

WHITEPAPER

openmaru

2026 예방점검 의무화, 자동화를 넘어 지능화로

OS·미들웨어·DB 예방점검과 운영 지식의 축적

공공 정보시스템 예방점검의 핵심은 점검 횟수가 아니라 판정과 증적입니다. OPENMARU APM 이 지표를 모으고, OpsKnow Agent 가 판정·조치·보고를 맡고, OpsKnow 가 그 결과를 운영 지식으로 쌓는 구조를 폐쇄망 안의 30B 수준 내부 sLLM 으로 구현하는 방법을 정리했습니다.

2026-11-16 · 작성자: OPENMARU · openmaru.io

목차

2026 예방점검 의무화, 자동화를 넘어 지능화로 — OS·미들웨어·DB 예방점검과 운영 지식의 축적

1장: 2026 예방점검 의무화로 바뀐 것 — 고시 제2026-23호와 15,033개 정보시스템의 A1~A4 등급

1.1 국민 영향 중심으로 다시 매긴 A1~A4 등급 — A1 73개부터 A4 14,132개까지

1.2 고시 제2026-23호가 기관에 부과하는 안정성 기준과 표준운영절차·SLA 체결 의무

1.3 일상점검부터 구조진단까지 3개 분야 점검이 만드는 우리 기관 연간 점검 물량

1.4 감사가 묻는 것은 점검 횟수가 아닌 판정 결과와 증적

2장: 예방점검 체계가 없을 때와 단순 자동화만 했을 때 — 판정 근거 없는 점검표의 감사 리스크

2.1 2023년 11월 행정전산망 마비와 라우터 포트 불량이 드러낸 노후 장비 점검 공백

2.2 스크립트 점검과 양식 보고서 — 단순 자동화가 채우는 범위와 비어 있는 판정

2.3 분기·연간 보고와 SLA 준수율 산출에서 드러나는 수작업 비용

2.4 감사 질문 기준으로 보는 우리 기관 예방점검 수준 자가진단 5문항

3장: 단순 자동화에서 지능화로 — 수집부터 학습까지 6단계에서 달라지는 산출물

3.1 체계 없음에서 AI 지능화까지 3단계 비교표 — 6개 운영 단계별로 남는 증적

3.2 수집·판정 단계 — 임계치 비교 대신 결정적 코드가 남기는 판정 근거

3.3 연관 단계 — 미들웨어 응답 지연과 DBMS 커넥션 풀, OS 자원을 한 사건으로 묶는 원인 설명

3.4 조치에서 학습까지 — 승인 기록과 운영 지식 축적이 남기는 증적

4장: 해법의 구조 — APM 수집, OpsKnow Agent 판정, OpsKnow 축적으로 나눈 예방점검 역할 분담

4.1 OPENMARU APM이 상시 수집하는 OS와 미들웨어(WEB/WAS), DBMS 지표의 범위

4.2 점검 결과를 판정하는 OpsKnow Agent와 9단계로 자동 적재하는 OpsKnow

4.3 기관 상황판 Wallboard와 범정부 단위 전국 Dashboard의 Hub & Spoke 구성

5장: AAS 위 OpsKnow Agent가 공공 운영에 맞는 이유 — 판정은 코드, 결정은 사람, 설명은 AI

5.1 LLM은 문장 해석과 설명만, 판정은 결정적 코드가 맡는 설계

5.2 30분 유효 1회용 승인 코드와 적용 직전 상태 재확인으로 보장하는 사람 결재

5.3 재부팅과 설정 변경, 제품 설치본 패치를 에이전트 범위 밖에 둔 작업 경계

5.4 AAS(AI Agent Studio)에서 정하는 에이전트별 권한 범위와 실행 기록이 남기는 책임 소재

6장: 소버린 AI Level 3로 구축하는 이유 — GPU 1대 폐쇄망에서 실행하는 30B 수준 내부 sLLM

6.1 운영 데이터의 외부 LLM 전송 위험과 국가정보원 N2SF 등급별 보안 통제

6.2 외부 LLM API 호출 0건을 판정 기준으로 삼는 소버린 AI Level 3 구성

6.3 GPU 1대에서 30B 수준 내부 sLLM으로 충분한 이유 — 판정을 코드가 맡는 구조

6.4 GPU 서버 1대 단위의 도입 비용과 외부 서비스 정책 변경에서 분리된 운영 안정성

7장: 보안 취약점 점검의 지능화 — KISA·CVE 권고 접수부터 승인 코드 적용과 재점검까지

7.1 KISA 보안공지와 CVE 발표 뒤 영향 서버를 가려내는 패키지 버전 대조

7.2 승인 코드 기반 OS 패키지 패치 적용과 재점검 결과의 증적화

7.3 지원 종료 OS와 수정본 없는 취약점을 「최신」으로 오판정하지 않는 별도 관리

7.4 분기 일괄 패치에서 권고 단위 처리로 — 공격 가능 기간을 줄이는 패치 일정

8장: 예방점검을 넘어 장애 대응과 정기 보고까지 — 한 저장소에 쌓는 운영 기록과 자동 처리·결재의 경계

8.1 Incident 케이스와 예방점검 결과, 정기 운영 보고를 한 저장소에서 연결하는 OpsKnow
8.2 원인부터 자동 RCA 시간까지 7개 필드로 남기는 Incident 기록
8.3 알림과 적재는 자동, 미들웨어(WAS) 재시작과 DB 풀 크기 변경은 결재로 나눈 경계 매트릭스
8.4 일일 보고부터 연간 보고까지 누적되는 정기 운영 보고와 SLA 산출 근거
9장: 관리자가 실제로 쓰는 장면 — 자연어 요청 한 줄에서 점검 보고서와 조치 승인까지
9.1 「이번 달 OS와 미들웨어, DBMS 점검 결과」 요청에 대한 영역별 판정 요약 보고
9.2 분기 감사 준비 장면 — 미이행 점검 항목 조회와 분기 운영 보고서 초안 검토
9.3 보안공지 발표일 장면 — 영향 서버 조회부터 승인 코드 발급과 재점검 회신까지
9.4 장애 직후 장면 — 과거 유사 Incident 검색과 결재가 필요한 조치 제안
10장: 경쟁 방식 비교와 도입 효과 — A사·B사 방식과 다른 판정·증적 범위, 도입 뒤 측정할 지표
10.1 일상점검 스크립트와 양식 보고 중심의 A사·B사 방식과 지능화 방식의 7개 기준 비교
10.2 도입 전후를 비교하는 측정 지표 — 점검 소요 시간부터 증적 누락 건수까지
10.3 차년도 사업 준비 순서 — 단계 자가 판정, 요구사항 선택, OpsKnow Agent 시연 요청
Appendix
References
함께 읽을 자료
Glossary

2026 예방점검 의무화, 자동화를 넘어 지능화로 — OS·미들웨어·DB 예방점검과 운영 지식의 축적

1장: 2026 예방점검 의무화로 바뀐 것 — 고시 제2026-23호와 15,033개 정보시스템의 A1~A4 등급

행정안전부는 2026년 8월 27일 공공 정보시스템 15,033개의 등급을 국민이 받는 서비스의 중요도에 따라 A1부터 A4까지 다시 매겼습니다[S01]. 같은 해 3월에는 「행정기관 및 공공기관 정보시스템 안정성 고시」 제정안이 행정예고되었고, 예방점검·장애 관리 기준과 서비스 수준 협약(SLA, Service Level Agreement) 체결 의무가 그 안에 들어갔습니다[S02]. 따라서 공공기관 정보화 부서는 2026년부터 예방점검을 권고 사항으로 다루기 어렵고, 고시 제2026-23호가 정한 기준에 맞춰 점검하고 그 결과를 남겨야 합니다. 등급 체계, 고시가 부과하는 의무, 2024년 예방점검체계에서 넘어온 3개 분야 점검을 차례로 확인하면, 의무화로 실제로 늘어난 책임이 판정 결과와 증적을 남기는 일이라는 점이 드러납니다. 이 결론이 차년도 예산을 판단하는 근거가 됩니다.

1.1 국민 영향 중심으로 다시 매긴 A1~A4 등급 — A1 73개부터 A4 14,132개까지

A1~A4 등급은 정보시스템이 멈췄을 때 국민이 받는 영향을 기준으로 매긴 공공 정보시스템 분류이며, 2026년 8월 행정안전부 발표로 15,033개 시스템 전체에 적용되었습니다[S01]. 등급이 중요한 이유는 같은 기관 안에서도 시스템마다 점검 강도와 복구 목표가 달라지기 때문입니다. 기관이 먼저 알아야 할 정보는 우리 기관 시스템 가운데 A1·A2에 속한 것이 몇 개이고, 그 시스템을 어떤 수준으로 점검해야 하는지입니다. 아래 세 항목은 등급별 분포, 기준이 바뀐 계기, 등급별 목표복구시간을 순서대로 다룹니다.

1.1.1 2026년 8월 확정된 15,033개 시스템의 등급별 분포와 비율

의사결정권자용 1장 핵심 확인 항목

1장의 판단 근거는 다음 세 가지로 모입니다. 해당 여부는 기관이 운영하는 시스템의 등급으로 정해지고, 의무의 범위는 정보시스템 안정성 고시가 정하며, 늘어난 업무량은 3개 분야 점검의 대상 장비 수와 주기로 계산합니다. 공식 발표로 나온 수치는 등급별 시스템 수이므로, 기관은 아래 표에 자기 기관 시스템 수를 대입해 비중을 직접 계산해야 합니다.

확인 항목	확인 위치	기관이 할 일
해당 여부	2026-08-27 등급 결과 통보	시스템별 A1~A4 등급 목록 작성
의무 범위	정보시스템 안정성 고시(고시 제2026-23호)	안정성 기준·SLA·표준운영절차 대조
업무량	일상점검·특별점검·구조진단 3개 분야	장비 수 × 항목 수 × 주기 산정
감사 대응	판정 근거·조치 이력·재점검 결과	증적 저장 방식 결정

A4 등급에 몰린 14,132개 시스템

행정안전부가 발표한 등급별 시스템 수는 A1 73개, A2 157개, A3 671개, A4 14,132개입니다[S01]. 비율로 바꾸면 A1이 0.5%, A2가 1.0%, A3가 4.5%이고 A4가 94.0%를 차지합니다. 전체 공공 정보시스템 가운데 A1과 A2를 합친 230개, 1.5%만이 가장 엄격한 복구 목표를 적용받습니다. A4 14,132개는 개별 시스템의 복구 목표가 느슨한 대신 수가 훨씬 많아서, 점검 업무량의 대부분은 A4 시스템에서 나옵니다. 등급 분포를 볼 때는 상위 등급의 엄격함과 하위 등급의 물량을 따로 계산해야 합니다.

등급	시스템 수	비율	등급 위치
A1	73개	0.5%	최상위 등급(112·산불 대응·정부24 포함)
A2	157개	1.0%	상위 등급
A3	671개	4.5%	중위 등급
A4	14,132개	94.0%	하위 등급
합계	15,033개	100%	2026-08-27 행정안전부 발표 기준

표의 비율은 발표된 등급별 시스템 수를 15,033개로 나눈 값입니다[S01].

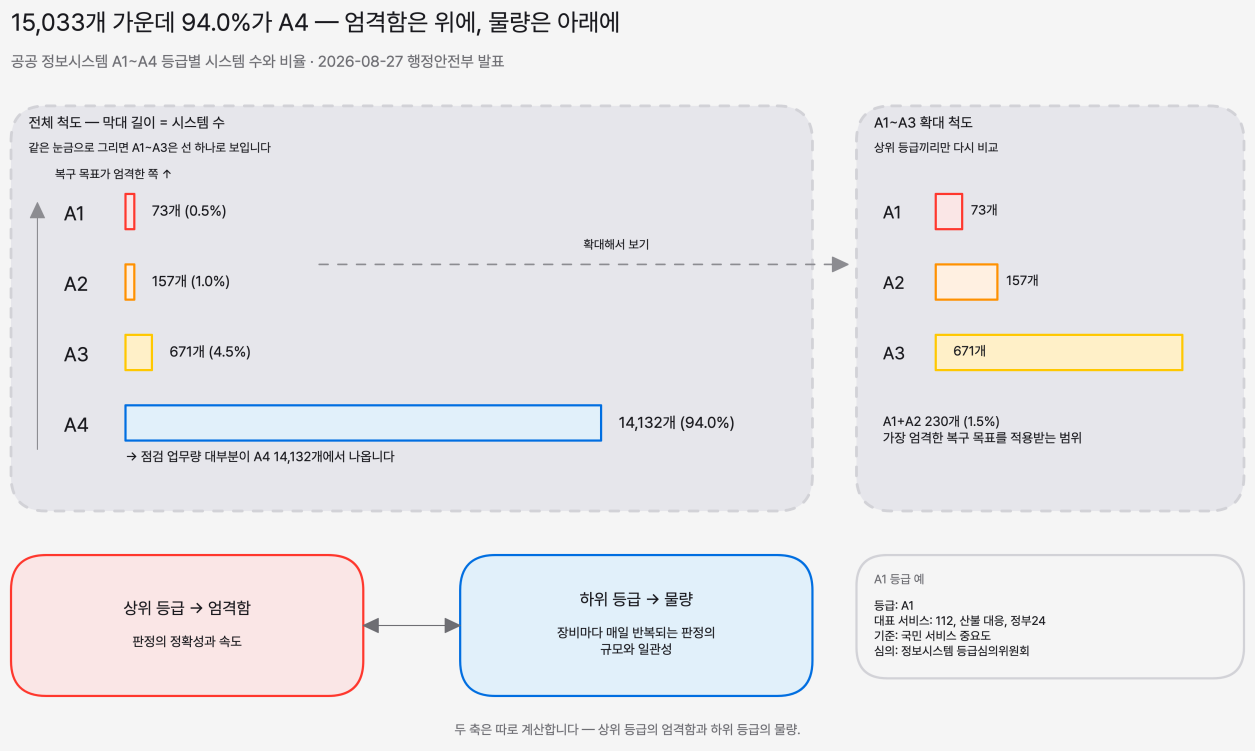


그림. 공공 정보시스템 15,033개의 A1~A4 등급별 시스템 수와 비율(2026-08-27 행정안전부 발표)

A1 등급에 들어간 대표 서비스

A1 등급에는 112 시스템, 산불 대응 시스템, 정부24처럼 국민 안전과 일상 민원에 직접 연결된 서비스가 들어갔습니다[S01]. 새 기준이 사용자 수가 아니라 국민이 받는 서비스의 중요도를 보기 때문에, 이 서비스들이 최상위

등급에 놓였습니다. 산불 대응 시스템은 평소 사용량과 관계없이 재난이 나면 대응 속도에 직접 영향을 주는 시스템입니다. 우리 기관 시스템의 등급을 확인할 때는 사용량 통계보다 그 시스템이 멈췄을 때 어떤 국민 서비스가 중단되는지를 먼저 봐야 합니다.

