



클라우드 기반 통합 관제 솔루션

InterMaxcloud

인터맥스 클라우드 제품소개서 ver 0.32

exem

The Next Revolution is Here

cloud

01

클라우드 환경 특징

-
1. 클라우드 개요/특징
 2. 모니터링 필요성

02

제품개요 및 특징점

-
1. 제품 개요
 2. 아키텍처
 3. 특징점

03

제품 기능 소개

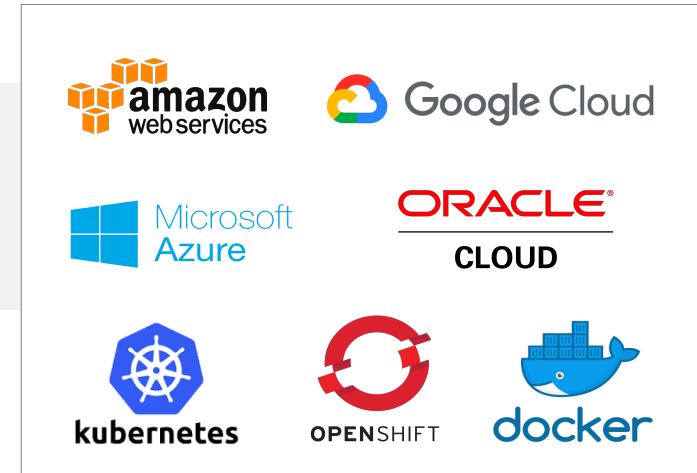
-
1. 실시간 대쉬보드
 2. 분석 화면
 3. 환경설정

다양한 클라우드 컴퓨팅 플랫폼 / 서비스 탄력성과 확장성

다양한 클라우드 컴퓨팅 플랫폼

- **Public Cloud** : AWS, Microsoft Azure, Google Cloud, Oracle Cloud Infra...
- **Private Cloud** : OpenStack, Kubernetes, Openshift, Docker...
- **Hybrid Cloud** : Public Cloud + Private Cloud
- **Multi Cloud** : 복수의 Public Cloud

- | | |
|----------|---|
| 이슈 | <ul style="list-style-type: none"> • Hybrid 클라우드 구성에 대한 On-Premise 및 Cloud 환경에 대한 All-in-One 통합 모니터링 • 하이브리드/멀티 클라우드, 멀티 클러스터 통합 모니터링과 Single View 제공 |
| 모니터링 트렌드 | <ul style="list-style-type: none"> • 단일 관리 플랫폼에 의한 이종의 클라우드 인프라(IaaS, SaaS, PaaS) 하나된 모니터링 |

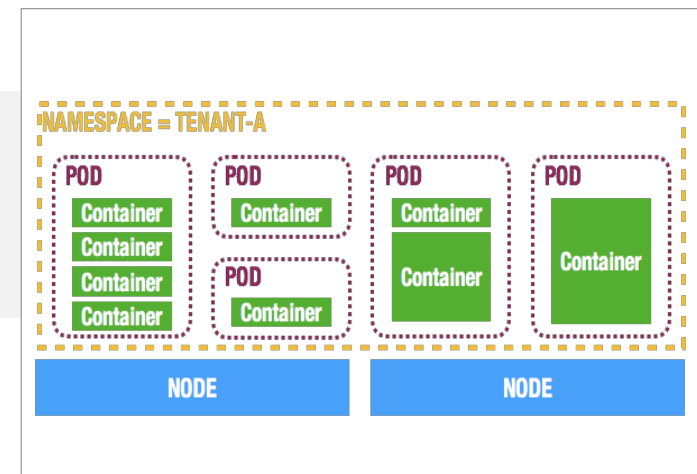


서비스 탄력성과 확장성

Rapid elasticity and scalability

- 컨테이너 기반 Auto Scale In / Out , Container Orchestration
- 모니터링 대상 그룹 및 인스턴스의 잦은 변동
- 워크로드 가상화와 자동화된 인프라 프로비저닝/스케줄링 (Terraform)
- VM, Pod, Container 등 가상 계층의 빠르고 잦은 할당/재할당 (탄력성)

- | | |
|----------|--|
| 이슈 | <ul style="list-style-type: none"> • 클라우드 전 계층에 대한 CPU, 메모리, 네트워크, 스토리지 리소스 통합 모니터링 • 민첩한 워크로드 실행을 위한 컨테이너 단위 Auto Scaling, 쿠버네티스 모니터링 |
| 모니터링 트렌드 | <ul style="list-style-type: none"> • QoS 보장을 위한 Auto Scaling, Provisioning, Container Orchestration 실시간 모니터링 |



컨테이너 기반 클라우드 네이티브 아키텍처 / 효율적이지만 복잡한 MSA

컨테이너 기반 클라우드 네이티브 아키텍처

Cloud Native architecture

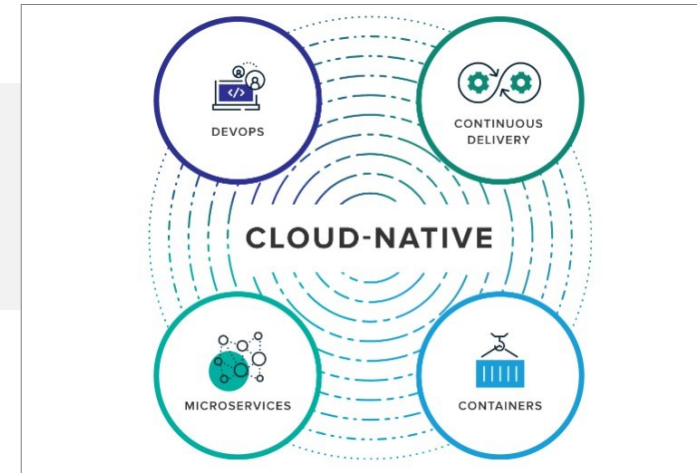
- 가상화, 컨테이너, 자동화, 오케스트레이션, 마이크로서비스 중심의 '클라우드 네이티브 인프라스트럭처 (Cloud Native-Infrastructure)'
- '클라우드 네이티브 인프라스트럭처' 기반의 재사용 가능한 독립적인 애플리케이션(Cloud Native- Application)

이슈

- Cloud Native 환경에서 Infra - 컨테이너 - App에 이르는 End-to-End 모니터링 한계
- 물리 노드에서 클라우드 애플리케이션 까지 클라우드 Full Stack 토폴로지 가시성 필요

모니터링
트렌드

- AI-powered, Cloud Native Full Stack에 대한 건강성 지표 자동 분석



효율적이지만 복잡한 MSA

Microservice Architecture

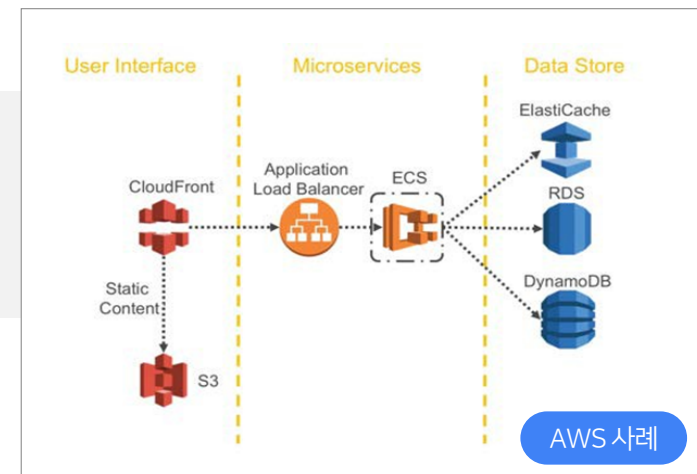
- 다양한 마이크로서비스 아키텍처 패턴 (현재, Serverless MSA로 진화 중)
- 수백 수천의 서비스 간 복잡한 호출 관계와 높은 상호 의존성 (Dependency)
- Restful API 기반 서비스 간 분산 트랜잭션
- 서비스 민첩성, 관리 용이성 (Container Orchestration)

이슈

- 중앙데이터베이스가 없어 각 서비스 별 로그에 의한 통합 모니터링 제약
- 서비스 간 높은 의존성에 따른 장애 추적 및 성능 모니터링의 어려움 (Blind Spot)

