AI 운영

AIOps

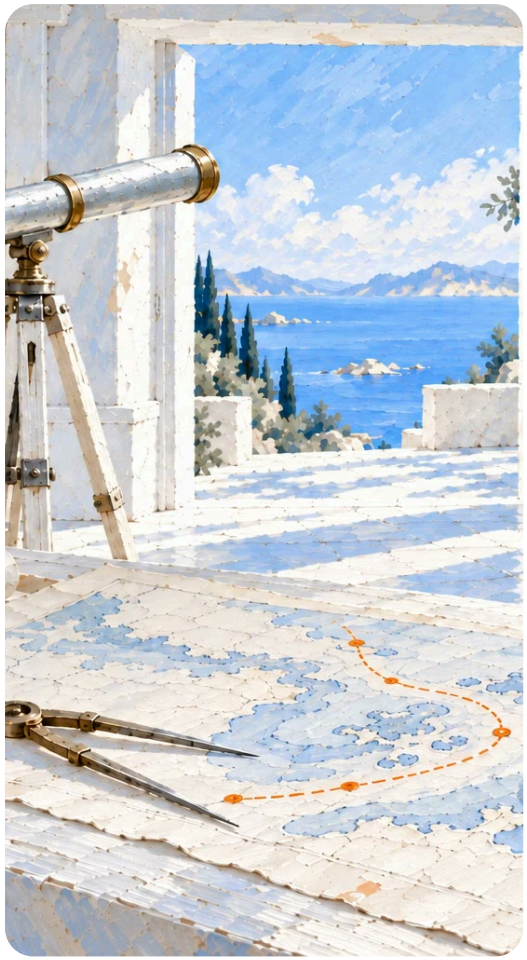
AI 기반 장애 예측 및 대응 솔루션

장애 조짐은 근거와 함께 먼저 알리고,
조치는 **귀사가 승인한 범위**에서 실행합니다.



이 브로슈어는 기능 목록이 아니라, 시스템을 운영하는 조직이 장애 앞에서 실제로 꺼내는 일곱 가지 고민에서 출발합니다.

고민 한 장, 해결 한 장씩
차례로 읽으실 수 있습니다.



들어가며

들어가며	장애 앞에서 운영 조직이 하는 일곱 가지 고민	p.03
솔루션	AIOps는 무엇이고, 어디에서 고민을 푸는가	p.04

일곱 가지 고민과 해결

고민 01	장애를 늘 사용자보다 늦게 안다	p.05
고민 02	알람이 너무 많아 진짜 원인이 묻힌다	p.07
고민 03	원인을 찾는 데 복구보다 오래 걸린다	p.09
고민 04	대응 노하우가 몇 사람에게만 있다	p.11
고민 05	AI가 운영 서버를 직접 건드리는 것이 불안하다	p.12
고민 06	AI의 판단을 무엇을 근거로 믿어야 할지 모른다	p.14
고민 07	지금 사용하는 모니터링 도구를 바꿔야 할까 걱정된다	p.16

적용과 도입

적용 예시	AIOps가 들어가면 달라지는 장애 대응	p.17
도입 순서	승인 모드로 시작해 넓혀 가는 도입 순서	p.19
문의	데모 영상 · 도입 문의	p.20

자주 듣는 고민

장애 앞에서 운영 조직이 가장 먼저 꺼내는 일곱 가지 고민

모니터링 도구는 충분히 갖췄는데, 장애는 왜 여전히 사람의 밤샘으로 해결될까요.

많은 장비를 적은 인력으로 운영하는 조직에서 반복해서 듣는 이야기를 모았습니다.



가장 마음에 걸리는 고민을 클릭하면, 그 고민을 AIOps가 어떻게 푸는지로 바로 이동합니다.

고민 01 "장애는 늘 사용자 전화로 먼저 알게 됩니다." 미리 알기 →

고민 02 "알람이 수백 개 울리면 무엇부터 봐야 할지 모릅니다." 알람 정리 →

고민 03 "원인을 찾는 데 복구보다 더 오래 걸립니다." 원인 분석 →

고민 04 "그 장애는 결국 그 사람만 압니다." 노하우 →

고민 05 "AI가 운영 서버를 직접 건드린다니, 더 불안합니다." 안전한 조치 →

고민 06 "AI가 틀리면요? 근거 없는 판단은 믿을 수 없습니다." 판단 근거 →

고민 07 "지금 사용하는 모니터링 도구를 다 바꿔야 하나요?" 기존 도구 →

하나의 흐름

AIOps는 이 고민들을 하나의 흐름으로 풉니다.

AI 기반 장애 예측 및 대응 솔루션

지금 사용하는 모니터링 위에 AI의 판단과 조치를 더해, 장애를 미리 알리고 발생하면 원인 분석부터 조치 · 회복 확인까지 이어 줍니다.



'고민' 아래 숫자 버튼을 클릭하면 해당 고민으로 이동합니다.

고민 01 · 사후 대응

장애는 늘 사용자 전화로 먼저 알게 됩니다.

인프라 운영 책임자에게서 가장 자주 듣는 이야기입니다.