1.1.2 기관·사용자 수에서 국민 서비스 중요도로 바뀐 등급 기준과 그 계기

옛 등급 기준이 남긴 저등급 판정 문제

이전 등급 기준은 기관의 규모와 시스템 사용자 수를 중심으로 등급을 매겼습니다[S01]. 이 방식은 측정이 쉽다는 장점이 있었지만, 사용자는 적어도 멈추면 국민 피해가 큰 시스템을 낮은 등급으로 분류하는 약점이 있었습니다. 실제로 2025년 9월 국가정보자원관리원 화재 때 사용자 수가 적다는 이유로 낮은 등급을 받은 시스템이 있었다는 문제가 드러났습니다[S01]. 행정안전부는 등급 기준을 국민이 받는 서비스의 중요도로 바꾸었고, 2026년 등급 결과는 이 새 기준을 처음으로 전체 시스템에 적용한 결과입니다.

구분	이전 기준	2026년 기준
판단 축	기관 규모·사용자 수	국민 서비스 중요도
등급 표기	기관 중심 분류	A1~A4
드러난 약점	사용자 적은 핵심 시스템의 저등급	국민 영향으로 재분류

민간 전문가가 참여하는 등급심의위원회

등급은 민간 전문가로 구성된 정보시스템 등급심의위원회가 심의하며, 2026년 등급 결과도 실무 검토와 전문가 검토, 분과위원회 심사를 거쳐 나왔습니다[S01][S02]. 이 절차가 기관에 주는 의미는 등급을 기관이 혼자 정하지 않는다는 점입니다. 다만 기관은 자기 시스템이 어떤 국민 서비스를 받치고 있는지 설명할 자료를 갖고 있어야 등급 심의 결과를 검토하고 설명할 수 있습니다. 시스템 구성, 연계 서비스, 장애 시 영향 범위를 평소에 정리해 두는 기관이 등급 변경 때도 근거를 빨리 낼 수 있습니다.

등급을 두고 기관 안에서 대화할 때는 어느 기준의 등급인지를 함께 말하는 습관이 필요합니다. 예방점검체계, 표준운영절차서, 기관 내부 분류처럼 등급을 다루는 문서가 여럿이고, 문서마다 등급을 묶는 방식이 같지 않기 때문입니다. 같은 시스템을 두고 한 사람은 내부 분류를, 다른 사람은 행정안전부가 확정해 통보한 등급을 말하면 의무 범위와 예산 논의가 엇갈립니다. 앞으로 대외적으로 쓰는 공식 등급은 확정 통보된 A1~A4 등급이므로, 회의 자료와 사업 문서에는 「확정 통보 기준 A2」처럼 근거를 붙여 적는 것이 안전합니다.

1.1.3 A1 1시간 이내부터 A4 3주 이내까지 등급별 목표복구시간

A1과 A4 사이에 벌어진 복구 목표의 폭

목표복구시간(RTO, Recovery Time Objective)은 장애나 재난으로 시스템이 멈춘 뒤 서비스를 다시 제공해야 하는 시간 한도입니다. 2026년 3월 행정예고 단계에서 알려진 목표복구시간은 A1이 1시간 이내, A4가 3주 이내였고, A2와 A3는 그 사이에 단계적으로 놓였습니다[S02][S03]. 같은 공공 정보시스템이라도 A1 시스템은 1시간 안에 돌아와야 하고, A4 시스템은 몇 주 단위의 복구 계획을 세울 수 있습니다. 이 수치는 행정예고안을 다룬 보도에서 확인한 값이므로, 기관은 계약서나 운영계획서에 옮기기 전에 시행된 고시 원문의 표와 대조해야 합니다.

목표복구시간이 점검 방식에 주는 영향

목표복구시간이 짧을수록 장애가 나기 전에 이상 징후를 잡아내는 예방점검의 비중이 커집니다. 예를 들어 1시간 안에 복구해야 하는 A1 시스템에서 디스크 부족이나 인증서 만료를 장애 발생 뒤에 알게 되면, 원인 확인에만 복구 시간 대부분을 쓰게 됩니다. A4 시스템은 복구 시간 여유가 있는 대신 대상 수가 많으므로, 사람이 하나씩 확인하는 방식으로는 점검 주기를 지키기 어렵습니다. 상위 등급은 판정의 정확성과 속도, 하위 등급은 판정의 규모와 일관성이 각각 점검 체계 설계의 기준이 됩니다.

1.2 고시 제2026-23호가 기관에 부과하는 안정성 기준과 표준운영절차·SLA 체결 의무

「행정기관 및 공공기관 정보시스템 안정성 고시」(고시 제2026-23호)는 공공 정보시스템의 장애 예방부터 복구까지 기관이 지켜야 할 기준을 한 고시로 묶은 규정입니다[S02]. 이전에는 예방점검체계와 표준운영절차가 행정안전부의 권고와 지침 형태로 운영되었습니다[S06]. 고시로 올라오면서 예방점검은 감사와 평가에서 이행 여부를 확인받는 의무로 바뀌었고, 위탁 운영 계약과 민간 클라우드 이용 계약에도 같은 기준이 들어가야 합니다. 아래 세 항목은 안정성 기준과 장애관리 계획, SLA 체결 의무, 백업과 훈련 의무를 차례로 다룹니다.

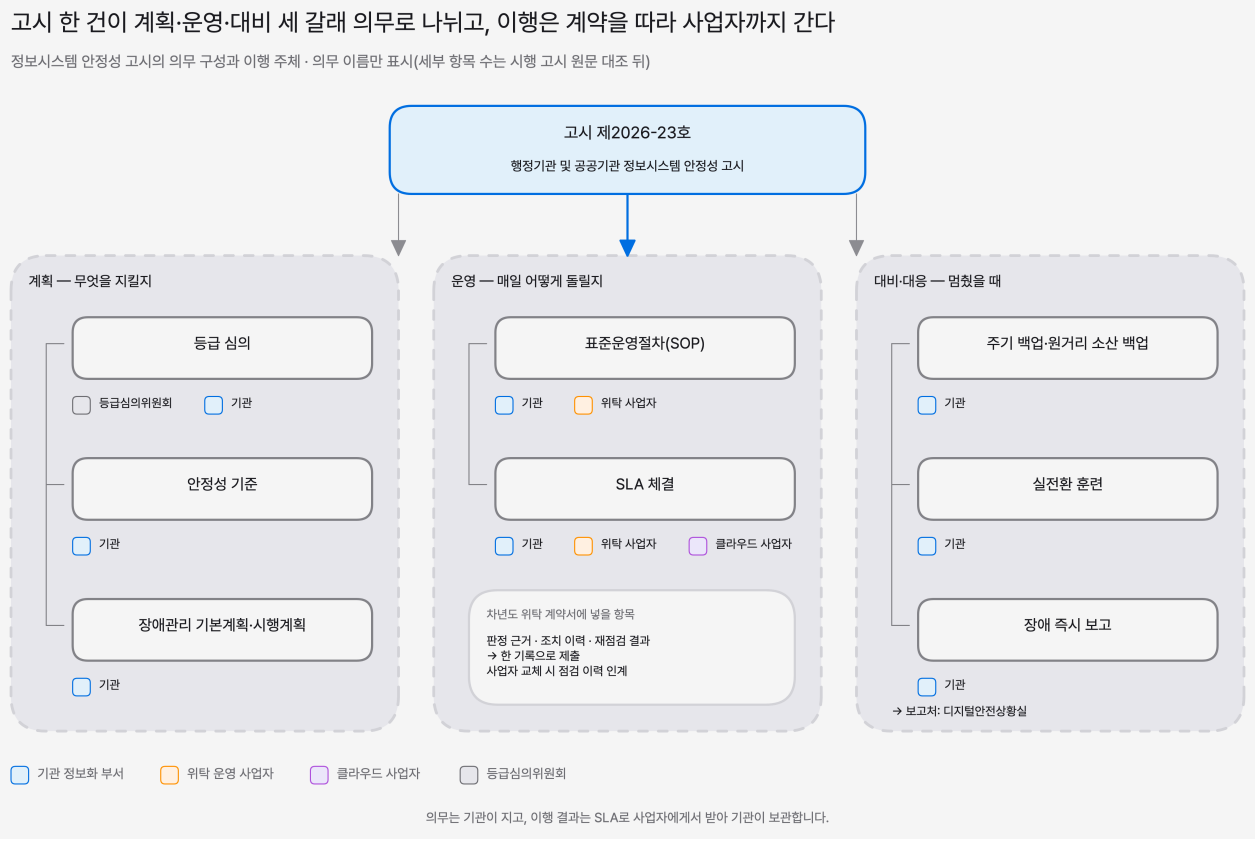


그림. 정보시스템 안정성 고시의 의무 구성 — 등급, 안정성 기준, 장애관리 계획, 표준운영절차, SLA, 백업·훈련

1.2.1 행정예고안 기준 46개 안정성 기준과 3년 단위 장애관리 기본계획

46개 안정성 기준의 범위

2026년 3월 행정예고된 제정안은 예방점검과 장애 관리 등을 다루는 46개 항목의 안정성 기준을 담은 것으로 알려졌습니다[S02]. 안정성 기준은 기관이 정보시스템을 설계하고 운영할 때 갖춰야 할 최소 조건을 항목별로 적은 목록이며, 예방점검은 그 가운데 운영 단계에서 매일 반복되는 항목입니다. 따라서 안정성 기준을 지켰는지 확인하려면 점검을 했다는 사실보다 각 항목을 어떤 근거로 정상 또는 이상으로 판정했는지가 남아 있어야 합니

다. 46개 항목의 분류와 별표 구성은 시행 고시 원문에서 확인한 뒤 기관 점검표와 한 줄씩 대조하는 것이 안전합니다.

3년 단위 기본계획과 연간 시행계획

행정예고안은 장애관리 기본계획을 3년 단위로 세우고, 해마다 연간 시행계획을 수립하도록 했습니다[S02]. 기본계획은 기관 전체 시스템의 장애 예방 방향과 투자 우선순위를 정하는 문서이고, 연간 시행계획은 그해 점검 일정과 예산, 담당 조직을 구체적으로 적는 문서입니다. 예방점검 체계를 바꾸려는 기관은 차년도 시행계획에 점검 방식과 증적 관리 방식을 함께 적어야 예산과 인력이 따라옵니다. 2026년 하반기는 다음 해 시행계획에 점검 자동화와 판정 기록 저장 방식을 반영할지 결정해야 하는 시기입니다.

계획	주기	담는 내용	예방점검과의 연결
장애관리 기본계획	3년	장애 예방 방향, 투자 우선순위	점검 체계 개편 방향
연간 시행계획	1년	점검 일정, 예산, 담당 조직	점검 방식·증적 관리 방식
안정성 기준	상시	46개 항목(행정예고안 기준)	항목별 판정 근거

1.2.2 민간 클라우드와 위탁 운영까지 넓어진 SLA 체결 의무와 차년도 계약 반영

SLA 체결 의무가 닿는 계약의 범위

행정예고안은 정보시스템 표준운영절차(SOP, Standard Operating Procedure)를 마련하고 서비스 수준 협약을 맺도록 했으며, 그 대상에 민간 클라우드와 위탁 운영을 포함했습니다[S02]. SLA는 서비스 제공자가 지켜야 할 가용성, 장애 대응 시간, 보고 의무를 수치와 절차로 정한 계약 문서입니다. 기관이 운영을 외부 사업자에게 맡기거나 민간 클라우드를 쓰더라도 예방점검 책임은 계약서를 따라 사업자에게 이어지고, 이행 결과는 기관이 받아 보관해야 합니다. 표준운영절차는 2024년 예방점검체계와 함께 처음 마련되었고, 응용프로그램 영역 절차도 뒤이어 배포되었습니다[S06].

차년도 위탁 계약서에 넣을 항목

SLA 체결 의무를 차년도 계약에 반영하려면 점검 결과를 어떤 형식으로 받을지를 계약서에 먼저 적어야 합니다. 예를 들어 사업자가 매일 점검표만 제출하면 기관은 판정 근거와 조치 이력을 따로 요청해야 하고, 감사 때 그 자료를 다시 모으는 일이 생깁니다. 계약서에 판정 근거와 조치 이력, 재점검 결과를 한 기록으로 제출하도록 적어 두면 사업자가 바뀌어도 기록이 끊기지 않습니다. 다음 목록은 차년도 위탁 계약서에 반영할 항목입니다.

