MSAP.ai

AI Native 옵저버빌리티 세미나 · Session 02

MSA 장애의 원인 서비스 찾기

토폴로지 맵과 자동 RCA 로 후보를 좁히고

AI 로그 분석으로 확인하는 방법

SESSION

Session 02 · MSA 장애 원인 서비스 찾기

문제 · 해결 · 데모 · 정리



© MSAP.ai. All rights reserved.

Session 02 구성

KEY INSIGHT

장애 대응 시간이 원인 찾기에서 늘어나는 이유를 먼저 보고

화면 세 가지로 그 시간을 줄이는 과정을 데모로 확인함

01 문제

MSA 에서 원인 서비스를 찾기 어려운 이유 — 장애 전파, 재시도 증폭, 팀별 화면

02 해결

토폴로지 맵 · 자동 RCA · 한 화면 4개 탭 — 그리고 RCA 와 CogentAI 의 역할 구분

03 데모

주문이 안 되는 쇼핑몰 장애의 원인을 세 단계로 확인

04 정리

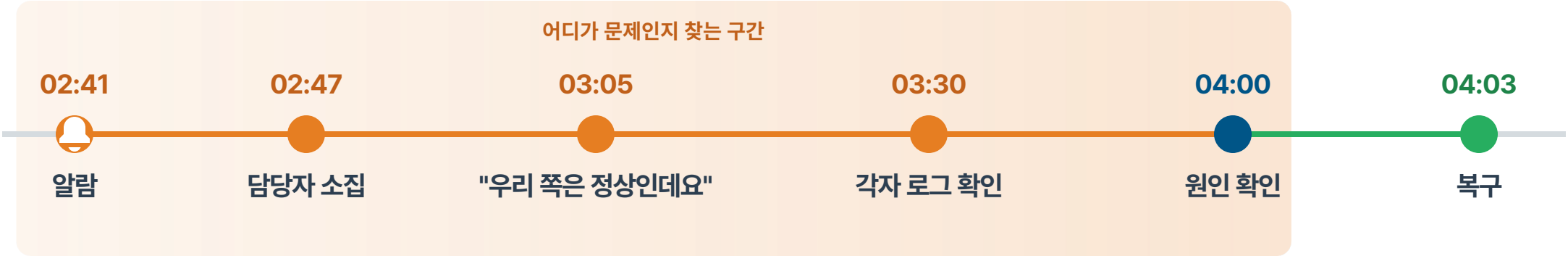
데모에서 찾은 원인과 원인 확인 절차, 다음 세션의 DNS·네트워크 진단

찾는 데 79분, 고치는 데 3분

KEY INSIGHT

장애 시간 대부분은 복구가 아니라 원인 위치를 찾는 데 쓰임

대응 시간을 줄이려면 찾는 단계를 줄여야 함



알람부터 복구까지 82분의 구성 (실제 길이 비율)

79분 어디가 문제인지 찾는 시간

실제 조치 **3분**

찾는 시간

담당자 소집 · 서로 확인 · 각자 로그 대조로 79분

고치는 시간

원인을 확인한 뒤 복구까지 3분

Source: 흔히 겪는 장애 대응 과정을 재구성한 예 (특정 고객 사례 아님)

업계 조사로 본 장애 대응 시간

KEY INSIGHT

인시던트 대부분은 코드 수정 없이 완화되지만 원인 범주를 알기까지 오래 걸림

진단 단계를 줄이는 쪽이 효과가 큼

175분

인시던트 평균 해결 시간

PagerDuty 2024
IT 리더 500명 설문

최대

50%

진단과 담당자 확인에 쓰이는 인시던트 시간

PagerDuty 2025
자사 데이터, 방법론 미공개

90% 이상

코드 변경 없이 완화된 고심각도 인시던트

롤백 · 설정 변경 · 재시작 등

Microsoft, SoCC 2022
Teams 고심각도 인시던트 152건

60%

원인이 코드가 아닌 인시던트

의존성 · 인프라 · 배포 · 설정

Microsoft, SoCC 2022
같은 152건 원인 분류

고치는 작업은 짧고, **원인 범주를 알아내는 진단 단계**가 대응 시간을 좌우함

Source: PagerDuty 설문 보도자료(2024-06, 미·영·호 IT 리더 500명) · PagerDuty 블로그(2025-08) · Ghosh 외, How to Fight Production Incidents?, ACM SoCC 2022

