

복잡한 쿠버네티스,
옵저버빌리티는
어떻게 하지?

모니터링 vs 옴저버빌리티

모니터링

What, When

옴저버빌리티

Why, How

**복잡한 쿠버네티스,
그리고 더 복잡한 쿠버네티스 모니터링**

복잡한 쿠버네티스,
그리고 더 복잡한 쿠버네티스 ~~모니터링~~
옵저버빌리티

쿠버네티스의 등장

MSA 등장

컨테이너 기술 확대

컨테이너 관리 필요성

쿠버네티스 등장

쿠버네티스의 복잡성

다양한
구성 요소 + 상태 관리 + 네트워킹

쿠버네티스 모니터링을 위한 옴저버빌리티 핵심 요소



Metrics

Trace

Log

일반적인 쿠버네티스 모니터링 환경 구축 여정



와탭 쿠버네티스 모니터링의 차별점

통합 모니터링 플랫폼, 와탭

서버, 애플리케이션, 데이터베이스, 브라우저, 쿠버네티스 상태를 한눈에 파악



와탭 쿠버네티스 모니터링 차별점 - 연관 분석

메트릭스, 로그, 트레이스를 실시간으로 연계하여 문제의 근본 원인을 신속하게 분석



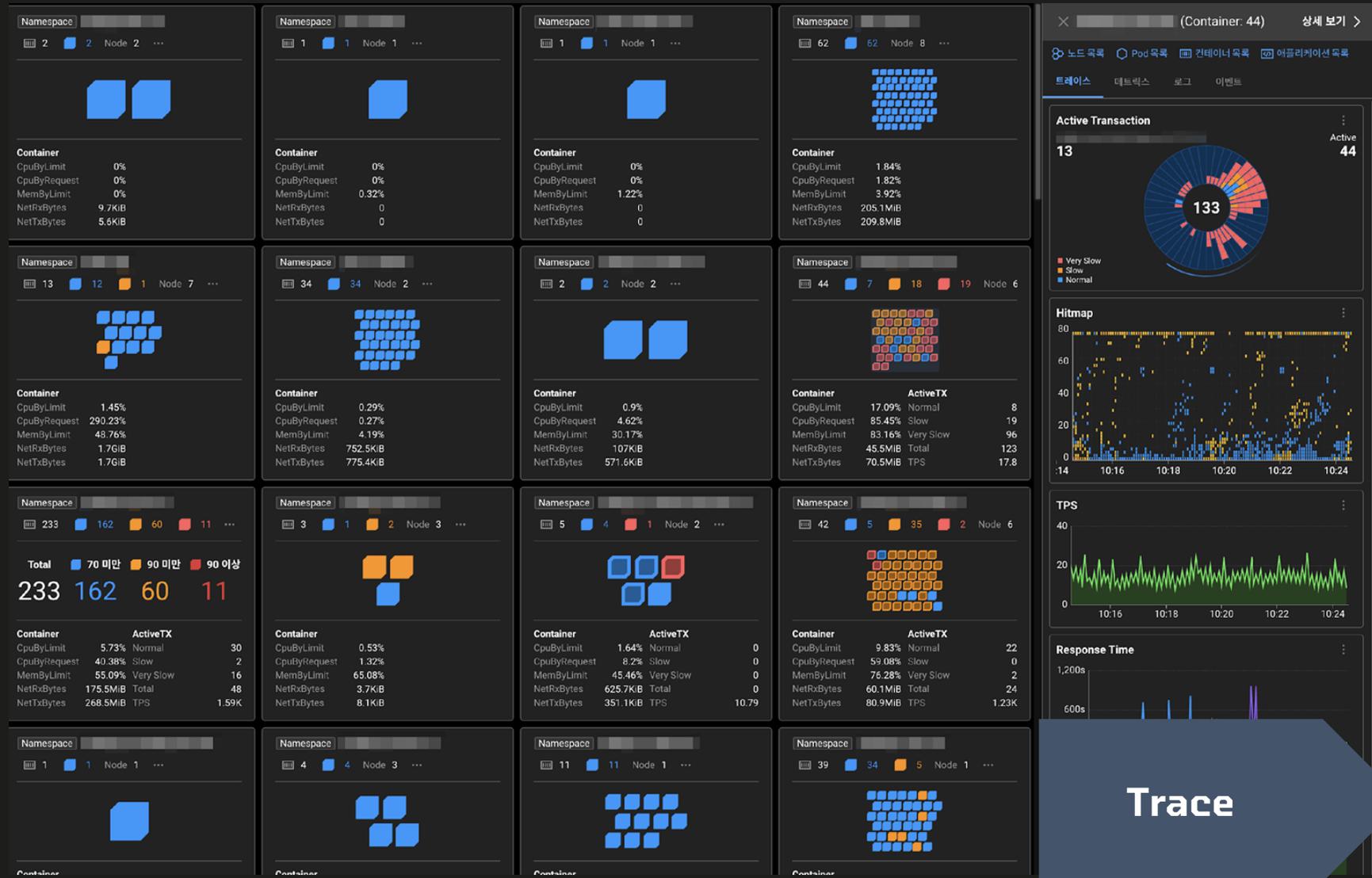
Metrics

Trace

Log

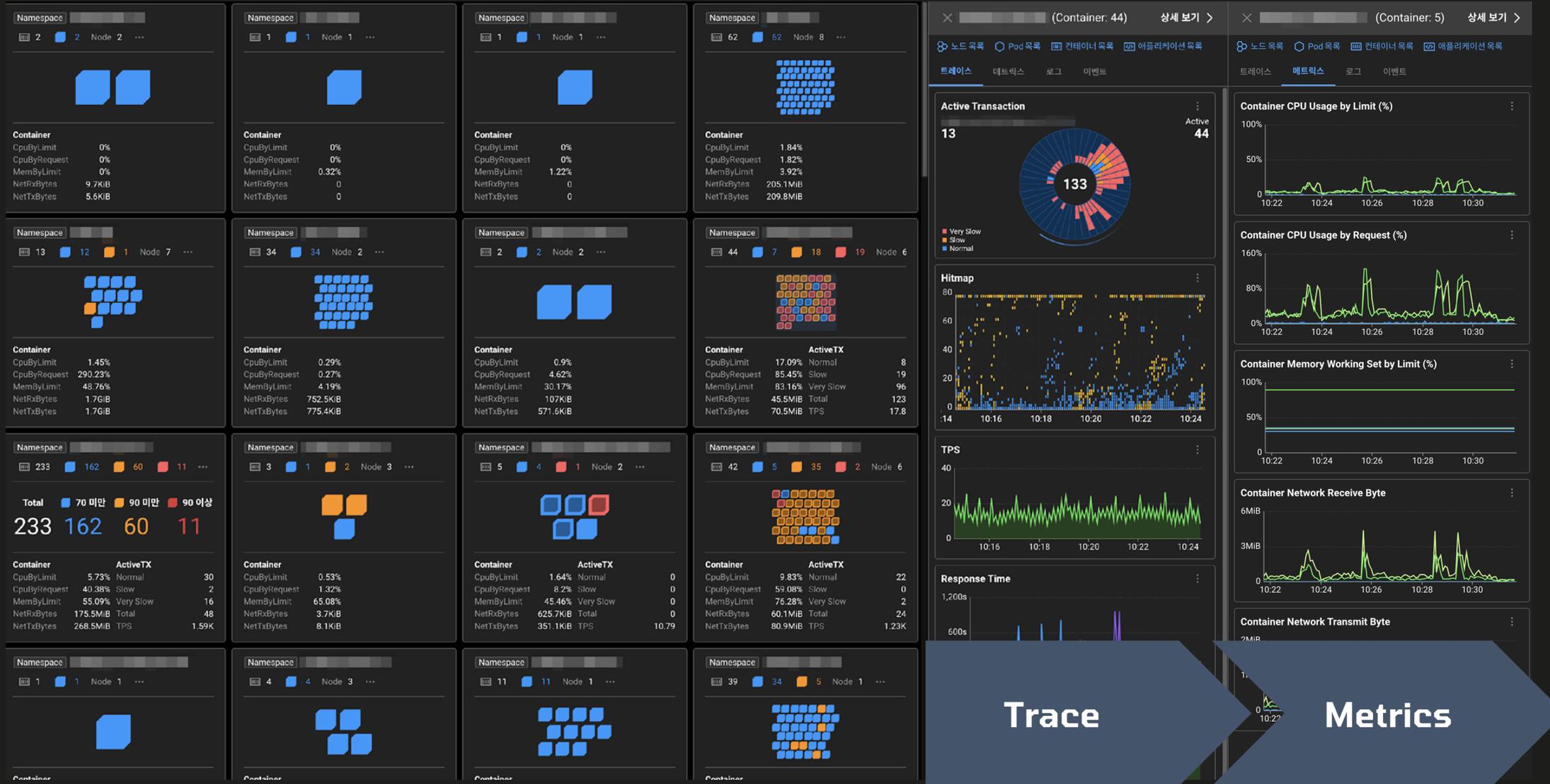
와탭 쿠버네티스 모니터링 차별점 - 연관 분석

메트릭스, 로그, 트레이스를 실시간으로 연계하여 문제의 근본 원인을 신속하게 분석



와탭 쿠버네티스 모니터링 차별점 - 연관 분석

메트릭스, 로그, 트레이스를 실시간으로 연계하여 문제의 근본 원인을 신속하게 분석



와탭 쿠버네티스 모니터링 차별점 - 연관 분석

메트릭스, 로그, 트레이스를 실시간으로 연계하여 문제의 근본 원인을 신속하게 분석

The dashboard displays a grid of 16 container metrics cards, each showing CPU, memory, and network usage. To the right, detailed views for three containers are shown:

- (Container: 44) 상세 보기**: Shows Active Transaction (13), Hitmap, TPS, and Response Time.
- (Container: 5) 상세 보기**: Shows Container CPU Usage by Limit (%), Container CPU Usage by Request (%), Container Memory Working Set by Limit (%), Container Network Receive Byte, and Container Network Transmit Byte.
- 전체 (Container: 20) 상세 보기**: Shows a log view for the container.

