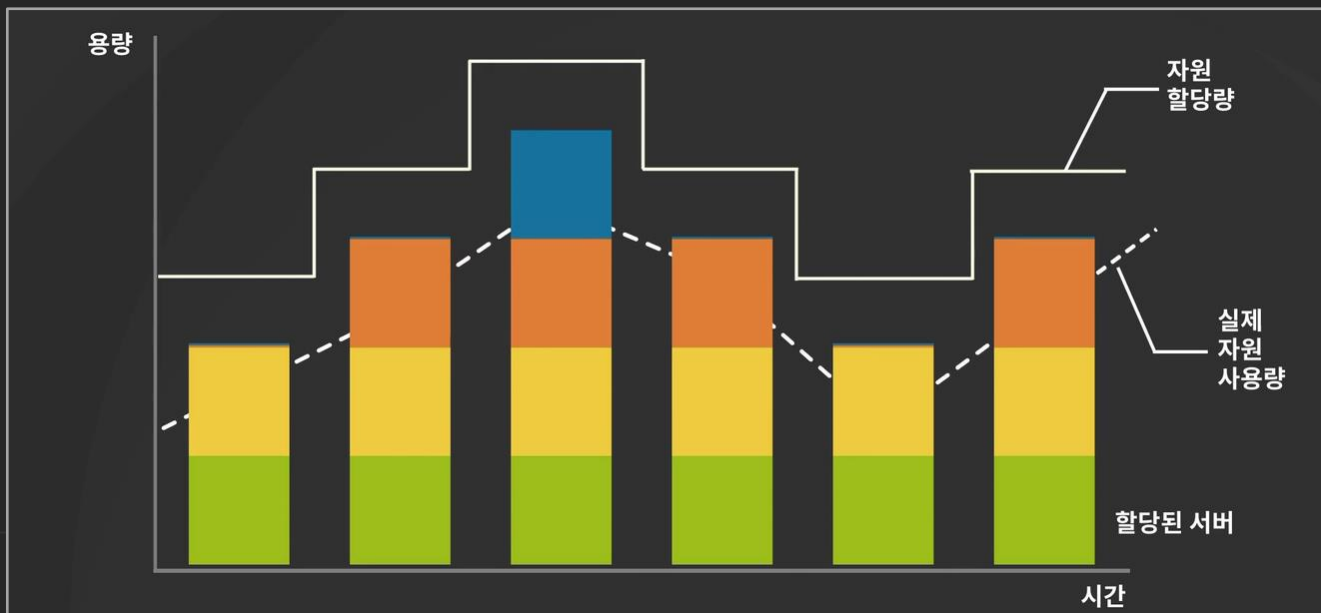
CoreOS 프로세스 상태 분석



Application Performance Management

PaaS 클라우드 미터링

서비스별 사용량 측정 기준



OPENMARU APM tachometer

실제 자원 CPU, 메모리 사용량을 기준으로
과금 계산함

대분류	중분류	사용량 측정 기준
소프트웨어서비스 (SaaS)	애플리케이션 서비스	• 사용자 수/ 트랜잭션 량
플랫폼 서비스 (PaaS)	엔터프라이즈 플랫폼	• 플랫폼 자원(Platform Resource)량 /트랜잭션 량
	호스팅 플랫폼 서비스	• 플랫폼 자원량/ 트랜잭션 량/데이터 크기
인프라 서비스 (IaaS)	서버 서비스	• 시스템 자원 양: CPU 코어(core)수, CPU Time, 메모리 양
	스토리지 서비스	• 스토리지 할당량 (디스크 할당량)
	백업 서비스	• 백업량 (백업 데이터 양)
	네트워크 서비스	• 포트 수/대역 폭