PART 01 · 문제

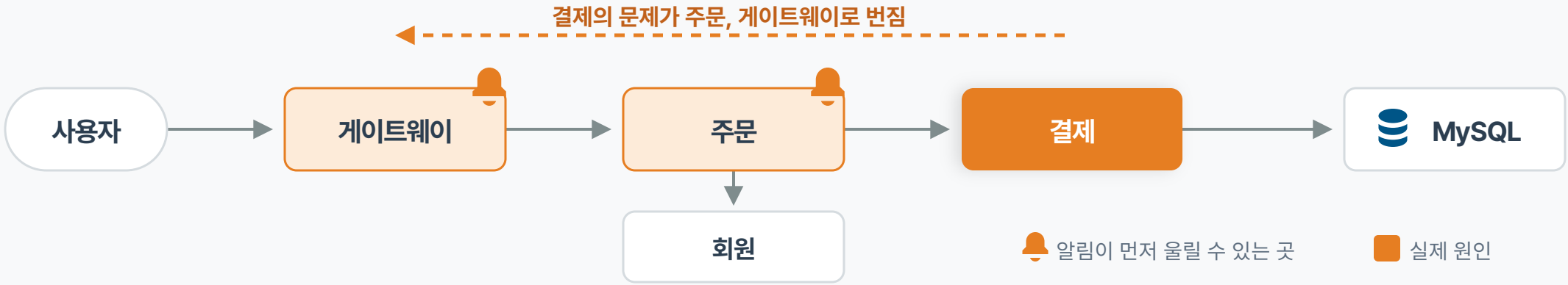
장애 전파와 알림 위치

요청 하나가 여러 서비스를 거칠 때 원인 서비스를 찾기 어려운 이유

MSA 에서 장애가 번지는 방향

KEY INSIGHT **장애는 호출하는 쪽으로 번져 알림이 먼저 울린 서비스가 원인이 아닐 수 있음**
요청이 거친 전체 경로를 봐야 원인을 가릴 수 있음