At the bottom, three large blue arrows labeled **Trace**, **Metrics**, and **Log** indicate the integration of these data sources.

와탭 쿠버네티스 모니터링 차별점 - 연관 분석

메트릭스, 로그, 트레이스 와 쿠버네티스 이벤트 정보를 실시간으로 연계하여 문제의 근본 원인을 신속하게 분석

The dashboard displays a grid of 16 namespace monitoring cards. Each card shows a visual representation of containers and their resource usage. The detailed view on the right shows:

- Active Transaction:** A circular gauge showing 133 active transactions.
- Container CPU Usage by Limit (%):** A line chart showing CPU usage over time.
- Container CPU Usage by Request (%):** A line chart showing CPU usage by request over time.
- Container Memory Working Set by Limit (%):** A line chart showing memory usage over time.
- Container Network Receive Byte:** A line chart showing network receive bytes over time.
- Container Network Transmit Byte:** A line chart showing network transmit bytes over time.
- Hitmap:** A heatmap showing request distribution.
- TPS (Transactions Per Second):** A line chart showing TPS over time.
- Response Time:** A line chart showing response times over time.
- Log View:** A scrollable log showing application output and system messages.
- Event View:** A scrollable list of Kubernetes events.

At the bottom, four large blue arrows labeled **Trace**, **Metrics**, **Log**, and **Event** indicate the interconnected nature of the data.