참고문헌 : 공공부문 클라우드 서비스 미터링 체계

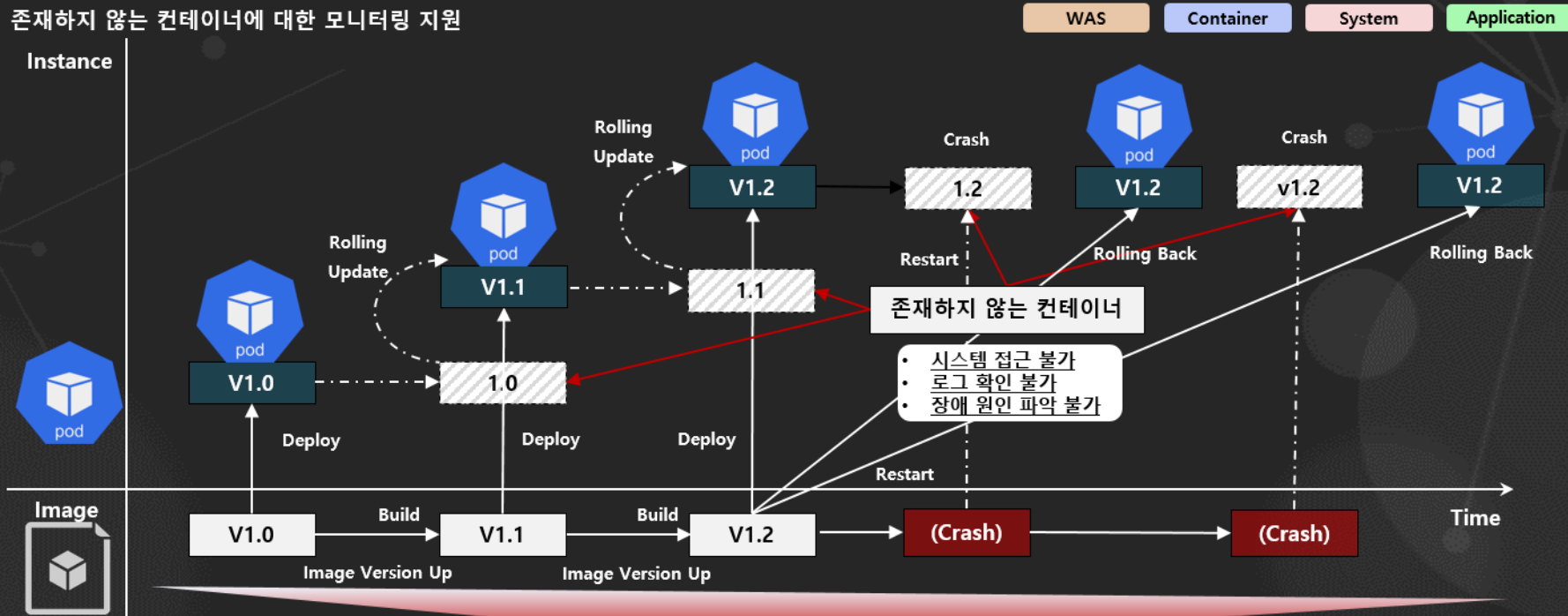
비상태 WAS 인스턴스에 대한 모니터링 기능

PaaS 환경의 비상태 WAS 인스턴스에 대한 모니터링 기능



- APM은 PaaS 환경에서 존재하지 않는 컨테이너가 중지된 후 장애원인 파악을 위한 정보를 파악할 수 있는 방법을 제공해야 한다.
- APM에서는 존재하지 않는 컨테이너에 대한 정보를 보관하고 있어 장애원인을 파악하여 정확한 조치를 취할 수 있다.