사용자 요청 하나의 호출 경로



01 서비스가 나뉨

기능마다 서비스를 따로 만들고
따로 배포함, 언어도 제각각

02 호출이 이어짐

사용자 요청 하나가
여러 서비스를 차례로 거침

03 장애가 번짐

한 서비스의 문제가 그 서비스를
부르는 쪽으로 전파됨

Source: 2026-10-28 클라우드네이티브TV 세미나 Session 02 발표자료

재시도 증폭과 2차 증상

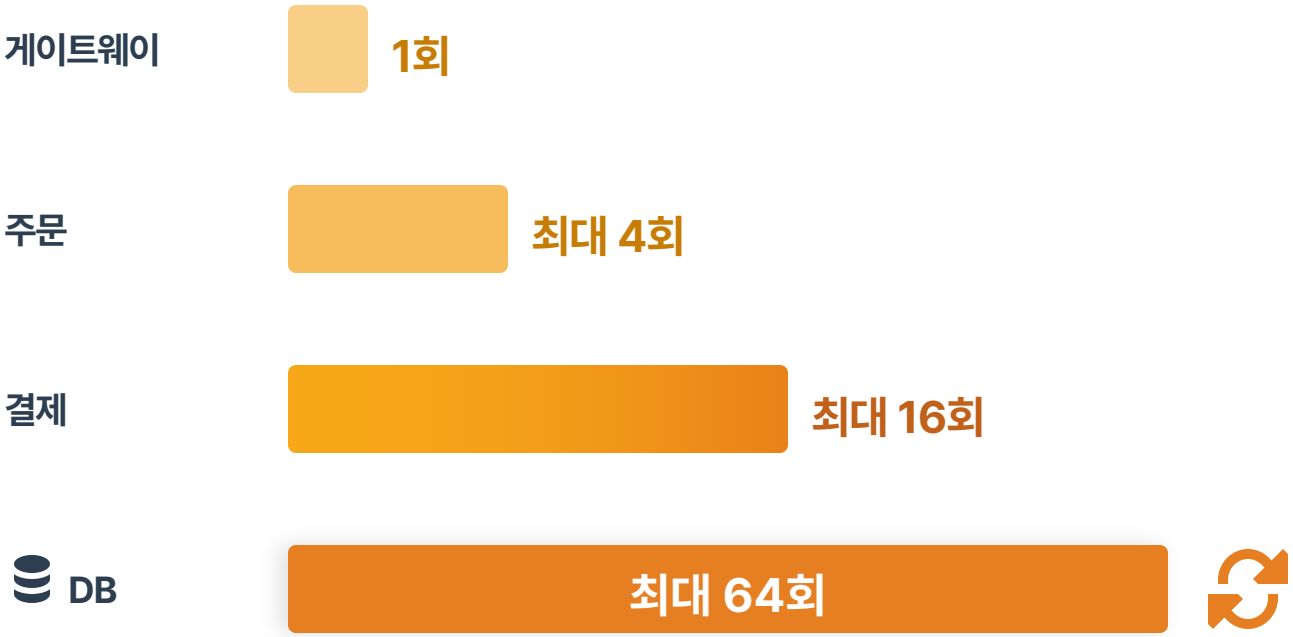
KEY INSIGHT

과부하가 만든 지연·헬스체크 실패가 처음 문제가 생긴 곳을 가림

알림 위치보다 먼저 이상해진 지점을 찾아야 함

사용자 동작 1번이 계층을 내려가며 늘어나는 시도 횟수

각 계층: 첫 시도 1회 + 재시도 3회



느려진 하위 서비스에 위 계층의 재시도가 곱해져 몰림

2차 증상이 원인을 가림

- 지연 증가와 헬스체크 실패
- 부하가 다른 서버로 옮겨 가며 연쇄 실패
- 증상이 보이는 곳과 원인이 있는 곳이 다름

국지 장애가 몇 분 안에 전체 장애로 커질 수 있음

담당 팀이 다르면 더 어려움

- 각 팀은 자기 서비스의 증상만 봄
- 원인이 사슬처럼 이어지면 장애 중 진단이 지연
- 상관관계를 인과로 착각하기 쉬움

Source: Google, Site Reliability Engineering 22장 Addressing Cascading Failures · 12장 Effective Troubleshooting (2016)

팀별 대시보드의 한계

KEY INSIGHT

담당자마다 다른 도구와 시간대를 보면 서로의 판단을 맞춰 보는 데 시간이 듦

같은 시간대의 전체 경로를 한 화면에서 봐야 대조 시간이 사라짐

서비스별로 보는 화면

게이트웨이 팀
도구 A · 14:00 기준



주문 팀
도구 B · 13:50 기준



결제 팀
도구 C · 13:45 기준



"우리 쪽은 정상인데요" × 3

- 팀마다 자기 서비스 지표만 확인
- 도구마다 서비스 이름과 시간대가 다름
- 서로의 화면을 맞춰 보는 회의가 길어짐

조직당 평균 **8개** 옵저버빌리티 도구 사용

요청 경로 전체를 보는 화면

같은 시간대 13:45 고정



세 서비스를 한 그림에서 비교

- 호출 관계 전체를 한 그림으로 확인
- 같은 시간대로 고정한 채 탭 이동
- 원인 후보를 근거와 함께 표시

대조하는 회의 없이 같은 그림을 보고 판단

Source: Grafana Labs Observability Survey 2025 (응답 1,255명, 조직당 평균 8개 기술) · 제품 문서 "제품 전체 — 왜 eBPF 기반인가"