와탭 쿠버네티스 모니터링 차별점 - 분석 시나리오

많은 사용자 경험 사례와 트러블슈팅 경험을 바탕으로 한 핵심 데이터와 문제 판단 기준을 제시

WhaTap
CP EKS-DEMO Pod 시작 분석
🔊 🔔 🗄️ 🗨️ 👤

- ☰ EKS-DEMO
- 📄 대시보드
- 📁 컨테이너
- 📁 워크로드
- 🌐 클러스터
- 📊 분석
- ★ Pod 시작 분석
- 📄 애플리케이션 히트맵
- 📄 멀티서비스 히트맵
- 📄 MSA 분석
- 📄 애플리케이션 스택
- 📄 메트릭스 차트
- 📄 메트릭스 이상 탐지
- 📄 메트릭스 조회
- 📄 로그
- 📄 보고서
- 🔍 애플리케이션
- 🔔 경고 알림
- ⚙️ 관리
- 🧪 실험실

조회 시각 22:45:32
30일 이내 생성된 Pod만 표시

Pod 시작 성능을 분석하여 컨테이너 자원을 효율적으로 관리하세요.

와탭이 제안하는 Pod 시작 성능 분석 데이터를 통해 **적절한 리소스 요청 및 제한을 설정**하세요. 시작 분석을 원하는 Pod를 선택하면 해당 Pod가 시작된 후 약 5분 동안의 시작 성능을 분석합니다. **Pending > Running > Stable Running**으로 나뉘어진 구간별 자원 사용량을 파악하여 효율적으로 자원을 사용하고, 안정적으로 클러스터를 운영할 수 있습니다.

*Pod 시작 성능 기능은 최근 30일 데이터를 이용하기 때문에 생성일로부터 30일이 지난 Pod의 시작 성능은 분석할 수 없습니다.

[WhaTap Docs에서 보기](#)