모니터링
트렌드

- AI-powered, 서비스 간 복잡한 상호 의존성을 고려한 이상탐지와 근본원인 자동 분석



01

클라우드 환경 특징

- 1. 클라우드 개요/특징
- 2. 모니터링 필요성

02

제품개요 및 특징점

- 1. 제품 개요
- 2. 아키텍처
- 3. 특징점

03

제품 기능 소개

- 1. 실시간 대쉬보드
- 2. 분석 화면
- 3. 환경설정

클라우드 환경 통합 성능관리 솔루션 - 핵심 기능

- **다차원 토폴로지**

Layer별 체계적 모니터링

- **AI 기반 Health Indicator**

이상탐지 및 신속한 이슈 감지

- **AI 기반 히스토그램 분석**

효과적 원인분석

- **멀티 클러스터 Single View**

클라우드 통합 모니터링

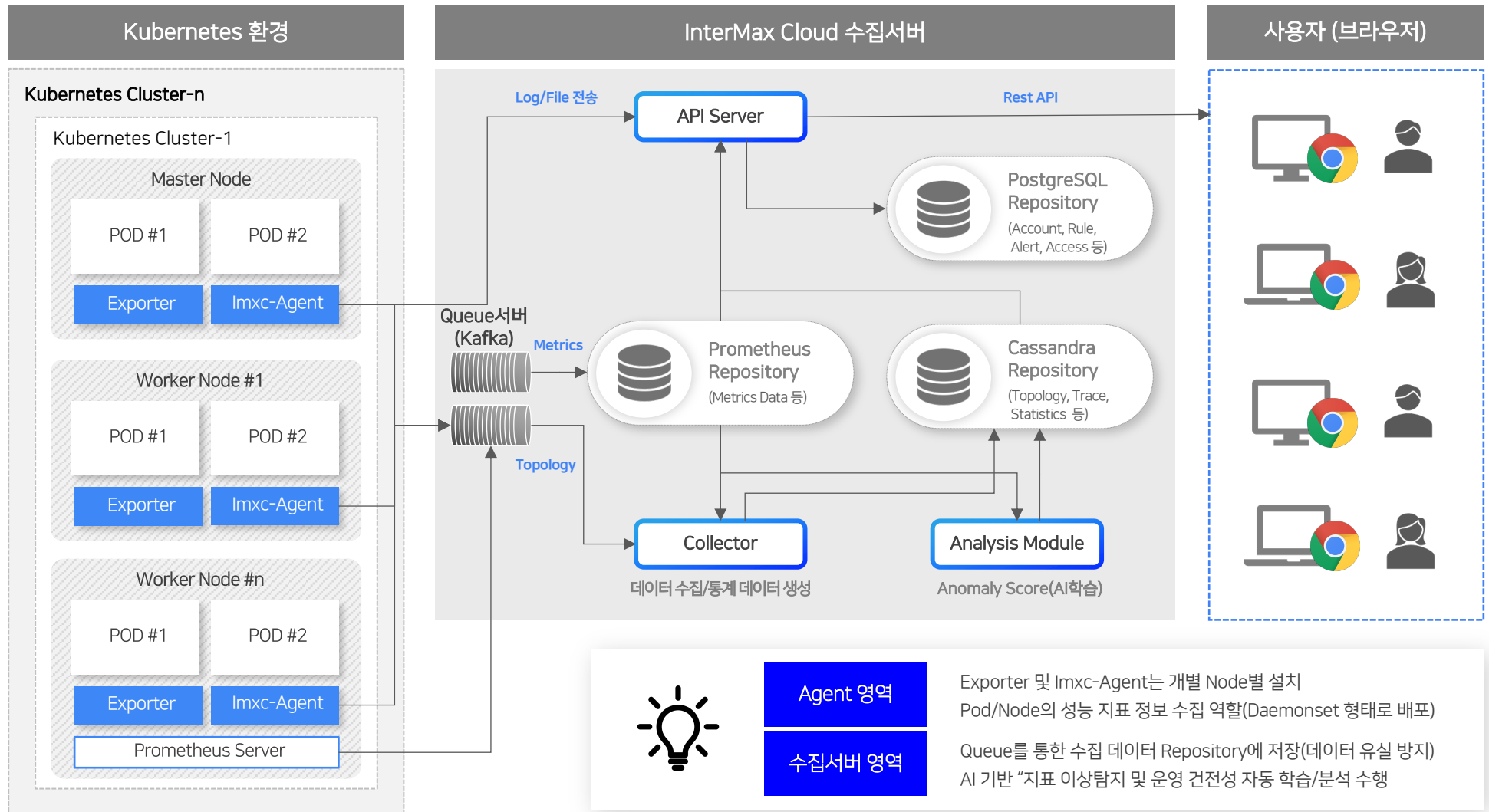
- **컨테이너 기반 Agent 설치**

손쉬운 설치 및 모니터링 대상 자동관리



대규모 클라우드 환경에 적합한 아키텍처 구성

설치 편의성, 장애방지, 확장성을 고려한 아키텍처 제공



클라우드 모니터링의 기능을 모두 담았습니다

클라우드 환경의 All-in-One 통합 모니터링

클라우드 환경의 대규모 관제에 효과적인 모니터링을 제공하여 운영 효율성을 극대화합니다.

클라우드 전체
Layer 가시성 확보

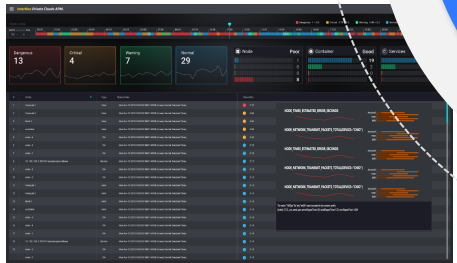


대규모 관제용
3D 아키텍처 토폴로지뷰

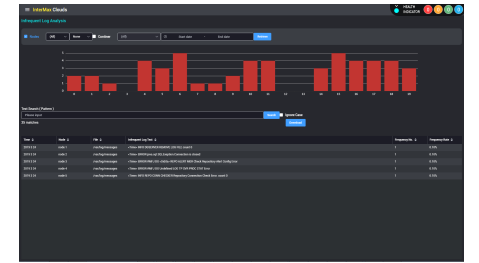


All-in-One

복잡한 환경을 관제하는
건강성지표



다수 컨테이너 관제에 필수적인
Smart 로그 분석



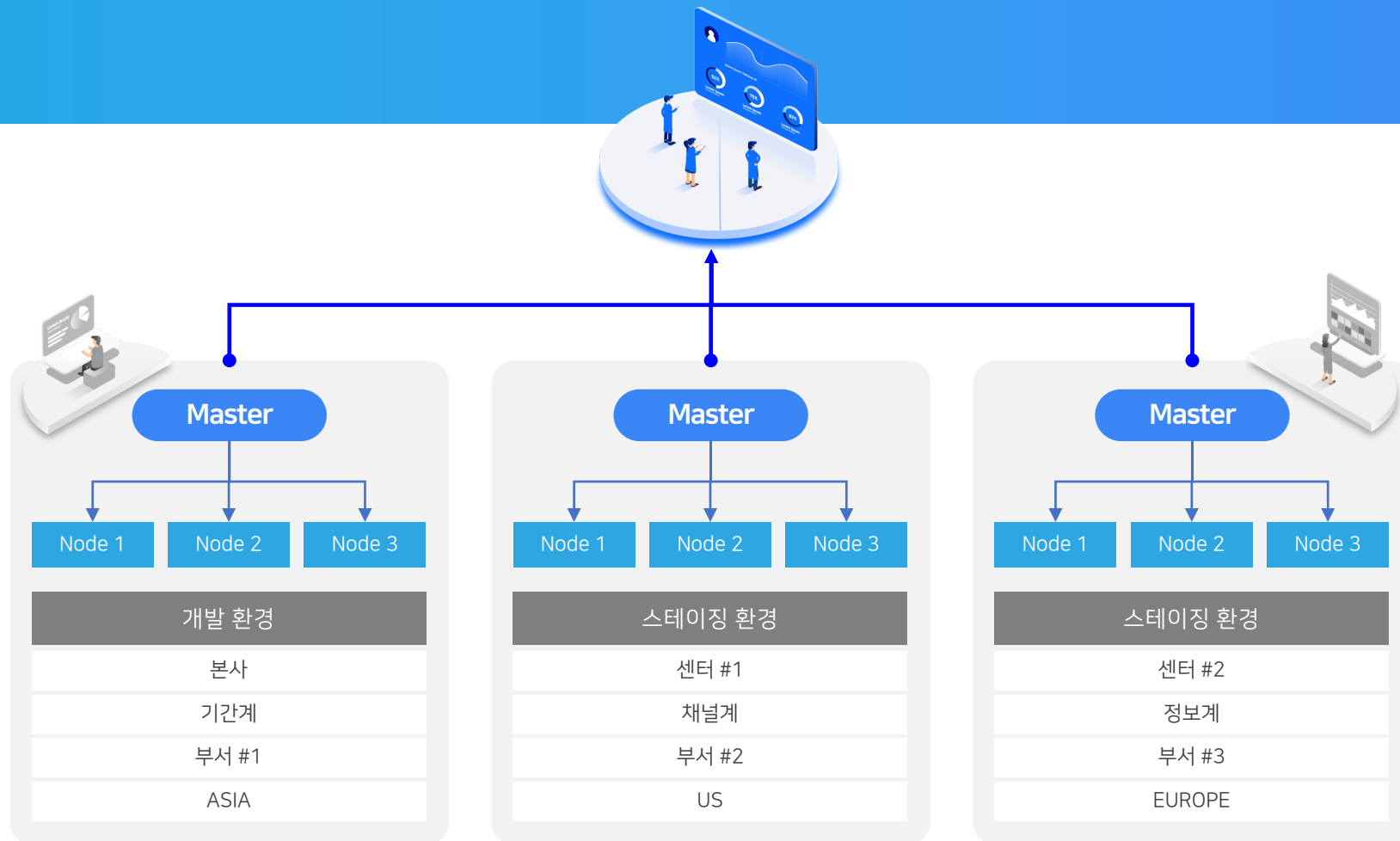
전체 서비스 Layer 가시성 확보

전체 서비스 Layer 가시성 확보 (클러스터,노드,Pod,컨테이너,어플리케이션)



