금감원 IT감사방향에 대응하는
와탭랩스 Observability

와탭랩스
최중락 · 김성조

Contents

2024 K-금융

rent rank	Previous	Current vs.	Company	Headquarters	Total assets (US$B)
1	1	NC	Industrial and Commercial Bank of China Ltd. (1398-SEHK)	China	6,303.44
2	3	▲	Agricultural Bank of China Ltd. (1288-SEHK)	China	5,623.12
3	2	▼	China Construction Bank Corp. (939-SEHK)1	China	5,400.28
4	4	NC	Bank of China Ltd. (3988-SEHK)1	China	4,578.28
5	5	NC	JPMorgan Chase & Co. (JPM-NYSE)	US	3,875.39
6	6	NC	Bank of America Corp. (BAC-NYSE)	US	3,180.15
7	8	▲	HSBC Holdings PLC (HSBA-LSE)2	UK	2,919.84
8	9	▲	BNP Paribas SA (BNP-ENXTPA)3	France	2,867.44
9	7	▼	Mitsubishi UFJ Financial Group Inc. (8306-TSE)	Japan	2,816.77
10	10	NC	Crédit Agricole Group4	France	2,736.95
11	12	▲	Postal Savings Bank of China Co. Ltd. (1658-SEHK)	China	2,217.86
12	11	▼	Citigroup Inc. (C-NYSE)5	US	2,200.83
13	13	NC	Sumitomo Mitsui Financial Group Inc. (8316-TSE)	Japan	2,027.34
14	17	▲	Banco Santander SA (SAN-BME)	Spain	1,986.36
15	15	NC	Bank of Communications Co. Ltd. (3328-SEHK)	China	1,982.89
16	16	NC	Wells Fargo & Co. (WFC-NYSE)	US	1,932.47
17	14	▼	Mizuho Financial Group Inc. (8411-TSE)	Japan	1,923.56
18	18	NC	Barclays PLC (BARC-LSE)6	UK	1,891.72
19	22	▲	Société Générale SA (GLE-ENXTPA)7	France	1,717.49
20	20	NC	UBS Group AG (UBSG-SWX)	Switzerland	1,717.25
21	21	NC	Groupe BPCE	France	1,706.80
22	26	▲	Goldman Sachs Group Inc. (GS-NYSE)	US	1,641.59
23	19	▼	JAPAN POST BANK Co. Ltd. (7182-TSE)	Japan	1,625.60
24	23	▼	Royal Bank of Canada (RY-TSX)*8	Canada	1,566.41
25	25	NC	China Merchants Bank Co. Ltd. (600036-SHSE)	China	1,555.30
26	27	▲	Deutsche Bank AG (DBK-XTRA)	Germany	1,450.57
27	28	▲	Industrial Bank Co. Ltd. (601166-SHSE)	China	1,432.59
28	24	▼	Toronto-Dominion Bank (TD-TSX)*	Canada	1,428.29
29	29	NC	China Citic Bank Corp. Ltd. (998-SEHK)	China	1,276.63
30	32	▲	Crédit Mutuel Group	France	1,262.95
31	30	▼	Shanghai Pudong Development Bank Co. Ltd. (600000-SHSE)**	China	1,207.18
32	31	▼	Morgan Stanley (MS-NYSE)	US	1,193.69
33	33	NC	Lloyds Banking Group PLC (LLOY-LSE)	UK	1,122.76
34	34	NC	China Minsheng Banking Corp. Ltd. (600016-SHSE)	China	1,082.37
35	36	▲	ING Groep NV (INGA-ENXTAM)	Netherlands	1,078.35
36	35	▼	Intesa Sanpaolo SpA (ISP-BIT)9	Italy	1,066.74
37	37	NC	Bank of Nova Scotia (BNS-TSX)*		
38	41	▲	Bank of Montreal (BMO-TSX)*		
39	39	NC	China Everbright Bank Co. Ltd. (601818-SI		
40	40	NC	NatWest Group PLC (NWG-LSE)		
41	38	▼	UniCredit SpA (UCG-BIT)10		
42	42	NC	Commonwealth Bank of Australia (CBA-AS		
43	46	▲	Banco Bilbao Vizcaya Argentaria SA (BBV/		
44	43	▼	Standard Chartered PLC (STAN-LSE)12		
45	44	▼	La Banque Postale SA		
46	45	▼	Ping An Bank Co. Ltd. (000001-SZSE)		
47	48	▲	State Bank of India (SBIN-NSEI)		
48	54	▲	ANZ Group Holdings Ltd. (ANZ-ASX)**13		
49	49	NC	Canadian Imperial Bank of Commerce (CM		
50	53	▲	DZ Bank AG		
64	73	▲	Itaú Unibanco Holding SA (ITUB4-BOVESPA)	Brazil	555.72
65	59	▼	KB Financial Group Inc. (A105560-KOSE)	South Korea	551.94
66	64	▼	Danske Bank A/S (DANSKE-CPSE)14	Denmark	542.81
67	62	▼	Truist Financial Corp. (TFC-NYSE)	US	535.35
68	66	▼	Shinhan Financial Group Co. Ltd. (A055550-KOSE)	South Korea	533.48
69	61	▼	Resona Holdings Inc. (8308-TSE)16	Japan	527.53
70	68	▼	Sumitomo Mitsui Trust Holdings Inc. (8309-TSE)	Japan	520.34
71	70	▼	Bank of Beijing Co. Ltd. (601169-SHSE)**	China	503.31
72	67	▼	China Guangfa Bank Co. Ltd.***	China	495.55
73	NR	▼	Charles Schwab Corp. (SCHW-NYSE)	US	493.18
74	NR	▼	HDFC Bank Ltd. (HDFCBANK-NSEI)17	India	464.34
75	76	▲	Bank of Jiangsu Co. Ltd. (600919-SHSE)**	China	457.25
76	72	▼	Hana Financial Group Inc. (A086790-KOSE)	South Korea	456.47
77	83	▲	Banco do Brasil SA (BBAS3-BOVESPA)	Brazil	447.72
78	98	▲	Nationwide Building Society (NBS-LSE)**18	UK	446.95
79	82	▲	China Zheshang Bank Co. Ltd. (2016-SEHK)	China	443.37
80	74	▼	Oversea-Chinese Banking Corp. Ltd. (O39-SGX)	Singapore	441.53
81	79	▼	Bank of Shanghai Co. Ltd. (601229-SHSE)	China	435.14
82	78	▼	ABN AMRO Bank NV (ABN-ENXTAM)	Netherlands	417.72
83	77	▼	Bank of New York Mellon Corp. (BK-NYSE)	US	409.88
84	84	NC	United Overseas Bank Ltd. (U11-SGX)11	Singapore	396.62
85	90	▲	Banco Bradesco SA (BBDC4-BOVESPA)	Brazil	394.76
86	75	▼	NongHyup Financial Group Inc.**	South Korea	394.46
87	85	▲	Nomura Holdings Inc. (8604-TSE)	Japan	388.42
88	80	▼	Woori Financial Group Inc. (A316140-KOSE)	South Korea	384.04
89	81	▼	KBC Group NV (KBC-ENXTBR)	Belgium	383.47
				Brazil	377.29
				Austria	372.67
				Germany	368.41
			*	China	365.96
			l) (SEB A-OM)	Sweden	358.79
				Switzerland	352.87
			. A-OM)	Sweden	351.79
			.)	South Korea	345.81

구분	~20	~100	Top Bank	US$B	KWN(조원)
CHINA	6	20	1위	6,303	8,194
US	4	12	5위	3,875	5,038
JAPAN	3	8	7위	2,817	3,662
KOREA	0	6	65위	552	712

디지털 금융 감독방향 변화 - 규제완화의 배경

금융분야 망분리 개선 방향과 기대효과

"**금융권 망분리 10년, 혁신과 보안의 새로운 균형으로의 도약**"

☞ 급변하는 **IT환경** 下 **금융산업 경쟁력 제고**, **금융소비자 효용 증진**, **금융보안체계 선진화**를 위해 **망분리 개선**은 선택이 아닌 **필수**

☞ 망분리 개선 과정에서 **보안**상 허점이 발생하지 않도록 **충분한 안전장치** 마련

□ **(검토배경) 금융권 망분리** 도입('13.12월) 후 **10년 넘게 경과***

* '13.3.20. 금융회사 대규모 전산망 마비를 계기로 '공공부문'의 물리적 망분리를 '금융권' 도입

○ ①망분리로 인한 **업무 비효율**, ②**연구·개발** 및 **신기술**(AI 등) 활용 **애로**, ③**해외 규제**와의 **괴리** 등에 따른 **규제개선 요청** 지속

□ **(그간의 경과)** 몇 차례 **규제개선**이 있었으나 시장의 기대에 **미흡** → **변화된 IT 환경**을 감안하여 규제 적정성 등 **종합적 재검토** 필요

① ('19.1월) **클라우드** 이용시 **물리적 망분리 예외** 허용, 단 소프트웨어 형태 클라우드(SaaS) 이용 불가
② ('22.11월) "**연구·개발 망분리 예외**" 허용, 다만 **제한조건**으로 인해 활용 미흡*
* ①개인신용정보 처리 불가, ②연구·개발망과 내부 업무망간 물리적 분리 등을 충족하는 연구개발 환경 구현 어려움
③ ('23.9월) **규제샌드박스**로 업무망 SaaS 이용을 **허용**했으나, **부가조건***으로 **활용 제한**
* ①고객 개인신용정보 처리 불가, ②일부 프로그램(보안, 개발, 대고객 프로그램 제외)에 한하여 허용 등