현장에서는 이런 일이 생깁니다

알람은 선을 넘은 뒤에야 울립니다

디스크 · 커넥션 · 메모리는 며칠에 걸쳐 차오르지만, 고정 임계값은 선을 넘어서야 알려 줍니다.

조짐은 있었지만 아무도 보지 못했습니다

수백 대 장비의 추세를 사람이 매일 들여다볼 수는 없습니다. 장애가 난 뒤에야 그래프를 다시 열어 봅니다.

늦게 안 만큼 길게 멈춥니다

이상 징후가 생긴 시점과 대응을 시작한 시점 사이의 공백이 그대로 서비스 중단 시간이 됩니다.

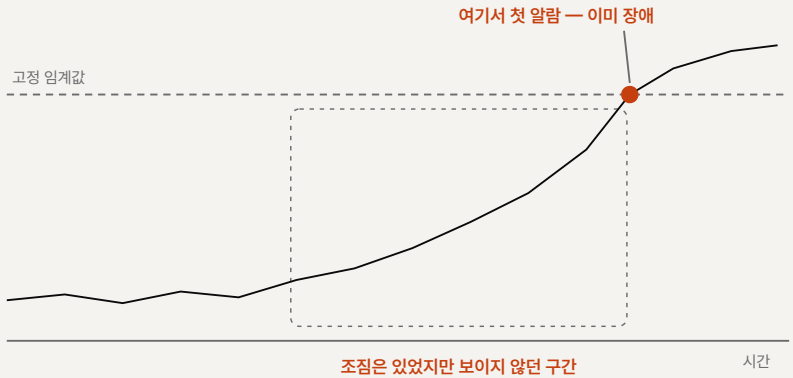
숨은 걱정 **"다음 장애도 결국 사고가 난 뒤에야 알게 되는 것"**

AIOps의 해결 보기 →

해결 01 · 임계에 닿기 전에

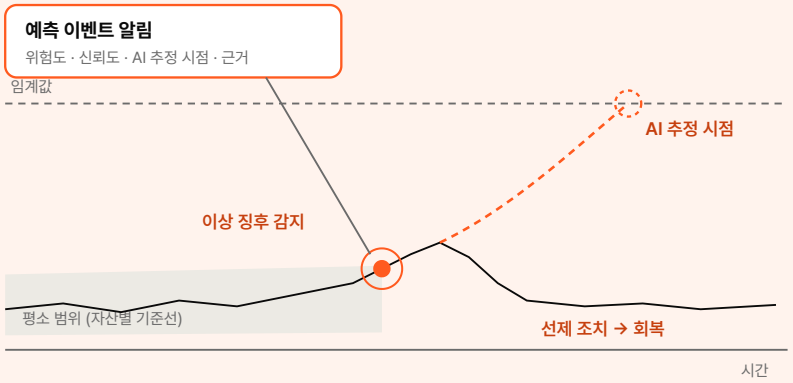
선을 넘은 뒤가 아니라,
선을 향해 가는 순간에 알립니다.

지금



알람은 임계를 넘은 뒤에야 울리고, 그 전의 조짐은 그냥 지나갑니다.

AIOps 도입 후



평소 범위를 벗어나는 순간 감지해, 장애 전에 조치할 시간을 확보합니다.

자산마다 다른 평소 범위

장비마다 평소의 움직임을 기준선으로 삼아, 얼마나 벗어났는지를 점수로 매깁니다.

추세 · 로그 · 과거 이력을 함께

오르는 추세에 관련 로그와 비슷했던 과거 장애를 겹쳐 보고, AI가 장애로 이어질지 판단합니다.

위험도 · 신뢰도 · 예상 시점

예측마다 위험도와 신뢰도, AI가 추정한 예상 시점을 함께 알립니다.

확인하고 미리 조치

운영자가 맞는 예측으로 판정하면, 검증해 둔 조치를 장애 전에 실행합니다.

고객이 얻는 것

- ✓ 장애가 나기 전에 대응을 시작합니다
- ✓ 매일 그래프를 들여다보지 않아도 됩니다
- ✓ 긴급 대응 대신 계획된 조치로 정리합니다

고민 02 · 알람 폭주

알람이 수백 개 울리면,
무엇부터 봐야 할지 모릅니다.

관제 · 당직 담당자들이 공통으로 하는 이야기입니다.



현장에서는 이런 일이 생깁니다

원인 하나가 알람 수십 개로 번집니다

DB 하나가 느려지면 그 위의 WAS · 웹 · 배치에서 알람이 한꺼번에 쏟아집니다.

중요한 알람이 노이즈에 묻힙니다

반복되는 경고에 익숙해지면, 정말 봐야 할 알람도 지나치게 됩니다.

같은 장애를 여러 사람이 따로 쫓습니다

담당자마다 다른 알람을 붙잡고 있어, 대응이 겹치거나 빈틈이 생깁니다.