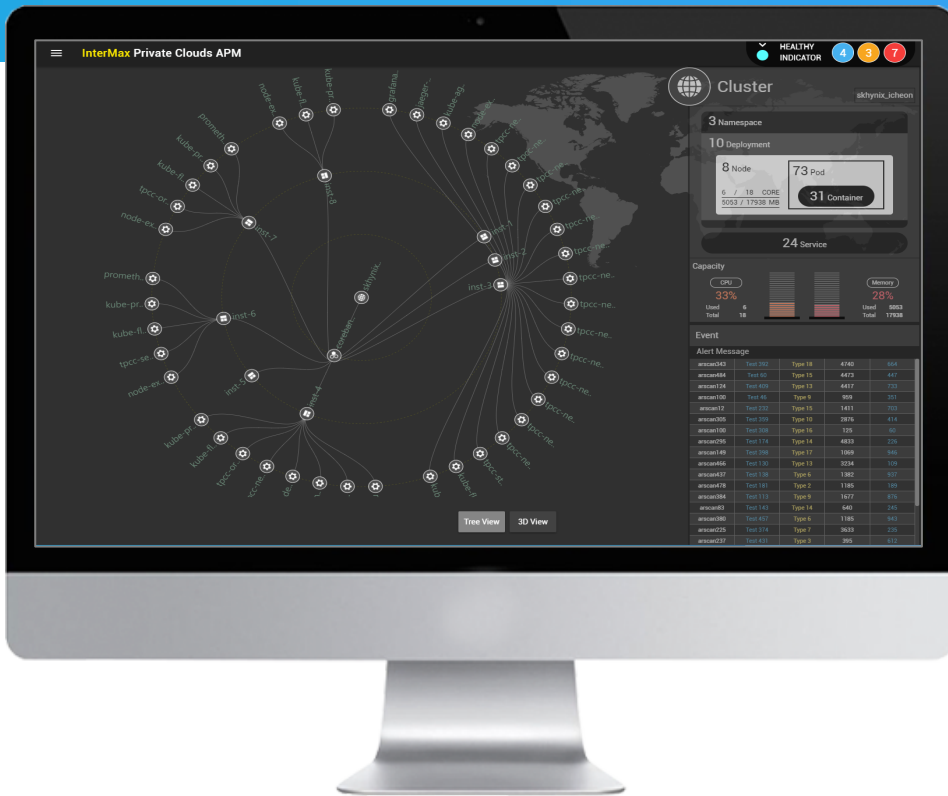
멀티 클러스터 Single View 제공

멀티 클러스터 Single View를 통한 전사 관점의 통합 관제 지원



Infra 기반 토폴로지뷰 제공

대규모 모니터링에 필수적인 2D / 3D 아키텍처 토폴로지뷰

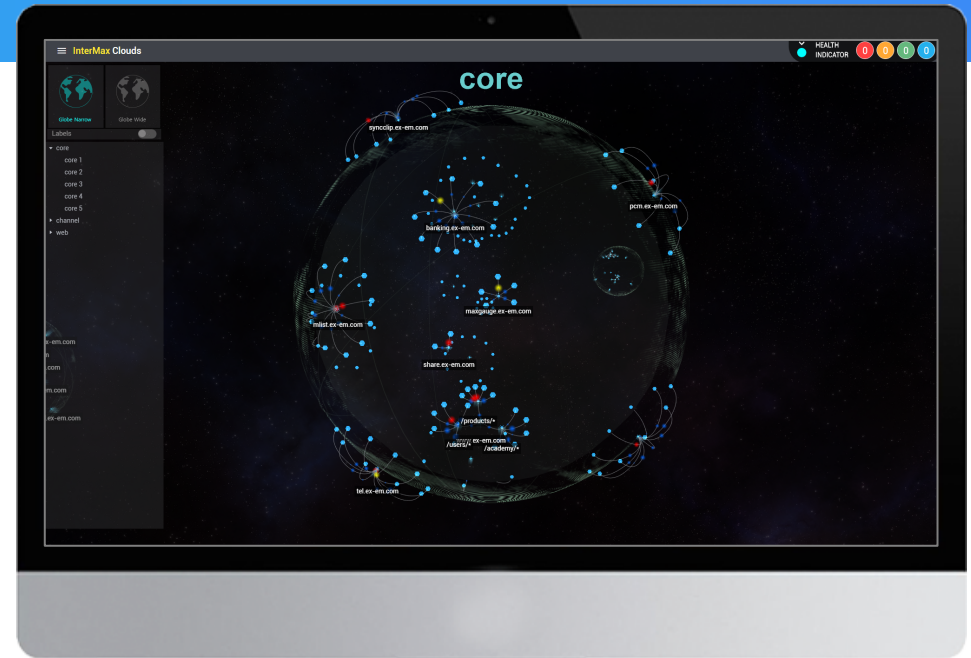


Services 기반 토폴로지뷰 제공

서비스의 연관 관계를 직관적으로 제공하는
3D 서비스 토폴로지뷰



전체 클러스터별 서비스 배치도



선택한 클러스터의 서비스 상세 배치도

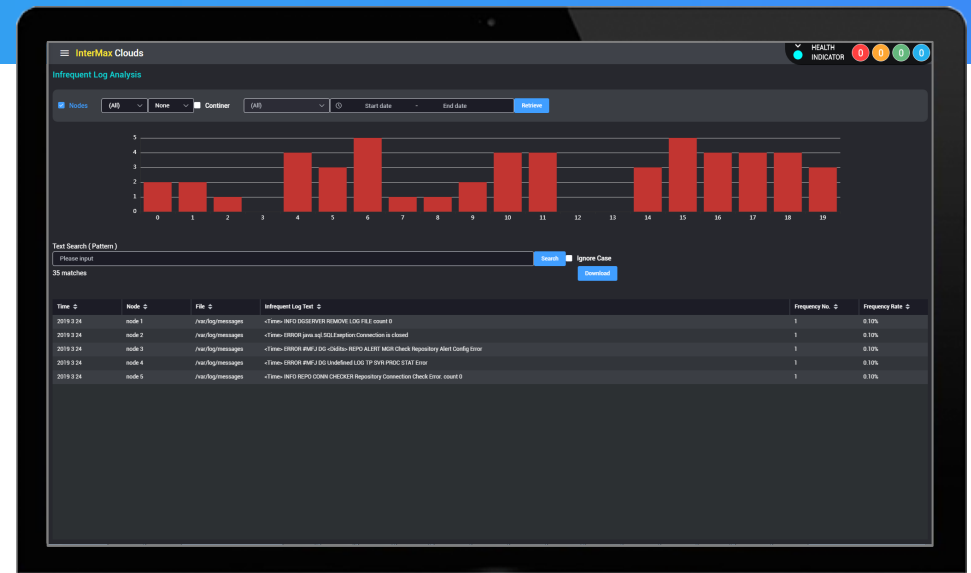
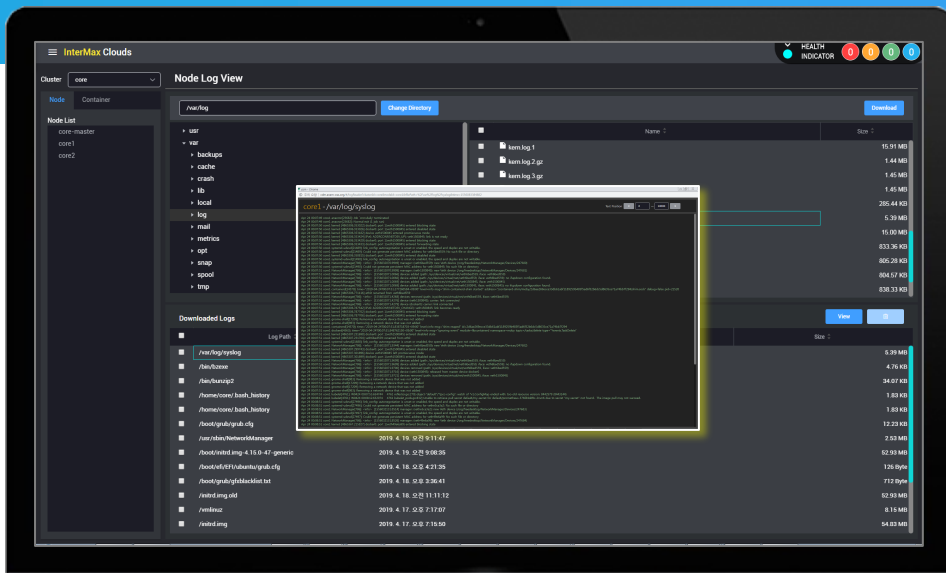
시인성 뛰어난 장애 인지 능력

시인성 뛰어난 **애니메이션 UI**를 통한 장애 발생 컨테이너 빠른 탐색과 인지



Machine Learning 기반의 효과적인 로그 데이터 분석 및 히소로그 탐지

다수 컨테이너 환경에 필수적인 **Smart 로그 분석 지원** (통합 Log View / 히소로그 분석)