[금융분야 망분리 개선 로드맵 (금융위원회,금융감독원 2024.08) 발췌]

□ **(필요성)** 현행 **망분리** 규제를 **고수**할 경우 ①급격하게 **변화**하는 **IT환경**에서 **생존**하기 **어려우며**, ②**금융보안**의 **발전**을 오히려 **저해**할 우려

① **소프트웨어**(S/W) **시장**이 자체구축형 → 클라우드 기반의 구독형 **SaaS**로 빠르게 **전환**되고*, 특히 **AI, 보안** 관련 **S/W**는 대부분 **SaaS**로 제공

* SaaS(Software as a Service) 시장규모 : 글로벌('19년 187조 → '24년 396.7조), 국내('19년 0.69조 → '24년 1.62조)

- **인터넷** 연결을 **일괄 차단**하는 **현행 망분리** 규제에 따라 **SaaS** 이용 **제한**

② 기존 자체구축 환경에서 **망분리**는 높은 보안성을 보장했으나, **클라우드 환경** 下 오히려 **보안성 약화*** **요인**으로도 작용

* 클라우드 보안 솔루션 및 운영체제(OS) 프로그램 등의 보안패치 다수가 SaaS 형태로 배포되는 반면, 망분리 환경은 실시간 업데이트가 어려워 최신 위협에 대응 곤란

- 금융회사는 망분리만을 준수할 뿐 **해외 선진 보안체계 도입**에 **소홀***,

* (예) EU, 미국 등의 경우 제로트러스트(ZeroTrust) 기법에 대한 도입 논의가 활발한 반면, 국내 금융회사 등은 망분리 체계를 전제로 논의 미흡

- **일부 회사**는 **망분리**라는 **규제 그늘**에 숨어 **필요·최소한**의 **보안체계**도 적절히 **갖추지 않는** 등 오히려 **보안수준**이 **낮아지는 부작용**

- **망규제 , 클라우드, AI 등 사용 규제로 IT 갈라파고스 등장**
- **국제금융에 기반한 경쟁력 확보가 우선**

국내 망분리와 해외 네트워크 세분화 비교

[국내금융환경]

[해외금융환경]

[금융분야 망분리 개선 로드맵 (금융위원회,금융감독원 2024.08) 발췌]

규제완화의 내용 – 망분리개선안

[금융분야 망분리 개선 로드맵 (금융위원회,금융감독원 2024.08) 발췌]

목적

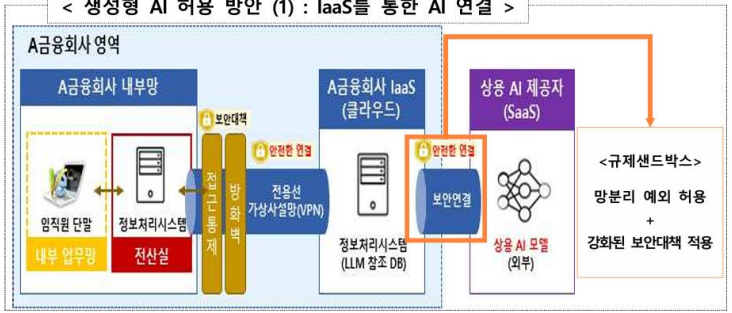
클라우드 기반의 응용프로그램(SaaS)활용도 제고

연구개발분야 망분리 규제 개선

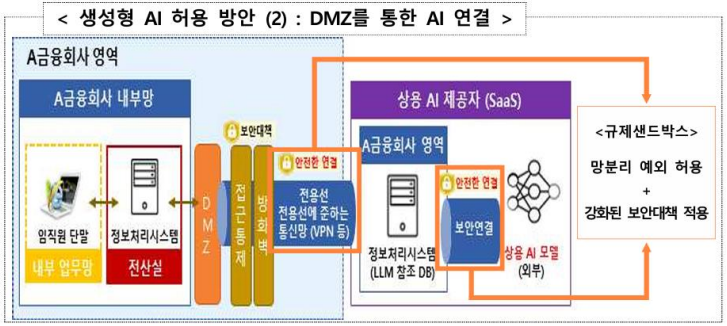
01	고객관리, 보안관리, 업무자동화 등 SaaS 활용 제고
02	가명정보 활용한 AI, SaaS 등 활용 제고
03	연구개발망과 업무망과 물리적 분리를 논리적 분리로 변경하여, AI 활용 허용을 제고함
04	연구개발망 등에서 개발한 결과물을 처리계에 이관할 수 있음

규제완화의 내용 – SaaS 활용안

< 생성형 AI 허용 방안 (1) : IaaS를 통한 AI 연결 >



< 생성형 AI 허용 방안 (2) : DMZ를 통한 AI 연결 >



[금융분야 망분리 개선 로드맵 (금융위원회,금융감독원 2024.08) 발췌]

생성형 AI 활용 허용

IaaS 를 통한 AI 연결

금융사 내부의 IaaS에 있는 정보계를 보안연결해서 SaaS AI 이용

DMZ를 통한 AI 연결

방화벽에서 전용선으로 SaaS AI 를 보안 연결하여 사용

규제완화의 내용 – 일정

구 분	~`24.3Q	~`24.4Q	~`25.上	~`25.下
[1] 생성형 AI 활용	규제샌드박스 접수 및 허용		규제샌드박스 성과 검증	규제샌드박스 정규 제도화
			규제샌드박스 추가 확대 방안 마련	규제샌드박스 추가 확대 접수 및 허용
[2] 업무망에서의 SaaS 이용	규제샌드박스 접수 및 허용		규제샌드박스 성과 검증	규제샌드박스 정규 제도화
			규제샌드박스 추가 확대 방안 마련	규제샌드박스 추가 확대 접수 및 허용
[3] SaaS 이용절차 개선	클라우드 이용절차 관련 「전자금융 감독규정」 개정			
[4] 연구개발 분야 망분리 개선	연구개발 관련 「전자금융 감독규정」 개정			
[5] 제3자 리스크 관리 강화	1단계	정보처리 업무위탁제도 정비방안 마련		
	2단계			
[6] 新 금융보안 체계 구축	연구용역 발주	「디지털 금융보안법(안)」 마련	「디지털 금융보안법(안)」 발의	입법 추진(계속)
			3단계	

[금융분야 망분리 개선 로드맵 (금융위원회,금융감독원 2024.08) 발췌]

☞ **자율보안-결과책임 원칙에 입각한 新 금융보안체계를 구축하겠습니다**

□ (현행) 세세한 **보안수단** 규정에 **열거**, "**규정**만 **지키면 면책**"이란 **인식** 만연

○ 금융회사는 **최소 기준**만을 준수할 뿐 적극적 **보안투자**에 **소홀**하고, **일률적·경직적** 규정으로 인해 IT 리스크에 **유연한 대응**이 **어려움**

□ (개선방향) 자율보안-결과책임 원칙에 입각한 **新금융보안체계 구축**을 위해 「**디지털금융보안법**(가칭)」을 **제정**, "규칙(Rule) → **원칙**(Principle)**중심**"으로 **규제 전환**

① **금융당국**은 법령을 통해 **주요 보안 원칙·목표**를 제시하고, **구체적·기술적 보안 통제**사항은 가이드로 **모범사례 제시**(준수 의무사항은 아님)

- **금융회사**는 업무환경, 인프라, 보안역량 등에 대한 **자체 리스크 평가**를 통해 **자율적**으로 **세부 보안통제**를 **구성**하고 당국에 **보고**

② 전산사고 등에 대한 **배상책임 강화**, 실효성 있는 **과징금 도입** 등 **금융회사**의 **책임 강화**를 위한 **법적 근거** 마련

- **중요 보안사항**의 **최고경영자**(CEO) 및 **이사회 보고**의무, **정보보호 최고책임자**(CISO) 역할 **확대** 등 금융회사의 **내부 보안체계**를 **강화**

③ **금융당국**은 금융회사의 **자율보안체계 수립·이행** 등을 **검증**하고,

- 점검 결과 **일정 수준 이하**의 **금융회사**의 경우 **보안수준 제고**를 위한 **시정요구·이행명령**을 부과, **불이행시 엄중제재** 및 **영업정지** 등 **조치**

- **2025년 부터 제도 및 입법 추진 예정**
- **금융회사별 자율보안체계 수립 이행 필요**

금융감독원 IT검사방향

	금융감독원	금융회사
IT감사・거버넌스 강화	• IT검사・거버넌스 체계 확립 요구 - CIO・CISO 간담회 - IT상시협의체 - IT부문 내부감사협의체	• 경영진 IT운영보고체계수립 - CIO-CISO 상호견제 - IT감사 및 자체감사 역량확충 - 정기적 감사 실시
자율시정 체계구축	① 맞춤형 자가진단 체크리스트 배포 ④ 자율점검 결과평가 및 高위험사 선정	② IT감사실시(자율점검/시정) ③ 점검결과(대표이사확인) 및 건의사항제출
핀포인트 IT검사 실시	• 핀포인트 수시감사 실시 • 기본사항 위반시 엄중제재 • 모범・취약사례 전파	• IT운영・통제체계 확립 - 기본사항 준수 및 자체점검 실시 • 전자금융사고억제력・복원력 제고