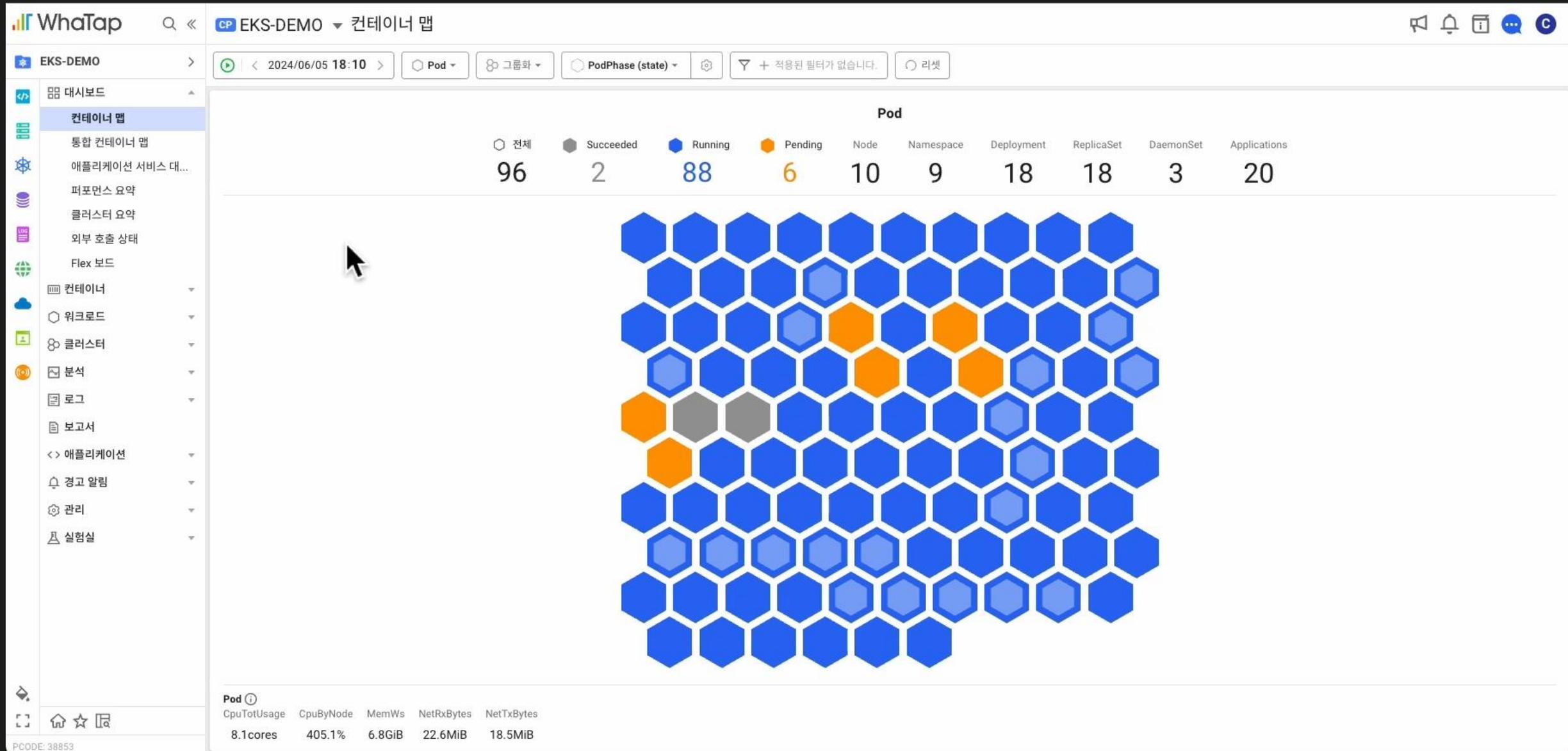
^ Pod 시작 분석 안내 보기

Name	QoS Class	Restart Count	CPU Limits	Memory Limits	CPU Requests	Memory Requests	Start Time
aws-node-5n2cc	Burstable	0	4000m	3.76GiB	50m	-	2024/06/05 20:47:32
aws-node-6mvs5	Burstable	0	4000m	7.52GiB	50m	-	2024/05/14 00:24:51
aws-node-bkd5p	Burstable	0	4000m	3.76GiB	50m	-	2024/06/05 20:56:08
aws-node-jkpq7	Burstable	0	4000m	3.76GiB	50m	-	2024/06/05 20:46:26
aws-node-jmvkf	Burstable	0	4000m	3.76GiB	50m	-	2024/06/05 20:37:42
aws-node-lv2jf	Burstable	0	4000m	3.76GiB	50m	-	2024/06/05 20:40:14
aws-node-pnltt	Burstable	0	4000m	3.76GiB	50m	-	2024/06/05 20:50:03
aws-node-qwkcl	Burstable	0	4000m	3.76GiB	50m	-	2024/06/05 20:48:16
aws-node-z8l42	Burstable	0	4000m	3.76GiB	50m	-	2024/06/05 20:54:07
aws-node-zdsmh	Burstable	0	4000m	3.76GiB	50m	-	2024/06/05 20:52:05
callee-java-deploy-5bff788776-...	BestEffort	0	2000m	1.88GiB	-	-	2024/06/05 20:57:32
callee-with-db-java-deploy-bc5...	BestEffort	0	2000m	1.88GiB	-	-	2024/06/05 20:55:41
caller-java-deploy-7855bc67b7...	BestEffort	0	2000m	1.88GiB	-	-	2024/06/05 20:47:04
catserver-576466bc8f-8c7kc	Burstable	0	200m	256MiB	200m	128MiB	2024/06/05 20:55:07

Pod 생성 후 약 5분 동안의 CPU Limit 대비 사용량 평균보다 낮은 CPU 사용량이 발생한 시점

와탭 쿠버네티스 모니터링 차별점 - 분석 시나리오

많은 사용자 경험 사례와 트러블슈팅 경험을 바탕으로 한 핵심 데이터와 문제 판단 기준을 제시



와탭 쿠버네티스 그룹 개발자의 개발 여정

에이전트 개발자의 고민 3가지

1 성능 영향 최소화

2 데이터 선별과 검증

3 속도 vs 안전성

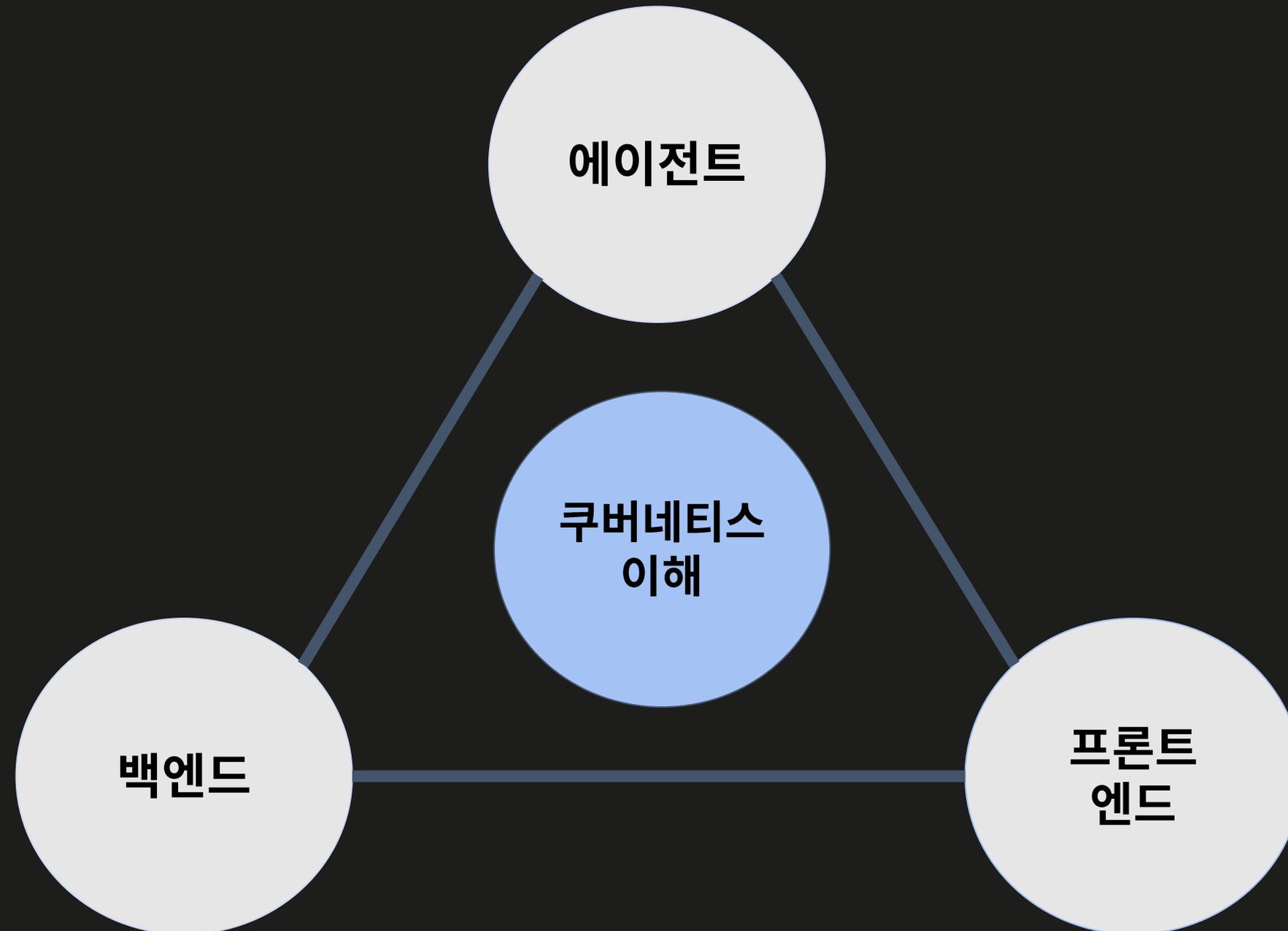
프론트엔드 개발자의 고민 3가지

1 무엇을 보여줘야 할까?

2 어떻게 보여줘야 할까?