PART 02 · 해결

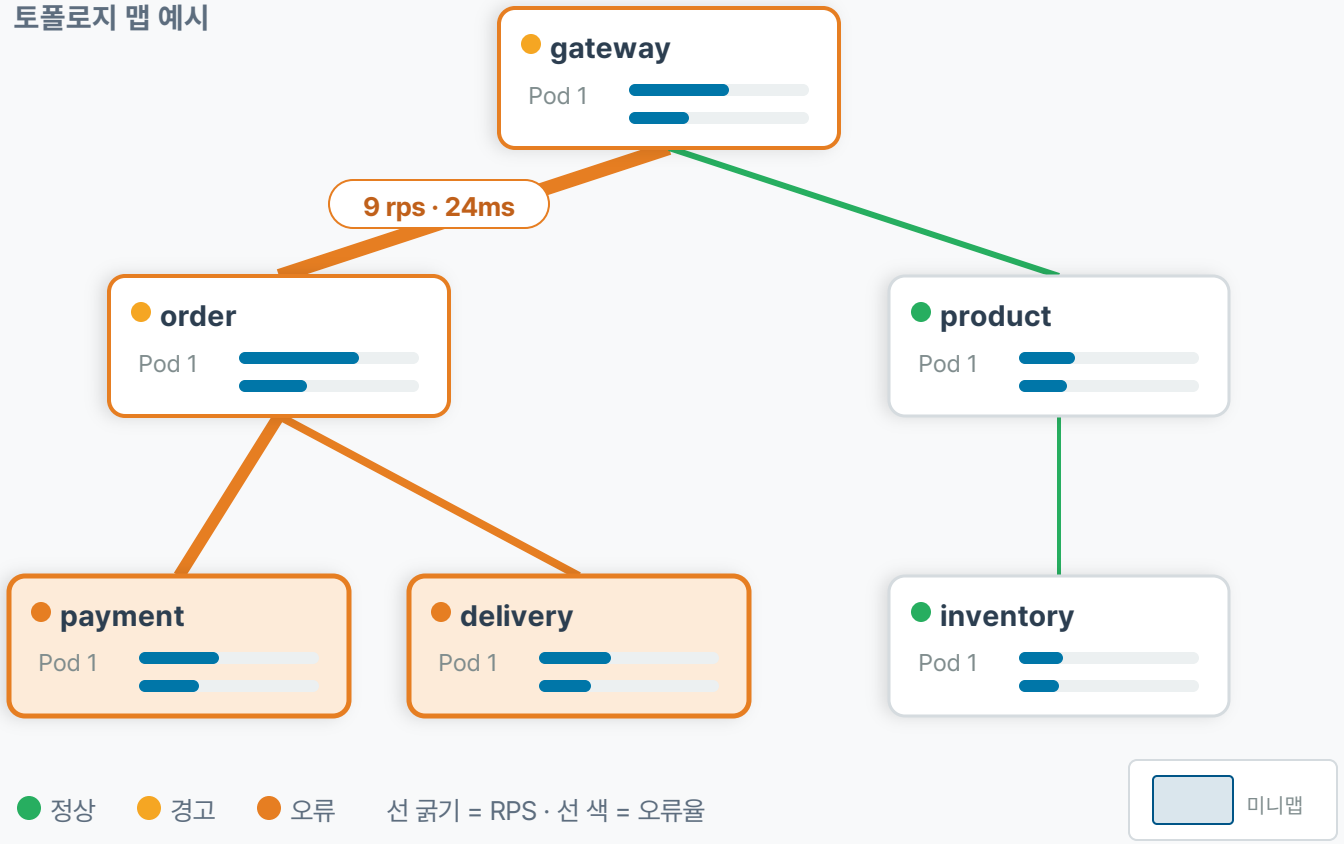
요청 경로 전체를 보는 세 가지 화면

토폴로지 맵 · 자동 RCA · 메트릭·로그·추적·프로파일링 통합 화면

토폴로지 맵의 표시 규칙

KEY INSIGHT **노드 에이전트가 eBPF 로 서비스 간 통신을 관찰해 맵을 자동으로 그림**
서비스를 등록하거나 관계를 설정하지 않아도 지금 상태가 반영됨

토폴로지 맵 예시



노드
서비스 이름 · 상태 · Pod 수 · CPU·메모리
게이지 75% 이상 노랑, 90% 이상 빨강

연결선
굵기는 초당 요청 수(RPS), 색은 오류율
오류율 1% 초과 노랑, 5% 초과 빨강

탐색
미니맵 · 네임스페이스 필터 · 이름 검색
처음 열 때 위험 노드로 자동 이동

갱신
10초마다 달라진 부분만 반영
새로 생기거나 사라진 서비스도 바로 표시

Source: MSAP.ai Observability 제품 문서 "토폴로지 맵" — 서비스 등록·관계 설정·코드 수정 없이 노드 에이전트가 소켓 통신을 관찰

인시던트 생성 기준: SLO 위반

KEY INSIGHT **경보 여러 건을 사건 하나로 묶어 대응자가 볼 대상을 하나로 줄임**
오류 예산 소비량이 대응 순서를 정함

같은 시각에 들어온 경보

- CPU
- 메모리
- 응답시간
- 오류율
- 로그
- 연결 실패

↓ SLO 위반으로 판정되면

인시던트 1건 자동 생성

가용성 · 응답시간 SLO 위반일 때만 열리고 자동 해결

경보 여섯 건이어도 사건은 하나

CPU 가 높아도 응답시간이 느려지지 않으면 인시던트 없음

1

인시던트

사용자가 받은 결과(SLO) 위반일 때 자동 생성
같은 시간대의 여러 증상을 한 건으로 묶음
원인이 사라지면 자동으로 해결 처리

2

알림

규칙 하나가 발동한 것 — 이상 탐지도 여기로 감
CPU · 메모리 · DNS · 로그 등 기본 규칙 약 40개
순간 튐은 대기 시간으로 걸러 냄

3

대응 순서

목록에 영향받은 요청 비율과 소비된 오류 예산 표시

같은 30분 인시던트라도

오류 예산 **12%** 와 **140%** 는 먼저 볼 순서가 다름

Source: MSAP.ai Observability 제품 문서 "인시던트·근본원인분석" — 인시던트를 여는 단일 기준은 SLO(가용성·응답시간) 위반