- 등급별 목표복구시간과 장애 대응 시간
- 일상점검·특별점검·구조진단의 수행 주기와 제출 형식
- 항목별 판정 근거와 이상 판정 시 조치 이력 제출
- 조치 후 재점검 결과와 기록 보관 기간
- 사업자 교체 시 점검 이력 인계 방식

1.2.3 주기 백업과 원거리 소산 백업, 연 1회 이상 실전환 훈련 의무

모든 시스템에 적용되는 백업 의무

행정예고안은 등급과 관계없이 모든 정보시스템에 주기 백업과 원거리 소산 백업을 의무로 정했습니다[S02]. 원거리 소산 백업은 데이터 사본을 운영 센터와 물리적으로 떨어진 장소에 보관하는 방식이며, 화재나 침수처럼 한 장소 전체가 멈추는 경우에 대비합니다. 2025년 국가정보자원관리원 화재가 등급 기준 개편의 계기가 된 만큼[S01], 백업 의무는 장애 예방과 같은 무게로 다뤄야 합니다. 일상점검 항목에 백업 성공 여부와 소산 사본의 최신성을 넣어 매일 판정하는 기관이 감사 때 이행 근거를 바로 낼 수 있습니다.

실전환 훈련과 디지털안전상황실 보고

행정예고안은 연 1회 이상 실제로 운영을 대체 환경으로 넘기는 실전환 훈련을 요구하고, 중요 시스템에 장애가 나면 행정안전부 디지털안전상황실에 즉시 보고하도록 했습니다[S02]. 실전환 훈련은 계획서상 복구 절차가 실제로 목표복구시간 안에 끝나는지를 가장 직접 확인하는 방법입니다. 이때 훈련 결과도 점검과 같은 형식으로 남겨야 하는데, 어느 단계에서 얼마나 걸렸고 어떤 문제가 나왔는지가 다음 해 시행계획의 근거가 되기 때문입니다.

의무	적용 대상	주기	남겨야 할 기록
주기 백업	모든 시스템	기관 계획에 따름	백업 성공 여부와 시각
원거리 소산 백업	모든 시스템	기관 계획에 따름	소산 사본 위치와 최신성
실전환 훈련	모든 시스템	연 1회 이상	단계별 소요 시간과 문제점
장애 즉시 보고	중요 시스템	장애 발생 시	보고 시각과 조치 경과

1.3 일상점검부터 구조진단까지 3개 분야 점검이 만드는 우리 기관 연간 점검 물량

정보시스템 예방점검은 일상점검, 특별점검, 구조진단 3개 분야로 나뉘며, 이 구조는 행정안전부가 2024년 10월 마련한 예방점검체계에서 시작되었습니다[S06][S07]. 3개 분야는 점검 주기와 깊이가 다르기 때문에 한 기관 안에서도 서로 다른 일정으로 겹쳐 돌아갑니다. 점검 대상은 서버 운영체제(OS)뿐 아니라 웹 서버, WAS(Web Application Server, 웹 애플리케이션 서버) 같은 미들웨어와 DBMS(Database Management System, 데이터베이스 관리 시스템)까지 이어집니다. 의무화로 늘어난 업무량은 장비 수, 영역별 항목 수, 주기를 곱한 판정 건수로 계산해야 실제 규모가 보입니다.

1.3.1 2024년 예방점검체계에서 고시로 올라온 3개 분야 8개 점검항목

3개 분야 8개 점검항목의 구성

2024년 예방점검체계는 일상점검, 특별점검, 구조진단 3개 분야에 8개 점검항목을 두고, 장애관리·변경관리 같은 표준운영절차를 함께 마련했습니다[S07]. 일상점검은 CPU·메모리·디스크 상태, 메인 화면 접속 여부와 접속 시간, 인증서 유효기간과 도메인 만료일처럼 매일 확인해야 하는 운영 상태를 봅니다[S07]. 반면 특별점검은 성능처럼 일상점검보다 깊은 항목을 들여다보고, 구조진단은 시스템 구성 전반이 장애에 견디는 구조인지를 점검합니다. 일상점검은 빈도가 높고, 특별점검과 구조진단은 깊이가 큰 분야입니다. 고시의 안정성 기준과 8개 점검항목의 대응 관계는 시행 고시 원문에서 확인해야 하지만, 세 분야 모두 이행 기록을 요구받는다고 보고 준비하는 편이 안전합니다.

이 체계가 그 이전의 대책과 다른 점은 시점입니다. 2023년 장애 뒤의 처방이 노후 장비 교체와 이중화처럼 사고 뒤 복구 능력을 높이는 쪽이었다면, 예방점검체계는 사고가 나기 전에 무엇을 얼마나 자주 들여다볼지를 정하

는 쪽으로 무게를 옮겼습니다. 세 분야는 점검 주기와 깊이가 서로 다르므로, 같은 운영 인력과 같은 도구로 세 분야를 똑같이 다룰 수 없습니다. 매일 넓게 보는 일상점검은 자동 수집에, 깊게 보는 특별점검과 구조진단은 전문 인력의 분석과 그 근거 자료에 맞춰 인력과 도구를 나눠 배치하는 것이 현실적입니다.

분야	보는 대상	빈도와 깊이	주요 대상 영역
일상점검	자원 상태, 접속, 인증서·도메인 만료	매일, 얕고 넓게	OS·웹 서버·WAS·DBMS
특별점검	성능 중심의 심화 항목	주기적, 깊게	OS·WEB/WAS·DBMS
구조진단	시스템 구성 전반	장기 주기, 구조 전체	시스템 아키텍처

미들웨어와 DBMS까지 이어지는 점검 대상

점검 항목을 영역별로 나누면 OS 영역에 더해 웹 서버, WAS, DBMS 영역이 각각 별도 항목을 가집니다. 예를 들어 WAS 영역의 스레드 사용률·커넥션 풀 상태나 DBMS 영역의 세션 수·테이블스페이스 사용률은 OS 점검으로는 보이지 않는 지표입니다. 미들웨어와 DBMS 지표를 매일 수집하는 장치가 없으면 점검표의 해당 칸은 사람이 콘솔에 접속해 채우게 됩니다. OPENMARU APM(Application Performance Management, 애플리케이션 성능 관리)은 WAS와 DBMS의 내부 상태를 상시 수집하는 제품이므로, 미들웨어·DBMS 일상점검의 원자료를 사람 손 없이 공급하는 수집 계층으로 쓸 수 있습니다.

영역별로 점검 결과를 따로 남기더라도, 한 점검 날짜 아래에 OS·웹 서버·WAS·DBMS 결과가 같은 형식으로 나란히 모여 있어야 담당자가 그날 어느 영역부터 들여다볼지를 정할 수 있습니다. 결과의 형식이 영역마다 다르면 담당자는 영역을 오갈 때마다 문서를 읽는 방법부터 다시 익혀야 하고, 영역 사이의 이상을 이어 보는 일은 더 어려워집니다. 4.2절에서 다루는 OpsKnow의 예방점검 결과 자산이 날짜와 영역을 축으로 문서를 쌓는 이유가 여기에 있습니다.

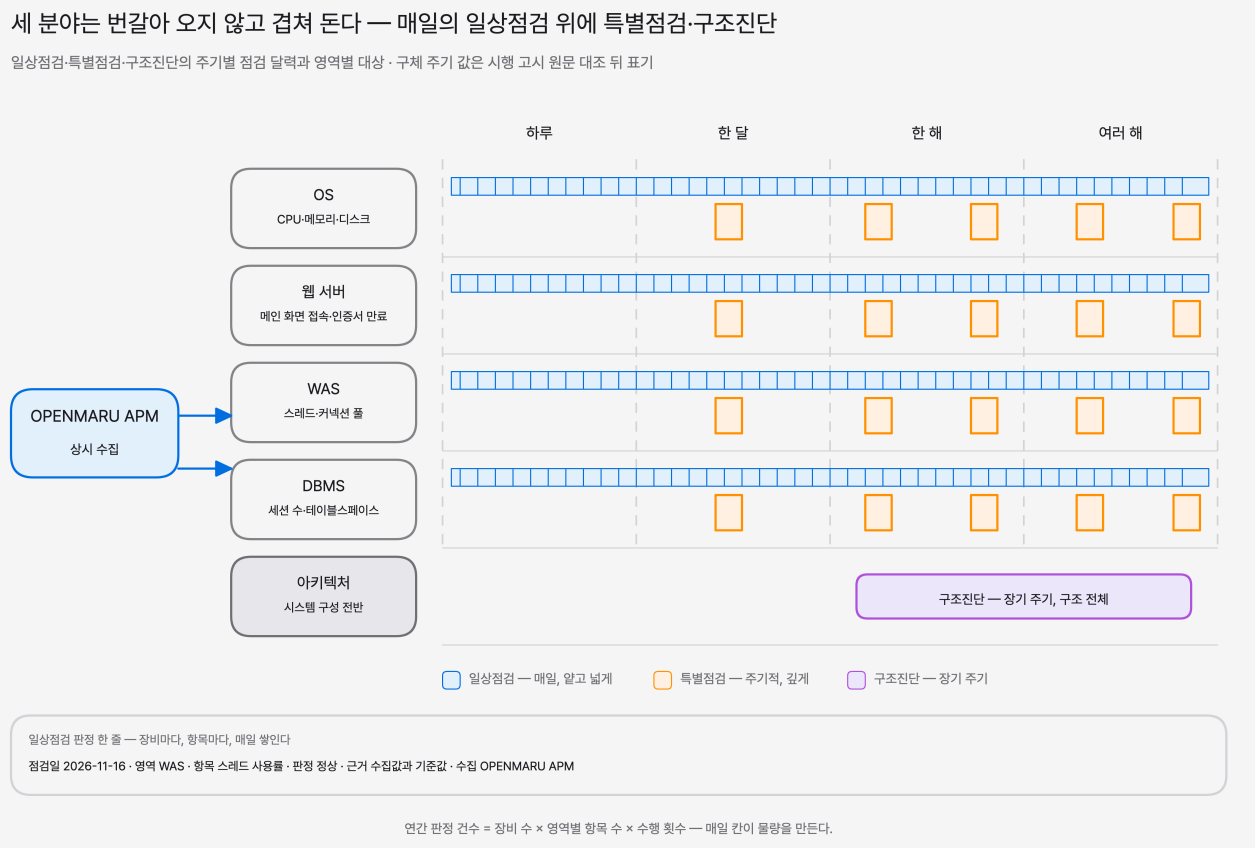


그림. 일상점검·특별점검·구조진단의 주기별 점검 달력과 영역별 대상

1.3.2 시스템 수와 항목 수, 주기를 곱해 계산하는 우리 기관 연간 점검 건수

연간 판정 건수를 구하는 산식

기관의 연간 점검 물량은 판정 건수로 세야 실제 업무량과 맞고, 점검표 장수로는 규모가 드러나지 않습니다. 판정 건수는 대상 장비 수에 영역별 점검 항목 수를 곱하고, 다시 연간 수행 횟수를 곱한 값을 3개 분야에 대해 더한 것입니다. 예를 들어 일상점검은 연간 수행 횟수가 업무일 수만큼 커지므로, 장비 수가 조금만 늘어도 판정 건수는 크게 늘어납니다. 기관은 아래 표에 자기 장비 수와 항목 수를 넣어 먼저 규모를 계산하고, 그 결과를 차년도 시행계획의 인력과 예산 산정 근거로 써야 합니다.

분야	대상 장비 수	영역별 항목 수	연간 수행 횟수	연간 판정 건수
일상점검	① 기관 입력	② 기관 점검표 기준	③ 업무일 수	① × ② × ③
특별점검	① 기관 입력	② 기관 점검표 기준	③ 고시 주기	① × ② × ③
구조진단	① 대상 시스템 수	② 진단 영역 수	③ 고시 주기	① × ② × ③
합계				세 분야 합

계산 결과를 해석하는 기준

산식으로 얻은 판정 건수가 크다는 사실 자체는 문제가 아닙니다. 문제는 각 판정에 정상 또는 이상의 근거가 붙어 있어야 하고, 이상 판정마다 조치와 재점검 기록이 이어져야 한다는 점입니다. 예를 들어 A4 시스템 14,132개가 공공 정보시스템의 94.0%를 차지한다는 등급 분포를 떠올리면[S01], 대부분 기관에서 판정 건수를 끌어

올리는 쪽은 하위 등급 시스템입니다. 산식의 결과는 판정과 기록을 사람 손 없이 남겨야 하는 범위를 정하는 데 쓰는 것이 맞고, 인력 증원 규모로만 해석하면 판정 근거를 남기는 문제가 그대로 남습니다.

1.4 감사가 묻는 것은 점검 횟수가 아닌 판정 결과와 증적

예방점검 의무화 이후 감사와 평가가 확인하는 대상은 점검표를 몇 번 냈는지보다 각 판정이 어떤 근거로 내려졌고 이상이 나왔을 때 무엇을 했는지입니다. 2023년 11월 행정전산망 장애에서 행정안전부는 라우터의 케이블 연결 모듈에 있는 포트 일부 이상을 원인으로 발표했고, 해킹 흔적은 없다고 밝혔습니다[S04][S05]. 이 결론이 나오기까지 초기에는 다른 장비가 원인으로 지목되기도 했는데[S05], 장비마다 상태를 판정한 기록이 평소에 쌓여 있으면 이런 원인 추정 단계를 줄이는 근거 자료로 쓸 수 있습니다. 아래 두 항목은 증적의 구성 요소를 정의하고, 등급별 의무를 예산 근거로 옮기는 방법을 정리합니다.