[금융분야 망분리 개선 로드맵 (금융위원회,금융감독원 2024.08) 발췌]

중점검사사항

중점검사사항

• 주요 전자금융사고 발생 원인인 제3자 리스크관리,전산시스템 성능관리,비상대책 및 프로그램 • 전산원장통제관리 등을 중점 점검

- IT감사•IT경영: IT감사조직구성,감사부서의 IT감사실적 및 정기적인 경영진 보고여부,시정처리의 적정성 등
- 전자금융사고대응체계 : 사고보고 및 대응체계의 적정성, 비상연락체계 관리 등
- 제3자 리스크관리 : 외부 단일장애지점,IT위탁•연계 서비스, SW공급망 등에 대한 인식(연관관계명세 등) 및 통제
- 전산시스템성능관리 : IPO,차세대 등 대형이벤트에 대비한 전산자원별 성능관리 임계치 설정 및 테스트 절차 준수 여부점검
- 프로그램•전산원장통제: 프로그램 •전산원장변경관리,제3자검증실시, 사전 영향도 분석 및 테스트, 절차준수 여부 점검 등
- IT부문비상대책수립•운용: 핵심업무선정•관리,대외기관연계 기능을 포함한 훈련실시, 전산센터화재예방 •대비

신규IT트렌드에 대한 선제적 대응

•IT신기술도입,제3자서비스증가 및 애자일 조직확산 등 금융권동향에 대한 신규IT리스크를 심층점검•모니터링하여 전자금융사고예방

(참고)주요금융IT 신기술활용사례및예상 리스크

신기술	활용	주요예상리스크
클라우드	IaaS,PaaS,SaaS	클라우드제공업체(CSP) 장애시 CSP내 대고객서비스 중단
데브옵스	S/W개발	• S/W 개발운영이 융합되어 각 직무간 경계•통제가 모호 • 자동화로 빠른 개발과 배포를 추구하여 통제 사각 발생
GPT	챗봇고객응대,투자조언,AI비서	GPT 사용으로 망분리,망간연결문제 등 신규 보안•장애 이슈

[금융분야 망분리 개선 로드맵 (금융위원회,금융감독원 2024.08) 발췌]

중점검사사항과 현실 GAP

현행 금융사 디지털금융 운영 현황

Legacy

Silo 시스템 모니터링 운영 (Legacy)
- Browser·AP·DB·Server 등 조직별 모니터링

Cloud Platform

신규 및 고도화 사업은 클라우드 & AI
- 클라우드 운영 플랫폼의 다양성 증대

Legacy & Cloud

클라우드 등 외부연계 확대증가
- Legacy 시스템-(PaaS,SaaS) 연계서비스 확대

Open Source S/W

오픈소스 활용한 S/W개발의 일상화
- 다양한 오픈소스의 활용으로 경험 부재의 오류 증가

Organization

DevOps 조직의 등장
- DevOps 개발/운영으로 MSA/Agile 증가

[주요 전자금융사고 발생 원인]

①IT감사·IT경영
②전자금융사고대응체계
③제3자 리스크관리
④전산시스템성능관리

[신규IT리스크 심층점검·모니터링]

① 클라우드 운용
② DevOps S/W개발
③ 애자일 조직 확산
④ 생성형 AI 연계

중점검사사항

금융사 모니터링 TO-DO

멀티 센터(클라우드) 통합 모니터링
- Legacy·각 클라우드 환경의 통합모니터링

오픈소스 소프트웨어 모니터링
- 다양한 오픈소스의 이해와 통합

DevOps를 고려한 모니터링
- 자동화, 빠른 개발·배포에 적응하고 경험 부족 보완

클라우드 네이티브 옵저버빌리티
- 컨테이너 마이크로 서비스의 통합 추적과 분석

K-금융 환경에서 모니터링 과제

Observability 확보하기

멀티 데이터센터(클라우드) 통합 모니터링

멀티 데이터센터와 클라우드 환경을 통합적으로 모니터링하면 네트워크 비용을 절감하고 효율적인 데이터 분석 및 보고 체계를 구축할 수 있습니다. 이를 통해 하이브리드 및 멀티클라우드 환경에서도 안정적이고 신뢰성 있는 운영이 가능합니다.

클라우드 인프라의 확산

금융 IT에서도 클라우드 인프라 활용이 지속적으로 확산

- 하이브리드 클라우드: 온프레미스 데이터센터와 클라우드를 혼합하여 운영
- 멀티 클라우드: AWS, Azure, GCP 등 다양한 클라우드 서비스 사용

데이터센터와 클라우드 통합 모니터링

클라우드 외부에서 직접 모니터링할 경우 네트워크 비용 증가 및 성능 저하 가능성이 존재하므로 데이터 수집 및 처리를 통합적으로 관리

- 개별 센터/클라우드 통합 모니터링
- 멀티 클라우드 통합 모니터링

하이브리드 클라우드

멀티 클라우드

개별 센터/클라우드 통합

- 각 데이터센터 및 클라우드 리전에 전용 모니터링 데이터 서버를 배치하여 수집
- 센터 내 데이터 전송으로 데이터 전송 최적화

멀티 클라우드 통합

- 대시보드, 알림 시스템, 보고서, 어카운트 및 권한을 중앙 통합 관리
- 분산 트랜잭션을 연결, 클라우드와 데이터센터 간 데이터 흐름 추적 및 분석.

오픈소스 소프트웨어의 확산

분산 시스템에서 데이터 통합 및 프로세스 관리를 위한
미들웨어

- Apache Kafka: 분산 스트리밍 플랫폼, 메시지 큐 및 로그 처리
- RabbitMQ: 메시지 브로커로 비동기 메시징 지원
- Redis: 인메모리 데이터 저장소로 캐시 및 메시지 큐로 사용
- Nginx: 리버스 프록시, 로드 밸런서 및 웹 서버 역할
- Tomcat: Java 기반 웹 애플리케이션 서버

비용성, 유연성, 확장성 측면에서 장점을 가진
관계형 데이터베이스

- PostgreSQL: 고성능, 확장성, ACID 준수
- MySQL: 경량화된 구조와 높은 성능
- MariaDB: MySQL의 포크 버전으로, MySQL과 호환 가능
- SQLite: 임베디드 데이터베이스로, 서버리스 환경에서 경량 데이터 저장소로 사용.

대규모 데이터 처리, 비정형 데이터, 고속 트랜잭션 처리
NO SQL 데이터베이스

- MongoDB: 문서 지향 데이터베이스로 JSON 유사 문서 저장
- Cassandra: 분산 컬럼형 데이터베이스로, 수평 확장성과 고가용성 제공
- Redis: 키-값 기반의 인메모리 데이터 저장소로, 빠른 응답 시간 제공. 세션 관리, 캐싱

오픈소스 확산으로 인한 성능 및 장애 관리 이슈

오픈소스 소프트웨어(OSS)의 확산은 전통적인 상용 소프트웨어에 비해 낮은 진입 장벽과 높은 유연성을 제공하지만, 운영 경험 부족으로 인해 성능 저하 및 장애 발생 가능성이 높아질 수 있습니다. 특히 전통적인 운영자들은 OSS의 특성과 복잡한 구성 관리에 익숙하지 않을 수 있으므로, 이에 따른 주요 모니터링 이슈를 성능과 장애 관리 중심으로 정리합니다.

주요 이슈 01

성능 관리

▸▸ 리소스 사용 최적화 부족

- OSS의 자원 소비 특성에 대한 경험 부족으로 CPU, 메모리, 디스크, 네트워크 사용이 최적화되지 않을 가능성

▸▸ 부하 테스트 및 확장성

- OSS를 도입한 시스템의 부하 처리 한계를 사전에 파악하지 못한 상태에서 운영 중 성능 저하 발생

▸▸ I/O 병목 현상

- OSS가 데이터베이스, 파일 시스템, 네트워크를 과도하게 사용할 경우 발생하는 I/O 병목

주요 이슈 02

장애 관리

▸▸ 구성 변경 관리 부족

- 운영자들이 OSS의 구성 옵션 및 업데이트 관리 경험이 부족하여, 잘못된 변경으로 인해 장애 발생 가능

▸▸ 디버깅 및 원인 분석 한계

- OSS 장애 발생 시 로그와 진단 정보를 분석하는 운영 경험 부족

▸▸ OSS 의존성 및 버전 호환성

- OSS 간 종속성 충돌로 인해 성능 저하 또는 장애 발생 가능

주요 이슈 03

개발 배포

▸▸ 운영 초기에 사용하지 않던 OSS가 유지보수 중간에 추가

- 시스템 모니터링과 네트워크 영역과 통합 모니터링

▸▸ 애플리케이션 내부에서 OSS 컴포넌트가 추가되면 운영자는 알 수 없음

- 빌드스크립트(POM.XML) 수정만으로 새로운 OSS 컴포넌트가 추가 가능, 그러나 운영자는 인지 할 수 없음

DevOps 조직에서 통합 운영하며 고려해야 하는 모니터링

DevOps 환경에서는 개발(Development)과 운영(Operation) 간의 경계를 허물어 지속적인 통합(CI), 지속적인 배포(CD), 자동화된 운영을 가능하게 합니다.
이를 지원하기 위한 모니터링 솔루션은 성능, 안정성, 효율성 뿐만 아니라 협업과 빠른 문제 해결을 위한 통합적인 기능을 갖추어야 합니다.