▶ 존재하지 않는 컨테이너에 대한 모니터링 지원



존재하지 않는 컨테이너

- 시스템 접근 불가
- 로그 확인 불가
- 장애 원인 파악 불가

실시간 데이터 수집
비상태 정보 기록
OPENMARU APM Server



Web Console

비상태 인스턴스 조회가능
사라진 장애 시점의 데이터 조회가능
인스턴스 장애 원인 파악가능

휘발성 인스턴스 모니터링 기록 추적 기능

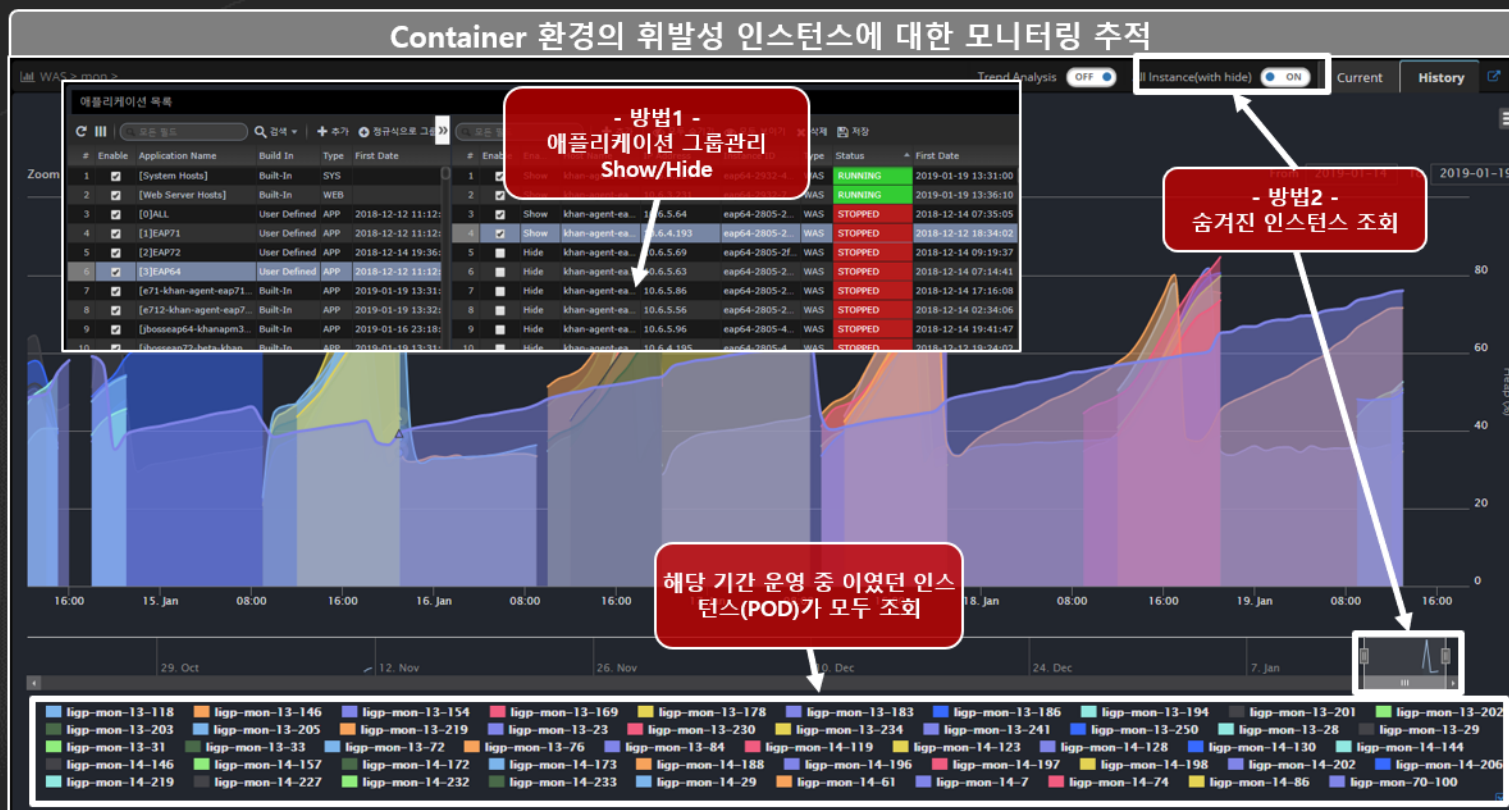
중지된 컨테이너에 대한 추적 기능



- Container 환경에서 중지되면 로그 분석 및 Container 장애 원인 추적이 불가능하다.
- APM에서는 휘발성 인스턴스의 모든 기록을 남겨놓고 있어, 과거의 Container 데이터를 조회하여 모니터링 기록을 추적, 장애 원인파악이 가능하다.