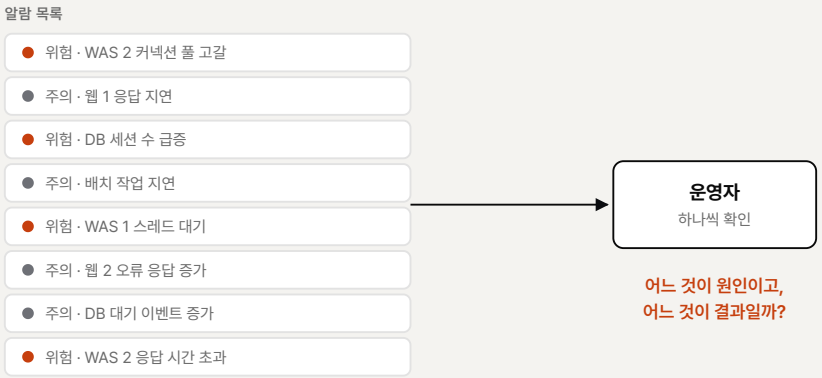
숨은 걱정 **“경보에 지쳐서 진짜 신호를 놓치는 것”**

AIOps의 해결 보기 →

해결 02 · 알람이 아니라 상황으로

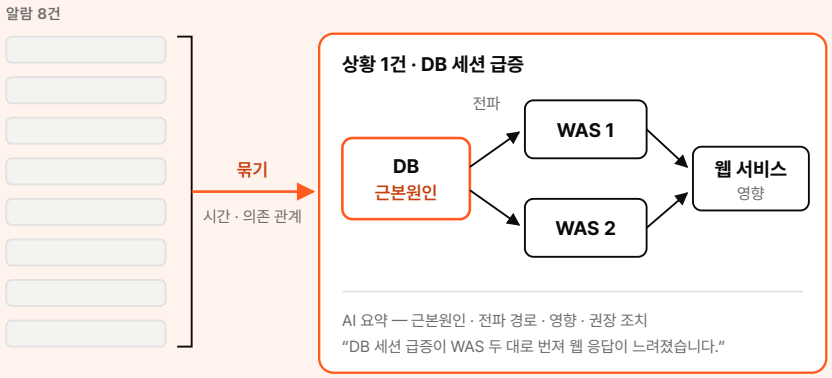
흩어진 알람을 원인 중심의 '상황' 하나로 묶어 보여 줍니다.

지금



원인과 결과가 뒤섞인 알람을 사람이 하나씩 풀어야 합니다.

AIOps 도입 후



원인 하나를 중심으로 묶인 상황 하나를 봅니다. (화면 예시)

시간과 의존 관계로 묶습니다

같은 시간대에, 서로 의존하는 자산에서 생긴 신호를 하나의 상황으로 모읍니다.

묶는 기준은 규칙으로

어떤 신호를 묶을지는 AI의 추측이 아니라 정해진 규칙으로 정해, 같은 상황이면 늘 같은 결과가 나옵니다.

공용 장비만 겹치면 따로

스토리지 · 네트워크처럼 모두가 함께 사용하는 장비만 겹칠 때는 묶지 않아, 무관한 장애가 섞이는 것을 줄입니다.

상황을 한 문단으로

묶인 신호가 둘 이상이면, AI가 원인과 전파 경로 · 영향 · 권장 조치를 한 문단으로 정리합니다.

고객이 얻는 것

- ✓ 알람 목록 대신 상황 몇 건만 확인합니다
- ✓ 원인과 결과를 처음부터 구분해서 봅니다
- ✓ 한 상황을 여러 팀이 같은 화면으로 봅니다

고민 03 · 원인 분석

원인을 찾는 데
복구보다 더 오래 걸립니다.

장애 회의 때마다 반복되는 이야기입니다.



현장에서는 이런 일이 생깁니다

도구를 오가며 대조합니다

지표는 여기, 로그는 저기, 변경 이력은 또 다른 곳에 있습니다. 담당자가 화면을 오가며 시간을 맞춰 봅니다.

어디까지 번졌는지 모릅니다

장애 지점은 찾아도, 영향을 받는 서비스가 어디까지인지는 구성도를 펼쳐 놓고 따져야 합니다.

직전에 바뀐 것을 놓칩니다

장애 직전의 배포나 설정 변경이 원인일 때가 많지만, 그 기록은 다른 시스템에 있습니다.

숨은 걱정 **"분석이 늦어지는 동안 서비스가 계속 멈춰 있는 것"**

AIOps의 해결 보기 →

해결 03 · 원인 · 영향 · 대응을 한 화면에

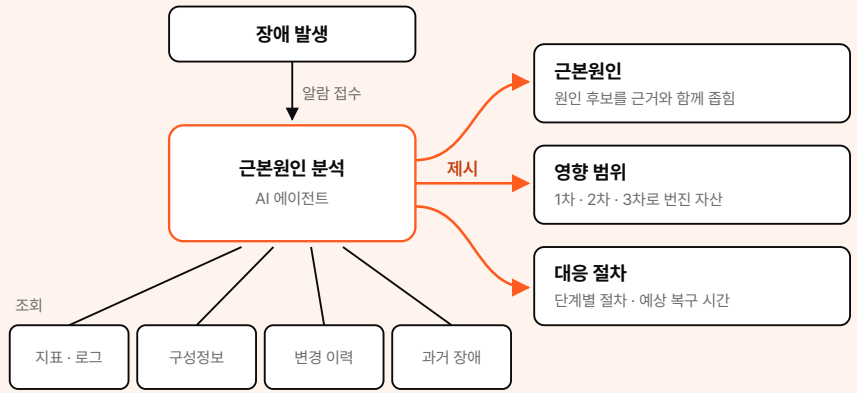
장애가 나면 원인 · 영향 범위 · 대응 절차가 한 화면에 함께 올라옵니다.