혼한 로그는 이벤트를 클러스터링 하여 스마트하게 패턴과 변칙을 식별

- 수십만의 로그 이벤트를 패턴화 하여 로그를 줄임
- 문제를 빠르게 분석, 해결하고 변칙과 간헐적인 에러들을 쉽게 식별함

다양한 시스템별 로그 뷰어 제공

다양한 시스템별 히소 로그 분석 뷰

애플리케이션 상세 Tracing 분석을 위한 InterMax 연동 기능

컨테이너 환경에서 **진정한 End-to-End 모니터링**을 위한
APM 제품(InterMax)과 연동 지원

InterMaxcloud
Cloud Performance Maximizer

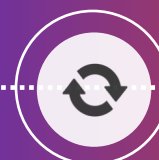


Cloud | Pod,Container,VM

InterMax
Application Performance Maximizer

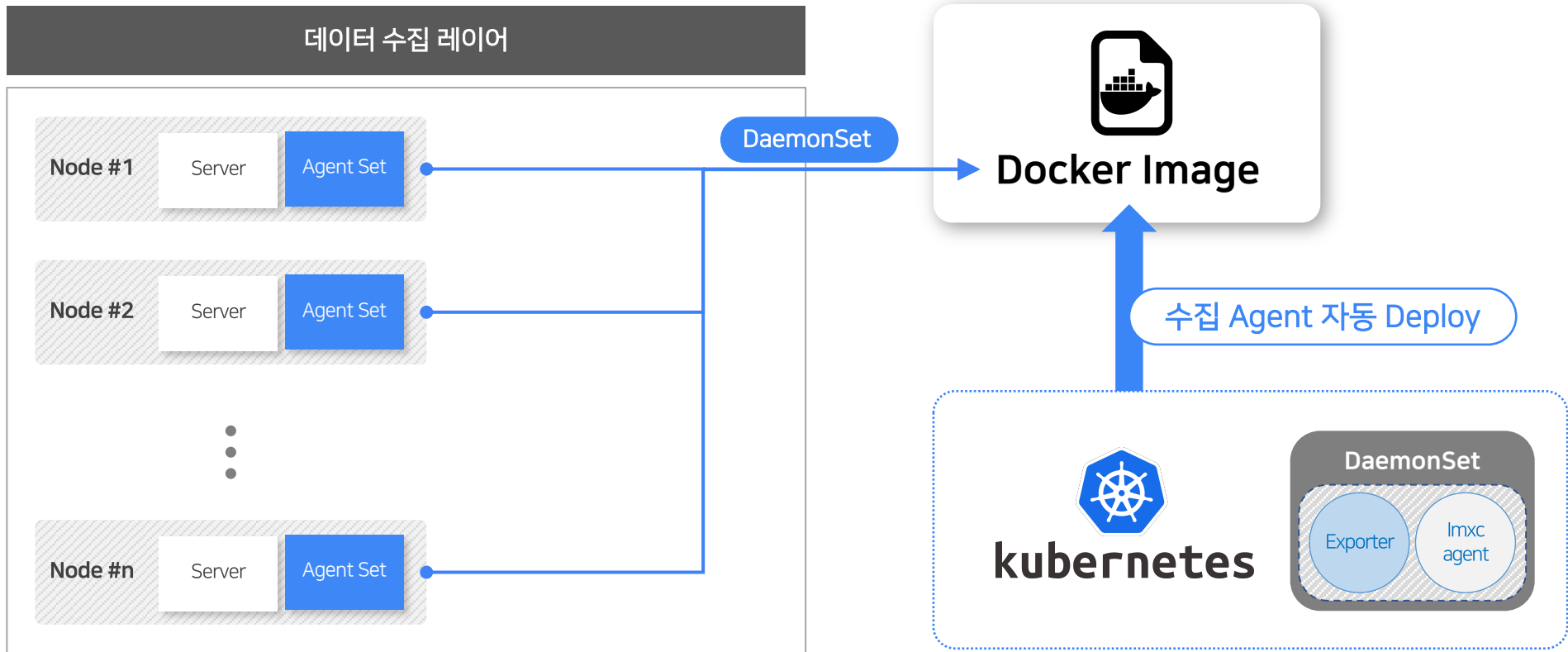


End-to-End | WEB,WAS,.NET,C



전사 시스템 관리 편의성 제공

수집 Agent 컨테이너 기반 자동설치 방식으로 전사 시스템 관리 편의성 제공



기대 효과

소수의 운영인력으로 대규모 클라우드 환경 통합 관제 지원



01

클라우드 환경 특징

- 1. 클라우드 개요/특징
- 2. 모니터링 필요성

02

제품개요 및 특징점

- 1. 제품 개요
- 2. 아키텍처
- 3. 특징점

03

제품 기능 소개

- 1. 실시간 대쉬보드
- 2. 분석 화면
- 3. 환경설정

Real-Time Monitoring

Part 1.

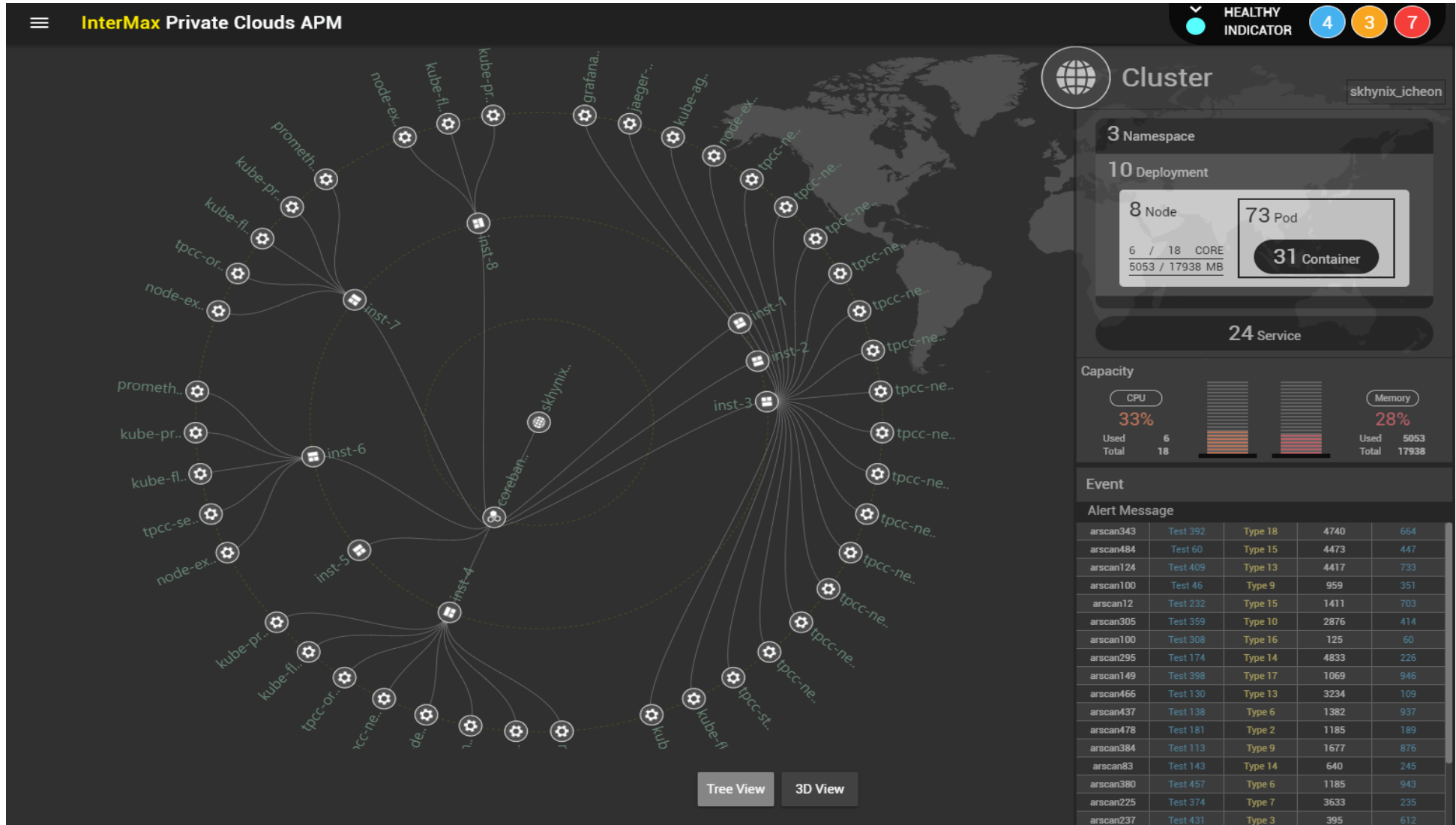
Real-time Monitoring

Key Features

Infrastructure	<ul style="list-style-type: none">• Topology(Cluster) 전체 클러스터 / Node / Pod / 컨테이너를 한눈에 볼 수 있게 Hierarchy형태로 구조화 된 토폴로지(2D, 3D)• Overview Pod생사 추이 및 주요부하 Pod를 모니터링하는 Overview 대시보드• Namespace Namespace별 현황(CPU, 메모리, 네트워크)• Nodes Node별 현황(CPU, 메모리, 네트워크, Disk)
Services	<ul style="list-style-type: none">• Cluster Service 서비스 호출로 인한 End-to-End 호출 Path View 정보 InterMax APM 어플리케이션 상세모니터링 연계• DataCenter Service 서비스 간의 호출을 3D 호출 Path View
Health Indicator	<ul style="list-style-type: none">• HI Detail Node / Pod / Container 건강성 현황 및 이상지표탐지
Workloads	<ul style="list-style-type: none">• Pods Pod별 상세 현황(Pod 가동현황, CPU, 메모리, 네트워크, Disk)• Deployment Deployment와 해당 Pod 현황(Pod 가동현황, CPU, 메모리, 네트워크, Disk)• DaemonSets DaemonSet과 해당 Pod 현황(Pod 가동현황, CPU, 메모리, 네트워크, Disk)• ReplicaSets Replicaset과 해당 Pod 현황(Pod 가동현황, CPU, 메모리, 네트워크, Disk)• Jobs Job을 수행중인 Pod 현황(Job성공/실패, 가동현황)• Cron Jobs 주기적으로 수행되는 Cron Job을 수행중인 Pod현황(성공/실패 및 수행결과 추이)• StatefulSets 상태유지형 StatefulSets으로 수행되는 Pod 현황(Pod 가동현황, CPU, 메모리, 네트워크, Disk)

Kubernetes기반 클라우드 환경에 대한 통합 모니터링 - Topology View

클라우드 기반 환경에 대한 통합 관제 - Kubernetes 환경의 토폴로지 뷰(2D, 3D 형태)



Kubernetes기반 클라우드 환경에 대한 통합 모니터링 - Topology View

클라우드 기반 환경에 대한 통합 관제 - Kubernetes 환경의 토폴로지 뷰(2D, 3D 형태)

Cluster skhynix_icheon

3 Namespace
11 Deployment
8 Node
75 Pod
6 / 18 CORE
4734 / 17938 MB
32 Container
23 Service

Capacity

CPU: 33%
Used: 6 / Total: 18
Memory: 26%
Used: 4734 / Total: 17938

Event

Alert Message				
arscan298	Test 202	Type 7	220	727
arscan280	Test 138	Type 19	3975	285
arscan315	Test 17	Type 13	1690	287
arscan274	Test 73	Type 8	4856	599
arscan165	Test 202	Type 3	420	863
arscan396	Test 347	Type 14	648	798
arscan412	Test 299	Type 19	281	674
arscan407	Test 468	Type 14	3276	721
arscan449	Test 70	Type 12	1693	193
arscan350	Test 256	Type 4	3401	92
arscan152	Test 83	Type 8	1452	977
arscan149	Test 207	Type 18	3763	61
arscan313	Test 24	Type 7	1444	723
arscan198	Test 74	Type 18	4665	957
arscan27	Test 49	Type 7	2576	121
arscan499	Test 114	Type 1	1404	869
arscan329	Test 466	Type 19	3908	768
arscan452	Test 57	Type 8	2628	130
arscan101	Test 260	Type 12	1609	677
arscan169	Test 435	Type 4	2545	538

HEALTHY INDICATOR

1	●	ubuntu1	106	31
2	●	ubuntu32	130	88
3	●	ubuntu38	81	94
4	●	ubuntu7	122	7
5	●	ubuntu38	142	49
6	●	ubuntu8	36	67
7	●	ubuntu3	65	97
8	●	ubuntu36	113	32
9	●	ubuntu36	162	78
10	●	ubuntu17	29	90
11	●	ubuntu14	50	42
12	●	ubuntu15	193	93
13	●	ubuntu14	146	55
14	●	ubuntu23	17	20