▶ Container 환경에서의 휘발성 인스턴스에 대한 모니터링 추적(홈 > WAS > 애플리케이션 그룹 관리)

WAS Container System Application



정산 금액 출력화면



Administrator

WAS

Web

System

Container

- 그룹1
- dev
- 월별 정산 통계**
- 도커 CPU
- 도커 메모리
- 도커 트래픽
- 도커

CUBRID

헬스체크

이벤트

보고서

설정

Home / Container / dev / 월별 정산 통계

Date: 2019 08 검색

년/월을 선택하고 검색을 클릭

일일 비용이 표시됨

일일 비용이 표시됨

정산 대상 컨테이너 그룹 목록이 표시됨

월별 정산 통계 상세 >

총 가격	860,600 (CPU: 199,400, MEM: 661,200)										가격 정책		정산일	말일 (2019-08-01 ~ 2019-08-31)								프로젝트 페딩	^dev-dk.* 목록
	08-15	08-16	08-17	08-18	08-19	08-20	08-21	08-22	08-23	08-24	08-25	08-26	08-27	08-28	08-29	08-30	08-31						
일 비용합계	0	0	0	14,100	38,400	38,700	69,000	103,800	110,400	88,500	48,000	48,000	49,700	50,400	50,400	50,400	50,400						
CPU 비용	0	0	0	2,700	7,200	7,200	19,200	31,200	31,200	23,900	9,600	9,600	9,600	9,600	9,600	9,600	9,600						
Memory 비용	0	0	0	11,400	31,200	31,500	49,800	72,600	79,200	64,600	38,400	38,400	40,100	40,800	40,800	40,800	40,800						
컨테이너 수	0	0	0	3	3	3	13	13	13	26	4	4	4	4	4	4	4						
총 사용시간	:00	00:00:00	00:00:00	27:06:00	71:55:30	71:58:30	186:09:20	311:51:20	312:00:00	236:02:50	96:00:00	96:00:00	95:57:20	96:00:00	95:57:30	96:00:00	96:00:00						
Avg CPU Usage	%00	0.0000	0.0000	0.0036	0.0036	0.0035	0.0039	0.0031	0.0030	0.0033	0.0035	0.0035	0.0035	0.0034	0.0035	0.0034	0.0034						
Avg MEM Usage	yte	0 Byte	0 Byte	613.8 MB	632.2 MB	658.0 MB	678.1 MB	704.4 MB	730.3 MB	719.9 MB	741.9 MB	758.3 MB	772.2 MB	778.1 MB	786.4 MB	797.5 MB	807.3 MB						
Details	14	08-15	08-16	08-17	08-18	08-19	08-20	08-21	08-22	08-23	08-24	08-25	08-26	08-27	08-28	08-29	08-30	08-31					

Record ID: 1

1-8 of 8

정산 대상 컨테이너 목록 확인(과금 월별)

opennaru | 오픈나루 | 한국어 | [Icons]