지금



원인을 찾는 일이 사람의 조회와 기억에 달려 있습니다.

AIOps 도입 후



에이전트가 흩어진 근거를 모아 원인 · 영향 · 대응을 한 번에 제시합니다.

근본원인까지 좁혀 갑니다

지표와 로그를 결합해 증상 너머의 원인 후보를 좁히고, 단계별 대응 절차와 예상 복구 시간을 제안합니다.

영향 범위를 지도로

구성정보의 의존 관계를 따라 1차 · 2차 · 3차로 번지는 자산을 한눈에 보여 줍니다.

직전 변경을 함께 점검

장애 직전의 변경 · 배포 이력을 대조해, 최근 변경이 원인인지 먼저 확인합니다.

비슷했던 과거 장애

과거 장애를 비슷한 순서로 찾아, 이미 검증된 대응을 다시 사용합니다.

고객이 얻는 것

- ✓ 누가 당직이어도 같은 근거로 분석을 시작합니다
- ✓ 영향받는 서비스에 바로 알릴 수 있습니다
- ✓ 원인을 찾던 시간을 복구에 사용합니다

고민 04 · 노하우

그 장애는 결국 그 사람만 압니다.



기록은 있는데 찾지 못합니다

장애 보고서와 런북은 쌓여 있지만, 급할 때 필요한 한 줄을 찾기 어렵습니다.

사람이 떠나면 방법도 떠납니다

숙련자가 자리를 옮기면, 그가 알던 대응 방법도 함께 사라집니다.

숨은 걱정

"담당자가 바뀔 때마다 대응 품질이 흔들리는 것"

해결 04 · 노하우를 회사의 지식으로

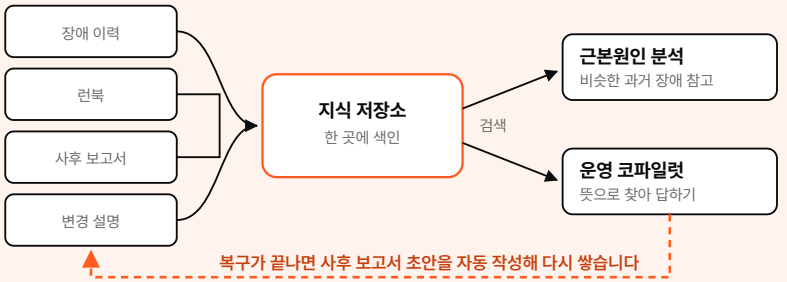
대응할수록 쌓이고, 누구나 꺼내 사용합니다.

지금



노하우가 사람과 흩어진 문서에 묶여 있습니다.

AIOps 도입 후



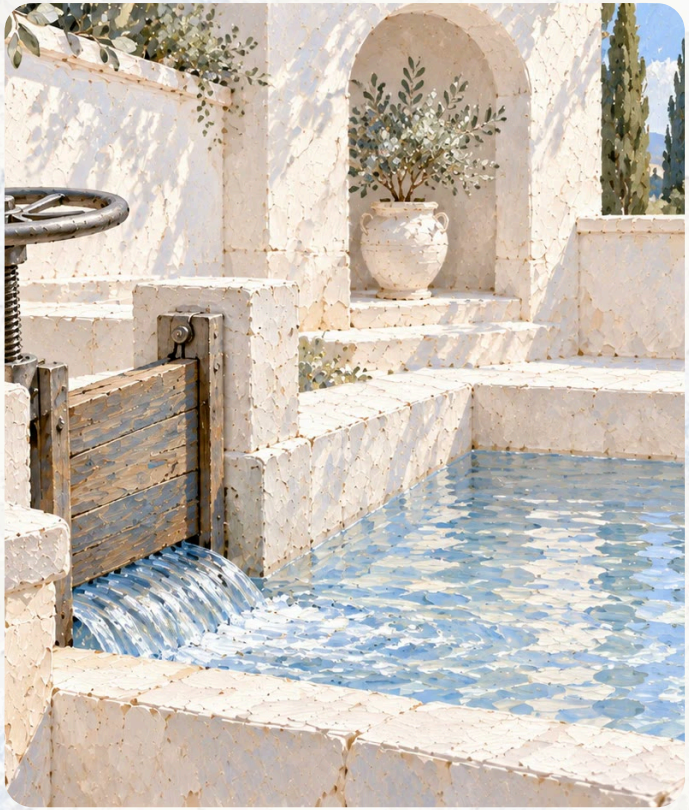
장애 하나가 끝날 때마다 회사의 지식이 늘어납니다.