1.4.1 정상·이상 판정 근거와 조치 이력, 재점검 결과로 구성되는 증적 3요소

예방점검 증적의 세 가지 구성 요소

예방점검의 증적은 점검을 수행했다는 사실을 넘어, 판정의 근거와 그 뒤의 처리 과정을 감사인이 다시 따라갈 수 있게 남긴 기록입니다. 첫째 요소는 정상·이상 판정 근거로, 어떤 지표를 어떤 기준값과 비교해 판정했는지를 담습니다. 둘째 요소는 조치 이력으로, 이상 판정 뒤 누가 무엇을 승인하고 실행했는지를 담습니다. 셋째 요소는 재점검 결과로, 조치 뒤 같은 항목을 다시 점검해 정상으로 돌아왔는지를 확인한 기록입니다.

증적 요소	담는 내용	빠지면 생기는 문제
판정 근거	지표 값, 기준값, 판정 시각	정상 판정의 근거를 설명하지 못함
조치 이력	승인자, 조치 내용, 실행 시각	이상 판정 뒤 처리 여부를 입증하지 못함
재점검 결과	조치 뒤 같은 항목의 판정	조치가 효과가 있었는지 알 수 없음

세 요소를 한 기록으로 잇는 방식

세 요소가 각각 다른 시스템에 흩어져 있으면 감사 때 담당자가 점검표, 메일, 작업 대장을 다시 맞춰야 합니다. 예를 들어 디스크 사용률 이상 판정은 모니터링 화면에, 정리 작업 승인은 메일에, 재점검 결과는 다음 날 점검표에 남아 있다면 세 기록을 잇는 것은 사람의 기억입니다. 판정, 조치, 재점검이 같은 기록 번호로 묶여 있으면 감사인은 한 번의 조회로 전체 경과를 확인할 수 있습니다. OPENMARU OpsKnow는 장애 예방점검 결과, Incident 케이스, 일정별 운영 보고(일·주·월·분기·연) 세 자산을 한 저장소에 쌓으므로, 판정부터 재점검까지를 같은 저장소에서 조회할 수 있습니다.

감사 현장에서 먼저 요구되는 자료의 모양을 떠올리면 세 요소가 왜 한 기록이어야 하는지가 분명해집니다. 점검 결과는 점검 일시, 정상·이상 판정, 측정값, 담당자, 근거가 된 로그나 지표의 위치가 한 줄에 담긴 표로 요구되는 경우가 많습니다. 이상으로 판정된 항목은 임시 조치, 근본 조치, 재발방지 조치로 나뉜 조치 이력이 뒤따라야 하고, SLA 기준을 밑돈 구간은 언제 시작해 어떻게 복구됐는지가 시간 순서로 남아 있어야 합니다. 감사 초기의 자료 제출 단계에서 이 기록을 여러 도구에서 따로 뽑아 맞추다 보면 시간이 걸릴 뿐 아니라 수치끼리 어긋나는 일이 생기므로, 처음부터 한 저장소에서 같은 형식으로 꺼낼 수 있게 해 두는 편이 감사 대응 부담을 줄입니다.

1.4.2 등급별 의무 차이를 예산 산정 근거로 옮기는 정리표

등급별로 달라지는 점검 강도와 기록 요건

등급별 의무 차이는 목표복구시간과 장애 보고 의무에서 가장 크게 드러나며, 백업과 실전환 훈련은 모든 등급에 공통으로 적용됩니다[S02]. 따라서 예산 요구서에는 상위 등급의 판정 속도와 하위 등급의 판정 규모를 구분해 적는 편이 설득력이 있습니다. 다음 표는 등급별 시스템 수와 행정예고 단계의 목표복구시간, 기관이 예산 요구서에 옮길 점검·기록 요건을 정리한 것입니다. 표의 마지막 열은 앞서 정리한 의무를 이행하기 위해 기관이 준비할 항목을 권고로 적은 것이며, 법령 문구를 그대로 옮긴 값은 아닙니다.

등급	시스템 수	목표복구시간(행정예고안 보도)	예산 요구서에 옮길 점검·기록 요건
A1	73개	1시간 이내	상시 판정, 장애 즉시 보고 기록, 실전환 훈련 단계별 기록
A2	157개	A1과 A3 사이	상시 판정, 장애 보고 기록, 실전환 훈련 기록
A3	671개	A2와 A4 사이	일상점검 자동 판정, 특별점검 판정 기록
A4	14,132개	3주 이내	대량 일상점검 자동 판정, 백업·소산 판정 기록

시스템 수는 2026-08-27 발표[S01], 목표복구시간은 행정예고안 보도[S02][S03] 기준입니다.

정리표를 예산 근거로 쓰는 순서

정리표를 예산 요구서에 옮길 때는 먼저 기관 시스템을 등급별로 나누고, 등급마다 위 표의 요건 가운데 현재 사람이 처리하는 항목을 표시합니다. 그다음 1.3.2의 산식으로 그 항목의 연간 판정 건수를 계산하면, 수작업으로 남길 수 없는 판정 규모가 숫자로 드러납니다. 미들웨어와 DBMS 판정은 지표 수집이 먼저 갖춰져야 하므로 OPENMARU APM 같은 수집 계층의 범위를 함께 적어야 합니다. 수집된 지표로 판정과 보고, 조치 승인을 이어 주는 OpsKnow Agent의 역할을 같은 요구서에 적어 두면, 수집·판정·기록이 한 사업으로 묶여 증적이 끊기지 않습니다.

예산 요구서에서 점검 항목 전부를 사람의 인건비로 환산할 필요는 없다는 점도 함께 적어 두면 심의 설명이 쉬워집니다. 값을 모으기만 하면 판정할 수 있는 항목은 수집 도구와 판정 코드에 맡기고, 사람이 판단해야 하는 항목만 인력 산정에 넣으면 증액 요구의 크기를 근거와 함께 줄일 수 있습니다. 이때 도구 도입비와 인건비를 한 줄로 합치지 않고 「자동 판정으로 넘기는 항목」과 「사람 확인으로 남는 항목」을 나누어 적어야, 심의 담당자가 늘어나는 점검 양에 비해 예산이 왜 그만큼만 늘어나는지를 따라갈 수 있습니다.

2장: 예방점검 체계가 없을 때와 단순 자동화만 했을 때 — 판정 근거 없는 점검표의 감사 리스크

행정안전부는 2024년 10월 일상점검·특별점검·구조진단 3개 분야와 8개 예방점검 항목으로 이루어진 정보시스템 예방점검체계를 마련했습니다[S06][S07]. 1장에서 살펴본 고시 제2026-23호는 이 체계 위에서 예방점검과 장애 관리를 기관이 지켜야 할 기준으로 묶었습니다. 행정예고 단계에 알려진 내용을 보면 이 고시는 46개 안정성 기준과 표준운영절차(SOP), 그리고 서비스 수준 협약(SLA) 체결 의무를 함께 담고 있습니다[S02].

의무가 되면 점검을 했다는 사실만으로는 부족한데, 감사에서는 점검 결과를 어떤 근거로 정상이나 이상으로 판정했는지, 이상이면 누가 승인해 무엇을 조치했는지, 조치 뒤 다시 점검한 결과가 무엇인지를 묻기 때문입니다. 즉 의무화 이후 예방점검의 산출물은 채워진 점검표 한 장보다 판정과 조치를 뒷받침하는 증적의 묶음에 가깝습니다. 지금 쓰는 점검 방식이 증적을 남기는지부터 따져 보아야 차년도 사업 범위를 정할 수 있는 이유도 여기에 있습니다.

이 장에서는 예방점검 체계가 없을 때 드는 비용을 2023년 11월 행정전산망 장애 사례로 정리했습니다. 스크립트 점검과 양식 보고서를 중심으로 한 단순 자동화가 채우는 범위와 비워 두는 범위도 함께 나누어 보았습니다. 끝에는 감사 질문 다섯 개로 우리 기관의 예방점검 수준을 스스로 판정하는 자가진단표를 실었습니다.

2.1 2023년 11월 행정전산망 마비와 라우터 포트 불량이 드러낸 노후 장비 점검 공백

2023년 11월 17일 지방행정전산서비스에 장애가 발생해 지방자치단체의 민원 업무가 멈췄습니다[S04]. 정부는 11월 25일 원인을 네트워크 장비인 라우터의 포트 불량으로 발표했고, 해킹 흔적은 없다고 밝혔습니다[S04]. 이 장애는 장비별 점검 이력과 증적이 없을 때 원인 규명에 시간이 더 든다는 점을 공공 정보시스템 운영 조직에 보여 준 사례입니다. 근본 원인을 두고 다른 시각의 보도도 이어졌으므로[S05], 아래 경과는 정부 발표를 기준으로 정리하고 원인 자체는 새로 판단하지 않습니다.

2.1.1 정부 발표 기준의 장애 경과 — 발생부터 포트 전환과 원인 발표까지

행정전산망 장애의 정부 발표 경과

정부 발표를 날짜순으로 놓으면 서비스 복구와 원인 규명이 서로 다른 속도로 진행된 것이 드러납니다. 장애는 2023년 11월 17일에 발생했고, 서비스는 11월 18일 오전 9시에 다시 열렸습니다. 11월 19일에는 문제가 된 포트를 교체해 연결을 바꾼 뒤 지연이 풀리면서 정상화되었습니다. 원인은 대전센터 라우터 가운데 광주센터와 연결된 포트의 불량으로 정리되었고, 정부는 이를 11월 25일 브리핑에서 발표했습니다[S04].

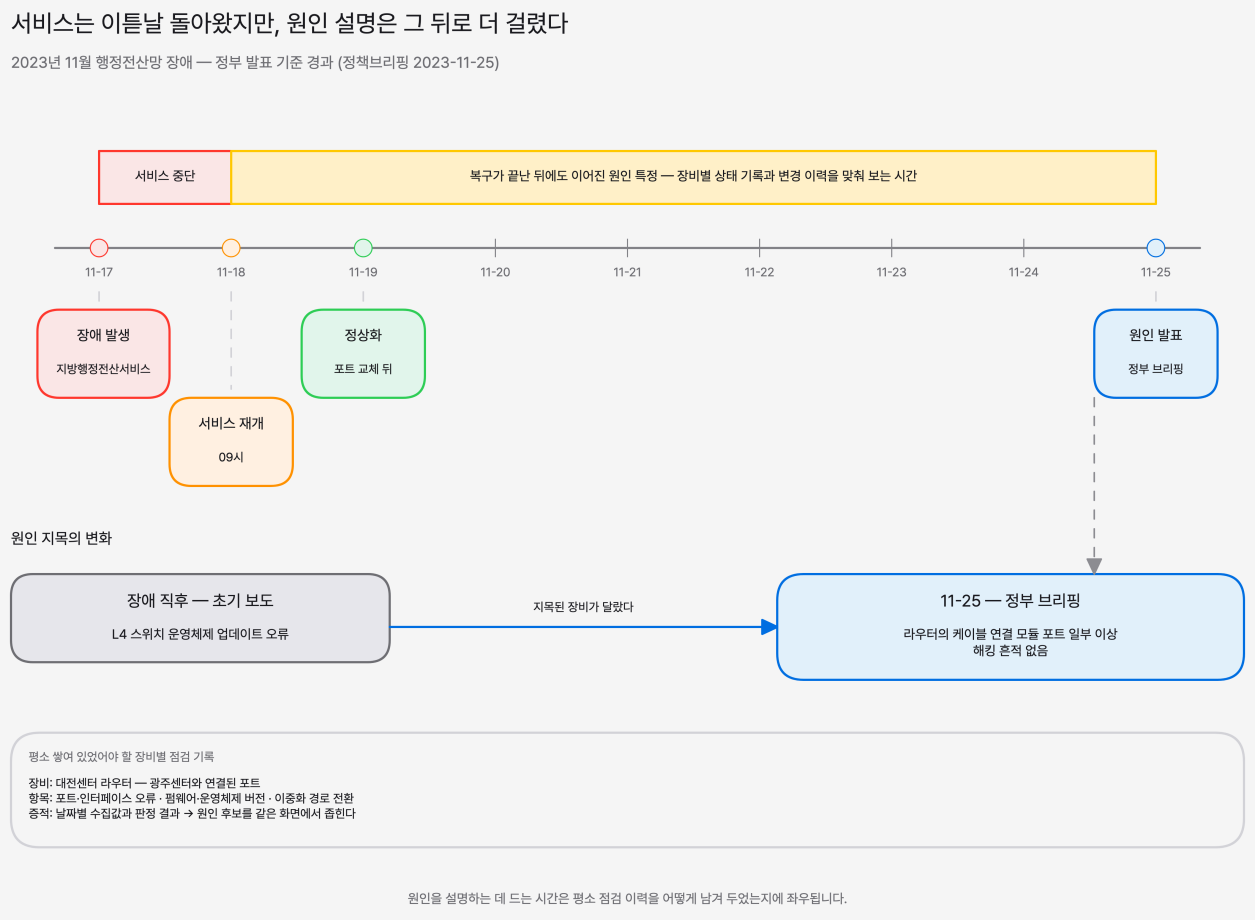


그림. 2023년 11월 행정전산망 장애의 정부 발표 기준 경과 — 발생, 서비스 재개, 포트 교체와 정상화, 원인 발표

서비스 재개와 원인 발표의 시차

복구 조치와 원인 특정 사이에 시차가 생기는 것은 대형 정보시스템 장애에서 드문 일이 아닙니다. 서비스를 다시 여는 일은 우회 경로나 장비 전환으로 먼저 할 수 있지만, 원인을 공식적으로 밝히려면 장비별 상태 기록과 설정 변경 이력을 맞춰 보아야 하기 때문입니다. 이때 장비마다 점검 결과가 날짜별로 한곳에 쌓여 있으면 원인 후보를 빠르게 좁힐 수 있습니다. 반면 점검 기록이 담당자별 파일이나 업체별 보고서에 흩어져 있으면 후보를 하나씩 배제하는 데 시간이 더 걸립니다. 복구 이후의 시간, 곧 원인을 설명하는 데 드는 시간은 평소 점검 이력을 어떻게 남겨 두었는지에 좌우됩니다.