홈 / Container / dev / 월별 정산 통계

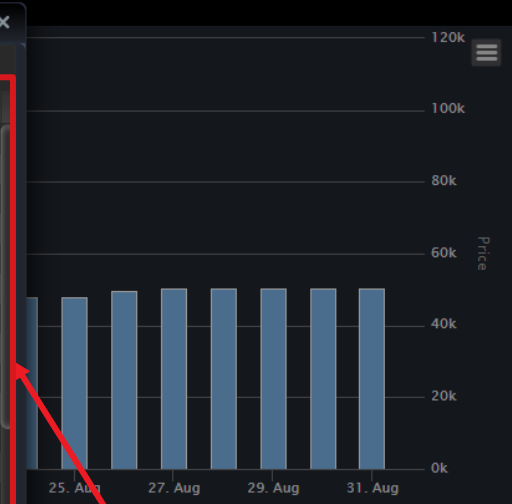
Date: 2019 08 | 검색 | 엑셀 다운로드

월별 컨테이너 목록 확인

#	Project	Container Na...	Start Date	End Date	Used Ti...	CPU Price	MEM Price	실행시간 CPU ...	실행시간 MEM...
1	dev-dk	httd-example...	2019-8-23 17:...	2019-8-23 17:...	00:00:20	0	0	0.0595	51.7 MB
2	dev-dk	s2i-tomcat8-8...	2019-8-20 12:...	2019-8-23 15:...	75:11:00	7600	14900	0.0031	247.1 MB
3	dev-dk	s2i-tomcat8-8...	2019-8-17 14:...	2019-9-1 00:00	344:57:20	34500	69000	0.0026	237.8 MB
4	dev-dk	s2i-tomcat8-8...	2019-8-20 12:...	2019-8-23 15:...	75:10:50	7600	7500	0.0031	210.1 MB
5	dev-dk	s2i-tomcat8-8...	2019-8-20 12:...	2019-8-23 15:...	75:11:00	7600	7500	0.0031	211.8 MB
6	dev-dk	s2i-tomcat8-8...	2019-8-20 12:...	2019-8-23 15:...	75:11:00	7600	14800	0.0031	218.0 MB
7	dev-dk	s2i-tomcat8-8...	2019-8-20 12:...	2019-8-23 15:...	75:11:00	7600	14900	0.0031	235.7 MB
8	dev-dk	s2i-tomcat8-8...	2019-8-20 12:...	2019-8-23 15:...	75:11:00	7600	12400	0.0031	213.9 MB
9	dev-dk	s2i-tomcat8-8...	2019-8-20 12:...	2019-8-23 15:...	75:11:00	7600	14700	0.0031	216.4 MB
10	dev-dk	s2i-tomcat8-8...	2019-8-20 12:...	2019-8-23 15:...	75:10:50	7600	12400	0.0031	215.4 MB
11	dev-dk	s2i-tomcat8-8...	2019-8-20 12:...	2019-8-23 15:...	75:11:00	7600	12400	0.0031	214.5 MB
12	dev-dk	s2i-tomcat8-8...	2019-8-20 12:...	2019-8-23 15:...	75:11:00	7600	14900	0.0031	227.0 MB
13	dev-dk	test1-1-build	2019-8-23 16:...	2019-8-23 16:...	00:01:50	0	0	0.0128	73.0 MB
14	dev-dk	test1-app-1-b...	2019-8-23 17:...	2019-8-23 17:...	00:00:50	0	0	0.0266	79.2 MB
15	dev-dk	test1-app-1-d...	2019-8-23 17:...	2019-8-23 17:...	00:00:20	0	0	0.0365	64.3 MB
16	dev-dk	test1-app-1-bx...	2019-8-23 17:...	2019-8-23 17:...	00:07:50	100	0	0.0624	221.6 MB

Close

프로젝트 페턴 ^dev-dk* 목록



월 별	08-26	08-27	08-28	08-29	08-30	08-31
CPU 비용	0	0	0	2,700	7,200	7,200
Memory 비용	0	0	0	11,400	31,200	31,500
컨테이너 수	0	0	0	3	3	3
총 사용시간	:00	00:00:00	00:00:00	27:06:00	71:55:30	71:58:30
Avg CPU Usage	100	0.0000	0.0000	0.0036	0.0036	0.0035
Avg MEM Usage	yte	0 Byte	0 Byte	613.8 MB	632.2 MB	658.0 MB
Details	14	08-15	08-16	08-17	08-18	08-19

Record ID: 1 | 1-8 of 8

정산 대상 컨테이너 목록 확인(과금 일별)