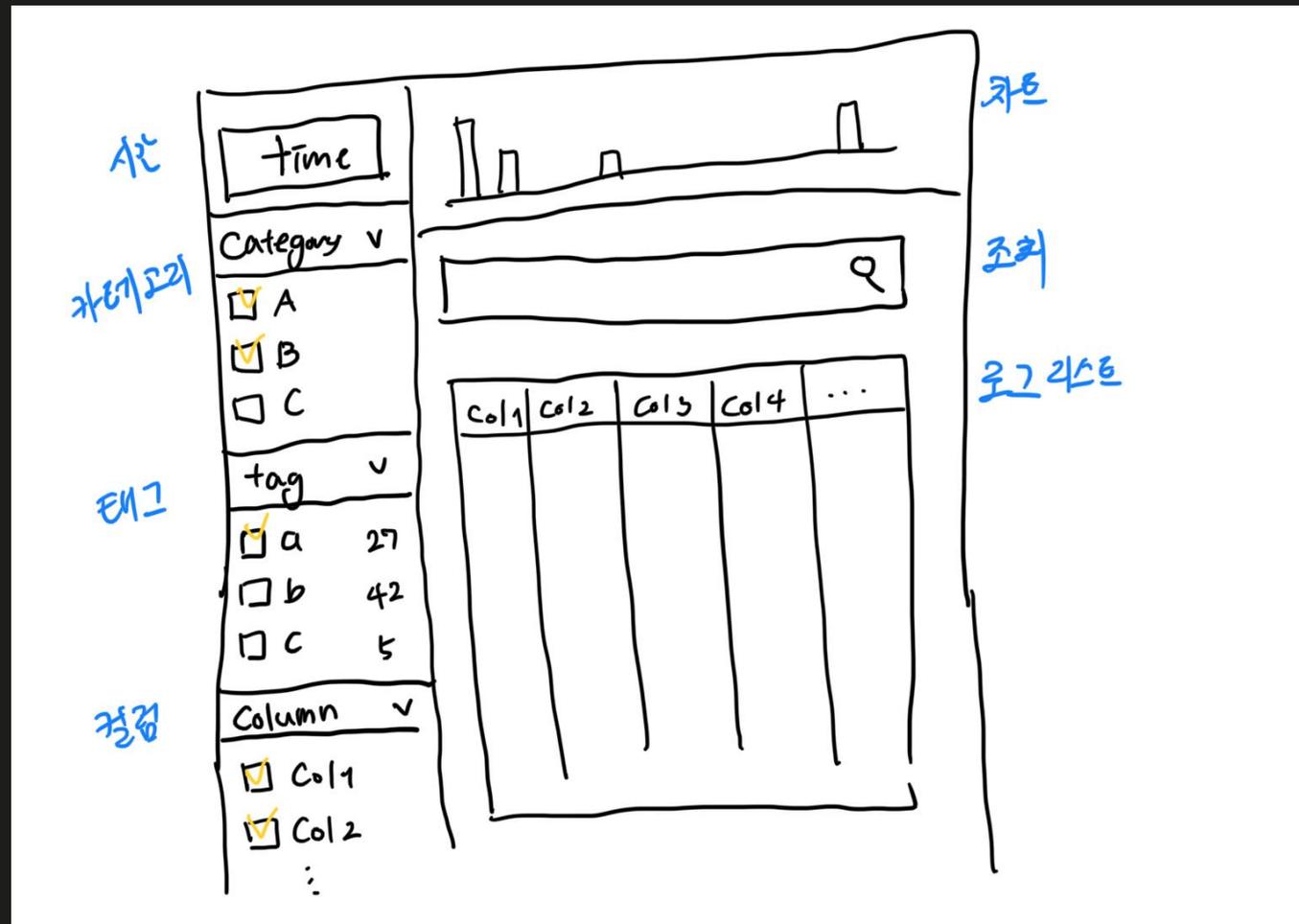
3 어디까지 보여줘야 할까?

무엇을 보여줘야 할까?



어떻게 보여줘야 할까?

어떻게 보여줘야 할까?



어떻게 보여줘야 할까?

색 시간
카테고리 Category v
 A
 B
 C
태그 tag v
 a 27
 b 42
 c 5
필드 Column v
 Col 1
 Col 2
 :

time	oname
2021-12	ote-keeper
[Error] this error from service	
2021-12	ote-keeper
[Info] this info from service this info from service	
2021-12	ote-keeper
[Warning] this warning from service	

time	oname	Contents
> 2021-12	ote-keeper	[Error] this...
> 2021-12	ote-keeper	[Info] this...
v 2021-12	ote-keeper	[Info] this from service. 2021-12-16
> 2021-12	ote-keeper	[Warn] this...

어디까지 보여줘야 할까?

한 눈에 보고 싶다

중요한게 무엇인지 알고 싶다

문제를 찾아 나가고 싶다

모든 정보가 모여있으면 좋겠다

내가 입맛에 맞게 구성하고 싶다

보고자하는 정보를 편하게 걸러내고 싶다

인터렉션 없이 보고 싶다

분석을 대신 해줬으면 좋겠다

있어보이고 싶다

차이를 비교하고 싶다

어디까지 보여줘야 할까?

한 눈에 보고 싶다

중요한게 무엇인지 알고 싶다

문제를 찾아 나가고 싶다

모든 정보가 모여있으면 좋겠다

내가 입맛에 맞게 구성하고 싶다

보고자하는 정보를 편하게 걸러내고 싶다

인터렉션 없이 보고 싶다

분석을 대신 해줬으면 좋겠다

있어보이고 싶다

차이를 비교하고 싶다

어디까지 보여줘야 할까?

한 눈에 보고 싶다

중요한게 무엇인지 알고 싶다

문제를 찾아 나가고 싶다

모든 정보가 모여있으면 좋겠다

너무 다양한 요구 사항

내가 입맛에 맞게 구성하고 싶다

어디까지 수용해야 할까?