2.1.2 초기 원인 지목과 최종 결론의 차이로 드러난 점검 이력 부재의 비용

장애 초기의 원인 지목 변화

장애 직후에는 L4 스위치(4계층 스위치, 서버 간 트래픽을 나눠 보내는 네트워크 장비)의 운영체제 업데이트 오류가 원인으로 지목되었습니다. 그러나 정부의 최종 발표는 라우터 포트 불량이었고, 장애 초기와 최종 결론이 가리킨 장비가 서로 달랐습니다[S05]. 아래 표는 공개 보도에 나타난 원인 지목의 변화를 시점별로 나눈 것입니다[S04][S05].

시점	지목된 원인	근거의 형태
장애 직후	L4 스위치 운영체제 업데이트 오류	초기 보도
2023-11-25 정부 발표	라우터 포트 불량, 해킹 흔적 없음	정부 브리핑
발표 이후	근본 원인에 대한 다른 시각	언론 보도

원인 특정 지연이 남기는 운영 비용

원인 지목이 바뀐 이유를 외부에서 단정할 수는 없지만, 장비별 점검 결과와 변경 이력이 같은 형식으로 쌓여 있었다면, 운영체제 업데이트가 있었던 장비와 포트 오류가 늘던 장비를 같은 화면에서 비교할 수 있었을 것입니다. 원인 특정이 늦어지면 재발 방지 대책의 범위도 늦게 정해지는데, 같은 모델의 라우터를 쓰는 다른 센터를 언제 점검할지, 교체 예산을 어느 장비에 먼저 쓸지 같은 결정이 원인 발표 뒤로 밀립니다. 점검 이력이 없는 상태의 비용이 장애 시간보다 원인 설명과 후속 결정의 지연에서 더 크게 드러나는 것도 이 때문입니다.

2.1.3 라우터 노후화 논란이 보여 주는 장비 수명 주기 점검 항목의 필요성

장비 도입 시점과 내구연한 정보

장애 뒤 보도에서는 장비 노후화를 원인으로 보는 시각도 소개되었습니다[S05]. 장비의 도입 시점과 내구연한은 장애가 난 뒤에야 공개 논의에 오르곤 하지만, 원래는 평소 점검표에 들어 있어야 하는 항목입니다. 장비가 내구연한에 가까워지면 부품 노후로 인한 간헐적 오류가 늘고, 제조사 지원이 끝나 펌웨어 수정본을 받지 못하는 경우도 생깁니다. 그래서 하드웨어 상태를 하루 단위로 보는 일상점검과 별도로, 장비의 수명 주기를 주기적으로 확인하는 점검 항목이 필요합니다.

수명 주기 점검 항목 예시

아래 항목은 특정 고시 별표를 옮긴 것이 아니며, 네트워크·서버 장비 운영에서 일반적으로 쓰는 수명 주기 점검 항목을 정리한 예시입니다. 핵심은 각 항목의 결과가 점검일과 함께 남아 추세로 비교할 수 있어야 한다는 점입니다.

점검 항목	확인 내용	남겨야 할 증적
도입 시점과 내구연한	내구연한 도래 여부와 교체 계획 반영 여부	자산 대장 갱신일, 교체 계획 문서
펌웨어·운영체제 버전	제조사 지원 종료 여부, 보안 수정본 적용 여부	버전 이력, 적용 승인 기록
포트·인터페이스 오류	오류 카운터의 주간 추세	날짜별 수집값과 판정 결과
이중화 경로 전환	예비 경로로 실제 전환되는지	전환 시험 일시와 결과

이 표에서 중요한 것은 마지막 열입니다. 같은 항목을 점검하더라도 결과가 날짜별 수집값으로 남지 않으면 오류가 늘어나는 추세를 볼 수 없고, 교체 우선순위를 근거와 함께 설명할 수도 없습니다. 수명 주기 점검의 가치는 한 번의 확인보다 이력의 누적에서 나옵니다.

2.2 스크립트 점검과 양식 보고서 — 단순 자동화가 채우는 범위와 비어 있는 판정

예방점검체계에 대응할 때 가장 먼저 떠올리는 방식은 단순 자동화입니다. 단순 자동화는 일상점검 항목을 스크립트로 매일 수집하고, 정해진 임계치와 비교한 결과를 양식 보고서에 자동으로 채우는 방식을 말합니다. 이 방식은 점검을 빠뜨리지 않았다는 사실을 남기는 데에는 효과가 있습니다. 다만 판정의 근거, 운영체제(OS)·미들웨어·DBMS 사이의 연관, 조치와 재점검의 기록은 여전히 사람이 따로 챙겨야 하는 일로 남습니다.

2.2.1 일상점검 항목을 매일 수집해 점검표 칸을 채우는 방식의 구조

일상점검 항목의 수집 대상

일상점검은 서버의 CPU·메모리·디스크 상태, 업무 시스템 메인 화면의 접속 여부와 접속 시간, 인증서 유효기간과 도메인 만료일 같은 항목을 봅니다[S07]. 여기에 웹 서버와 WAS의 스레드·세션 상태, DBMS의 커넥션 수와 테이블스페이스 사용률 같은 미들웨어·DBMS 지표가 더해집니다. 이 가운데 미들웨어와 DBMS 지표는 이미 운영 중인 성능 관리 도구가 상시 수집하는 경우가 많습니다. 예를 들어 OPENMARU APM은 WAS와 DBMS의 응답 시간, 스레드, 커넥션 풀 지표를 상시 수집하는 제품이므로, APM을 운영하는 기관이라면 미들웨어·DBMS 지표 수집은 이미 갖춘 셈입니다.

스크립트 수집에서 양식 보고서까지의 흐름

단순 자동화의 구조는 수집, 임계치 비교, 양식 채움, 보고서 보관의 네 단계로 정리됩니다. 점검 스크립트는 정해진 시각에 대상 서버에서 값을 읽고, 각 값을 미리 정한 임계치와 비교해 정상 또는 이상 표시를 붙입니다. 스크립트는 그 표시를 점검표 양식의 해당 칸에 채우고, 완성된 보고서 파일을 날짜별 폴더에 보관하는 데까지 처리합니다. 이 구조를 쓰면 장비마다 매일 반복되는 점검의 양을 사람 손 없이 처리할 수 있으므로, 점검 누락이 줄어듭니다.

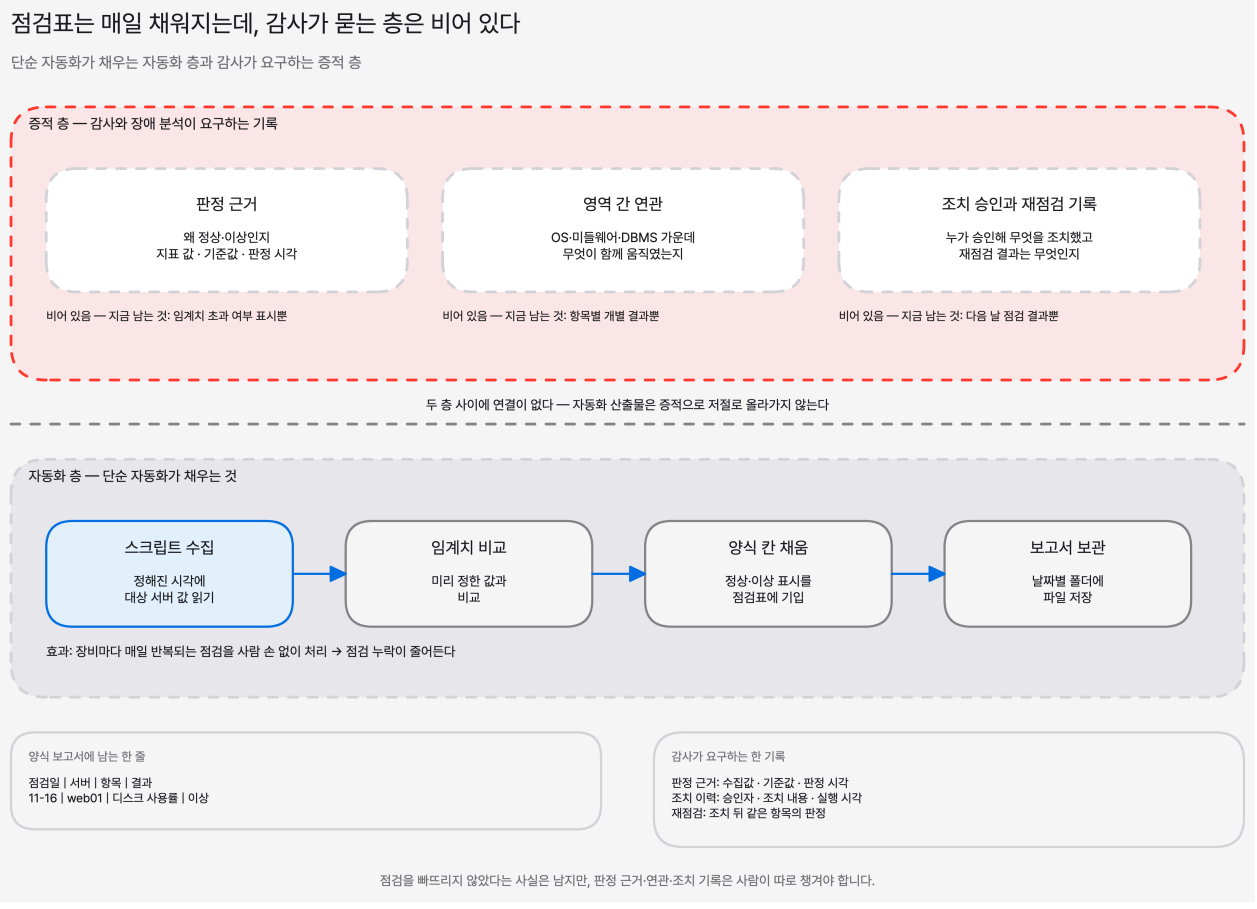


그림. 단순 자동화가 채우는 자동화 층과 감사가 요구하는 증적 층의 두 층 구조

2.2.2 임계치 초과 여부만 남고 원인 연관과 조치 이력이 빠지는 세 가지 빈칸

양식 보고서에 남지 않는 세 가지 기록

양식 보고서에 남는 것은 결국 「임계치를 넘었는가」라는 한 줄의 결과입니다. 그런데 감사와 장애 분석에서 필요한 기록은 그 결과를 둘러싼 맥락이고, 이 맥락이 단순 자동화 산출물에서 세 군데 비어 있습니다. 아래 표는 그 빈칸을 단순 자동화 산출물에 남는 기록과 감사에서 요구하는 기록으로 나누어 대조한 것입니다.

빈칸	단순 자동화 산출물에 남는 것	감사와 장애 분석에서 요구하는 것
판정 근거	임계치 초과 여부 표시	왜 정상 또는 이상인지, 어떤 기준과 이력으로 판단했는지
영역 간 연관	OS·미들웨어·DBMS 항목별 개별 결과	한 영역의 이상이 다른 영역의 어느 지표와 함께 움직였는지
조치와 재점검	다음 날 점검 결과	누가 승인해 무엇을 조치했고, 재점검 결과가 무엇인지

임계치 설정만으로 감사에 대응하기 어려운 이유

세 빈칸이 생기는 이유는 스크립트가 값을 비교할 수는 있어도 그 비교 결과를 해석하지는 못하기 때문입니다. 예를 들어 WAS 응답 시간이 임계치를 넘은 날 DBMS 커넥션 대기도 함께 늘었다면, 두 결과는 같은 원인의 두 증상일 수 있습니다. 하지만 항목별로 따로 채운 점검표에는 두 칸이 각각 「이상」으로만 남고, 둘 사이의 관계는 담당자의 기억에만 남습니다. 공공 예방점검 시장의 경쟁 업체 자료 두 건(A사·B사)도 「임계치 설정만으로는 감사 대응이 안 된다」는 점을 공통으로 강조합니다. 단순 자동화를 도입한 기관도 판정 근거와 조치 이력을 남기는 별도의 체계를 함께 갖추어야 감사 질문에 답할 수 있습니다.

스크립트나 모니터링 도구의 임계치로 점검을 대신하는 방식 자체가 잘못된 것은 아닙니다. 값을 모으기만 하면 끝나는 항목은 수집 수단에 맡기고, 판단이 필요한 항목에 사람의 시간을 쓰는 것이 운영 인력을 아끼는 길입니다. 다만 이 방식이 점검을 수행한 것으로 설명되려면 도구를 갖췄다는 사실만으로는 부족하고, 임계치를 어떤 근거로 정했고 언제 누가 바꿨으며 그 관리 결과가 어떻게 남았는지를 보여 줄 수 있어야 합니다. 임계치 비교 결과만 남는 단순 자동화가 감사에서 약한 이유는 바로 이 관리 기록이 점검표 밖에 흩어져 있기 때문입니다.