고객이 얻는 것

- ✓ 숙련자가 아니어도 같은 기준으로 대응합니다
- ✓ 사람이 바뀌어도 대응 방법은 남습니다

목차로 돌아가기

고민 05 · 자동 조치



AI가 운영 서버를 직접 건드린다니, 그게 더 불안합니다.

자동화를 검토하는 운영 책임자가 결재 직전에 묻는 질문입니다.

현장에서는 이런 일이 생깁니다

새벽의 수작업은 실수를 부릅니다

피곤한 당직자가 명령어를 손으로 입력하다 보면,
복구 작업이 또 다른 장애가 되기도 합니다.

그렇다고 맡기기는 두렵습니다

근거를 알 수 없는 자동 실행이
운영 서버에 무엇을 할지 예상할 수 없습니다.

실행했다고 끝난 것이 아닙니다

스크립트가 성공으로 끝나도, 서비스가 실제로
회복됐는지는 다시 사람이 확인해야 합니다.

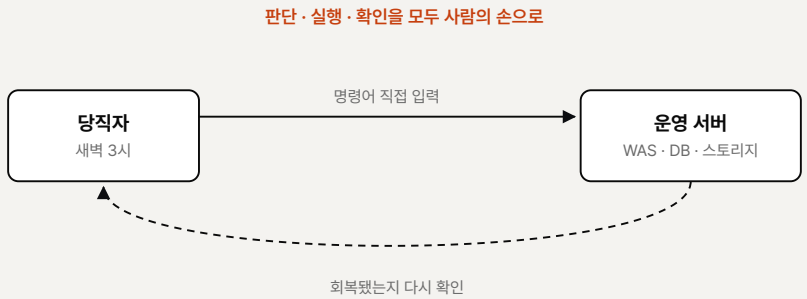
숨은 걱정 **"자동화가 사고를 키웠을 때 그 책임을 떠안는 것"**

AIOps의 해결 보기 →

해결 05 · 통제된 자동화

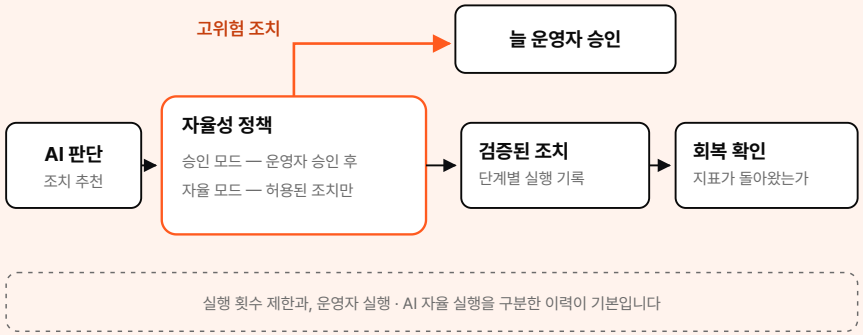
AI는 사람이 정한 범위 안에서만 실행하고, 서비스가 회복됐는지까지 확인합니다.

지금



실수와 확인 누락이 생기기 쉬운 구조입니다.

AIOps 도입 후



무엇을 자동으로 맡길지는 조직이 정하고, 결과는 지표로 확인합니다.

승인 모드로 시작

처음에는 모든 조치를 운영자가 승인하고, 신뢰가 쌓인 유형부터 자율 범위를 넓힙니다.

고위험 조치는 늘 승인

조치마다 위험도와 자동 실행 허용 여부를 정해 두어, 고위험 작업은 자율 모드에서도 사람이 승인합니다.

검증된 절차만 실행

AI가 명령을 새로 지어내지 않습니다. 미리 검증해 둔 조치 플레이북 가운데서 골라 실행합니다.

효과까지 확인

실행이 끝났는지가 아니라, 조치 후 대상 지표가 실제로 회복됐는지를 기준으로 판단합니다.

고객이 얻는 것

- ✓ 야간 · 휴일에도 같은 절차로 대응합니다
- ✓ 손으로 입력하던 실수가 줄어듭니다
- ✓ 실행 이력이 남아 책임 소재가 분명합니다

고민 06 · AI 판단의 신뢰



AI가 틀리면요?
근거 없는 판단은 믿을 수 없습니다.

AI 도입을 검토하는 운영 조직이 가장 경계하는 부분입니다.

현장에서는 이런 일이 생깁니다

왜 그렇게 판단했는지 모릅니다

결과만 알려 주는 AI는 운영자가 맞는지 확인할 방법이 없습니다.

틀린 판단이 되풀이됩니다

오탐을 알려 줘도 다음 판단에 반영되지 않으면, 결국 알람이 하나 더 늘어날 뿐입니다.