보고자 하는 정보를 편하게 걸러내고 싶다

인터렉션 없이 보고 싶다

분석을 대신 해줬으면 좋겠다

있어보이고 싶다

차이를 비교하고 싶다

어디까지 보여줘야 할까? - 대시보드 형

한 눈에 보고 싶다

중요한게 무엇인지 알고 싶다

문제를 찾아 나가고 싶다

모든 정보가 모여있으면 좋겠다

내가 입맛에 맞게 구성하고 싶다

보고자하는 정보를 편하게 걸러내고 싶다

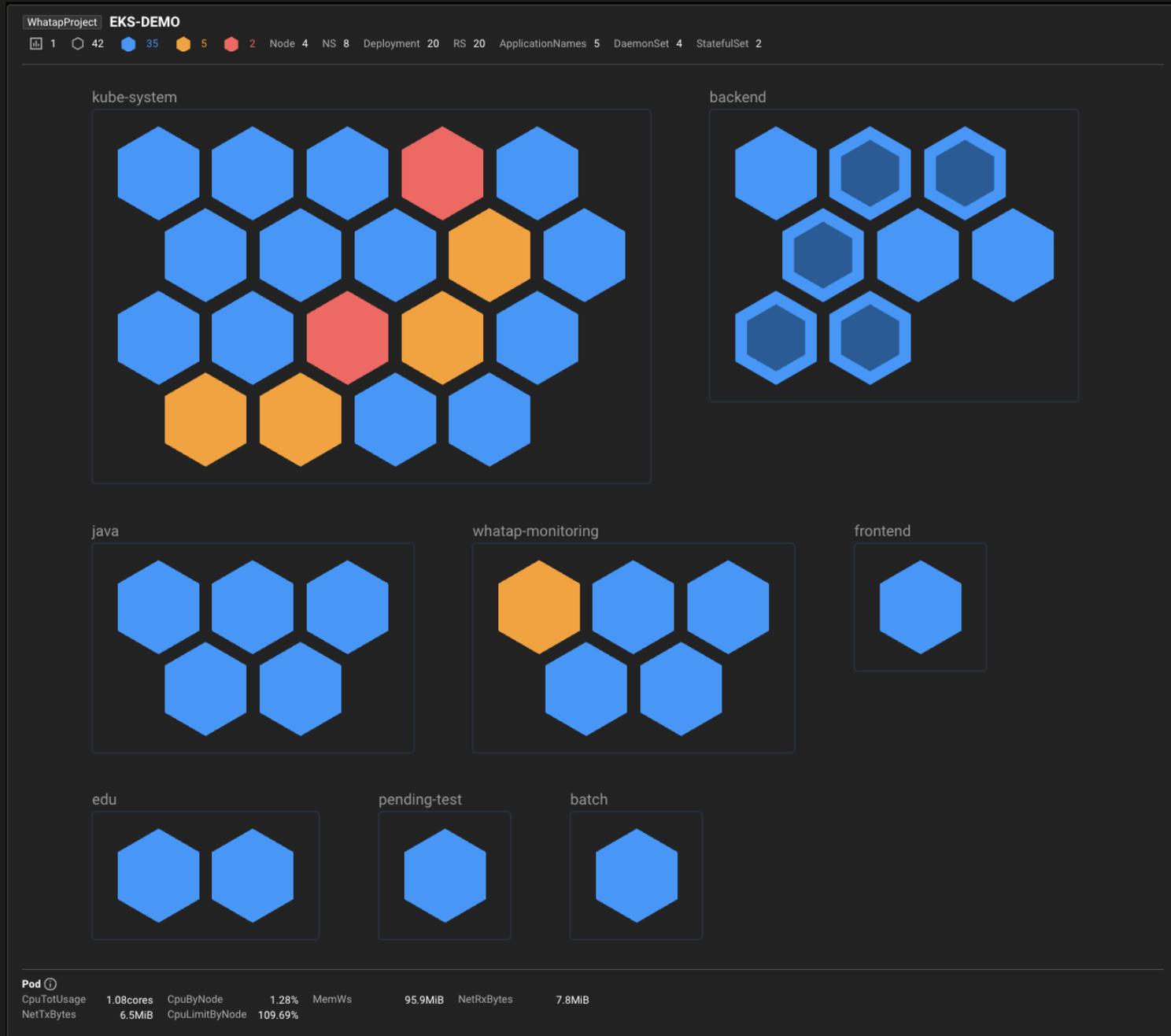
인터렉션 없이 보고 싶다

분석을 대신 해줬으면 좋겠다

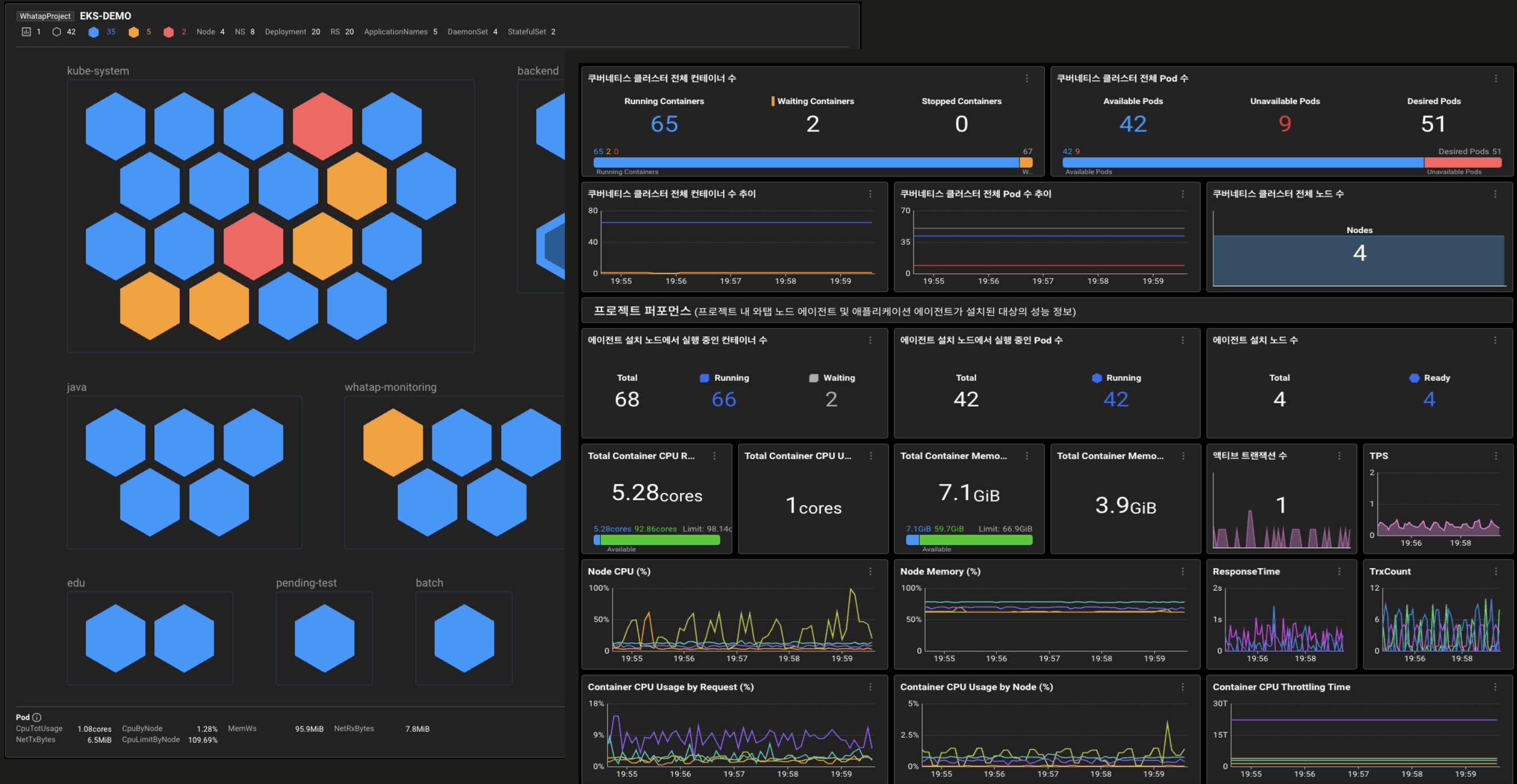
있어보이고 싶다

차이를 비교하고 싶다

어디까지 보여줘야 할까? - 대시보드 형



어디까지 보여줘야 할까? - 대시보드 형



어디까지 보여줘야 할까? - 분석 형

한 눈에 보고 싶다

중요한게 무엇인지 알고 싶다

문제를 찾아 나가고 싶다

모든 정보가 모여있으면 좋겠다

내가 입맛에 맞게 구성하고 싶다

보고자하는 정보를 편하게 걸러내고 싶다

인터렉션 없이 보고 싶다

분석을 대신 해줬으면 좋겠다

있어보이고 싶다

차이를 비교하고 싶다

어디까지 보여줘야 할까? - 분석 형

상태	시간	그룹	전체	Guaranteed	BestEffort	Burstable
LIVE 19:56:00	그룹 없음	전체 9	▲ Guaranteed 3	▲ BestEffort 3	▲ Burstable 3	
+ 적용된 필터가 없습니다.						
Guaranteed	batch-67dbd97847-hb577					
🕒	2024-05-31 17:06:26					
PodScheduled	Unschedulable: 0/4 nodes are available: 2 Insufficient memory, 4 Insufficient cpu. preemption: 0/4 nodes are available: 4 No preemption victims found for incoming pod.					
Namespace:	batch	Deployment:	batch	ReplicasetName:	batch-67dbd97847	
BestEffort	node-selector-pending					
🕒	2024-06-05 15:39:42					
PodScheduled	Unschedulable: 0/4 nodes are available: 4 node(s) didn't match Pod's node affinity/selector. preemption: 0/4 nodes are available: 4 Preemption is not helpful for scheduling.					
Namespace:	pending-test					
BestEffort	pvc-pending					
🕒	2024-06-05 15:39:42					