01 전구간 커버리지

전체 스택 모니터링

브라우저부터 데이터베이스까지 서버, 네트워크, 쿠버네티스, 애플리케이션, 로그 등 전체 구성 요소를 모니터링

분산 아키텍처 지원

마이크로서비스, 컨테이너 기반 인프라 등 복잡한 환경에서도 통합된 관찰 가능성 제공

사용자 경험 관점의 모니터링

실제 사용자 활동(Real User Monitoring, RUM) 및 시뮬레이션 기반 부하 테스트

02 진화적 기능 확장

지속적인 기능 확장

새로운 소프트웨어, 서버, OSS 컴포넌트를 위한 모니터링 기능의 빠른 업데이트 및 확장

OSS 컴포넌트 지원 강화

최신 OSS 구성 요소에 대한 지속적인 업데이트 및 프레임워크(F/W) 지원

모듈화된 설계

필요에 따라 기능을 확장하거나 축소할 수 있는 유연한 모니터링 아키텍처

03 스케일링

수평적 확장

대규모 트래픽 환경에서 리소스 소비를 최소화하며 효율적으로 데이터 수집 및 분석 가능

부하 분산

워크로드 증가에 따라 모니터링 인프라 확장/축소 조정

멀티클라우드 및 하이브리드 환경 지원

클라우드 기반 인프라와 온프레미스 시스템 간 통합

04 데이터 보안

데이터 암호화

데이터 전송 시 TLS, 저장 시 AES-256 등 최신 암호화 기술 적용

접근 제어

역할 기반 접근 제어(RBAC) 및 다중 인증(MFA) 지원으로 보안 강화

모니터링 보안 관리

민감한 모니터링 데이터를 보호하고, 이상 접근 시 경고

05 자동화

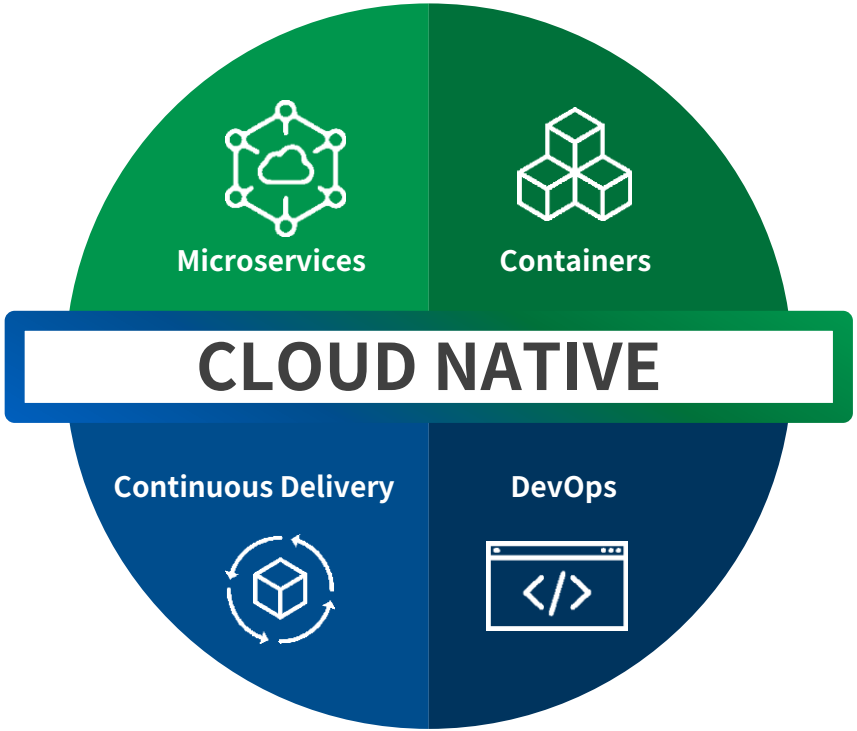
작업 자동화

반복적인 모니터링 작업을 자동화하여 운영 효율성 극대화

CI/CD 파이프라인 연동

지속적 통합/배포(CI/CD)에서 발생하는 성능 및 장애 데이터를 피드백 루프로 제공

클라우드 네이티브 아키텍처

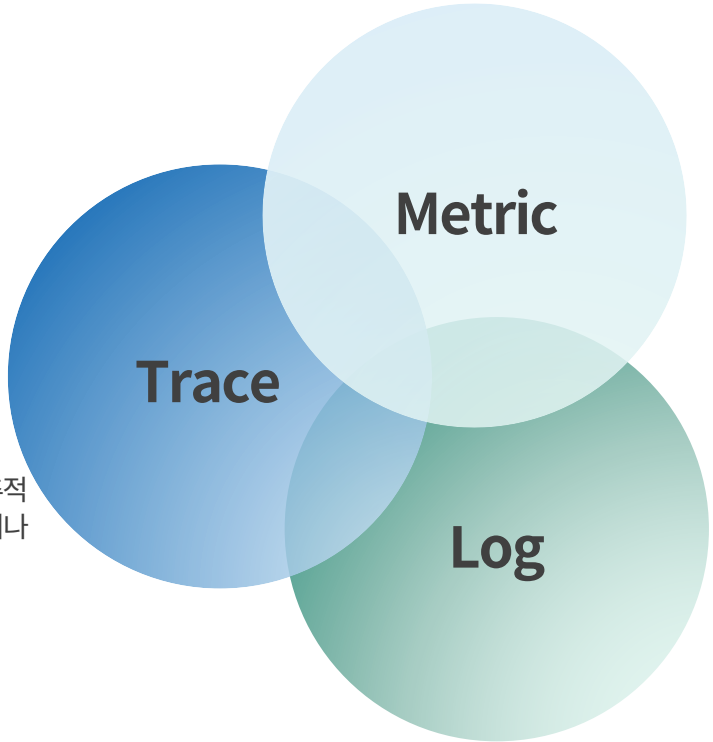


클라우드 네이티브 아키텍처의 목적
새로운 기술(AI, IOT, 블록 체인, 메타버스 등등)을 빠르게 비즈니스에 적용하자

Microservices	애플리케이션을 마이크로서비스로 분할하고
Containers	컨테이너를 적용하고 쿠버네티스로 관리하고
Continuous Delivery	지속적으로 빠르게 배포하고
DevOps	이 전반의 변화를 DevOps에 의해 관리하고 자동화 하자

옵저버빌리티

옵저버빌리티(Observability)는 시스템의 출력 데이터를 분석하여 내부 상태를 파악할 수 있는 능력을 의미합니다. 이는 전통적인 모니터링을 넘어 시스템의 전반적인 동작을 더 깊이 이해하고 내부 및 외부 문제의 근본 원인을 분석할 수 있게 합니다.



Metrics, 다층적 성능 지표

시스템의 성능, 자원 사용량 또는 애플리케이션 상태를 측정하는 정량적인 데이터
예: CPU 사용량, 메모리 소비량, 요청 대기 시간

Traces, 분산 추적

요청 또는 트랜잭션이 시스템의 여러 컴포넌트를 통과하며 진행되는 과정을 추적
시스템 간 의존성을 이해하고 병목현상이나 오류 지점을 정확히 파악하는 데 필수

Logs, 다층적 비정형 로그

시스템 내에서 발생하는 개별 이벤트를 시간순으로 기록한 데이터
시스템 작동과 동작에 대한 맥락 있는 정보를 제공

옵저버빌리티 vs 모니터링

Monitoring

“시스템 성능을 관찰하고 상태를 추적하는 행위(동사)”

Observability

“시스템의 내부 상태를 분석하고 이해할 수 있는 능력(명사)”

옵저버빌리티는 모니터링을 통해서 확보

옵저버빌리티는

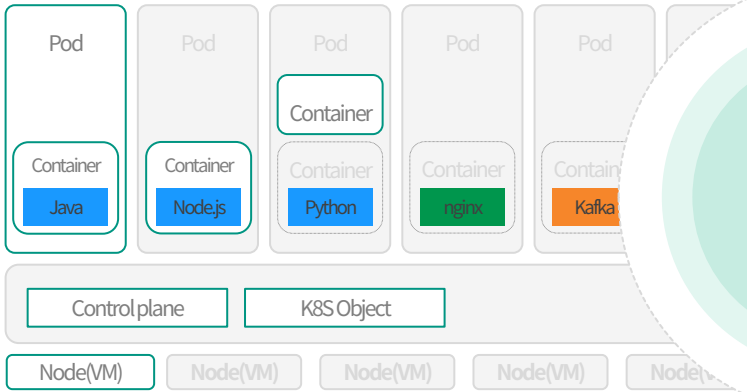
모던 IT시스템을 모니터링하기 위해 전략으로
쿠버네티스를 시작으로 발전

옵저버빌리티 구현을 위한 핵심 요건

01	**모든 영역에 대한 모니터링** 모든 IT컴포넌트가 적절한 수준에서 모니터링
02	**통합 모니터링** 모든 영역의 모니터링 데이터가 통합 분석
03	**지속적인 모니터링 업데이트** 시스템 변화 및 새로운 문제에 대응하기 위해 모니터링 능력을 지속적으로 수정하고 확장

쿠버네티스 모니터링(옵저버빌리티)

쿠버네티스 환경에서 효과적인 모니터링은 옵저버빌리티(Observability) 를 필수적으로 요구합니다. 특히, 컨테이너 중단시 로그 삭제와 분산된 구조로 인해 지표를 통합적으로 분석하는 체계가 필요합니다.