AI가 한 일이 보이지 않습니다

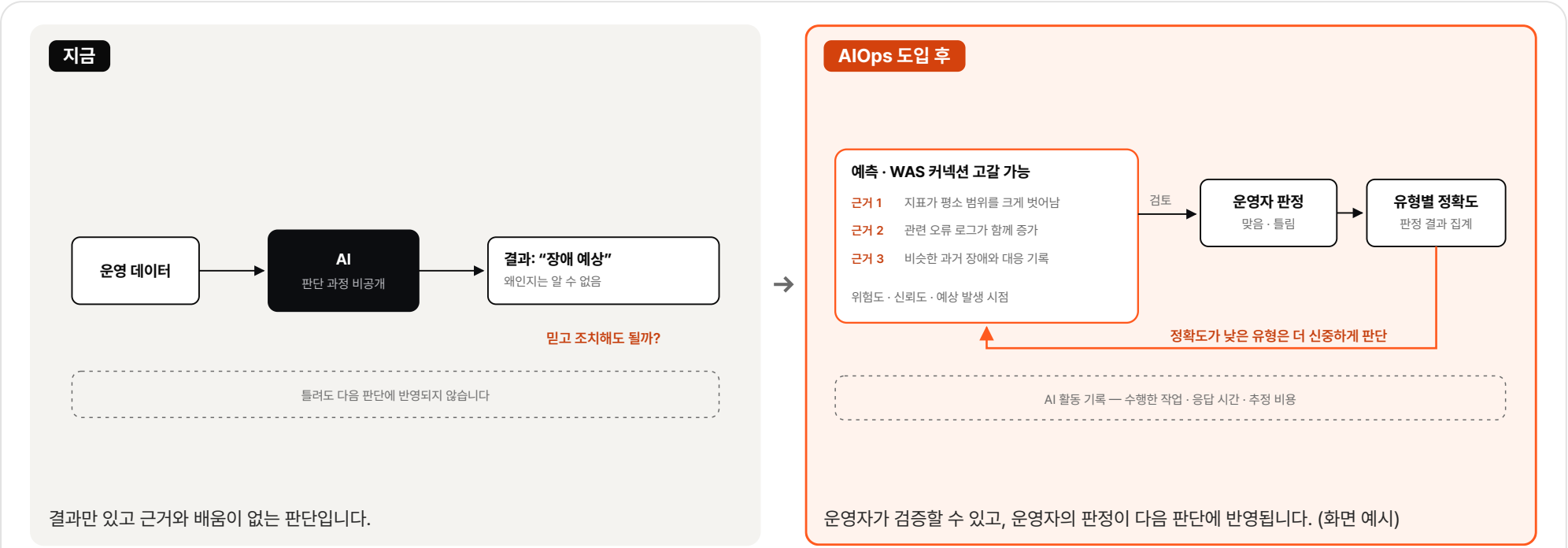
AI가 무엇을 얼마나 처리했고, 비용이 얼마나 들었는지 확인하기 어렵습니다.

숨은 걱정 **"설명할 수 없는 판단에 운영을 맡기는 것"**

AIOps의 해결 보기 →

해결 06 · 설명할 수 있는 판단

모든 판단에 근거가 붙고, 운영자의 판정이 다음 판단을 바꿉니다.



판단마다 근거를 함께

지표가 평소보다 얼마나 벗어났는지, 관련 로그, 비슷한 과거 장애를 예측과 함께 제시합니다.

판정이 정확도로 쌓입니다

운영자의 맞음 · 틀림 판정이 유형별 정확도로 집계되고, 정확도가 낮은 유형은 더 신중하게 판단하도록 반영됩니다.

규칙과 AI의 역할 분리

늘 같은 결과가 나와야 하는 신호 묶기는 규칙으로, 설명과 예측은 AI로 나눠 맡깁니다.

AI가 한 일도 기록으로

AI의 작업 · 응답 시간 · 추정 비용을 남깁니다. 운영 코파일럿은 분석과 추천만 하고 직접 실행하지 않습니다.

고객이 얻는 것

- ✓ 운영자가 직접 검증할 수 있는 판단
- ✓ 운영자의 판정이 반영되는 예측
- ✓ AI 사용량과 추정 비용이 보입니다

고민 07 · 기존 도구

지금 사용하는 모니터링 도구를 다 바꿔야 하나요?



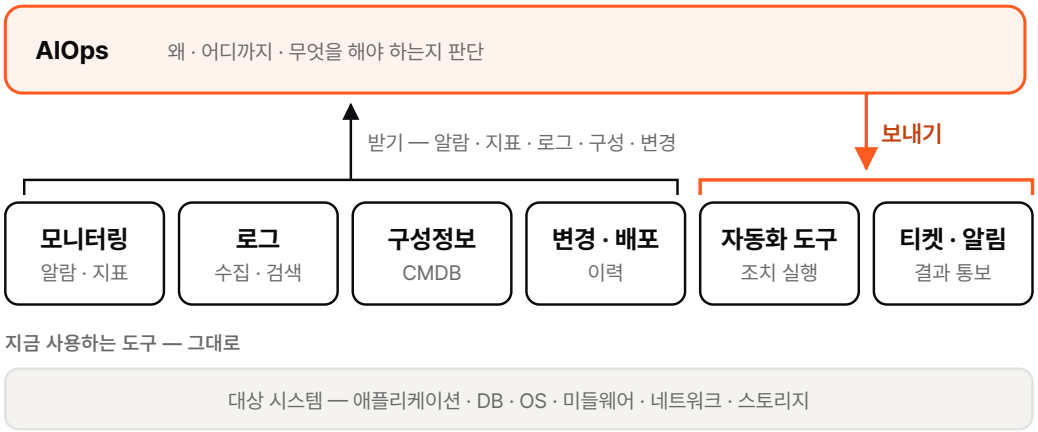
이미 투자한 도구가 많습니다
모니터링 · 로그 · 자동화 도구마다 들인 시간과 비용이 있습니다.