Tree View 3D View

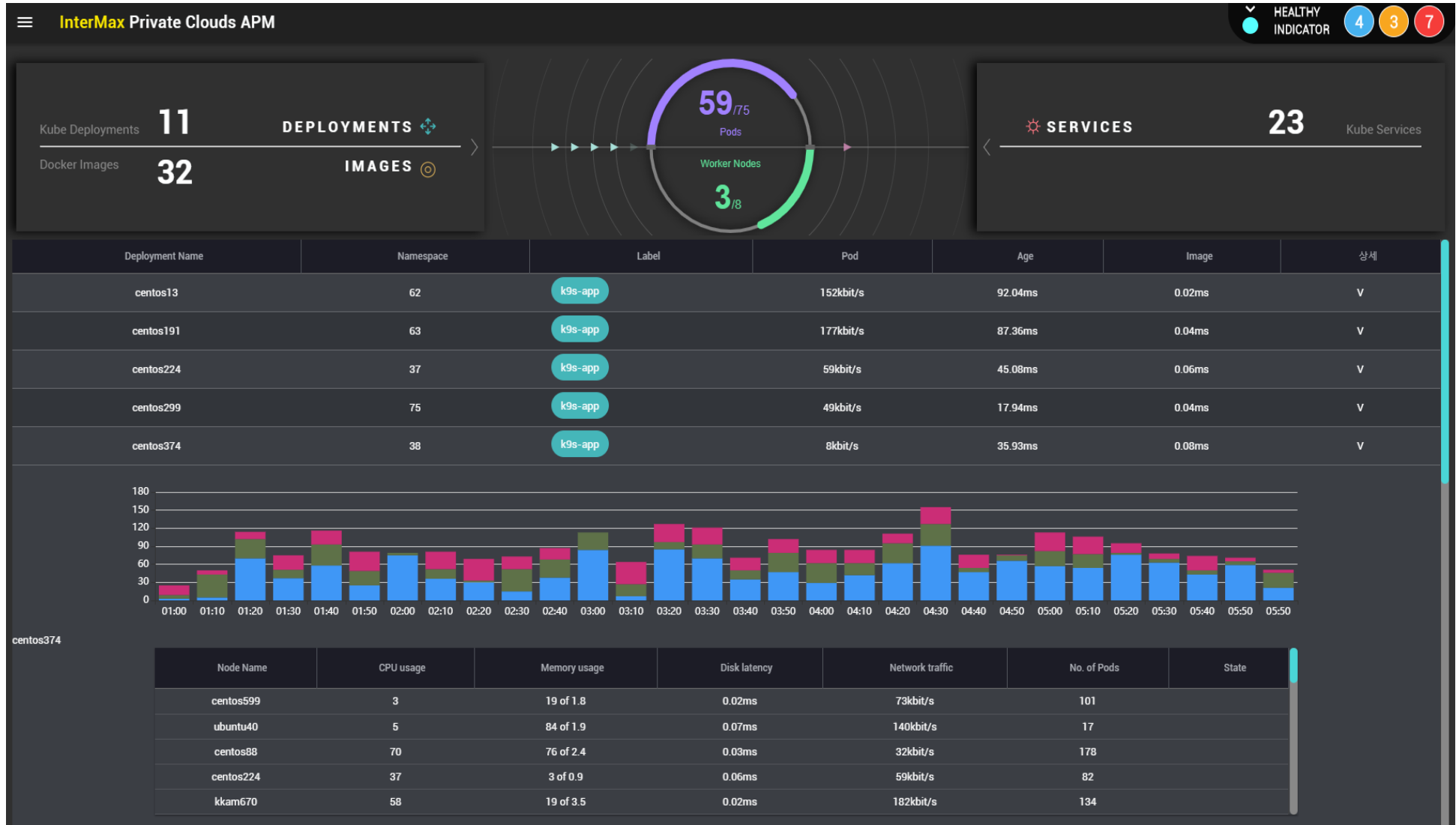
전체 성능 Metrics 지표들에 대한 Overview 제공

Clusters, Hosts(node), Pod, Containers 에 대한 통합 모니터링(성능 지표 - Overview)



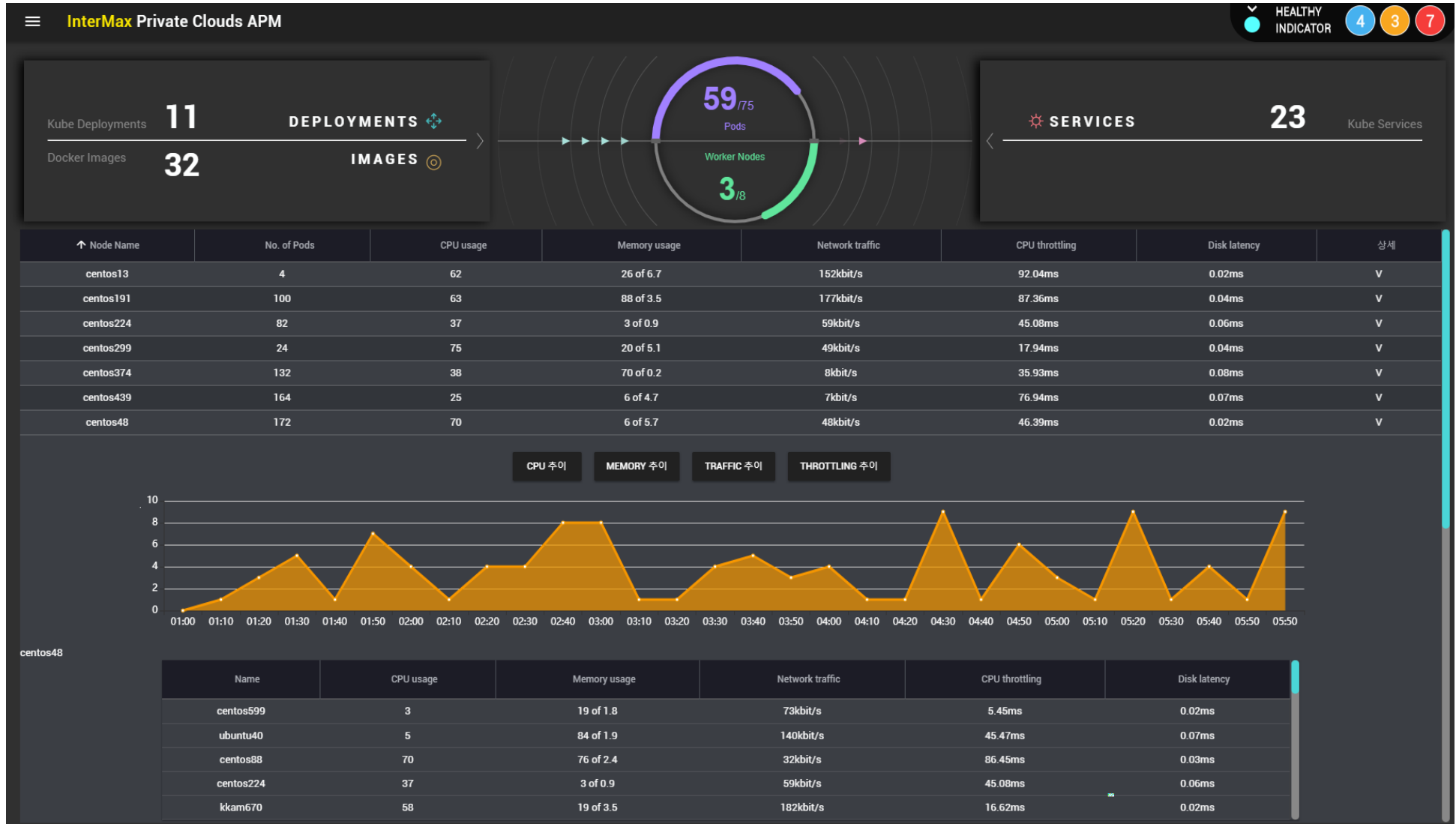
Deployments별 모니터링 현황 - hosts별 디플로이먼트 사용 현황