opennaru | 오픈나루 | 한국어

홈 / Container / dev / 월별 정산 통계

Date: 2019 08 검색 액셀 다운로드

일별 사용 컨테이너 목록 확인

#	Project	Container Name	Start Date	End Date	Used Ti...	CPU Price	ME...	실행시간 CPU ...	실행시간 MEM...
1	dev-dk	httpd-example-1-b...	2019-8-23 17:38	2019-8-23 17:38	00:00:20	0	0	0.0595	51.7 MB
2	dev-dk	s2i-tomcat8-8-5pt...	2019-8-23 00:00	2019-8-23 15:46	15:46:30	1600	3100	0.0029	247.5 MB
3	dev-dk	s2i-tomcat8-8-cq5...	2019-8-23 00:00	2019-8-23 23:59	24:00:00	2400	4800	0.0026	241.9 MB
4	dev-dk	s2i-tomcat8-8-fbnhs	2019-8-23 00:00	2019-8-23 15:46	15:46:30	1600	1600	0.0030	211.6 MB
5	dev-dk	s2i-tomcat8-8-hxrpr	2019-8-23 00:00	2019-8-23 15:46	15:46:30	1600	1600	0.0029	213.0 MB
6	dev-dk	s2i-tomcat8-8-ppsp...	2019-8-23 00:00	2019-8-23 15:46	15:46:30	1600	3100	0.0028	218.7 MB
7	dev-dk	s2i-tomcat8-8-mcjr	2019-8-23 00:00	2019-8-23 15:46	15:46:30	1600	3100	0.0028	236.4 MB
8	dev-dk	s2i-tomcat8-8-rs5rh	2019-8-23 00:00	2019-8-23 15:46	15:46:30	1600	3100	0.0029	215.0 MB
9	dev-dk	s2i-tomcat8-8-s6gpr	2019-8-23 00:00	2019-8-23 15:46	15:46:30	1600	3100	0.0029	217.7 MB
10	dev-dk	s2i-tomcat8-8-sft2w	2019-8-23 00:00	2019-8-23 15:46	15:46:30	1600	3100	0.0029	216.7 MB
11	dev-dk	s2i-tomcat8-8-slbjq	2019-8-23 00:00	2019-8-23 15:46	15:46:30	1600	3100	0.0029	215.2 MB
12	dev-dk	s2i-tomcat8-8-vxstq	2019-8-23 00:00	2019-8-23 15:46	15:46:30	1600	3100	0.0030	227.9 MB
13	dev-dk	test1-1-build	2019-8-23 16:56	2019-8-23 16:58	00:01:50	0	0	0.0128	73.0 MB
14	dev-dk	test1-app-1-build	2019-8-23 17:07	2019-8-23 17:08	00:00:50	0	0	0.0266	79.2 MB
15	dev-dk	test1-app-1-deploy	2019-8-23 17:08	2019-8-23 17:08	00:00:20	0	0	0.0365	64.3 MB
16	dev-dk	test1-app-1-lx76t	2019-8-23 17:08	2019-8-23 17:16	00:07:50	100	0	0.0624	221.6 MB

Close

총 가격

일	8-26	8-27	8-28	8-29	8-30	8-31
CPU 비용	0	0	0	2,700	7,200	7,200
Memory 비용	0	0	0	11,400	31,200	31,500
컨테이너 수	0	0	0	3	3	3
총 사용시간	:00	00:00:00	00:00:00	27:06:00	71:55:30	71:58:30
Avg CPU Usage	100	0.0000	0.0000	0.0036	0.0036	0.0035
Avg MEM Usage	yte	0 Byte	0 Byte	613.8 MB	632.2 MB	658.0 MB

프로젝트 패턴 ^dev-dk* 목록

일	8-26	8-27	8-28	8-29	8-30	8-31
49,700	50,400	50,400	50,400	50,400	50,400	50,400

월별 정산 통계

일별 정산

일	8-26	8-27	8-28	8-29	8-30	8-31
CPU 비용	0	0	0	2,700	7,200	7,200
Memory 비용	0	0	0	11,400	31,200	31,500
컨테이너 수	0	0	0	3	3	3
총 사용시간	:00	00:00:00	00:00:00	27:06:00	71:55:30	71:58:30
Avg CPU Usage	100	0.0000	0.0000	0.0036	0.0036	0.0035
Avg MEM Usage	yte	0 Byte	0 Byte	613.8 MB	632.2 MB	658.0 MB

Details: [14](#) [08-15](#) [08-16](#) [08-17](#) [08-18](#) [08-19](#) [08-20](#) [08-21](#) [08-22](#) [08-23](#) [08-24](#) [08-25](#) [08-26](#) [08-27](#) [08-28](#) [08-29](#) [08-30](#)

정산 데이터 엑셀 다운로드



Administrator

Home / Container / 그룹1 / 월별 정산 통계

Date: 2019 08 검색 엑셀 다운로드

월별 정산 통계

자동 저장 | bill_2019-08_그룹1 (2)... | 11500

그룹1 월 비용 보고서

기간	2019-08-01 ~ 2019-08-31
총비용	12,500
총CPU비용	1,000
총메모리비용	11,500

그룹1 월 비용 보고서 상세 (1/2)