2.2.3 특별점검·구조진단과 보안 패치를 범위 밖에 두는 시중 도구의 한계

시중 점검 도구의 범위 비교

예방점검체계는 일상점검만이 아니라 특별점검과 구조진단까지 3개 분야로 이루어져 있습니다[S07]. 그런데 시중의 점검 도구 자료를 보면 일상점검 스크립트와 양식 보고서에 초점이 모여 있고, 나머지 분야는 범위 밖으로 두거나 기록 지원에 그칩니다. 아래 표는 경쟁 업체 자료 두 건의 범위 서술을 익명으로 요약한 것입니다.

점검 범위	A사 자료	B사 자료
일상점검 스크립트와 양식 보고서	범위 안	범위 안
특별점검	범위 밖	결과 기록만 지원

점검 범위	A사 자료	B사 자료
구조진단	범위 밖	외부 전문 기관에 위임
보안 취약점 진단과 패치	범위 밖	범위 밖
장애·변경 관리	ITSM 연동에 위임	ITSM 연동에 위임

도구 범위 밖에서 사람이 처리하는 업무

표의 아래쪽 세 행은 기관이 의무화 이후에도 사람의 수작업으로 메워야 하는 업무입니다. 예를 들어 한국인터넷진흥원이 운영하는 보호나라&KrCERT/CC는 보안공지와 CVE(Common Vulnerabilities and Exposures, 공개 취약점 식별 번호) 기반의 취약점 정보를 제공합니다[S10]. 이 권고를 우리 기관의 서버 목록과 대조해 조치 대상을 고르고 패치 적용을 승인받는 일은 점검 도구 밖에서 담당자가 직접 처리해야 합니다. 장애·변경 관리를 ITSM(IT 서비스 관리) 도구에 맡긴 경우에도 점검 결과와 장애 기록이 서로 다른 시스템에 남아, 두 기록을 잇는 작업이 보고서를 만들 때마다 반복됩니다. 두 경쟁 자료 모두 AI나 LLM(대규모 언어 모델)을 쓰는 판정 기능을 다루지 않는다는 점도 같은 한계를 보여 줍니다.

2.3 분기·연간 보고와 SLA 준수율 산출에서 드러나는 수작업 비용

예방점검의 결과는 일일 점검표에서 끝나지 않고 주간·월간·분기·연간 보고로 이어집니다. 2026년 8월 27일 행정안전부가 등급을 정한 공공 정보시스템은 15,033개이고, 그중 A1 73개, A2 157개, A3 671개, A4 14,132개입니다[S01]. 시스템마다 점검 결과가 매일 쌓이고 이를 주기마다 다시 묶어야 하므로, 보고 단계의 수작업이 운영 조직의 감사 대응력을 좌우합니다. SLA 체결이 의무가 되면[S02] 준수율 산출 근거를 원자료까지 거슬러 보여 줄 수 있는지가 새로운 점검 대상이 됩니다.

2.3.1 엑셀·문서 양식으로 취합하는 분기 보고서의 작성 부담

보고 주기별 수작업 단계

분기 보고서는 그보다 짧은 주기의 기록을 모두 다시 읽어야 만들 수 있습니다. 일일 점검표는 자동으로 만들어지더라도, 주간·월간 보고는 그 점검표에서 이상 항목을 골라 내고 장애 기록과 조치 내역을 다른 시스템에서 찾아 붙이는 방식으로 작성됩니다. 아래 표는 보고 주기별로 담당자가 손으로 처리하는 단계를 정리한 것입니다.

보고 주기	담당자의 수작업 단계	근거가 흩어지는 곳
일일	이상 표시 항목 확인, 조치 필요 여부 메모	점검표 파일, 메신저 대화
주간·월간	이상 항목 발췌, 장애 기록과 조치 내역 대조, 지표 차트 캡처	ITSM 도구, 결재 문서, 모니터링 화면
분기·연간	하위 보고서 취합, 추세 요약, SLA 준수율 계산, 양식 문서 편집	엑셀 집계표, 위탁 운영사 보고서

취합 작업이 감사 대응력에 미치는 영향

이 흐름에서 분기·연간 보고서는 원자료를 직접 읽지 않고 하위 보고서를 다시 요약한 결과가 됩니다. 감사에서 특정 수치의 근거를 물으면 담당자는 분기 보고서에서 월간 보고서로, 다시 일일 점검표와 장애 기록으로 거꾸로

찾아 내려가야 합니다. 담당자가 바뀌었거나 중간 보고서의 집계 방식이 달라졌다면 같은 수치를 다시 만들어 내지 못하는 경우도 생깁니다. 보고 수작업의 비용은 작성 시간만이 아니라 보고서의 수치를 원자료로 다시 설명할 수 있는지에서도 드러납니다.

수작업 취합이 남기는 또 하나의 비용은 담당자 교체 때 드러납니다. 순환보직으로 담당자가 바뀌면 어떤 점검표 파일과 어떤 메신저 대화가 어느 보고서의 근거였는지에 대한 기억도 함께 사라지고, 후임자는 다음 분기 보고서를 쓰기 위해 이전 보고서의 계산 과정을 처음부터 다시 추적해야 합니다. 보고서가 원자료와 연결되지 않은 채 사람의 손을 거쳐 만들어지는 한, 보고 품질은 담당자 개인의 숙련도에 묶여 있게 됩니다.

2.3.2 흩어진 SLA 준수율 산출 근거가 만드는 감사 지적 위험

SLA 준수율 산출 근거의 구성

행정예고안에 따르면 SLA 체결 의무는 기관이 직접 운영하는 시스템뿐 아니라 민간 클라우드와 위탁 운영 시스템에도 적용됩니다[S02]. SLA 준수율은 측정 기간 동안 약속한 서비스 수준을 지킨 비율이며, 이 값을 계산하려면 장애의 시작·종료 시각, 계획된 작업 정지의 제외 근거, 측정 방법 같은 원자료가 필요합니다. 이 원자료가 점검 도구, ITSM 도구, 위탁 운영사 보고서에 나뉘어 있으면 준수율은 엑셀 집계표에서 사람이 맞춰 넣은 값이 됩니다. 이때 집계표의 수치와 원자료 사이의 연결이 끊어지면 수치 자체가 맞더라도 감사에서 근거를 보이기 어렵습니다.

SLA 산출 과정의 감사 지적 사례 유형

아래 표는 산출 근거가 흩어져 있을 때 감사에서 자주 나올 수 있는 질문 유형을 정리한 예시입니다. 어느 경우든 문제의 핵심은 계산 실수보다 준수율의 계산 과정을 원자료와 함께 다시 보여 줄 수 없다는 데 있습니다.

위험 유형	감사에서 받는 질문
장애 시작 시각이 문서마다 다름	준수율 계산에 어느 시각을 썼고, 그 근거 기록은 무엇인가
계획 정지 제외 근거 부족	제외한 정지 시간이 사전에 승인된 작업이었다는 기록이 있는가
위탁 운영사 보고와 기관 측정값 불일치	두 값이 다른 이유와 최종 값을 정한 근거는 무엇인가
재계산 불가	같은 원자료로 같은 준수율을 다시 산출할 수 있는가

이런 위험은 준수율 목표치를 얼마로 정하든 같은 방식으로 나타납니다. SLA 의무화에 대비하는 첫 작업은 목표치 협상보다 장애·작업·점검 기록을 같은 저장소에서 같은 시각 기준으로 남기는 일입니다.

위탁 운영 구조에서는 이 문제가 한 겹 더 복잡해집니다. 기관, 운영을 맡은 위탁 사업자, 장비와 소프트웨어를 유지보수하는 사업자가 각자 다른 도구로 기록을 남기면, SLA를 밑돈 구간이 언제 시작해 누구의 조치로 끝났는지를 세 곳의 기록을 맞춰 보며 재구성해야 합니다. 실무를 위탁하더라도 서비스 수준에 대한 최종 책임은 기관에 남으므로, 계약 단계에서 기록을 남길 저장소와 시각 기준, 기록 형식을 기관 쪽으로 정해 두어야 합니다. SLA 미달 기록이 판정 시점에 자동으로 남는 구조라면 위탁 사업자의 월간 보고와 기관의 측정값이 처음부터 같은 원자료에서 나오므로, 수치가 엇갈리는 일 자체가 줄어듭니다.

2.4 감사 질문 기준으로 보는 우리 기관 예방점검 수준 자가진단 5문항

2023년 행정전산망 장애와 단순 자동화의 세 빈칸, 보고 수작업의 비용은 결국 감사 질문 다섯 개로 모입니다. 「행정기관 및 공공기관 정보시스템 안정성 고시」는 예방점검과 장애 관리를 안정성 기준으로 묶었으므로[S02], 기관의 예방점검 수준도 점검 횟수보다 이 질문에 증적으로 답할 수 있는지로 판정하는 편이 정확합니다. 아래 자가진단은 판정 근거, 승인 기록, 재점검 결과, 이력 연결, 보고서 재생성의 다섯 문항으로 이루어지며, 문항별 충족 여부로 기관의 단계를 나눕니다.

2.4.1 판정 근거부터 보고서 재생성까지 감사에서 받는 5가지 질문

감사에서 받는 질문 다섯 가지

다섯 문항은 각각 행정전산망 장애 사례와 단순 자동화 분석에서 확인한 빈칸 하나씩에 대응하며, 문항마다 「예」로 답하려면 말로 설명하는 수준을 넘어 해당 기록을 바로 꺼내 보일 수 있어야 합니다.

번호	감사 질문	「예」로 답하기 위한 증적
1	점검 결과의 판정 근거를 보여 줄 수 있는가	판정 기준, 비교한 이력 값, 판정 사유가 결과와 함께 저장됨
2	조치마다 승인 기록이 있는가	승인자, 승인 시각, 승인 대상이 조치 기록과 연결됨
3	조치 뒤 재점검 결과가 남는가	같은 항목의 조치 전후 값과 재판정 결과
4	장애 이력과 점검 이력이 연결되는가	장애 기록에서 직전 점검 결과로, 점검 결과에서 관련 장애로 이동 가능
5	보고서가 원자료에서 다시 생성되는가	같은 기간을 지정하면 같은 수치의 보고서가 다시 만들어짐

승인 기록 문항이 가르는 수준 차이

다섯 문항 가운데 2번 승인 기록은 단순 자동화와 지능화를 나누는 기준이 되는 경우가 많습니다. 결재 문서나 메신저 대화에 남은 승인은 조치 기록과 자동으로 연결되지 않기 때문에, 감사 때마다 두 기록을 사람이 맞춰야 합니다. 반면 조치 요청마다 한 번만 쓸 수 있는 승인 코드를 발급하고 그 코드가 있어야 서버를 변경하도록 하면, 승인과 조치가 같은 기록 안에서 이어집니다. 적용 직전에 서버 상태를 다시 확인해 점검 때와 다르면 멈추도록 하면 3번 재점검 문항까지 같은 흐름에서 채워집니다. 즉 승인 기록 문항은 사람의 결정을 시스템이 증적으로 남기는지를 묻는 문항입니다.

2.4.2 자가진단 결과로 구분하는 체계 없음·단순 자동화·지능화 단계

충족 문항 수에 따른 단계 판정

충족한 문항 수를 세면 기관의 예방점검 단계를 체계 없음, 단순 자동화, 지능화의 세 가지로 나눌 수 있고, 아래 표의 미대응 리스크 열은 그 단계에 머물 때 가장 먼저 드러나는 문제를 적은 것입니다.

충족 문항 수	단계	현재 모습	미대응 리스크
0~1개	체계 없음	담당자별 수기 점검, 기록 형식이 제각각	장애 때 원인 규명 지연
2~3개	단순 자동화	스크립트 수집과 양식 보고서, 승인·재점검은 별도 문서	판정 근거와 승인 기록에 대한 감사 지적
4~5개	지능화	판정·승인·재점검·보고가 한 저장소에서 이어짐	판정 기준이 낡지 않도록 주기적 검토 필요

단계별 다음 행동과 운영 지식 소실 위험

체계 없음 단계의 기관은 점검 결과를 같은 형식으로 남기는 일부터 시작해야 하며, 이 단계에서는 2023년 장애와 같은 상황에서 원인 설명이 늦어지는 위험이 가장 큽니다. 단순 자동화 단계의 기관은 수집과 보고서 생성은 이미 갖추었으므로, 판정 근거와 승인 기록을 점검 결과에 붙이는 일이 다음 과제입니다. 이 단계에서 가장 자주 놓치는 위험은 운영 지식의 소실입니다. 장애 원인과 조치 요령이 담당자의 기억과 개인 문서에만 남으면, 담당자가 바뀔 때마다 같은 장애를 처음부터 다시 분석해야 하기 때문입니다.

지능화 단계는 판정과 조치, 보고가 한 저장소에서 이어지는 상태를 말합니다. 예를 들어 OPENMARU OpsKnow는 Incident 케이스, 예방점검 결과, 일·주·월·분기·연 단위 운영 보고를 한 저장소에 쌓는 운영 지식 저장소이고, OpsKnow Agent는 수집된 지표를 판정하고 조치 승인과 보고서 작성으로 잇는 AI 에이전트입니다. 이 구성에서는 다섯 문항의 증적이 운영 과정에서 함께 남으므로, 감사 대응을 위해 따로 문서를 만드는 일이 줄어듭니다. 따라서 자가진단 결과가 단순 자동화 단계로 나온 기관이라면, 차년도 사업 요구사항은 수집 범위를 넓히는 일보다 판정 근거와 승인 기록을 남기는 기능을 먼저 담는 편이 감사 리스크를 줄이는 데 효과적입니다.