디플로이먼트 현황과 각 hosts별 디플로이먼트 사용 현황을 추이로 상세 분석 제공



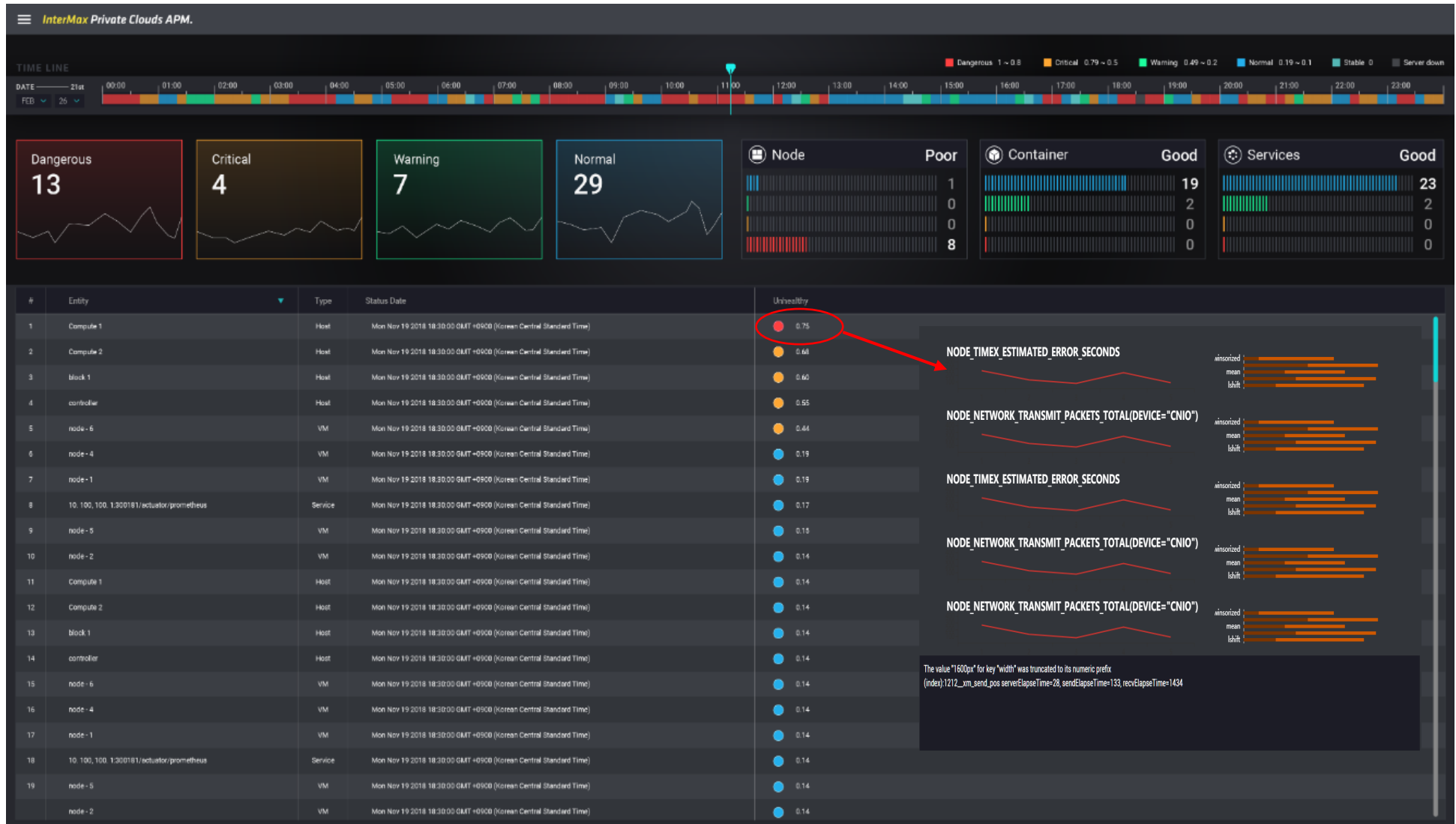
Hosts(Node)별 모니터링 현황 - Pod/Docker/Container 상세 추이 모니터링

Pod/Docker/Container에 대한 성능 지표 추이 모니터링(CPU/Memory/Network Traffic/CPU Throttling 추이 등)



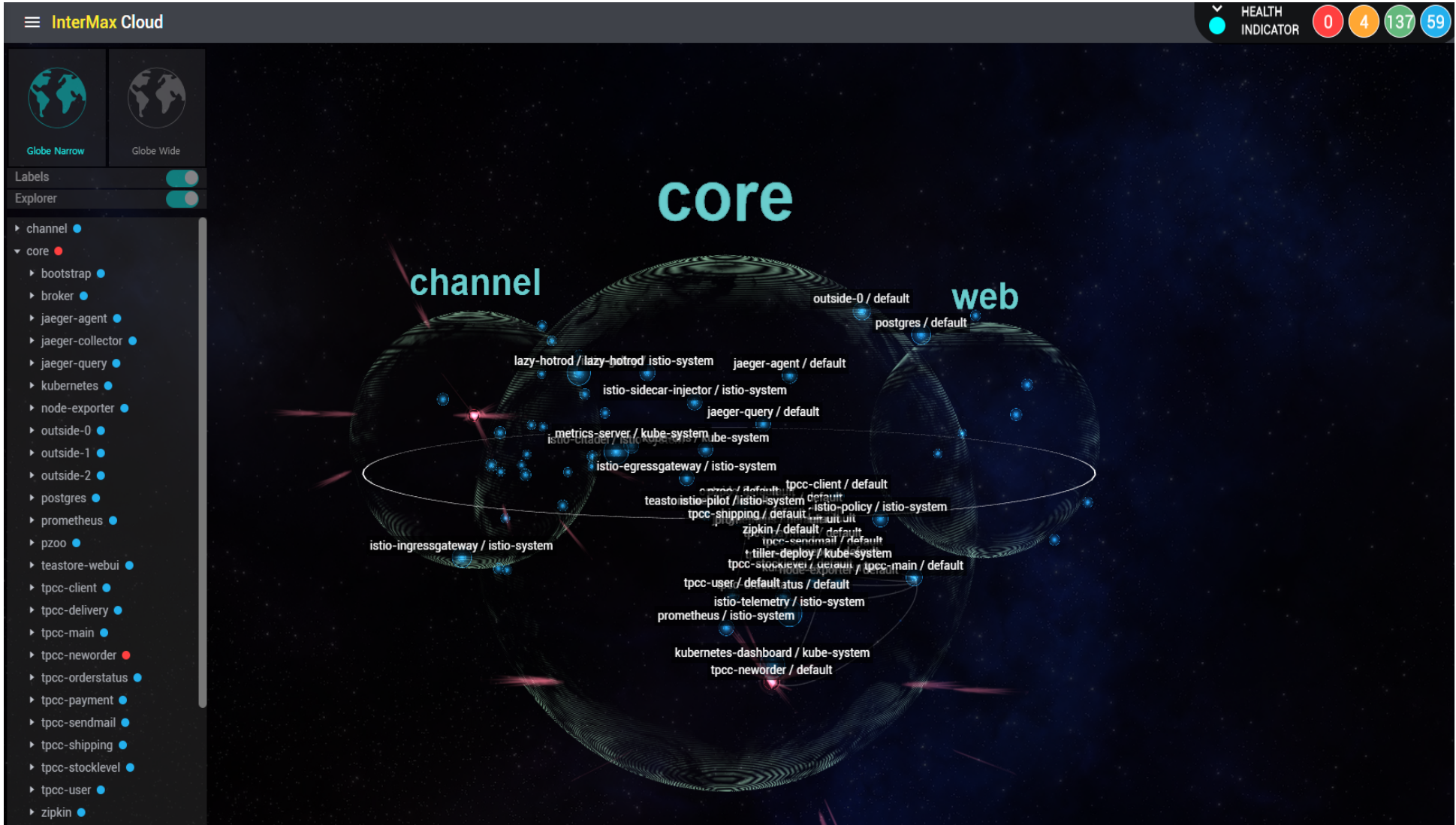
전체 Entity별, Metrics별 - 건강성 지표 모니터링

Entity별, Metrics별 scoring을 통한 건강성 판단을 위한 상세 모니터링 제공 - Alert 생성 및 원인 분석 기반



서비스 관점의 통합 모니터링

Application 레벨의 서비스 관점 모니터링(TPS, 응답시간, 서비스 부하량, 오류량 등)



서비스 레벨 상세 Trace 연계 분석 기능

Application 레벨의 상세 trace 분석 기능 제공 - InterMax와의 연동 키 맵핑을 통한 근본 원인 분석 기능

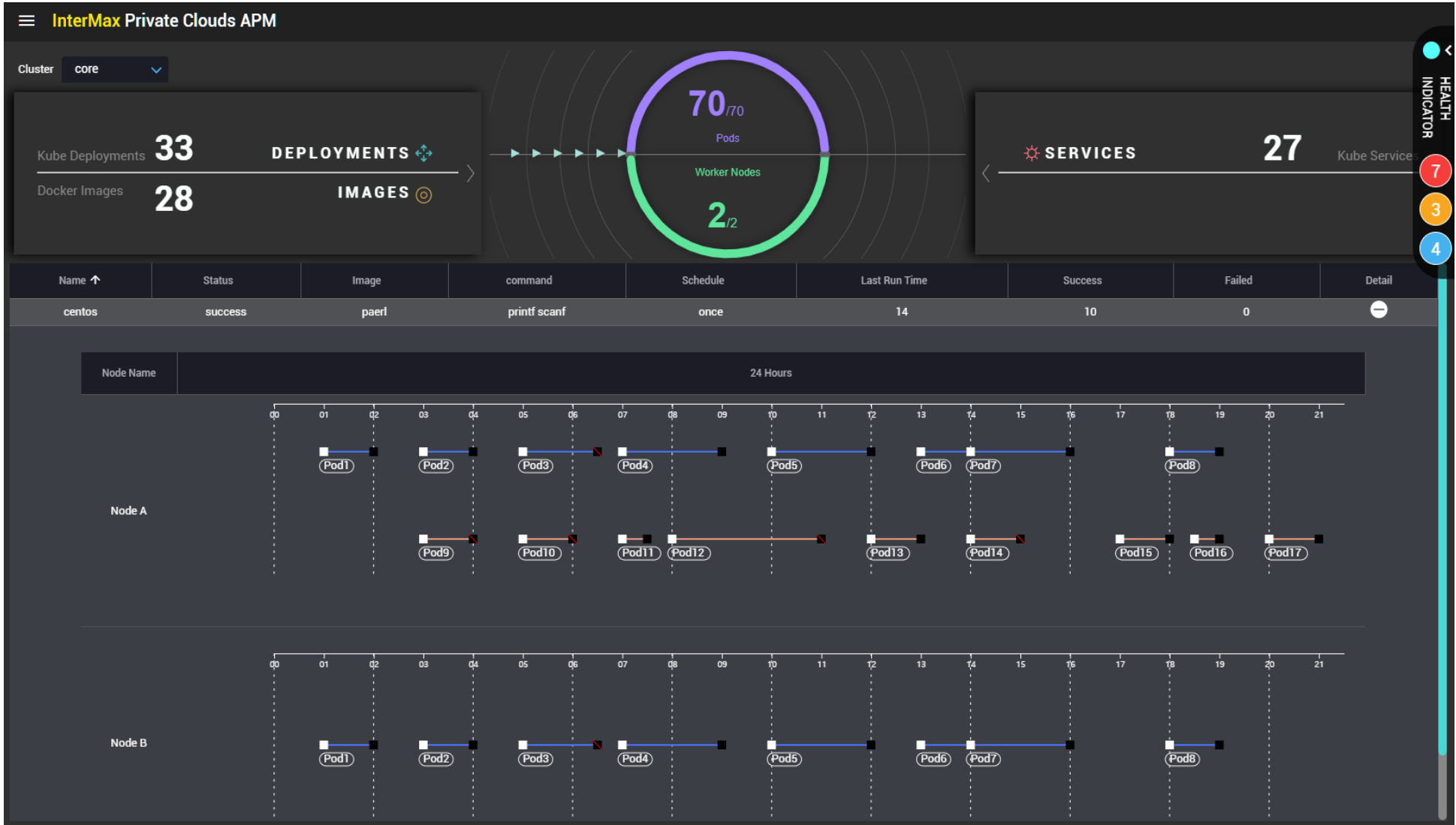
The screenshot displays the InterMax Cloud monitoring interface. It is organized into three main zones:

- Service Zone:** A network diagram showing the relationship between various services, including 'tpcc-neworder' and 'tpcc-sendmail'.
- Pod Zone:** A detailed view of a specific pod, 'tpcc-neworder-79b9dd4775-svtj'. It provides Pod Info (IP: 10.244.1.98, status: Running, count: 24, port: 30081, tps: 0.4), Pod health indicator (errorCount: 12, HIScore: 0.338, HITime: 2019-04-28 19:00), and buttons for 'Trace Analysis' and 'Log'.
- Node Zone:** A view of the underlying nodes, labeled 'core1' and 'core2'.