다양한 모니터링 대상

노드 레벨
클러스터를 구성하는 물리적/가상 머신의 CPU, 메모리, 디스크, 네트워크 상태를 모니터링
노드 장애 또는 리소스 부족 시 문제를 빠르게 감지

컨테이너/파드 레벨
컨테이너 및 파드의 상태, 리소스 사용량, 로그를 모니터링하여 워크로드의 성능 및 안정성을 확인

애플리케이션 레벨
컨테이너/파드 그룹에 따른 애플리케이션 모니터링 필요
애플리케이션의 응답 속도, 오류율, 트랜잭션 성능을 추적하여 사용자 경험과 직접적으로 연결된 문제를 식별

컨트롤플레인
kube-apiserver, etcd, kube-scheduler 등 핵심 구성 요소의 상태와 성능을 점검하여 클러스터 관리의 안정성 확보

로그

쿠버네티스의 동적 환경에서는 로그가 컨테이너와 함께 사라집니다.
로그 조회하기 위해 중앙 집중형 로그 관리가 필요하며,

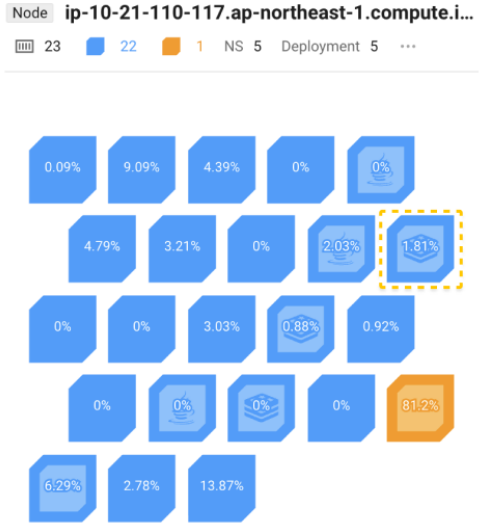
통합 분석

노드, 컨테이너/파드, 애플리케이션의 지표를 개별적으로 분석하는 것은 한계가 있습니다. 다양한 레벨의 데이터를 통합적으로 분석하여 상관 관계를 파악하고, 문제 원인을 정확히 진단하는 체계가 필요합니다.

쿠버네티스 외부 IT 컴포넌트 모니터링

쿠버네티스 모니터링은 클러스터 내부의 상태를 확인하는 것을 넘어,옵저버빌리티 관점에서 외부의 다양한 컴포넌트까지 통합 모니터링해야 함을 의미합니다.

모던 컴포넌트	OSS 미들웨어	Kafka, RabbitMQ
	NoSql 데이터베이스	Redis, MongoDB, Apache Cassandra, Couchbase
	Non-Java 애플리케이션	Node.js, Python, Go 등
	Spring Boot 및 경량 프레임워크	마이크로서비스와 경량 애플리케이션
	브라우저/모바일	사용자 경험
레거시 컴포넌트	데이터베이스 서버, 네트워크, 스토리지	오라클, PostgreSQL, MySQL 등 전통적인 IT 인프라 및 네트워크 관리 시스템
통합 모니터링	쿠버네티스 환경 확장 단일 뷰 제공 가시성 확보	외부 컴포넌트와 연계 증가 현대적(모던) 및 전통적(레거시) 시스템을 하나로 통합 전방위적인 시스템 상태를 추적하여 장애 및 성능 이슈 식별



옵저버빌리티에서 모니터링 데이터 폭증에 유연한 확장성

현대 IT 시스템에서 옵저버빌리티는 시스템의 상태와 성능을 깊이 이해하고 문제를 진단하는 데 필수적인 개념으로 자리 잡았습니다. 그러나 옵저버빌리티를 실현하는 과정에서 모니터링 데이터의 폭발적 증가라는 새로운 도전에 직면하고 있습니다.

컴팩트한 데이터 수집

- 중복제거 (해시처리)
- 역방향 트랜잭션 상세 (필요 시)
- 컴팩트한 자체 개발 데이터베이스

마이크로서비스로 인한 TPS 증가

- 단일 모놀리식 아키텍처에서 마이크로서비스 아키텍처로의 전환은 애플리케이션을 더 작고 독립적인 서비스
- 분산 아키텍처는 서비스 간의 상호작용이 늘어나며, 전체 **TPS(초당처리 요청 수)**가 기하급수적으로 증가
- 각 서비스에서 요청의 흐름을 추적하고 상태를 분석하기 위해 더 많은 메트릭스와 트레이스 데이터가 생성

Performance
UP

스케일어블 모니터링서버

- 클러스터링
- 스케일아웃과 샤딩
- 디스크 확장

로그데이터의 폭발적 증가

- 현대 시스템은 다양한 계층(애플리케이션, 인프라, 네트워크 등)에서 세분화된 로그 생성
- 특히 분산 시스템에서는 요청이 여러 컴포넌트를 거치며 생성되는 로그의 양이 방대
- 장애 발생 시 근본 원인을 찾기 위해서는 트랜잭션 로그와 상태 로그의 세부 분석이 필요하며, 이로 인해 로그 저장 및 처리 부담 증가

여러 이용자가 사용하기 때문에 내부 보안 강화 필요

옵저버빌리티를 위해 모니터링 통합되어야 합니다. 그런데 외부 클라우드나 AI 서비스를 사용하기 위해서는 망분리를 할 수없습니다. 따라서 통합 모니터링 자체에 대한 보안도 강화되어야 합니다.

데이터 암호화

모니터링 데이터 전송 암호화
모니터링 시스템에서 전송되는 데이터를 암호화하여 네트워크를 통한 데이터 탈취를 방지

계정 및 인증 보안 강화

패스워드 관리
- 패스워드의 정기적인 변경을 통해 보안 사고 위험 최소화
- 복잡한 패스워드 정책을 적용해 비밀번호 강도를 강화

다중 인증(MFA)

권한 관리 체계의 분산화

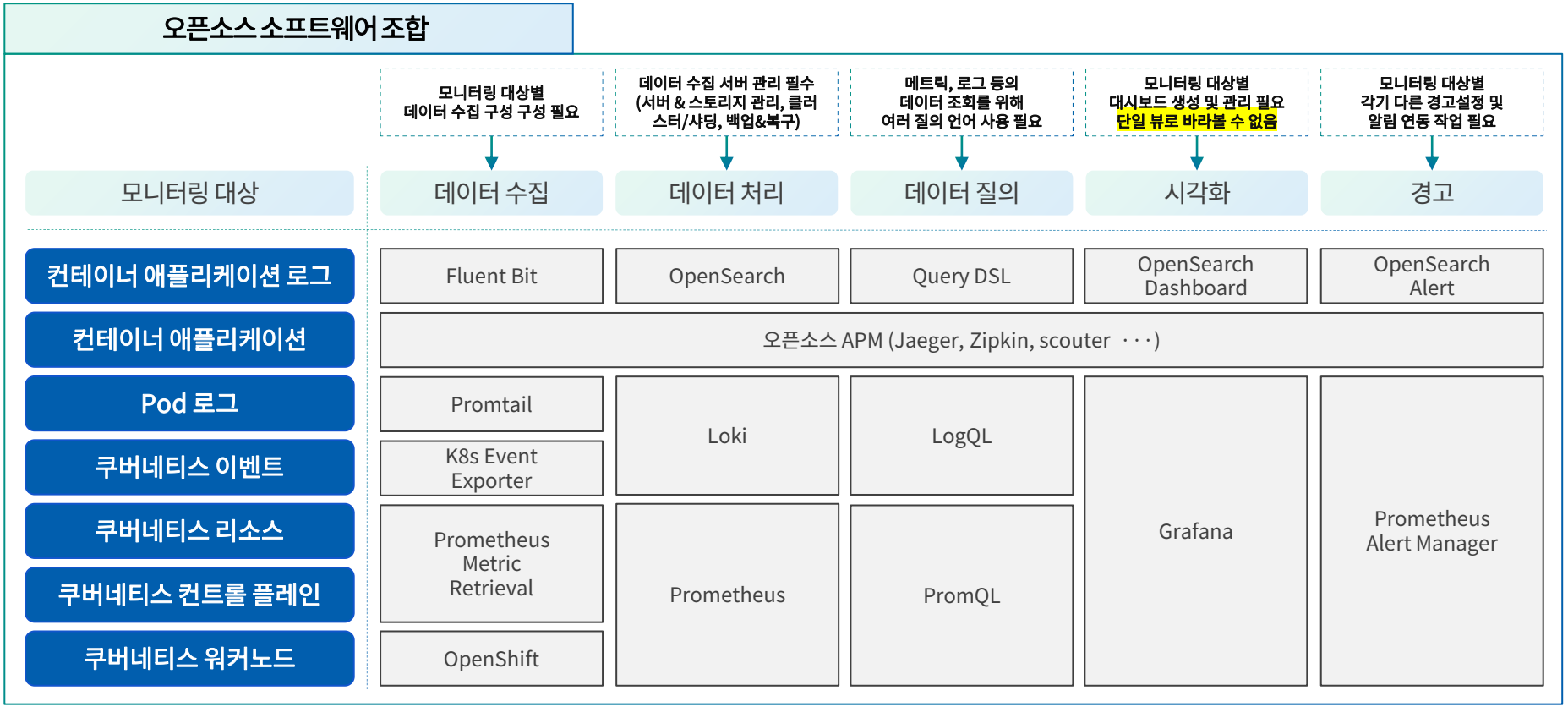
단일 관리자의 권한 집중 문제
- 하나의 슈퍼 유저(super user)가 모든 시스템을 통제하는 것은 보안상 취약점을 초래
- 단일 계정 해킹 시 전사적 시스템에 대한 접근 가능성이 발생