항목	2019 08 01	2019 08 02	2019 08 03	2019 08 04	2019 08 05	2019 08 06	2019 08 07	2019 08 08	2019 08 09	2019 08 10
일비용합계	1,700	1,800	1,800	1,900	1,700	400	700	800	800	900
CPU비용	100	100	100	100	100	100	100	100	100	100
메모리비용	1,600	1,700	1,700	1,800	1,600	300	600	700	700	800
컨테이너수	3	3	3	3	3	10	3	3	3	3

Record ID: 1

정산 데이터 엑셀 다운로드

Application Performance Management

대규모 클라우드에 최적화

대규모 모니터링을 위한 그룹 대시보드 기능



- ‘그룹별 대시보드’**
- 개별 인스턴스들을 수개 ~ 수 십개 단위의 그룹으로 묶어 그룹별 실시간 서비스 지연 현황 확인
 - 그룹별 TPS, 평균 응답시간, 액티브 사용자수 확인 가능
 - WAS 프로세스 현황 확인 가능
 - N개 그룹에 대한 전체 모니터링 가능함
 - 링크를 클릭하여 상세 화면 바로 이동

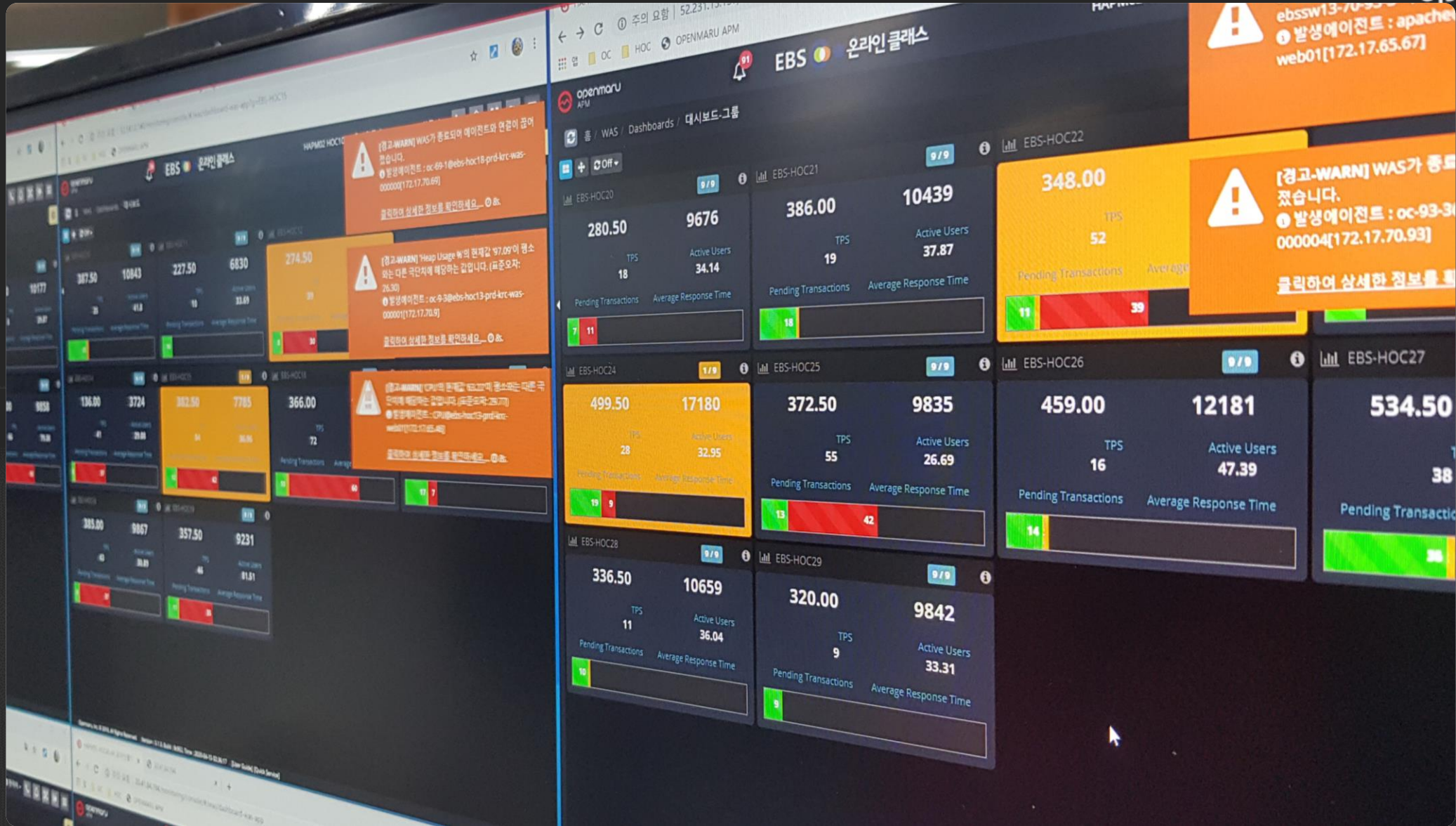
기관 그룹명

그룹 총 동시 접속자

그룹 총 평균 응답시간

전체 지연 큐 그래프

코로나로 인한 온라인 개학 – EBS 온라인클래스, KERIS 디지털교과서 표준 APM



MSA 환경 호출 추적 기능



Transaction for 172312e142b-313b4a6276bbbe36bac5

Agent	IP address	Instance ID	URL	Status	Duration(ms)	SQL Time(ms)	Fetch Gab	Fetch Count
WAS	10.131.1.207	iws52-10-mddtv	/testapp/get.jsp	200	2,417	0	0	0
WAS	10.131.1.208	eap72-egov-24-	└ /EgovTop.do	404	0	0	0	0
WAS	10.131.1.208	eap72-egov-24-	└ /EgovLeft.do	404	0	0	0	0
WAS	10.131.1.208	eap72-egov-24-	└ /EgovContent.do	404	0	0	0	0

MAS 환경에서 HTTP 호출 관계 추적

Transaction Detail | Trace | StackTrace

- HTTP Response Code Error - 404 : GET http://test-eap72-egov2.test1.svc:8080/EgovTop.do null
- HTTP Response Code Error - 404 : GET http://test-eap72-egov2.test1.svc:8080/EgovLeft.do null