루트 원인 분석 탭의 다섯 구성

KEY INSIGHT

급할 때는 위쪽 요약과 전파 경로만 보고 근거가 필요하면 아래 보고서까지 내려감

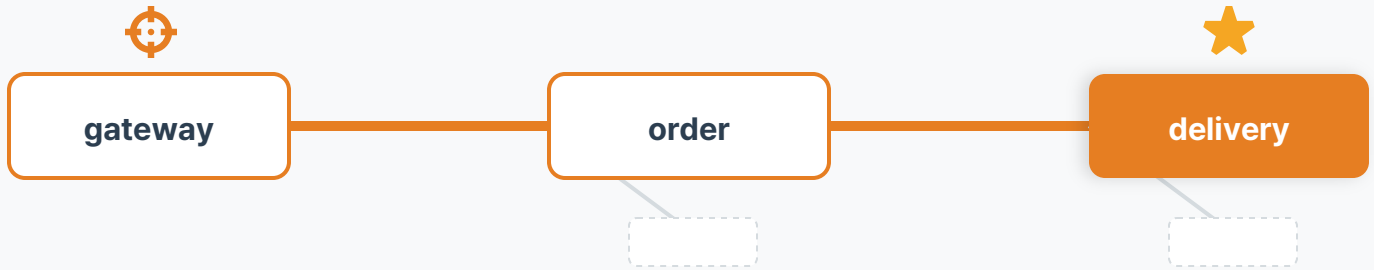
어디서 시작해 어디까지 번졌는지를 사람이 추측하기 전에 먼저 보여 줌

인시던트 → 인시던트 상세 → 루트 원인 분석 탭 · 위에서 아래로 읽는 순서

01 SLI	02 분석 요약	03 이슈 전파 경로	04 인과 타임라인	05 상세 보고서
인시던트 기간의 서비스 상태 차트	추정 근본 원인과 영향받은 서비스 수	**관련 서비스만 남긴 의존 관계 그림**	같은 이벤트를 시간순으로, 맨 위가 추정 근본 원인	원인 항목 트리와 항목별 근거 차트·로그

급할 때 보는 범위 — **근거가 필요할 때 내려가는 곳**

03 이슈 전파 경로 읽는 법



- 인시던트가 난 서비스
- 추정 근본 원인 서비스
- 둘을 잇는 경로 강조

호출하는 쪽을 따라 거꾸로 올라가 계산

Source: MSAP.ai Observability 제품 문서 "인시던트·근본원인분석" — 수집된 호출 관계만 그리며 감시 대상 밖 구간은 그림에 없음

RCA 의 원인 후보 선정 규칙

KEY INSIGHT **가장 먼저 발동한 판정을 근본 원인 후보로 올리고 계산에 언어 모델을 쓰지 않음**
같은 입력에 같은 결과가 나와 왜 그 후보인지 화면에서 되짚을 수 있음

근거 **애플리케이션마다 이미 돌고 있는 17개 판정**

SLO · CPU · 메모리 · 네트워크 · DNS · DB · JVM · 배포 등

↓

여섯 규칙으로 후보 모으기

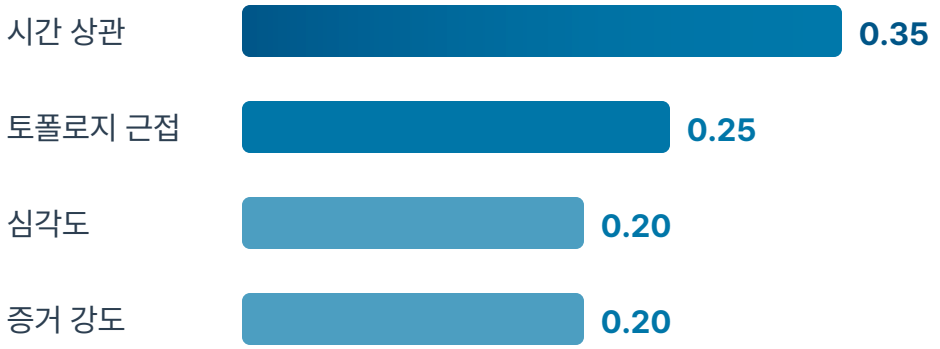
배포 상관	업스트림 장애	리소스 포화
로그 패턴	SLO 연쇄	DB 이슈

↓

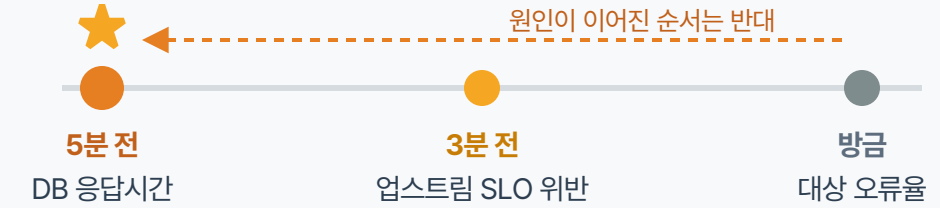
원인 후보와 순서

언어 모델 없이 규칙과 점수만 사용
애플리케이션 상세 화면과 같은 판정을 근거로 씀

후보 순서 = 네 요인 가중 합산



예: 판정이 발동한 시각



가장 먼저 발동한 판정이 근본 원인 후보가 됨

Source: MSAP.ai Observability 제품 문서 "인시던트·근본원인분석" — 17단계 감사 판정 · 여섯 규칙 · 네 요인 가중치