3장: 단순 자동화에서 지능화로 — 수집부터 학습까지 6단계에서 달라지는 산출물

행정안전부가 2024년 10월에 마련한 정보시스템 예방점검체계는 일상점검·특별점검·구조진단 3개 분야와 8개 점검항목으로 짜여 있습니다[S06][S07]. 이 가운데 일상점검은 CPU·메모리·디스크 상태, 메인 화면 접속 여부와 접속 시간, 인증서 유효기간과 도메인 만료일처럼 장비마다 매일 확인해야 하는 항목을 다룹니다[S07]. 여기에 고시 제2026-23호가 더해졌고, 행정예고 단계에서 알려진 내용만 보아도 예방점검과 장애 관리를 포함한 안정성 기준 46개와 서비스 수준 협약(SLA) 체결 의무가 들어 있습니다[S02]. 따라서 예방점검은 기관이 형편에 따라 고르는 관리 활동에서, 이행 결과와 근거를 확인받아야 하는 의무로 성격이 바뀌었습니다.

의무가 된 점검은 점검을 했다는 기록만으로 끝나지 않습니다. 감사나 장애 보고 때 확인하는 내용은 어떤 값과 규칙으로 정상이라고 판정했는지, 이상이 나왔을 때 누가 승인해 무엇을 바꿨는지, 바꾼 뒤 다시 확인했는지입니다. 즉 점검의 산출물은 점검표 한 장보다 판정 근거, 승인 기록, 재점검 결과가 이어진 증적이어야 합니다. 행정안전부가 2026년 8월 공공 정보시스템 15,033개의 등급을 새로 정했으므로[S01], 장비마다 매일 쌓이는 점검 결과를 사람이 하나씩 판정해 증적으로 묶는 방식은 규모 면에서 유지하기 어렵습니다.

이 장에서는 예방점검 업무를 수집·판정·연관·조치·보고·학습 6단계로 나누고, 체계 없음·단순 자동화·AI 지능화 세 수준이 단계마다 어떤 산출물을 남기는지 비교합니다. 단순 자동화는 점검 스크립트가 값을 모아 양식 보고서의 칸을 채우는 수준을 말합니다. 지능화는 이 문서 전체에서 한 가지 뜻으로 씁니다. 판정은 같은 입력에 늘 같은 결과를 내는 결정적 코드가 맡고, 조치 여부는 사람이 결정하며, AI는 판정 결과와 근거를 사람이 읽을 수 있는 문장으로 설명합니다. 이 정의로 보면 세 수준의 차이는 수집보다 판정 이후 단계에서 크게 벌어지며, 3.1의 비교표는 차년도 사업 요구사항의 기능 골격으로 그대로 옮겨 쓸 수 있습니다.

비교에 등장하는 제품은 세 가지입니다. OPENMARU APM은 WAS와 DBMS의 지표, 트랜잭션 추적 정보를 수집하는 제품이므로 이 장의 수집·연관 단계에서 근거 자료를 공급합니다. OpsKnow Agent는 그 지표와 OS 점검 결과를 판정하고 보고서를 만들며, 서버를 바꾸는 조치를 사람의 승인 코드와 연결합니다. 마지막으로 OPENMARU OpsKnow(이하 OpsKnow)는 Incident 기록, 예방점검 결과, 정기 운영 보고를 쌓는 운영 지식 저장소이고, 학습 단계에서 다음 판정의 근거를 돌려줍니다.

3.1 체계 없음에서 AI 지능화까지 3단계 비교표 — 6개 운영 단계별로 남는 증적

예방점검의 세 수준은 같은 장비를 같은 주기로 점검하더라도 단계마다 남기는 산출물이 다릅니다. 2024년 체계가 일상점검·특별점검·구조진단을 나누어 정했고[S07], 이 가운데 일상점검 항목을 스크립트로 모으는 일은 기존 도구로도 할 수 있는 작업입니다. 그러나 고시 제2026-23호가 안정성 기준의 이행을 요구하므로, 기관이 어느 수준에 있는지는 수집 방식보다 판정·승인·조치·보고·학습에서 무엇이 기록으로 남는지로 판단해야 합니다. 아래 표는 승인을 별도 행으로 떼어 7행으로 정리한 결과입니다.

운영 단계	체계 없음	단순 자동화	AI 지능화
수집	담당자가 장비에 접속해 값을 확인하고 수기 점검표에 적음	스크립트가 정해진 주기로 수치를 모아 파일로 저장	같은 수집에 APM 트랜잭션 단위 지표를 더하고 원자료를 보존

운영 단계	체계 없음	단순 자동화	AI 지능화
판정	담당자 경험으로 정상·이상을 판단하며 근거가 남지 않음	임계치 초과 여부만 표시	결정적 코드가 판정하고 입력 값·적용 규칙·상태 구분을 함께 기록
연관	영역별 담당자가 따로 확인	OS·WAS·DBMS 경보가 각각 발생	트랜잭션 ID로 영역별 지표를 묶어 원인 후보를 제시
승인	구두나 메신저로 처리	시스템 밖 결재 문서로 처리해 점검 결과와 분리됨	1회용 승인 코드로 승인자·시각·대상을 점검 기록에 연결
조치	담당자가 직접 변경하고 기록이 빠지기 쉬움	스크립트로 실행할 수 있으나 승인과 연결되지 않음	승인 뒤 실행하고, 적용 직전 상태를 다시 확인하고, 적용 뒤 재점검
보고	사람이 문서를 새로 작성	양식 보고서의 칸을 자동으로 채움	저장된 원자료에서 보고서를 다시 생성하고 AI가 설명문을 작성
학습	담당자 기억에 의존	보고서 파일을 보관	Incident 기록이 OpsKnow에 쌓여 다음 판정의 근거로 조회됨

> 🎯 *결정 포인트 — 세 수준의 차이는 판정 행부터 벌어집니다. 차년도 요구사항은 판정 근거 기록, 승인과 조치의 연결, 원자료 기반 보고, 장애 기록의 지식 축적 네 가지를 중심으로 적으면 됩니다.*

3.1.1 수집은 이미 자동화된 영역이고 차이가 판정 이후에 벌어지는 이유

일상점검 항목 수집은 세 수준 사이 차이가 작다

일상점검이 보는 CPU·메모리·디스크 사용률과 화면 접속 시간, 인증서 만료일은 운영체제 명령이나 간단한 접속 시험으로 얻을 수 있는 값입니다[S07]. 체계를 갖추지 못한 기관도 셸 스크립트나 기존 모니터링 도구로 이 값을 모으는 데는 큰 어려움이 없고, 단순 자동화 단계의 기관은 이미 정해진 주기로 같은 값을 파일에 쌓고 있습니다. 지능화 단계도 수집 자체는 같은 값을 같은 주기로 가져옵니다. 다만 OPENMARU APM이 WAS 스레드, DB 커넥션 풀, SQL 수행 시간을 트랜잭션 단위로 함께 수집하므로, 뒤의 연관 단계에서 쓸 연결 정보가 수집 시점부터 붙는다는 차이가 있습니다.

어떤 점검 항목을 자동 수집에 맡길 수 있는지는 그 항목이 다루는 데이터의 성격으로 가려낼 수 있습니다. 자원 사용률이나 응답 시간처럼 시간에 따라 변하는 수치는 지표(Metric)로, 오류 로그나 접속·접근 기록처럼 사건이 일어난 시각과 내용을 담은 텍스트는 로그(Log)로, 요청 하나가 WAS와 DBMS를 거쳐 가는 경로는 트랜잭션 추적(Trace)으로, 임계치를 넘을 때 나가는 알림은 경보(Alarm)로 나누어 보면 수집 수단이 하나씩 짝지어집니다. 네 가지 어디에도 들어가지 않는 항목, 예를 들어 절차가 지켜졌는지나 문서가 최신인지처럼 사람이 확인해야 하는 항목이 남는데, 이런 항목은 자동 수집 대상이 아니라 사람 확인 결과를 같은 저장소에 기록하는 대상으로 따로 관리합니다. 이렇게 나누어 두면 수집 도구에 맡길 범위와 판정·확인 기록이 필요한 범위가 처음부터 갈라집니다.

경쟁 방식이 판정 이후를 범위 밖으로 두는 이유

시장에 나와 있는 예방점검 제품 자료 두 건(A사·B사)도 일상점검 항목을 매일 스크립트로 점검하고 양식 보고서를 남기는 데 초점을 맞춥니다. 두 자료에는 AI 모델에 대한 언급이 없고, 장애·변경·문제 관리는 별도 ITSM 제품과의 연동에 맡깁니다. A사 자료는 특별점검과 구조진단을 제품 범위 밖으로 두고, B사 자료는 특별점검을 기록 지원 수준으로만 다룹니다. 즉 시장의 일반적인 접근은 수집과 양식 보고를 자동화하는 데서 멈추고, 판정 근거와 승인 기록을 하나로 잇는 일은 기관의 수작업으로 남겨 둡니다. 두 자료 모두 임계치 설정만으로는 감사 대응이 어렵다고 강조하는데, 이 문제를 해결하는 기능은 범위에 넣지 않았습니다.

판정 이후 4단계에서 벌어지는 증적 차이

다음 표는 단계별로 세 수준 사이의 차이가 얼마나 큰지를 정리한 것입니다. 수집과 보고 양식은 이미 상당 부분 자동화되어 있으므로 차이가 작거나 중간이고, 판정·연관·승인과 조치·학습은 단순 자동화와 지능화 사이에서 산출물의 종류 자체가 달라집니다.

운영 단계	체계 없음과 단순 자동화 사이	단순 자동화와 AI 지능화 사이
수집	큼	작음
판정	작음	큼
연관	작음	큼
승인·조치	작음	큼
보고	큼	중간
학습	작음	큼

이 표에서 읽을 내용은 투자 순서입니다. 체계 없음에서 단순 자동화로 넘어갈 때는 수집과 보고 양식에 투자하면 효과가 바로 보입니다. 반면 단순 자동화에서 지능화로 넘어갈 때는 수집 도구를 바꾸는 일보다 판정 규칙을 코드로 옮기고 승인 기록을 점검 결과에 붙이는 일이 효과를 결정합니다. 따라서 이미 점검 스크립트를 갖춘 기관이 새 수집 도구부터 다시 구매하면, 의무화 요건 가운데 비어 있는 판정·승인·학습 부분은 그대로 남게 됩니다.

투자 순서를 정할 때는 자동화의 단계도 함께 보아야 합니다. 값을 모아 쌓는 단계, 임계치를 넘으면 알리는 단계, 평소와 다른 움직임을 찾아내는 단계, 원인과 조치를 설명하는 단계는 앞 단계의 기록이 뒤 단계의 재료가 되는 순서로 이어집니다. 앞 단계의 기록이 빠짐없이 쌓여 있지 않으면 뒤 단계의 판정과 설명은 근거가 얇아지므로, 지능화로 넘어가려는 기관일수록 먼저 수집과 판정 기록이 같은 형식으로 끊김 없이 쌓이고 있는지를 확인하는 것이 순서입니다.

3.1.2 단계마다 남는 산출물 열로 읽는 비교표 활용법

비교표 한 칸을 요구사항 한 문장으로 바꾸는 방법

3.1 비교표의 AI 지능화 열은 각 칸이 「무엇이 기록으로 남는가」를 적고 있으므로, 칸 하나를 제안요청서(RFP, Request for Proposal)의 기능 요구사항 한 문장으로 옮길 수 있습니다. 요구사항 문장에는 기능 이름보다 산출물을 적는 편이 검수하기 쉽습니다. 예를 들어 「AI 판정 기능」이라고 쓰면 사업자마다 다르게 해석하지만, 「판정마다 입력 값과 적용 규칙을 함께 저장해야 한다」라고 쓰면 검수 때 저장된 기록을 열어 확인할 수 있습니다.

고시 제2026-23호에 따른 예방점검 이행 기록과 SLA 이행 보고가 같은 기록에서 나오도록 요구사항을 맞추면, 사업 종료 뒤에도 증적 체계가 유지됩니다[S02].

비교표의 산출물	요구사항 문장 예
판정 근거 기록	점검 결과마다 입력 값, 적용 규칙, 판정 상태를 함께 저장하고 같은 입력에는 같은 판정을 낼 것
원인 후보	OS·WAS·DBMS 지표를 트랜잭션 단위로 연결해 원인 후보를 제시할 것
승인 기록	서버를 바꾸는 조치는 1회용 승인 코드가 있을 때만 실행하고 승인자·시각·대상을 기록할 것
재점검 결과	적용 직전에 대상 상태를 다시 확인하고 적용 뒤 재점검 결과를 같은 기록에 남길 것
원자료 기반 보고	모든 보고서를 저장된 원자료에서 다시 생성할 수 있게 할 것
운영 지식	장애 처리 기록을 원인·조치·재발방지 항목으로 저장하고 다음 판정에서 조회할 수 있게 할 것

자가 판정에 비교표를 쓰는 순서

기관의 현재 수준을 판정할 때는 행마다 실제 산출물을 하나씩 열어 보면 됩니다. 수집 행은 지난달 점검 원자료 파일이 남아 있는지, 판정 행은 그 원자료에서 정상 판정의 근거 규칙을 찾을 수 있는지를 확인합니다. 승인 행은 최근 서버 변경 한 건에 대해 승인자와 승인 시각을 점검 기록에서 바로 찾을 수 있는지를 봅니다. 한 행이라도 기록을 찾지 못하면 그 행의 수준은 한 단계 아래로 적는 것이 정직한 판정입니다. 이렇게 판정한 결과는 2장의 자가진단과 10장의 실행 순서에서 그대로 다시 씁니다.