- HTTP Response Code Error - 404 : GET http://test-eap72-egov2.test1.svc:8080/EgovContent.do null

Method Traces

- Legend : ■ 0 <= time < 1,000ms ■ 1,000ms <= time < 3,000ms ■ 3,000ms <= time

```

=====
[Num.] [ Start Time | Elapsed | % | Exclusi | A-Gab | CPUTime ]          Method Call
-----
[ 1][17:23:38.563] 2,417| 100|    0|    0|    3.7|+ com.openmaru.khan.test.filter.CommonEncodingFilter.doFilter()
[ 2][17:23:38.563] 2,417| 100|    0| 2,417|    3.6| + org.apache.jasper.servlet.JspServlet.service()
[ 3][17:23:38.563] 2,417| 100|    0| 2,417|    3.6| + org.apache.jasper.servlet.JspServletWrapper.service()
[ 4][17:23:38.563] 2,417| 100| 2,412| 5|    3.5| + org.apache.jsp.get_jsp._jspService()
[ 5][17:23:38.563] 2| 0|    2| 0|    1.1| + org.apache.commons.httpclient.HttpClient.executeMethod()
> Infos:
  GET http://test-eap72-egov2.test1.svc:8080/EgovTop.do 404
  Child Transaction Detail
[ 6][17:23:40.977] 2| 0|    2| 0|    0.9| + org.apache.commons.httpclient.HttpClient.executeMethod()
> Infos:
  GET http://test-eap72-egov2.test1.svc:8080/EgovLeft.do 404
  Child Transaction Detail
[ 7][17:23:40.979] 1| 0|    1| 0|    0.6| + org.apache.commons.httpclient.HttpClient.executeMethod()
> Infos:
  GET http://test-eap72-egov2.test1.svc:8080/EgovContent.do 404
  Child Transaction Detail
=====
    
```



openmaru

제품 / 서비스에 관한 문의

- 콜 센터 : 02-469-5426 (휴대폰 : 010-2243-3394)
- 전자 메일 : sales@openmaru.com