A red arrow indicates the flow from the Pod Zone to a detailed trace analysis window. This window shows a table of SQL queries with columns for Instance, SID, TIME, SQL Text, TID, and various performance metrics. A second window shows a detailed view of a specific SQL query, including its text and execution plan.

Job/CronJobs에 대한 진행 현황 모니터링

Batch 프로세스와 같은 Jobs/CronJobs에 대한 모니터링 제공



Performance Analysis & Configuration

Part 2.

Performance Analysis

Key Features

Statistics & Analysis

- **Performance Trend** 다양한 성능 지표에 대한 장기 성능 추이 분석
- **Alert Analysis** 알람 발생 추이 분석
- **HI Analysis** 건강성 지표 분석
- **Job History** Job수행 이력 분석
- **Infrequent Log Analysis** 특이한 상황에서 발생하는 희소로그 분석
- **Log Viewer** 다수 컨테이너에 대한 원격 로그뷰어

Report

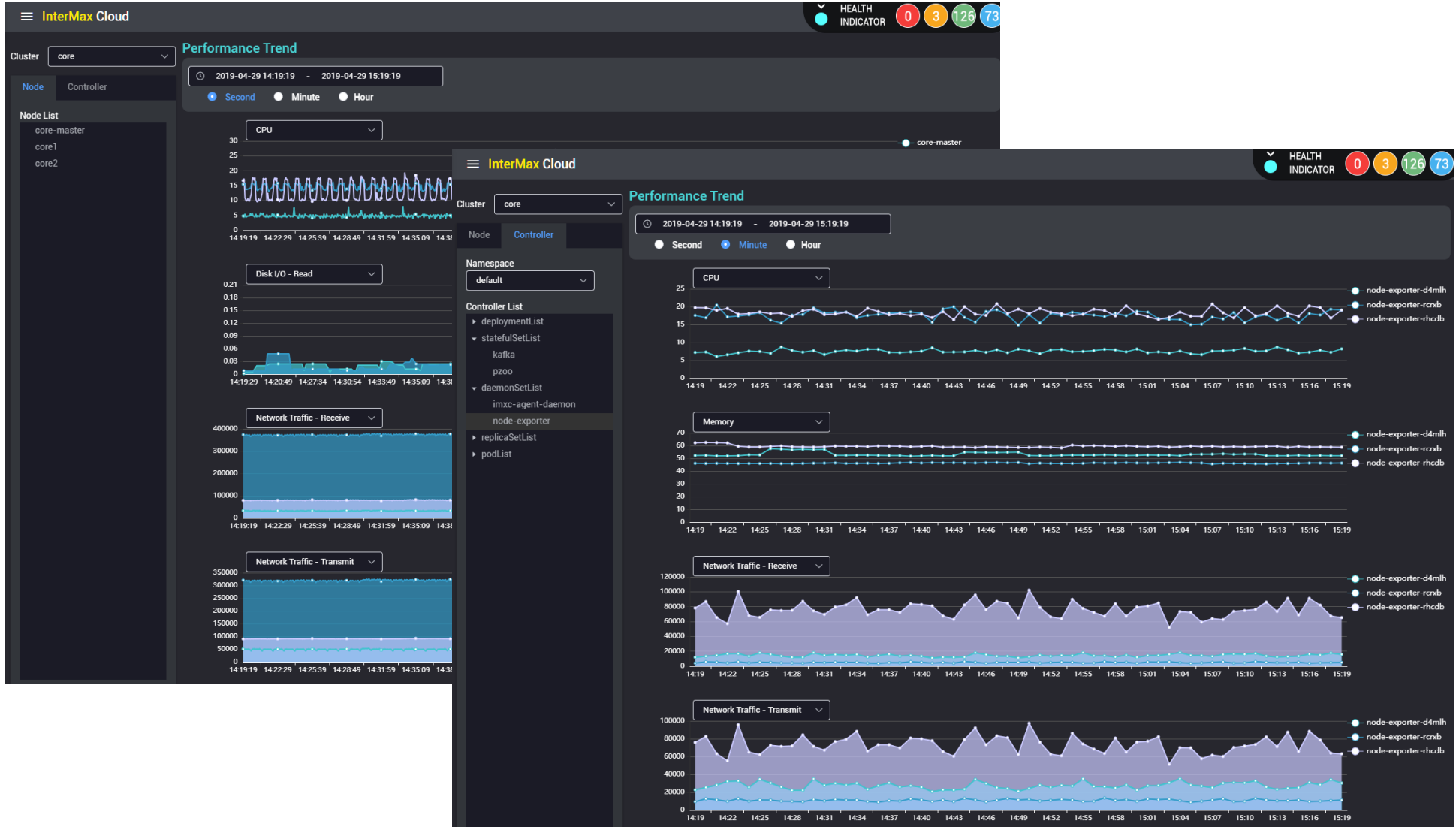
- **Daily Report** 일일보고서 / 일일 종합 현황, 시간별 현황, 일일 알람현황
- **Weekly Report** 주간보고서 / 주간 종합 현황, 일별 현황, 주간 알람현황
- **Monthly Report** 월간보고서 / 월간 종합 현황, 일별 현황, 월간 알람현황

Configuration

- **User Setting** 모니터링 그룹/사용자 추가 / 삭제
그룹/사용자별 메뉴/기능 권한 설정
- **Alert Setting** Node / Container 별 임계치 기반 알람 설정
- **Infrequent Log** 희소로그 분석 대상 / 주기 지정

Performance Trend 분석 기능

Cluster별, Node별, Controller별 시스템 리소스 및 성능 지표에 대한 추이 분석 제공



Health Indicator 분석 기능

특정 기간 동안 발생된 모든 건강성 지표(HI)에 대한 현황 분석 기능 - 특정 문제 발생 시점에 대한 상세 분석 가능

The screenshot displays the InterMax Cloud Health Indicator Analysis interface. At the top, there are filters for Cluster (core) and Namespace (All), along with a date range (2019-04-22 to 2019-04-29). A heatmap shows health indicator scores over time, with a red circle highlighting a specific issue at 10:00 on 2019-04-29. A red callout box with the text "문제 발생 구간을 drag하여 해당 시점의 HI 지표 분석" (Drag the problem occurrence interval to analyze the HI indicator at that time) points to this area.

Below the heatmap is a table listing various containers and their health scores. The table has columns for #, Node Id, Type, Namespace, Pod Id, Container Name, Max Metric Id, Max Score, and Score. The score for the selected issue (core-master) is 0.41.

#	Node Id	Type	Namespace	Pod Id	Container Name	Max Metric Id	Max Score	Score
61	core1	Container	default	postgres-78f...	postgres	postgres-78f78bf...	0.556	0.42
62	core1	Container	default	postgres-78f...	postgres	postgres-78f78bf...	0.556	0.42
63	core1	Container	istio-system	istio-sidecar...	sidecar-injector-we...	istio-sidecar-injec...	1.000	0.42
64	core1	Container	istio-system	istio-sidecar...	sidecar-injector-we...	istio-sidecar-injec...	1.000	0.42
65	core1	Container	default	postgres-78f...	postgres	postgres-78f78bf...	0.556	0.42
66	core-master	Container	kube-system	kube-apiserv...	kube-apiserver	kube-apiserver-co...	0.500	0.41
67	core1	Container	default	postgres-78f...	postgres	postgres-78f78bf...	0.556	0.41
68	core-master	Container	kube-system	kube-apiserv...	kube-apiserver	kube-apiserver-co...	0.500	0.41
69	core1	Container	default	prometheus...	prometheus	prometheus-576f...	0.639	0.41
70	core1	Container	default	prometheus...	prometheus	prometheus-576f...	0.639	0.41
71	core1	Container	default	prometheus...	prometheus	prometheus-576f...	0.636	0.41
72	core1	Container	default	pzoo-1	zookeeper	pzoo-1\$zookeeper	0.998	0.41
73	core1	Container	default	pzoo-1	zookeeper	pzoo-1\$zookeeper	0.998	0.41
74	core1	Container	default	prometheus...	prometheus	prometheus-576f...	0.631	0.41
75	core1	Container	istio-system	istio-sidecar...	sidecar-injector-we...	istio-sidecar-injec...	1.000	0.41

The detailed analysis for the selected issue (core-master) shows three metrics: container_cpu_system_seconds_total (mean: 13.45), container_cpu_user_seconds_total (mean: 11.58), and container_fs_usage_bytes (mean: 36.80). Each metric has a Top5 Unhealth Score bar chart and a data log. The data log shows a truncated value for 'width' and other server-related metrics.