추정 근본 원인과 신뢰도 표시

KEY INSIGHT

RCA 결과는 확인할 후보이며 근거 화면으로 바로 이어짐

틀린 결론을 확신 있게 따르는 위험을 줄임

화면 표시 예

추정 근본 원인 delivery · 오류 로그 패턴 급증

신뢰도 높음 | 중간 | 낮음

신뢰도 = 근거가 갖춰진 정도

화면이 알려 주는 것

- 원인 항목 라벨은 "추정 근본 원인"
- 신뢰도 높음 · 중간 · 낮음 (근거 충족도)
- 항목을 누르면 근거 차트와 로그 표본
- 추적 탭이 그 시간대로 자동 필터링

급할 때는 후보를 보고, 확인할 때는 근거로 내려감

RCA 가 하지 않는 일

- 여섯 규칙 밖의 원인 찾기 (앱 내부 로직 결함 등)
- 상관관계를 인과로 확정하기
- 감시하지 않는 구간 그리기
- 코드 수준의 조치 방안 제시

"어느 서비스가, 언제부터"까지가 RCA 의 몫

Source: MSAP.ai Observability 제품 문서 "인시던트·근본원인분석" — 결과는 확인할 후보이며 정확도 수치는 제시하지 않음

서비스 상세 화면의 4개 탭

KEY INSIGHT

네 가지 데이터가 같은 시간 범위로 묶여 있어 도구 전환 없이 원인 코드까지 내려감

시각을 맞추고 대상을 다시 찾는 대조 시간이 사라짐

METRICS

메트릭

무엇이 이상한가

SLO · CPU · 메모리

네트워크 · DNS 탭

탭 이름 앞 상태 표시등

이상해진 시점 먼저 확인

LOGS

로그

무슨 일이 있었나

로그 뷰어 · 전문 검색

패턴 묶기 (3만 건 → 한 줄)

CogentAI 로그 분석

로그의 추적 보기로 이동

TRACES

분산추적

어디서 지연됐나

응답시간 히트맵

스팬 폭포수 차트

느린 요청 드래그로 조회

요청 1건이 거친 구간

PROFILING

프로파일링

어떤 코드가 느린가

플레임 그래프

함수 단위 CPU 비중

재빌드·재시작 없이 사용

원인 코드 위치

탭을 옮겨도 시간 범위 유지

메트릭 → 로그 → 분산추적 → 프로파일링

조사 화면은 주소(URL)로 공유

Source: MSAP.ai Observability 제품 문서 "애플리케이션" · "로그 뷰어" · "분산추적" — 메트릭·로그·추적·프로파일링을 하나의 플랫폼에서 통합

RCA 와 CogentAI 의 역할 구분

KEY INSIGHT

후보 선정은 규칙이, 로그 해석과 확인 순서는 AI 가 맡음

사람이 판단할 재료가 근거와 설명으로 함께 나옴

RCA (루트 원인 분석)

규칙 · 판정 기반

- 어느 서비스가, 언제부터 이상했나
- 원인 후보와 이슈 전파 경로
- 같은 입력에 같은 결과

결과물: 확인할 서비스와 시각

후보

확인

CogentAI 로그 분석

버튼 한 번으로 요청

- 긴 스택 트레이스를 표로 정리
- 무엇이 실패했는지 핵심 한 줄
- 확인 순서와 kubectl 명령 제시

결과물: 원인 설명과 점검 절차

예: DNS 조회 실패 로그를 분석하면 안내하는 점검 순서

1. 도메인·서비스 등록 여부
2. 클러스터 DNS 동작
3. 네트워크 정책 차단
4. Pod DNS 설정 · 재시작

Source: MSAP.ai Observability 제품 문서 "인시던트·근본원인분석" · 연구소 영상 해설 "CogentAI 로그 분석" — CogentAI 는 연동된 환경에서 표시