단위 시스템별 권한 관리
- 시스템 단위로 관리자를 분리하여 책임과 권한을 분산
- 각 관리자는 자신이 담당하는 시스템에만 접근할 수 있도록 제한

멀티 테넌트 관리 체계
- 여러 사용자가 동일한 모니터링 플랫폼을 사용할 경우, 사용자별 데이터와 권한을 명확히 분리
- 테넌트 기반 접근 제어로 불필요한 데이터 노출 방지
- 모니터링 프로젝트 Owner가 모든 내부 권한 통제, 전체 운영 Super User도 개별 데이터 조회 불가

클라우드 네이티브(쿠버네티스) 환경에서 옵저버빌리티를 확보하기 위해 해야 할 일들

오픈소스와 전문 모니터링 솔루션 조합의 경우



클라우드 네이티브(쿠버네티스) 환경에서 옵저버빌리티를 확보의 어려움

옵저버빌리티 플랫폼의 필요성 (옵저버빌리티가 플랫폼으로 구현되어야 하는 이유)

효과적인 모니터링과 운영 환경의 안정성을 유지하기 위해서는 옵저버빌리티를 통합 플랫폼으로 구현해야 합니다. 이는 급격히 증가하는 데이터와 다양한 모니터링 요구사항을 효율적으로 처리하고, 지속적으로 발전하는 IT 환경에 대응하기 위한 필수적인 접근 방식입니다.

옵저버빌리티 플랫폼에서 보안 로그

현대 IT 시스템에서는 옵저버빌리티와 보안(SIEM) 모두에서 로그 데이터를 필수적으로 활용합니다. 그러나 두 영역에서 별도로 로그를 수집할 경우 비효율성, 데이터 중복, 관리 복잡성이 발생할 수 있습니다. 이를 해결하기 위해 로그 수집의 일원화가 필요하며, 단일화된 로그 관리 체계는 성능 모니터링과 보안 위협 탐지를 동시에 충족할 수 있습니다.

보안을 위한 로그 분석 (SIEM)

보안 로그는 시스템 보호와 규정 준수를 위해 중요한 정보를 제공

위협 탐지 및 사고 대응을 위한 로그 수집 및 보안 이벤트 통합 관리
법적 요구사항과 보안 정책을 준수하기 위한 데이터 확보로 규정 준수 지원

옵저버빌리티를 위한 로그 분석

옵저버빌리티 로그는 시스템의 성능과 상태를 분석하는 데 중점

트랜잭션 추적 및 연계 분석으로 요청의 흐름을 추적하여 병목현상이나 장애 원인 파악
검색과 통계 기반 분석으로 시스템 이상 상태를 빠르게 탐지하고 성능 최적화

로그 수집의 일원화가 필요

운영 효율성

단일 로그 수집 체계로 관리 복잡성을 줄이고,
데이터 품질과 일관성을 보장

비용 절감

중복 로그 수집 제거로 스토리지 및 처리 비용 감소

다목적 활용

한 번 수집된 로그 데이터를 보안과 성능 분석
모두에 활용하여 통합된 가시성 제공

옵저버빌리티 플랫폼에서 Gen AI

데이터 통합 이후의 분석과 의사결정 과정을 단순화 | 문제 해결 속도를 크게 개선 | 실시간 데이터 분석과 예측 기능 | 비즈니스 연속성을 보장 | 효율적인 리소스 사용