CronJob History 분석 기능

Batch형태의 Job에 대해 CronJob 형태로 실행한 job history를 분석할 수 있는 기능(Succeeded, Failed 건수 등)

InterMax Cloud
HEALTH INDICATOR 0 0 0 0

Job History

2019-04-22 00:00:00 - 2019-04-26 00:00:00 Cluster: channel Namespace: default Cron Job Job Search

Job Statistics Per Date

Name	Date	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22
long2	2019-04-22 00:00:00	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0
long2	2019-04-23 00:00:00	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0
long2	2019-04-24 00:00:00	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0
long2	2019-04-25 00:00:00	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0
long2	2019-04-26 00:00:00	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0
failed-job	2019-04-22 00:00:00	0/52	0/56	0/45	0/52	0/50	0/46	0/48	0/47	0/52	0/45	0/48	0/46	0/54	0/41	0/50	0/43	0/49	0/59	0/60	0/58	0/55	0/59	0/59
failed-job	2019-04-23 00:00:00	0/60	0/60	0/60	0/55	0/59	0/59	0/57	0/58	0/59	0/53	0/56	0/60	0/59	0/51	0/37	0/20	0/53	0/57	0/60	0/60	0/57	0/55	0/59
failed-job	2019-04-24 00:00:00	0/57	0/58	0/60	0/59	0/58	0/57	0/60	0/57	0/58	0/57	0/60	0/58	0/56	0/59	0/50	0/59	0/59	0/54	0/52	0/59	0/57	0/58	0/60
failed-job	2019-04-25 00:00:00	0/58	0/59	0/60	0/57	0/57	0/57	0/58	0/58	0/59	0/60	0/60	0/59	0/60	0/59	0/58	0/57	0/59	0/56	0/58	0/58	0/59	0/58	0/55
failed-job	2019-04-26 00:00:00	0/58	0/59	0/60	0/57	0/56	0/57	0/55	0/59	0/57	0/56	0/58	0/58	0/58	0/58	0/57	0/60	0/58	0/59	0/59	0/56	0/59	0/58	0/58
long	2019-04-22 00:00:00	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0
long	2019-04-23 00:00:00	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0
long	2019-04-24 00:00:00	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0
long	2019-04-25 00:00:00	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0
long	2019-04-26 00:00:00	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0

Job Statistics Per Day

Succeeded

240

Failed

6723

Running

477

Log Viewer를 통한 다양한 로그 파일 분석 기능

문제 발생시 대상 서버 개별 접근 대신 해당 로그 파일 뷰어를 통하여 관련 오류/이슈 로그를 직접 수집/분석 할 수 있음

The screenshot displays the InterMax Cloud Node Log View interface. The main window shows a file browser for the path `/root/works`. A sidebar on the left lists nodes: `core-master`, `core1`, and `core2`. Below the file browser, a table lists downloaded logs:

Log Path	Size
<code>/root/works/nohup-loss.out</code>	3.89 KB
<code>/root/works/make_loss_50percent.sh</code>	1.79 KB
<code>/root/works/make_delay_500ms.sh</code>	2.04 KB
<code>/var/log/bootstrap.log</code>	55.42 KB
<code>/var/log/auth.log.1</code>	42.48 KB
<code>/var/log/alternatives.log</code>	242 Byte
<code>/var/log/alternatives.log.1</code>	897 Byte

A detailed log viewer window is open, showing the content of `/var/log/bootstrap.log` for node `core1`. The log content includes:

```
gpgv: Signature made Thu Apr 26 23:38:40 2018 UTC
gpgv: using RSA key 3B4FE6ACC0B21F32
gpgv: Good signature from "Ubuntu Archive Automatic Signing Key (2012) <ftpmaster@ubuntu.com>"
gpgv: Signature made Thu Apr 26 23:38:40 2018 UTC
gpgv: using RSA key 3B4FE6ACC0B21F32
gpgv: Good signature from "Ubuntu Archive Automatic Signing Key (2012) <ftpmaster@ubuntu.com>"
dpkg: warning: parsing file '/var/lib/dpkg/status' near line 5 package 'dpkg':
missing description
dpkg: warning: parsing file '/var/lib/dpkg/status' near line 5 package 'dpkg':
missing architecture
Selecting previously unselected package base-passwd.
(Reading database ... 0 files and directories currently installed.)
Preparing to unpack .../base-passwd_3.5.44_amd64.deb ...
Unpacking base-passwd (3.5.44) ...
dpkg: base-passwd: dependency problems, but configuring anyway as you requested:
base-passwd depends on libc6 (>= 2.0); however:
Package libc6 is not installed.
base-passwd depends on libdebconfclient0 (>= 0.145); however:
Package libdebconfclient0 is not installed.
Setting up base-passwd (3.5.44) ...
dpkg: warning: parsing file '/var/lib/dpkg/status' near line 24 package 'dpkg':
missing description
```

Infrequent Log 분석 뷰를 통하여 희소 로그 분석 제공

수 많은 로그 파일 중에 평상시와 다른 형태의 로그 패턴이나 메시지를 자동으로 분석할 수 있는 기능 - 장애 근본 원인 추적용

InterMax Cloud

HEALTH INDICATOR 0 0 0 0

Infrequent Log Analysis

Nodes
 Continer

-

Node	Frequency
0	3
1	2
2	1
3	1
4	3
5	1
6	3
7	4
8	1
9	1
10	4
11	0
12	2
13	2
14	3
15	0
16	1
17	2
18	3
19	3

Text Search (Pattern)

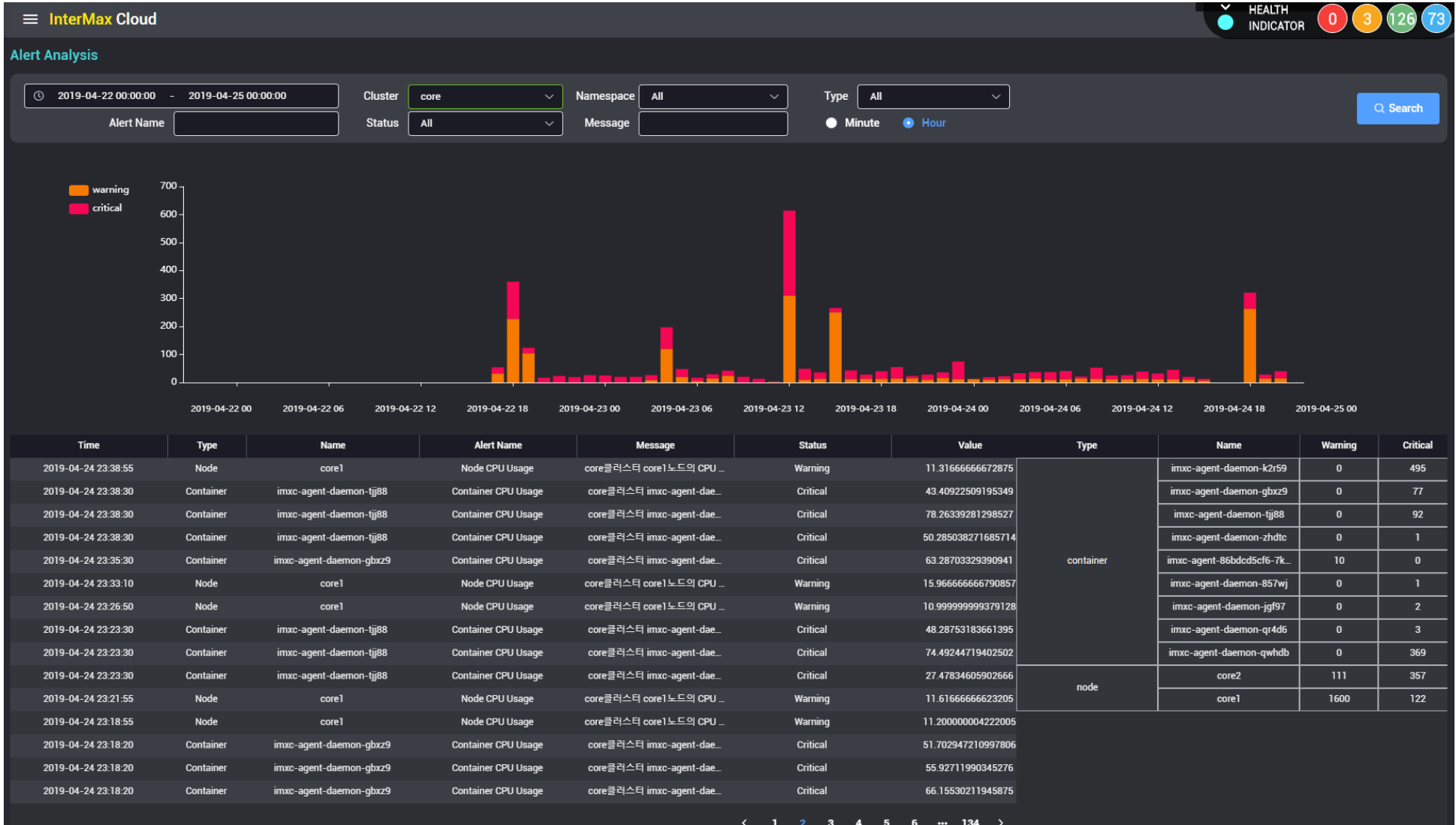
 Ignore Case

35 matches

Time	Node	File	Infrequent Log Text	Frequency No.	Frequency Rate
2019 3 29	node 1	/var/log/messages	<Time> INFO DGSERVER REMOVE LOG FILE count 0	1	0.10%
2019 3 29	node 2	/var/log/messages	<Time> ERROR java.sql.SQLException:Connection is closed	1	0.10%
2019 3 29	node 3	/var/log/messages	<Time> ERROR #MFJ DG <Didits> REPO ALERT MGR Check Repository Alert Config Error	1	0.10%
2019 3 29	node 4	/var/log/messages	<Time> ERROR #MFJ DG Undefined LOG TP SVR PROC STAT Error	1	0.10%
2019 3 29	node 5	/var/log/messages	<Time> INFO REPO CONN CHECKER Repository Connection Check Error. count 0	1	0.10%

Alert 분석 기능

특정 기간동안 발생한 알람 현황에 대한 상세 분석 및 통계 정보 제공



감사합니다.



(주)엑셈은 대한민국을 대표하는 최고의 성능 관리 전문가 집단입니다.

The Next Revolution is Here

cloud