PART 03 · 데모

쇼핑몰 주문 장애 데모

주문이 안 되는 상황에서 토폴로지 맵 · RCA · 4개 탭으로 원인을 좁히는 과정

S2 쇼핑몰 장애 원인 추적 세 단계

KEY INSIGHT

알림에서 원인 서비스까지 도구를 한 번도 바꾸지 않음

시작점 확인, 후보 좁히기, 근거 확인이 한 제품 안에서 이어짐

상황

사용자는 "주문이 안 된다"고 함

오류율 **약 45%**

사용자 만족도 **절반**

심각 인시던트 **3건**



STEP 1

토폴로지 맵 · 인시던트

어디서 시작됐나

· 빨간 연결선이 있는 구간 확인

· 인시던트가 열린 순서 확인

STEP 2

RCA

원인 후보와 전파 경로

· 어느 서비스까지 번졌는지 확인

· 인과 타임라인과 상세 보고서 근거

STEP 3

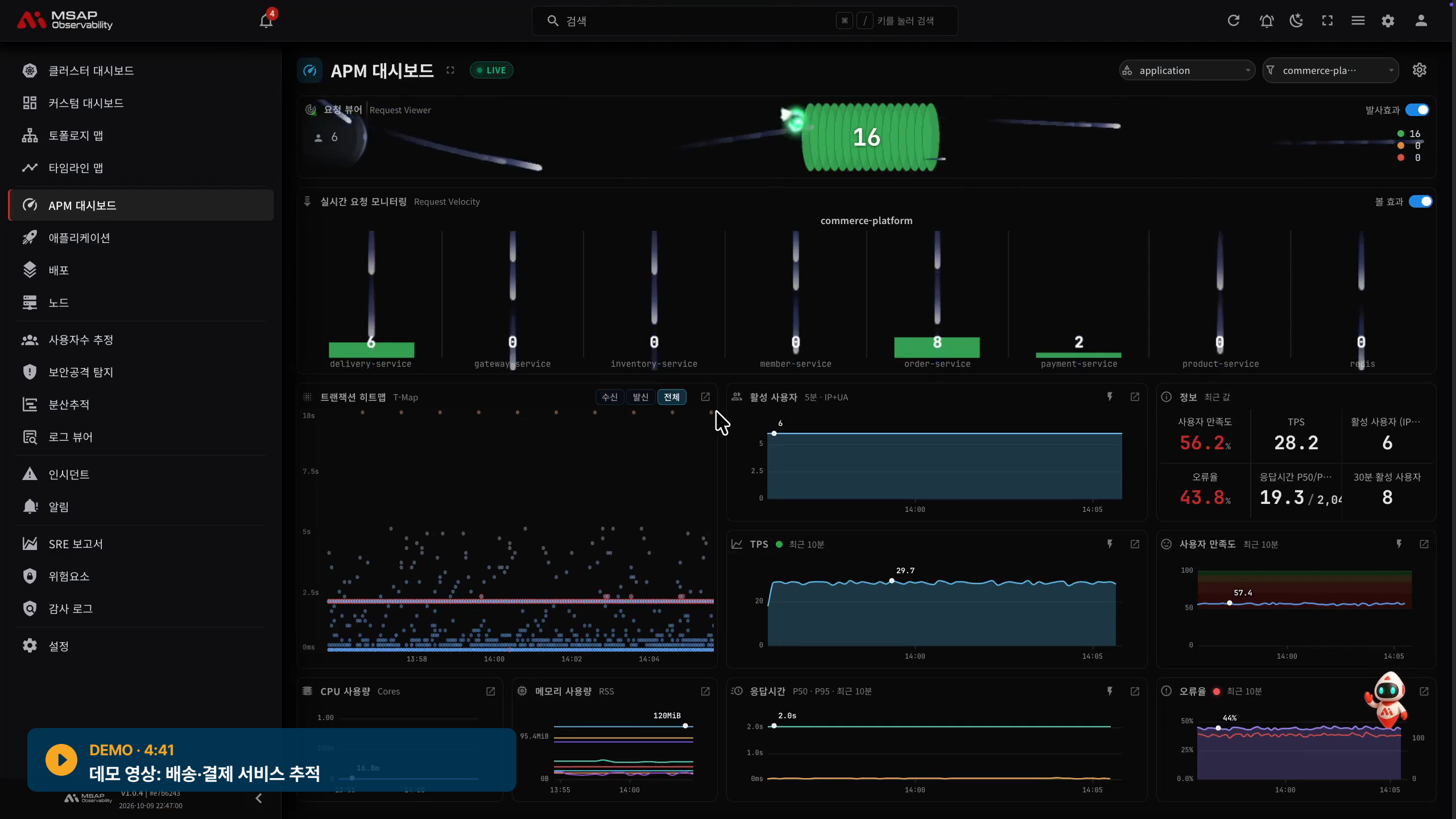
메트릭 · 로그 · CogentAI

배송·결제에서 원인 확인

· 네트워크·DNS 탭과 로그 패턴

· CogentAI 로 원인과 점검 순서

Source: 가상 쇼핑몰 데모 환경 (Session 01 데모와 같은 서비스 구성)


MSAP Observability
검색
⌘ / 키를 눌러 검색
클러스터 대시보드
커스텀 대시보드
토폴로지 맵
타임라인 맵
APM 대시보드
애플리케이션
배포
노드
사용자수 추정
보안공격 탐지
분산추적
로그 뷰어
인시던트
알림
SRE 보고서
위험요소
감사 로그
설정
APM 대시보드
LIVE
application
commerce-pla…
요청 뷰어 Request Viewer
6
16
발사효과
16
0
0
실시간 요청 모니터링 Request Velocity
볼 효과
commerce-platform
6
delivery-service
0
gateway-service
0
inventory-service
0
member-service
8
order-service
2
payment-service
0
product-service
0
redis
트랜잭션 히트맵 T-Map
수신
발신
전체
10s
7.5s
5s
2.5s
0ms
13:58
14:00
14:02
14:04
활성 사용자 5분 · IP+UA
6
5
2.5
0
14:00
14:05
정보 최근 값
사용자 만족도
56.2%
TPS
28.2
활성 사용자 (IP…
6
오류율
43.8%
응답시간 P50/P…
19.3 / 2,04
30분 활성 사용자
8
TPS 최근 10분
29.7
20
0
14:00
14:05
사용자 만족도 최근 10분
100
50
0
57.4
14:00
14:05
CPU 사용량 Cores
1.00
500m
0
16.8m
13:55
14:00
메모리 사용량 RSS
120MiB
95.4MiB
0B
13:55
14:00
응답시간 P50 · P95 · 최근 10분
2.0s
2.0s
1.0s
0ms
14:00
14:05
오류율 최근 10분
50%
25%
0.0%
44%
100
0
14:00
14:05
DEMO · 4:41
데모 영상: 배송·결제 서비스 추적
MSAP Observability
v1.0.4 | #e7b6243
2026-10-09 22:47:00