Gen AI를 통한 모니터링 및 분석

Gen AI(생성형 인공지능)는 이러한 어려움을 해결할 수 있는 핵심 도구로, 다음과 같은 역할을 수행할 수 있습니다

데이터 정제 및 구조화
통합된 데이터를 정리하고 분석 가능한 형태로 변환

+

패턴 식별 및 예측
데이터 내 숨겨진 상관 관계와 이상 패턴을 자동으로 감지

+

실시간 알림
모니터링 시스템에 이상 상황이 발생하면 즉시 경고를 생성

+

설명 가능성 제공
분석 결과에 대한 자연어 설명을 통해 이해도를 높임

데이터 통합의 필요성

다양한 소스, 비정형 데이터
데이터 통합은 모니터링 뿐 아니라 분석의 기반

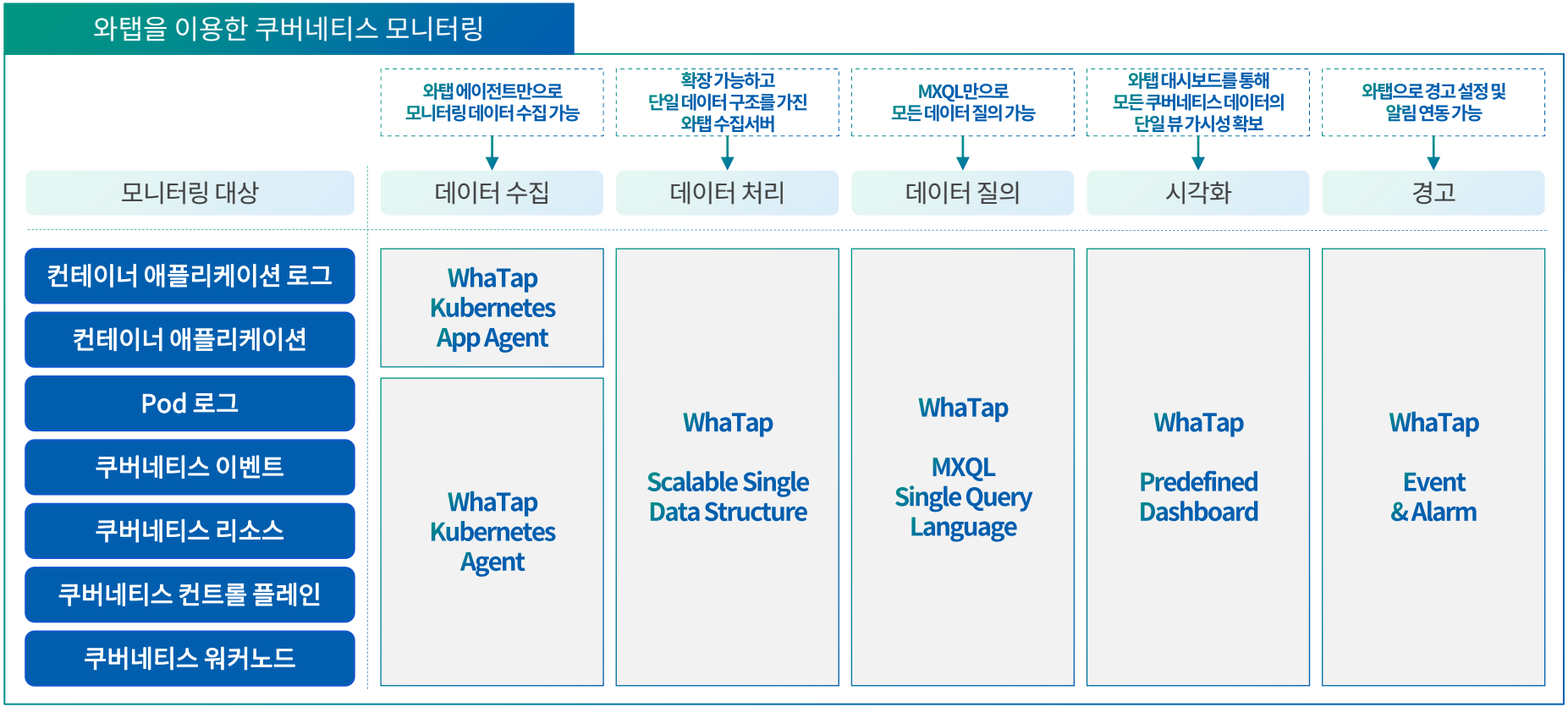
데이터 분석의 어려움

방대한 데이터의 패턴을 식별하거나 인사이트를 도출하기 위한
분석 작업의 기술적 제약 존재, 기존 방법론의 한계 존재

통합 옵저버빌리티 플랫폼 사용 예시

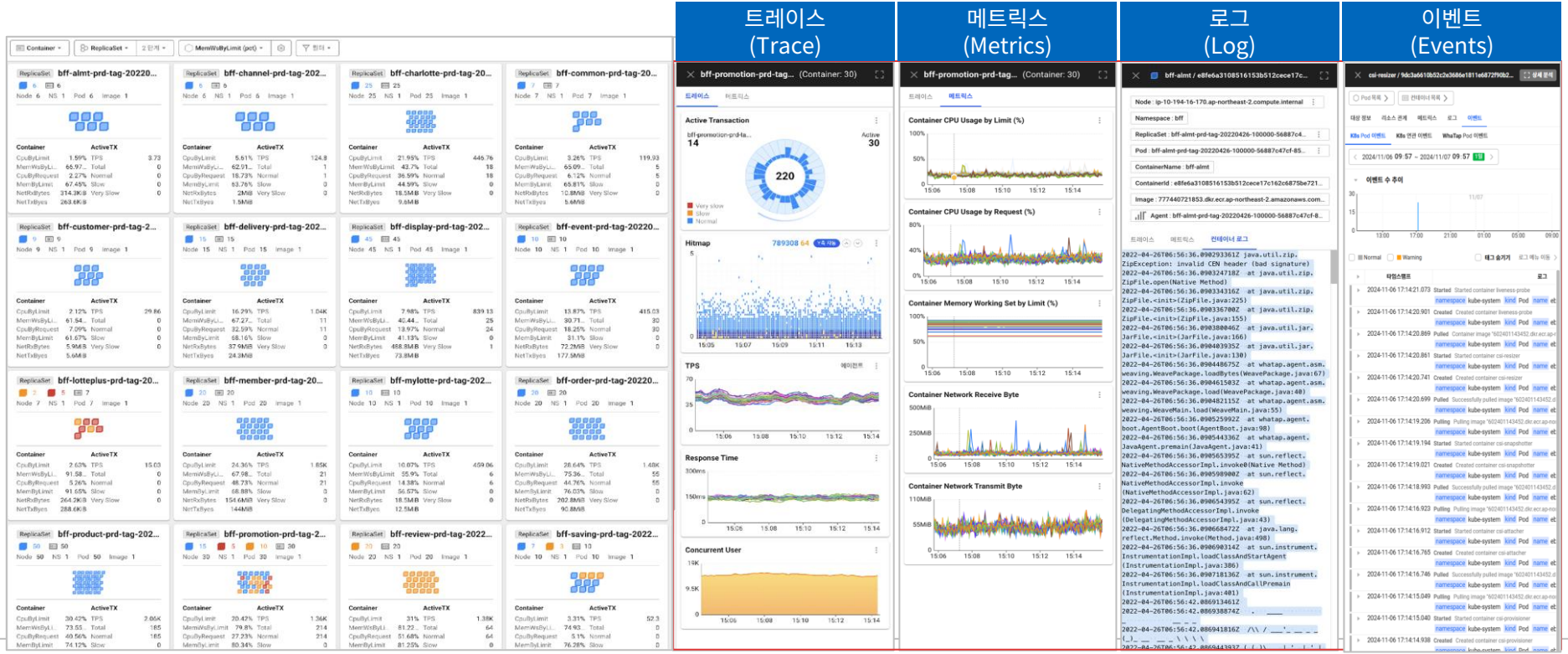
클라우드 네이티브(쿠버네티스) 환경에서 옵저버빌리티를 확보하기 위해 해야 할 일들

통합 옵저버빌리티 플랫폼을 이용하는 경우

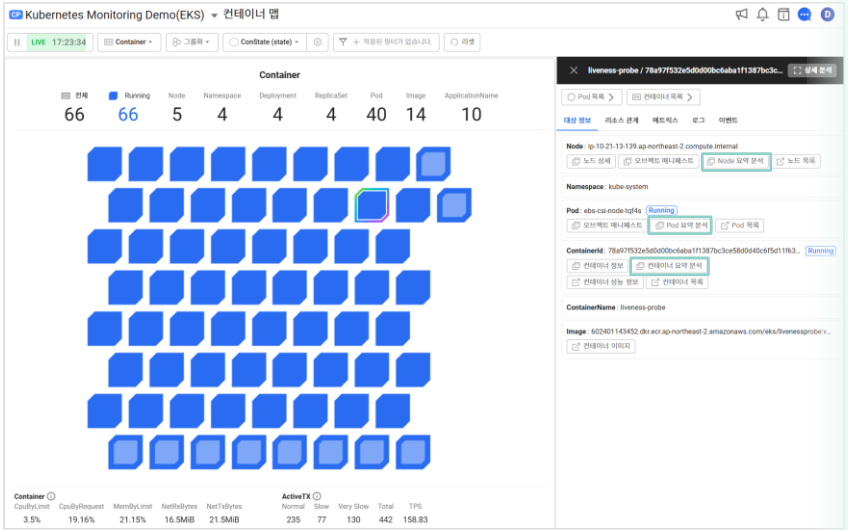


통합 옵저버빌리티 대시보드 예시 (와탭 솔루션 화면 예시)

쿠버네티스 환경에서 효과적인 모니터링은 옵저버빌리티(Observability) 를 필수적으로 요구합니다. 특히, 컨테이너 중단시 로그 삭제와 분산된 구조로 인해 지표를 통합적으로 분석하는 체계가 필요합니다.



DevOps 엔지니어를 위한 요약 분석



쿠버네티스 운영이 익숙하지 않은 사용자 분들을 위해
각 컨테이너, Pod, 노드에 대해 쉽게 말로 설명된 요약 분석 리포트를 제공

컨테이너 요약 분석

2024/11/07 17:25:17 복사

`78a97f532e5d0d00bc6aba1f1387bc3ce58d0d40c6f5d11f63e39caa93804ddf` (은)는 `v1.30.4-eks-a737599` 클러스터 및 `kube-system` 네임스페이스에 속하는 `ebs-csi-node-tqf4s` Pod 내에서 `2024/11/06 16:34:02` 에 기동된 컨테이너입니다.

컨테이너의 상태는 다음과 같습니다.

- 컨테이너 상태 : `RUNNING`

컨테이너 시작 후에 재기동된 적은 없습니다.

컨테이너 이미지는 `602401143452.dkr.ecr.ap-northeast-2.amazonaws.com/eks/livenessprobe@sha256:2fbd620d25bda985b20e07c56f1fae939158e3c469686389bae7c9fd426c8e49` 인데 모니터링 프로젝트 내에 이 이미지를 사용하여 실행 중인 컨테이너는 `7` 개이며, 해당 컨테이너들이 사용하고 있는 전체 CPU는 `0` millicore, 전체 메모리는 `48` MiB 입니다.

컨테이너가 실행되고 있는 노드는 `2024/11/06 16:34:27` 에 기동된 `ip-10-21-13-139.ap-northeast-2.compute.internal` 입니다. 이 노드의 OS는 `linux` , 커널 버전은 `5.10.219-208.866.amzn2.x86_64` , 아키텍처는 `amd64` , IP 주소는 `2406:da12:817:106::5144` 입니다. 노드의 CPU는 `2` 개(core)이고 메모리는 `1,924` MiB 입니다. 컨테이너 런타임 버전은 `containerd://1.7.11` 입니다.

그리고 이 노드는 Amazon EKS 상의 `SPOT` 타입으로 실행 중입니다.

Pod 요약 분석

2024/11/07 17:24:47 복사

`ebs-csi-node-tqf4s` (은)는 `v1.30.4-eks-a737599` 클러스터 및 `kube-system` 네임스페이스에 속하며 `2024/11/06 16:34:03` 에 기동된 Pod입니다. 현재 Phase는 `Running` 입니다.

최근 24시간 내에 이 Pod에 발생한 쿠버네티스 이벤트는 없습니다.

`ebs-csi-node-tqf4s` (은)는 `ebs-csi-node` DaemonSet에 의해 실행되었습니다.
`ebs-csi-node` DaemonSet에 의해 생성된 전체 Pod 수는 총 `5` 개 입니다.

Pod가 실행되고 있는 노드는 `2024/11/06 17:14:44` 에 기동된 `ip-10-21-11-241.ap-northeast-2.compute.internal` 입니다. 이 노드의 OS는 `linux` , 커널 버전은 `5.10.219-208.866.amzn2.x86_64` , 아키텍처는 `amd64` , IP 주소는 `2406:da12:817:105::1965` 입니다. 노드의 CPU는 `2` 개이고 메모리는 `1,924` MiB 입니다. 컨테이너 런타임 버전은 `containerd://1.7.11` 입니다.

그리고 이 노드는 Amazon EKS 상의 `SPOT` 타입으로 실행 중입니다.

Pod에 할당된 IP 주소는 `2406:da12:817:106:aa51::1` 입니다.

Pod 내에는 `3` 개의 컨테이너가 실행 중이며, 컨테이너는 각각 `ebs-plugin` , `liveness-probe` , `node-driver-registrar` 입니다.
단 애플리케이션 모니터링 에이전트가 설치된 컨테이너는 없습니다.

사용자별 화면 그룹핑 – 컨테이너뷰, 파드뷰, 노드뷰

그룹화 기능을 이용하여 사용자별 관심사에 따른 가시성 확보할 수 있습니다. 서버 담당자는 Node 상태를, 애플리케이션 담당자는 Deployment와 Pod 상태를 주로 모니터링합니다. 다양한 요구를 충족하기 위해 리소스를 그룹화하여 모니터링할 수 있는 기능이 제공됩니다.