데이터는 도구마다 흩어져 있습니다
지표 · 로그 · 구성정보 · 변경 이력이 따로 있어, 한 번에 판단하기 어렵습니다.

숨은 걱정
"운영해야 할 도구가 하나 더 늘어나는 것"

해결 07 · 바꾸지 않고 더합니다

지금의 도구는 그대로, 그 위에 판단을 더합니다.



모니터링이 '무엇이 임계를 넘었는가'를 알려 주면, AIOps는 '왜, 어디까지, 무엇을 해야 하는가'를 판단합니다.

표준 방식으로 연결
웹훅 · API · syslog 같은 표준 방식을 기준으로 연결을 설계합니다.

흩어진 데이터를 한 곳에
지표 · 로그 · 구성정보 · 변경 이력을 하나의 기준으로 정리합니다.

연동 범위는 함께 정합니다
현황 점검 단계에서 연결 대상과 방식을 확정합니다.

고객이 얻는 것

- ✓ 이미 투자한 도구를 그대로 사용합니다
- ✓ 흩어진 데이터가 하나의 판단으로 모입니다

목차로 돌아가기

업무 적용 예시

AIOps가 들어가면, 장애 대응이 이렇게 달라집니다.

DB 운영 · 용량 관리

가득 차기 전에 미리 알게 됩니다

고객이 얻는 것 — **업무 시간 중에, 계획된 조치로** 용량 문제를 정리합니다.

지금은

테이블스페이스 사용률이 조금씩 오르다 임계를 넘는 순간 알람이 울리고, 그때부터 긴급 증설을 시작합니다.

→

AIOps가 맡는 일

평소 범위와 비교해 포화로 이어질 추세를 미리 알리고, 관련 로그와 과거 이력을 근거로 붙입니다.

→

담당자가 하는 일

근거를 확인해 승인하면, 확장 조치와 회복 확인이 정해진 절차대로 이어집니다.

이 장면을 만드는 AIOps: 해결 01 미리 알기 · 해결 06 판단 근거 · 해결 05 통제된 조치

연결하는 것: DB 지표 · 로그 · 장애 이력 · 조치 플레이북

서비스 운영 · 연쇄 장애

알람 수십 개가 아니라 상황 하나로 받습니다

고객이 얻는 것 — **원인과 결과를 처음부터 구분해** 대응 순서를 정합니다.

지금은

DB 세션이 급증하자 WAS · 웹 · 배치에서 알람이 한꺼번에 울리고, 담당자들이 각자 다른 알람을 쫓습니다.

→

AIOps가 맡는 일

같은 시간대 · 의존 관계의 신호를 상황 하나로 묶고, 근본원인인 DB와 영향받은 WAS를 함께 표시합니다.

→

담당자가 하는 일

상황 하나를 기준으로 담당 팀이 함께 원인 조치에 집중합니다.

이 장면을 만드는 AIOps: 해결 02 상황 단위 관제 · 해결 03 영향 범위 · 해결 07 기존 도구 연동

연결하는 것: 모니터링 알람 · 구성정보 · 의존 관계 · 로그

업무 적용 예시입니다. 실제 범위와 효과는 귀사의 시스템과 운영 방식에 맞춰 함께 설계합니다.

업무 적용 예시 · 이어서

같은 방식으로, 다른 업무에서도 일합니다.

인프라 운영 · 스토리지 장애

어디까지 번졌는지 구성도를 펼치지 않습니다

고객이 얻는 것 — **장애 공지와 복구를 동시에** 시작합니다.

지금은
스토리지 경로 장애가 나면, 영향받는 서버와 서비스를 구성도와 문서를 뒤져 하나씩 찾아냅니다.

→

AIOps가 맡는 일
근본원인과 함께 영향받는 자산을 단계별로 표시하고, 직전 변경 이력 · 비슷한 과거 장애 · 대응 절차를 제시합니다.

→

담당자가 하는 일
영향받는 서비스에 알리고, 제시된 복구 조치를 승인해 실행합니다.

이 장면을 만드는 AIOps: 해결 03 원인 분석 · 해결 04 지식 재사용 · 해결 05 통제된 조치

연결하는 것: 구성정보(CMDB) · 변경 · 배포 이력 · 스토리지 지표 · 로그

운영 관리 · 사후 보고

장애가 끝난 뒤의 보고서를 처음부터 작성하지 않습니다

고객이 얻는 것 — **장애 하나가 끝날 때마다** 운영 조직의 지식과 근거가 늘어납니다.

지금은
복구 뒤에 경위 · 원인 · 조치를 정리한 보고서를 작성하고, 재발 방지 대책을 따로 챙겨야 합니다.

→

AIOps가 맡는 일
복구가 끝나면 사후 보고서 초안을 만들어 지식으로 저장하고, 정확도 · 복구 시간을 지표로 남깁니다.