PodScheduled	Unschedulable: 0/4 nodes are available: persistentvolumeclaim "nonexistent-claim" not found. preemption: 0/4 nodes are available: 4 Preemption is not helpful for scheduling.					
Namespace:	pending-test					
Burstable	resource-pending					
🕒	2024-06-05 15:39:42					
PodScheduled	Unschedulable: 0/4 nodes are available: 4 Insufficient cpu, 4 Insufficient memory. preemption: 0/4 nodes are available: 4 No preemption victims found for incoming pod.					
Namespace:	pending-test					
Burstable	resourcequota-pending					
🕒	2024-06-05 15:39:43					
PodScheduled	Unschedulable: 0/4 nodes are available: 4 Insufficient memory. preemption: 0/4 nodes are available: 4 No preemption victims found for incoming pod.					
Namespace:	pending-test					
Guaranteed	travi-6cfd4d5fcc-w4jf7					
🕒	2024-06-19 09:48:19					
PodScheduled	Unschedulable: 0/4 nodes are available: 4 Insufficient memory. preemption: 0/4 nodes are available: 4 No preemption victims found for incoming pod.					
Namespace:	batch	Deployment:	travi	ReplicasetName:	travi-6cfd4d5fcc	
BestEffort	caller-java-arm-d99cd45c7-2msnh					
🕒	2024-06-21 06:46:30					
PodScheduled	Unschedulable: 0/4 nodes are available: 4 node(s) didn't match Pod's node affinity/selector. preemption: 0/4 nodes are available: 4 Preemption is not helpful for scheduling.					
Namespace:	java	Deployment:	caller-java-arm	ReplicasetName:	caller-java-arm-d99cd45c7	
Guaranteed	travi-574bd6f5b6-fjqfp					
🕒	2024-07-24 18:35:50					

어디까지 보여줘야 할까? - 분석 형

|| LIVE 19:56:00 그룹 없음 전제 9 ▲ Guaranteed 3 ▲ BestEffort 3 ▲ Burstable 3

+ 적용된 필터가 없습니다.