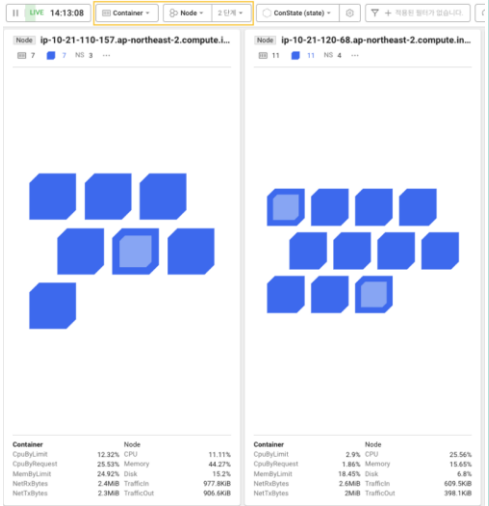
DevOps Engineer

Backend Developer

Product Manager

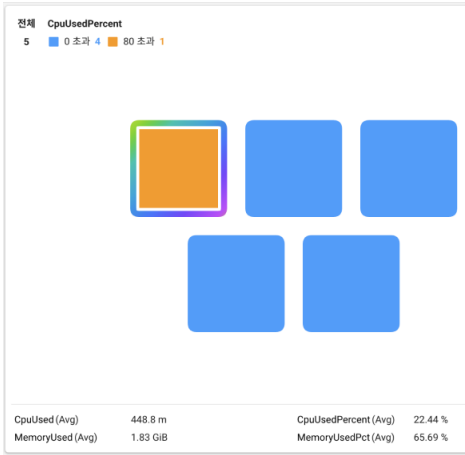
Frontend Developer

컨테이너 기준 뷰 + Node 그룹화

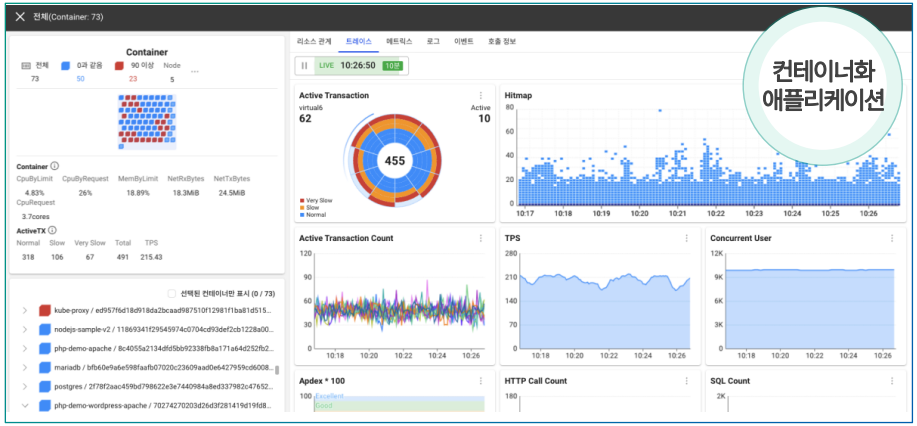


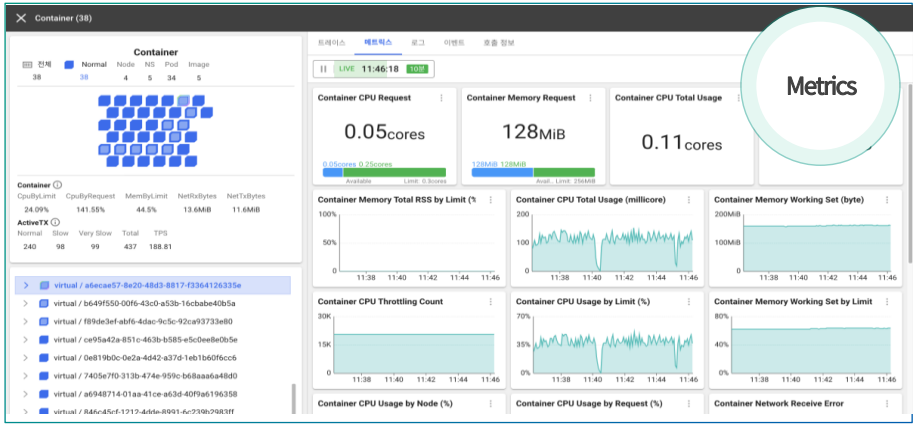
Pod 기준 뷰 + Deployment 그룹화

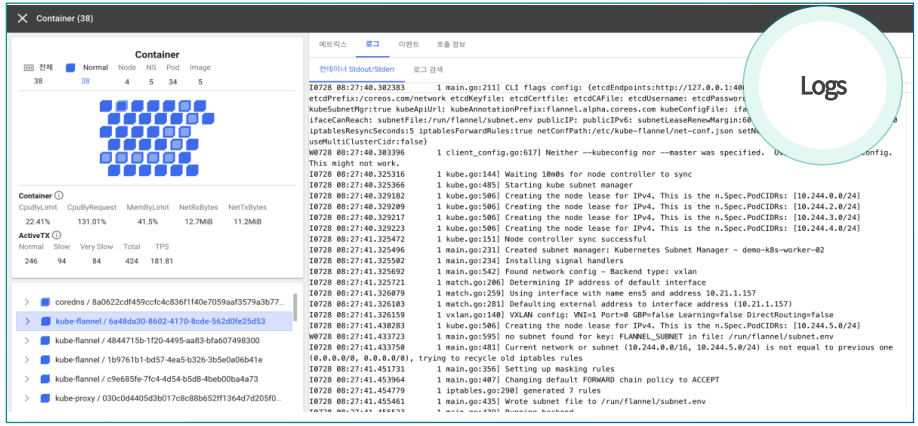
권한 관리 체계의 분산화



주요 화면 예시

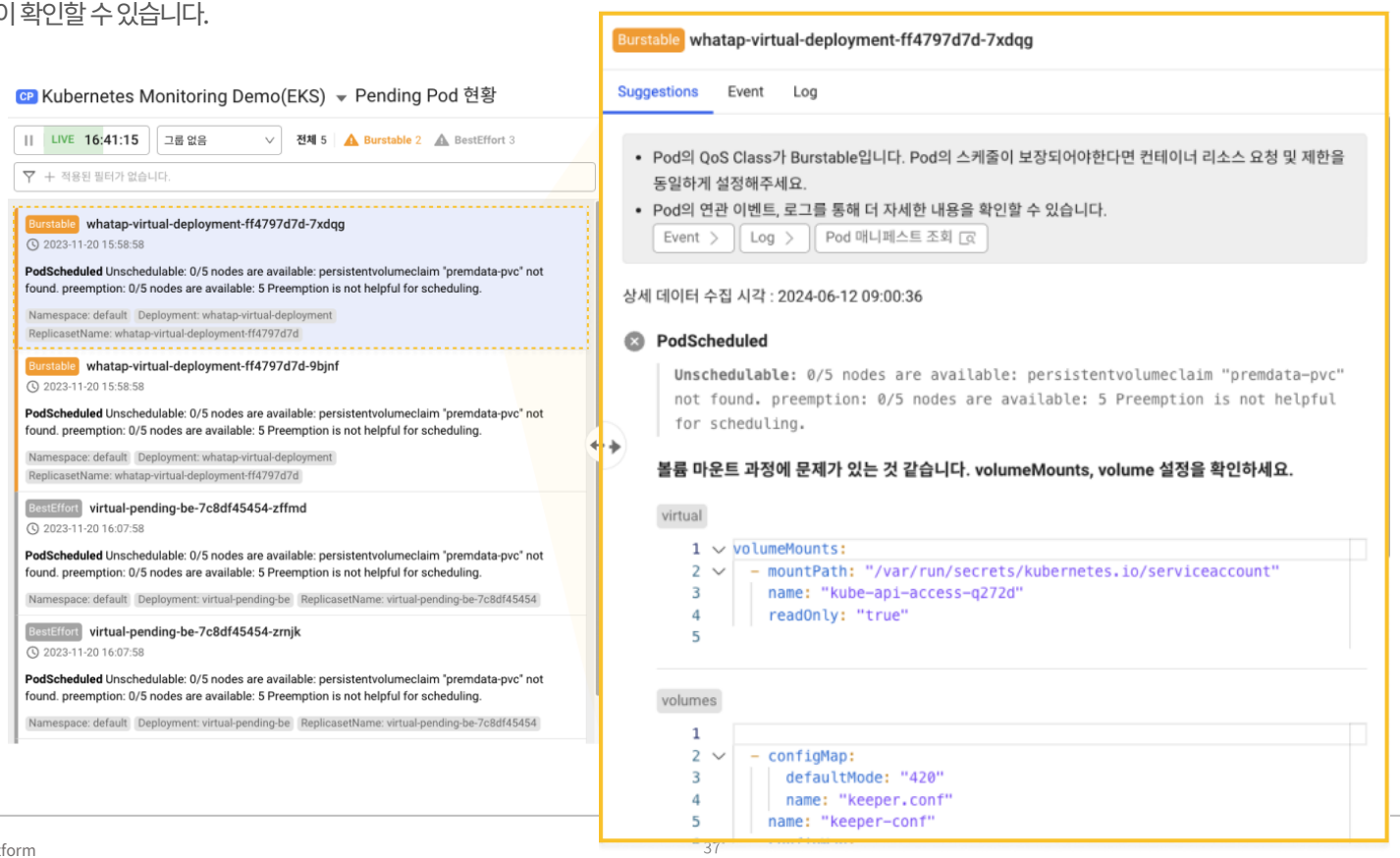






Gen AI 활용 예시, Pending Pods 분석의 Suggestion

Pending 상태의 Pod 현황을 조회하여 각 Pod의 대기 원인을 종합적으로 분석할 수 있습니다. Pending Pod 목록에서 Pod 선택 시 해당 Pod에 대한 상세 정보를 Suggestions, Event, Log 탭을 통해 다음과 같이 확인할 수 있습니다.



감사합니다.