PART 04 · 정리

원인 서비스 확인 절차

토폴로지 맵 · RCA · 4개 탭이 각각 맡은 일과 장애 원인

데모 장애의 원인은 외부 도메인 DNS 조회 실패

KEY INSIGHT

배송과 결제는 언어가 달라도 같은 원인으로 실패함

코드가 아니라 인프라 구간에서 시작된 장애도 같은 화면 흐름으로 짚을 수 있음

1 토폴로지 맵

게이트웨이 → 결제
구간의 연결선이
전부 오류 표시

2 인시던트 순서

인시던트 열린 순서
배송 → 주문 →
게이트웨이

3 RCA

배송 장애가
주문까지 영향

근거: 로그 이상과
SLO 연쇄

4 네트워크·DNS 탭

13:45부터 택배사
연결 시도 없음

도메인 조회가
서버 실패로 끝남

5 로그 · CogentAI

IP 는 비어 있고
실패 단계는 DNS

결제는 PG사 도메인
조회 실패로 판정

원인

13시 45분부터 외부 도메인 DNS 조회 실패

배송 → 택배사 도메인

결제 → PG사 도메인

Source: 가상 쇼핑몰 데모 환경 · Session 02 데모 영상 06:14~10:56 구간

원인을 근거로 확인하는 세 단계

KEY INSIGHT

담당자 경험과 추측에 기대던 조사가 화면에 남은 증거를 따라가는 절차로 바뀜

같은 장애에 필요한 사람과 시간이 줄어듦

단계	79분 사례의 방식 담당자 경험과 추측	오늘 본 방식 화면에 남은 증거를 따라감
시작점 어디서 시작됐나	각 팀이 자기 서비스 화면을 확인 "우리 쪽은 정상인데요"	**토폴로지 맵과 인시던트 순서로 확인** 오류 구간과 처음 열린 인시던트
원인 후보 어디가 의심되나	회의와 질문으로 추측 담당자 경험에 따라 순서가 달라짐	**RCA 가 근거와 함께 후보 제시** 이슈 전파 경로 · 인과 타임라인
원인 확인 왜 실패했나	도구를 오가며 로그를 대조 도구마다 시간대·이름을 다시 맞춤	**한 화면 탭 이동과 CogentAI 설명** 네트워크·DNS·로그 탭 → 점검 순서

Source: 흔히 겪는 장애 대응 과정을 재구성한 예와 가상 쇼핑몰 데모 비교

인프라 구간 장애와 DNS·네트워크 탭

KEY INSIGHT

코드에 문제가 없어도 서비스가 느려지거나 멈출 수 있음

응답 전에 끊긴 호출처럼 보이지 않는 곳까지 따로 짚어야 원인이 나옴

DNS·네트워크 탭이 보는 것

- TCP 왕복 시간과 연결 설정 지연
- 연결 실패와 실패한 목적지
- DNS 질의·응답 지연
- 도메인별 NXDOMAIN · ServFail 건수

원인의 60% 가 코드 밖에 있다는 조사도 있음

보이지 않는 구간

- 응답 전에 끊긴 외부 호출의 상세
- 연결이 막혀 요청이 없는 구간의 오류율
- 감시 대상 밖을 지나는 구간

이런 구간은 "연결 실패"와 지연 지표로 짚음

NEXT · SESSION 03

놓치기 쉬운 인프라 장애, DNS·네트워크 구간 지연 진단

코드에 문제가 없는데 서비스가 느려지거나 멈출 때 볼 곳

Source: MSAP.ai Observability 제품 문서 "제품 전체" · Ghosh 외, ACM SoCC 2022 (Teams 인시던트 152건 중 원인 60% 가 코드 외)

THANK YOU

도입 문의와 연락처

알림에서 원인 서비스까지 한 화면에서 확인하는 옵저버빌리티

MSAP.ai Observability

https://www.msap.ai/

02-6953-5427

hello@msap.ai

© MSAP.ai. All rights reserved.