→

담당자가 하는 일
초안을 검토해 확정하고, 지표를 보며 자동화를 넓힐 장애 유형을 정합니다.

이 장면을 만드는 AIOps: 해결 04 지식 자산화 · 해결 06 판정 반영 · 해결 05 자율 범위

연결하는 것: 장애 이력 · 사후 보고서 · 성과 지표

업무 적용 예시입니다. 실제 범위와 효과는 귀사의 시스템과 운영 방식에 맞춰 함께 설계합니다.

도입 순서

처음부터 AI에게 맡기지 않습니다. 승인 모드로 시작해 확인하고, 넓혀 갑니다.

1

현황을 함께 점검

사용 중인 운영 도구와 자산 구성을 살펴보고, 연결 지점과 대상 범위를 정합니다.

고객이 얻는 것

연결할 곳과 먼저 다룰 장애 유형

2

승인 모드로 시작

핵심 자산부터 예측과 원인 분석을 적용하고, 모든 조치는 운영자가 승인한 뒤 실행합니다. 승인 과정의 판정이 쌓여 유형별 정확도가 보입니다.

고객이 얻는 것

AI 판단이 귀사 환경에서 맞는지,
귀사 지표로 확인한 결과

3

저위험 조치부터 자율

정확도가 안정된 유형부터, 저위험 조치의 자동 실행을 허용합니다. 고위험 조치는 계속 승인을 거칩니다.

고객이 얻는 것

야간 · 휴일 대응 부담이 덜어지는 첫 구간

4

대상과 범위 넓히기

회복 확인 결과와 쌓인 지식을 근거로, 대상 자산과 자동화 범위를 넓혀 갑니다.

고객이 얻는 것

예측 정확도 · 복구 시간을
귀사 지표로 보며 넓히는 운영

첫 상담은 아래 자료 일부만 있어도 충분합니다

사용 중인 운영 도구

모니터링 · 로그 · 자동화 · 티켓 도구 목록

최근 반복된 장애

지난 몇 달 동안 되풀이된
장애 유형 두세 가지

자산 · 구성 자료

서버 · 장비 목록과 의존 관계
(CMDB나 엑셀)

표준 조치 절차

지금 사용하는 런북과
점검 · 조치 스크립트

SaaS로 가볍게 시작

대상 자산 규모와 연결 범위를 정한 뒤,
핵심 자산부터 작게 시작합니다.

전용 환경 협의

AI에 전달할 데이터 범위와 고객 전용 클라우드 구성을 보안 요건에 맞춰 협의합니다.

승인 모드가 기본

모든 조치는 운영자 승인으로 시작하고, 자율 범위는 언제든 조정합니다.

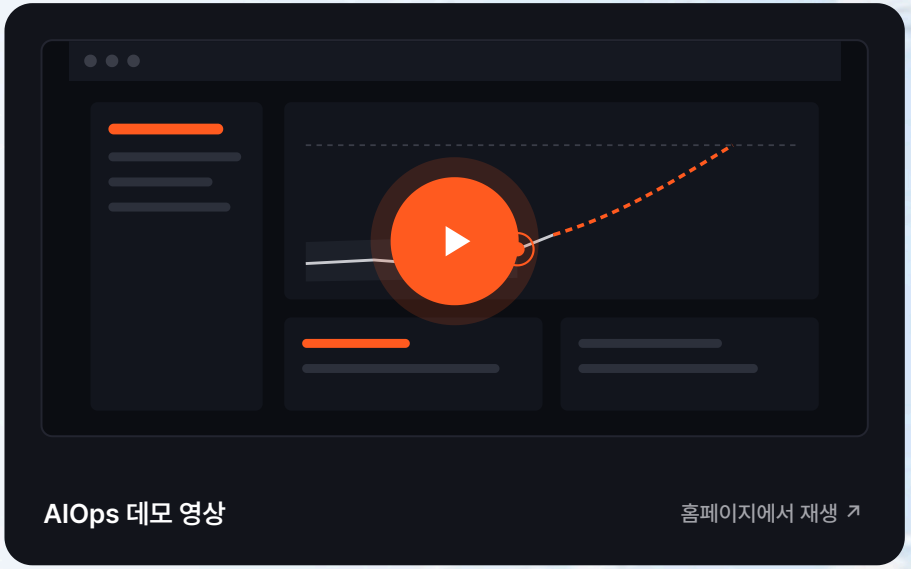
상담 안내

귀사의 고민부터 들려주세요.

최근 되풀이된 장애 하나를 알려 주시면, 비슷한 장애가
AIOps에서 어떻게 미리 알려지고 승인 · 조치되는지
데모로 보여 드립니다. 자료는 19쪽 목록 일부면 충분합니다.
먼저 살펴보고 싶다면 데모 영상을 확인해 보세요.

도입 문의하기 →

▶ 데모 영상 보기



문의 때 가장 급한 고민 하나를 알려 주세요

미리 알기 · 알람 정리 · 원인 분석 · 노하우 · 안전한 조치 · 판단 근거 · 기존 도구

이메일
sales@vicsnc.com

웹사이트
vicsnc.com