Guaranteed batch-67dbd97847-hb577
🕒 2024-05-31 17:06:26
PodScheduled Unschedulable: 0/4 nodes are available: 2 Insufficient memory, 4 Insufficient cp
Namespace: batch Deployment: batch ReplicasetName: batch-67dbd97847

BestEffort node-selector-pending
🕒 2024-06-05 15:39:42
PodScheduled Unschedulable: 0/4 nodes are available: 4 node(s) didn't match Pod's node affi
Namespace: pending-test

BestEffort pvc-pending
🕒 2024-06-05 15:39:42
PodScheduled Unschedulable: 0/4 nodes are available: persistentvolumeclaim "nonexistent-cla
Namespace: pending-test

Burstable resource-pending
🕒 2024-06-05 15:39:42
PodScheduled Unschedulable: 0/4 nodes are available: 4 Insufficient cpu, 4 Insufficient memor
Namespace: pending-test

Burstable resourcequota-pending
🕒 2024-06-05 15:39:43
PodScheduled Unschedulable: 0/4 nodes are available: 4 Insufficient memory. preemption: 0/4
Namespace: pending-test

Guaranteed travi-6cfd4d5fcc-w4jf7
🕒 2024-06-19 09:48:19
PodScheduled Unschedulable: 0/4 nodes are available: 4 Insufficient memory. preemption: 0/4
Namespace: batch Deployment: travi ReplicasetName: travi-6cfd4d5fcc

BestEffort caller-java-arm-d99cd45c7-2msnh
🕒 2024-06-21 06:46:30
PodScheduled Unschedulable: 0/4 nodes are available: 4 node(s) didn't match Pod's node affini
Namespace: java Deployment: caller-java-arm ReplicasetName: caller-java-arm-d99cd45c7

Guaranteed travi-574bd6f5b6-fjqfp
🕒 2024-07-24 18:35:50

오브젝트 매니페스트 상세

Kind Node Name ip-10-21-140-165.ap-northeast-1.compute.internal uid 92133671-8d70-47e4-a790-7ab3db794071 CreatedTime 2024-08-19T21:46:10Z

2024/08/14 ~ 2024/08/20 7일

비교하기

현재 시각

2024/08/20 09:01

2024/08/20 06:46 Warning

```
119 topology.kubernetes.io/zone: "ap-northeast-1"
120 node.kubernetes.io/instance-type: "t3.small"
121 eks.amazonaws.com/sourceLaunchTemplateVersion: "1"
122 kubernetes.io/hostname: "ip-10-21-140-165.ap-northeast-
123 alpha.eksctl.io/cluster-name: "an1-eks-demo"
124 alpha.eksctl.io/nodegroup-name: "eks-ng-jins-127-worker
125 kubernetes.io/os: "linux"
126 eks.amazonaws.com/capacityType: "SPOT"
127 topology.kubernetes.io/zone: "ap-northeast-1d"
128 beta.kubernetes.io/os: "linux"
129 eks.amazonaws.com/nodegroup: "eks-ng-jins-127-workers-s
130 k8s.io/cloud-provider-aws: "2fe7c991f54f724e1fcc1b50171
131 eks.amazonaws.com/sourceLaunchTemplateId: "lt-0112e883d

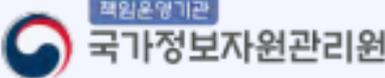
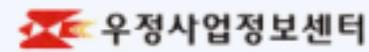
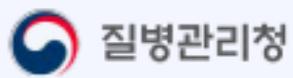
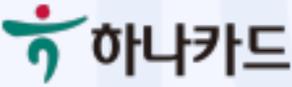
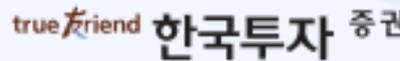
132 spec:
133   providerID: "aws:///ap-northeast-1d/i-08ed74845227f8d94"

134 status:
135   daemonEndpoints:
136     kubeletEndpoint:
137       Port: "10250"
138   allocatable:
139     ephemeral-storage: "76224326324"
140     memory: "1483028Ki"
141     cpu: "1930m"

2024/08/20 06:46 Warning
44 eks.amazonaws.com/sourceLaunchTemplateVersion: "1"
45 kubernetes.io/hostname: "ip-10-21-140-165.ap-northeast-
46 alpha.eksctl.io/cluster-name: "an1-eks-demo"
47 alpha.eksctl.io/nodegroup-name: "eks-ng-jins-127-worker
48 kubernetes.io/os: "linux"
49 eks.amazonaws.com/capacityType: "SPOT"
50 beta.kubernetes.io/os: "linux"
51 eks.amazonaws.com/nodegroup: "eks-ng-jins-127-workers-s
52 eks.amazonaws.com/sourceLaunchTemplateId: "lt-0112e883d
53+ apiVersion: "v1"
54+ kind: "Node"
55 spec:
56   providerID: "aws:///ap-northeast-1d/i-08ed74845227f8d94"
57+ taints:
58+   - effect: "NoSchedule"
59+     value: "true"
60+     key: "node.cloudprovider.kubernetes.io/uninitialized"
61+   - effect: "NoSchedule"
62+     key: "node.kubernetes.io/not-ready"

63 status:
64   daemonEndpoints:
65     kubeletEndpoint:
66       Port: "10250"
67   allocatable:
68     ephemeral-storage: "76224326324"
69     memory: "1483028Ki"
70     cpu: "1930m"
```

공공&금융 기관, 대기업, 중소기업까지 다양한 분야의 기업들이 와탭 모니터링 서비스를 선택했습니다.

Government	Financial Institutions	Large Enterprises	SMEs	Startups
 책임운영기관 국가정보자원관리원	 KB 국민은행	 LG U+	 11	 bithumb
 우정사업정보센터	 신한은행	 HYUNDAI	 STARBUCKS™	 Jobis
 건강보험심사평가원 HEALTH INSURANCE REVIEW & ASSESSMENT SERVICE	 롯데카드	 LOTTE	 Pulmuone	 여기어때.
 KERIS 한국교육학술정보원	 신한카드	 CJ	 MISSHA	 직방
 질병관리청	 하나카드	 ASIANA AIRLINES	 FILA	 wadiz
 법무부	 true friend 한국투자 증권	 DOOSAN	 아프리카TV	 opensurvey
 KLID 한국지역정보개발원	 키움증권	 SK	 I.I:mall	 true balance
 EWP 한국투자증권	 KYOBO 교보생명	 kt	 CERAGEM	 MUNPIA

감사합니다