3.2 수집·판정 단계 — 임계치 비교 대신 결정적 코드가 남기는 판정 근거

판정 단계의 결과물은 점검 항목마다 붙는 「정상」 또는 「이상」 표시이고, 감사 재현성은 이 표시가 같은 입력에서 늘 같게 나오는지에 달려 있습니다. 임계치 초과 여부만 표시하는 단순 자동화는 표시 자체는 남기지만, 어떤 값과 어떤 규칙으로 그 표시가 나왔는지를 함께 남기지 않습니다. 결정적 코드 판정은 같은 입력에 같은 결과를 내도록 만든 판정 코드이고, 판정 결과와 함께 입력 값, 적용 규칙, 상태 구분을 기록합니다. 고시 제2026-23호가 안정성 기준을 정하고 그 이행을 확인하는 구조이므로[S02], 판정의 재현 가능성은 증적의 기본 조건입니다.

3.2.1 수치 수집과 임계치 초과 표시에 머무는 단순 자동화의 판정

디스크 사용률 임계치 판정이 남기는 기록

단순 자동화의 판정은 수집한 값을 미리 정한 임계치와 비교해 초과하면 이상, 아니면 정상으로 표시하는 방식입니다. 예를 들어 디스크 사용률 임계치를 80%로 정한 기관에서 사용률이 78%로 나오면 보고서에는 「정상」 한 칸이 남습니다. 그런데 이 기록만으로는 임계치가 언제 누구의 결정으로 80%가 됐는지, 지난주에는 몇 %였는지, 같은 서버의 다른 파일시스템은 확인했는지를 알 수 없습니다. 판정 결과는 남지만 판정 근거는 스크립트 안에 묻혀 있고, 스크립트가 바뀌면 과거 판정을 다시 만들 방법이 사라집니다.

수집 실패를 정상으로 기록하는 단순 자동화의 결함

임계치 비교에는 값이 없을 때의 처리 규칙이 빠지기 쉽습니다. 인증서 유효기간을 확인하는 스크립트가 접속 시간 초과로 값을 가져오지 못하면, 많은 스크립트는 비교 대상이 없으므로 오류를 내지 않고 넘어가고 보고서 칸은 빈칸이나 정상으로 남습니다. 일상점검 항목에 인증서 유효기간과 도메인 만료일이 들어 있으므로[S07], 이런 처리는 실제로 만료가 임박한 인증서를 정상으로 기록하는 결과로 이어질 수 있습니다. 따라서 단순 자동화의 판정은 「확인하지 못함」과 「확인했고 정상임」을 구분하지 못한다는 구조적 결함을 안고 있습니다.

임계치 변경 이력이 점검 기록과 분리되는 문제

운영 중에는 오경보를 줄이려고 임계치를 조정하는 일이 자주 생깁니다. 임계치 변경은 스크립트나 설정 파일을 고치는 방식으로 이뤄지고, 변경 이력은 점검 결과와 다른 곳에 흩어집니다. 그 결과 감사에서 「지난 분기에 이 서버는 왜 한 번도 이상이 없었는가」라는 질문이 나오면, 담당자는 임계치가 그 사이 바뀌었는지를 기억이나 별도 문서로 답해야 합니다. 같은 입력에 같은 판정이 나온다는 재현성은 판정 규칙의 버전까지 기록에 남아야 성립합니다.

3.2.2 판정 근거와 상태 구분을 함께 남기는 결정적 코드 판정

OpsKnow Agent에서 판정과 설명을 맡는 주체

OpsKnow Agent는 사용자의 문장을 받아 판정 결과를 전달하는 일에만 LLM(Large Language Model, 대규모 언어 모델)을 씁니다. 문장 해석, 실행할 점검 명령의 선택, 결과 판정은 모두 결정적 코드가 맡습니다. 같은 서버를 같은 시각의 같은 데이터로 다시 판정하면 언제나 같은 결과가 나오고, 모델의 답변 방식이 바뀌어도 판정은 달라지지 않습니다. 이 구조는 「판정은 코드, 설명은 AI」라는 이 문서의 지능화 정의를 제품 동작으로 옮긴 것이며, 5장에서 다루는 OPENMARU AAS(AI Agent Studio) 위 에이전트 설계의 출발점입니다. AAS는 OpsKnow Agent가 실행되는 에이전트 실행 환경이므로, 판정 코드와 모델의 역할 분리는 AAS의 도구 호출 구조 안에서 유지됩니다.

보안 취약점 판정의 다섯 가지 상태 구분

판정 결과를 「정상」과 「이상」 두 가지로만 나누면 조치가 필요한 이유가 서로 다른 경우가 한 칸에 섞입니다. 예를 들어 한국인터넷진흥원(KISA)이 운영하는 KNVD(취약점 정보포털)에서 CVE 권고가 나왔을 때[S10], 해당 패키지가 없는 서버와 수정본이 아직 저장소에 없는 서버는 둘 다 「조치 안 됨」으로 보이지만 해야 할 일이 다릅니다. OpsKnow Agent는 OS 패키지 기준의 보안 취약점 판정 결과를 다음 다섯 가지 상태로 나누어 기록합니다.

판정 상태	뜻	보고서와 다음 행동
이미 조치됨	권고가 요구하는 수정본이 설치돼 있음	조치 완료로 집계하고 근거 버전을 기록
설치돼 있지 않음	권고 대상 패키지가 서버에 없음	해당 없음으로 집계
저장소에 수정본 없음	대상 패키지는 있으나 내려받을 수정본이 없음	조치 불가 사유와 함께 보고하고 저장소 확인을 요청
보안 정보 없음	판정에 필요한 권고 정보가 없음	판정 보류로 따로 집계

판정 상태	뜻	보고서와 다음 행동
지원 종료 OS	운영체제 공급사 지원이 끝남	패치 대상에서 빼고 교체 검토 대상으로 보고

판정 기록에 함께 남는 항목

증적으로 쓰려면 각 판정에 상태와 함께 대상 서버, 점검 시각, 확인한 패키지 버전, 기준이 된 권고 번호, 적용한 판정 규칙이 붙어 있어야 합니다. 그래야 담당자가 보고서의 한 줄에서 원래 기록으로 거슬러 올라가 왜 그 상태가 나왔는지를 확인할 수 있으므로, 차년도 요구사항에도 이 항목을 그대로 적으면 됩니다. 미들웨어와 DBMS 지표도 같은 원칙을 따릅니다. 예를 들어 OPENMARU APM이 수집하는 WAS 스레드 사용률, DB 커넥션 풀 대기, SQL 수행 시간을 판정에 쓸 때는 원래 값과 수집 시각을 판정 결과 옆에 함께 남겨야 판정 근거를 한 기록 안에서 추적할 수 있습니다. 판정 근거가 기록에 붙어 있으면 감사 담당자가 같은 판정을 직접 재현할 수 있습니다.

3.2.3 확실하지 않으면 정상이나 최신으로 판정하지 않는 원칙

근거가 부족한 판정을 보류로 기록하는 원칙

결정적 코드 판정은 근거가 부족할 때 판정을 미루도록 만들어야 합니다. OpsKnow Agent는 판정에 필요한 정보가 모자라면 해당 서버를 「최신」이나 「정상」으로 표시하지 않고, 판정 보류와 그 사유를 기록합니다. 예를 들어 권고 정보를 가져오지 못했거나 서버 접속이 실패한 경우에는 마지막으로 확인한 결과를 재사용하지 않고 이번 점검에서 확인하지 못했다는 사실을 남깁니다. 이 원칙이 있어야 앞에서 본 「수집 실패가 정상으로 기록되는 결함」이 판정 단계에서 차단됩니다.

판정 보류를 보고서에 드러내는 방식

판정 보류 건은 보고서에서 정상 건과 따로 집계하고, 사유별로 묶어 보여 줍니다. 관리자는 정상 비율 하나 대신 정상, 이상, 판정 보류의 건수를 각각 보게 되고, 판정 보류가 늘어나면 수집 경로나 저장소 연결에 문제가 생겼다는 신호로 읽을 수 있습니다. 감사 대응에서도 판정 보류를 숨기지 않은 기록이 더 믿을 만한 증적이 됩니다. 판정 보류를 그대로 적은 기록은, 모든 칸이 정상인 기록보다 재현 가능한 근거를 더 많이 담고 있기 때문입니다.

판정 보류에서 조치까지 이어지는 다음 단계

판정 보류는 다음 확인 작업의 시작점입니다. 운영 규칙으로는 보안 정보가 없어서 보류된 서버를 권고 정보가 들어온 뒤 다시 판정하고, 접속 실패로 보류된 서버는 접속 경로 점검을 담당자 작업으로 넘기도록 정해 두면 됩니다. 재판정과 작업 배정은 서버 상태를 바꾸지 않는 일이므로 승인 없이 진행할 수 있고, 서버를 바꾸는 조치는 3.4에서 설명하는 승인 절차를 거칩니다. 이 분리 덕분에 판정 단계의 자동화는 넓게 쓰면서도 서버 변경에 대한 통제는 사람에게 남길 수 있습니다.

3.3 연관 단계 — 미들웨어 응답 지연과 DBMS 커넥션 풀, OS 자원을 한 사건으로 묶는 원인 설명

연관 단계는 서로 다른 영역에서 나온 점검 결과와 경보를 하나의 장애 원인 후보로 묶는 단계입니다. 공공 정보 시스템의 서비스 하나는 OS, 미들웨어, DBMS, 네트워크 장비를 함께 거치므로, 한 영역의 이상이 다른 영역의 지표로 먼저 드러나는 경우가 흔합니다. 2023년 11월 행정전산망 장애도 처음에는 다른 장비가 원인으로 거론

됐다가, 정부 발표에서 라우터의 케이블 연결 모듈에 있는 포트 일부 이상으로 원인이 정리됐습니다[S04] [S05]. 즉 원인 판단에는 영역별 지표를 따로 보는 것보다 같은 요청이 지나간 경로를 따라 지표를 연결하는 근거가 필요합니다.

3.3.1 단일 지표 경보와 계층 연관 판정이 같은 장애를 다르게 보여 주는 예

민원 조회 서비스 응답 지연의 가상 예시

다음은 설명을 위해 만든 가상 예시입니다. 어느 기관의 민원 조회 서비스에서 응답이 느려졌고, 같은 시간대에 WAS 스레드 사용률 경보, DB 커넥션 풀 대기 경보, 서버 CPU 사용률 경보가 함께 발생했다고 가정합니다. 단일 지표 경보 방식에서는 세 경보가 각각 WAS 담당자, DBA, 시스템 담당자에게 따로 전달되고, 담당자마다 자기 영역의 지표를 확인한 뒤 「내 쪽은 결과일 뿐 원인이 아니다」라는 답을 내기 쉽습니다. 반면 계층 연관 판정에서는 같은 시간대의 느린 트랜잭션을 따라가 특정 SQL 하나의 수행 시간이 길어졌고, 그 때문에 커넥션 반환이 늦어져 풀이 고갈됐으며, 커넥션을 기다리는 WAS 스레드가 쌓였다는 순서로 원인 후보를 제시합니다.

비교 항목	단일 지표 경보	계층 연관 판정
담당자가 받는 정보	영역별 경보 세 건	원인 후보 한 건과 근거 지표 묶음
첫 확인 대상	영역마다 각자 정함	수행 시간이 늘어난 SQL
근거의 형태	경보 시각과 임계치	트랜잭션 경로를 따라 이어진 지표
Incident 기록에 남는 원인	영역별 소견이 따로 남음	원인 후보와 관련 트랜잭션 ID

원인 후보 하나가 조치 시간을 줄이는 이유

두 방식의 차이는 담당자들이 원인을 좁혀 가는 순서에서 생깁니다. 단일 지표 경보에서는 세 담당자가 동시에 조사를 시작하지만 서로의 결과를 맞춰 보기 전까지 원인을 정할 수 없고, 맞춰 보는 과정은 회의나 메신저에 의존합니다. 계층 연관 판정은 이 맞춰 보는 과정을 판정 단계로 끌어와, 담당자에게 처음부터 하나의 원인 후보와 그 근거를 줍니다. 조사 순서가 정해진 상태로 조치가 시작되고, 원인 후보가 틀렸을 때도 어느 근거가 틀렸는지를 기록에서 바로 확인할 수 있습니다. 다만 연관 판정이 내는 결과는 어디까지나 후보이며, 최종 원인은 조치 결과와 재점검으로 확인합니다.

3.3.2 APM 트랜잭션 ID로 영역별 지표를 연결하는 구조

APM 트랜잭션 추적이 원인 연결의 근거가 되는 방식

OPENMARU APM은 사용자 요청 하나가 WAS에 들어와 SQL을 호출하고 응답을 돌려줄 때까지의 경로를 트랜잭션 단위로 추적하고, 경로마다 트랜잭션 ID를 붙입니다. WAS 스레드 상태와 DB 커넥션 사용, SQL 수행 시간이 같은 ID 아래 묶이고, OS 자원 지표는 서버 이름과 수집 시각으로 그 묶음에 연결됩니다. OpsKnow Agent는 이 연결 정보를 판정의 입력으로 받아 원인 분석을 진행하고, 느린 트랜잭션의 ID를 근거 지표와 함께 결과에 남깁니다. 따라서 원인 후보가 어느 트랜잭션의 어떤 지표에서 나왔는지를 나중에 같은 ID로 다시 확인할 수 있습니다.

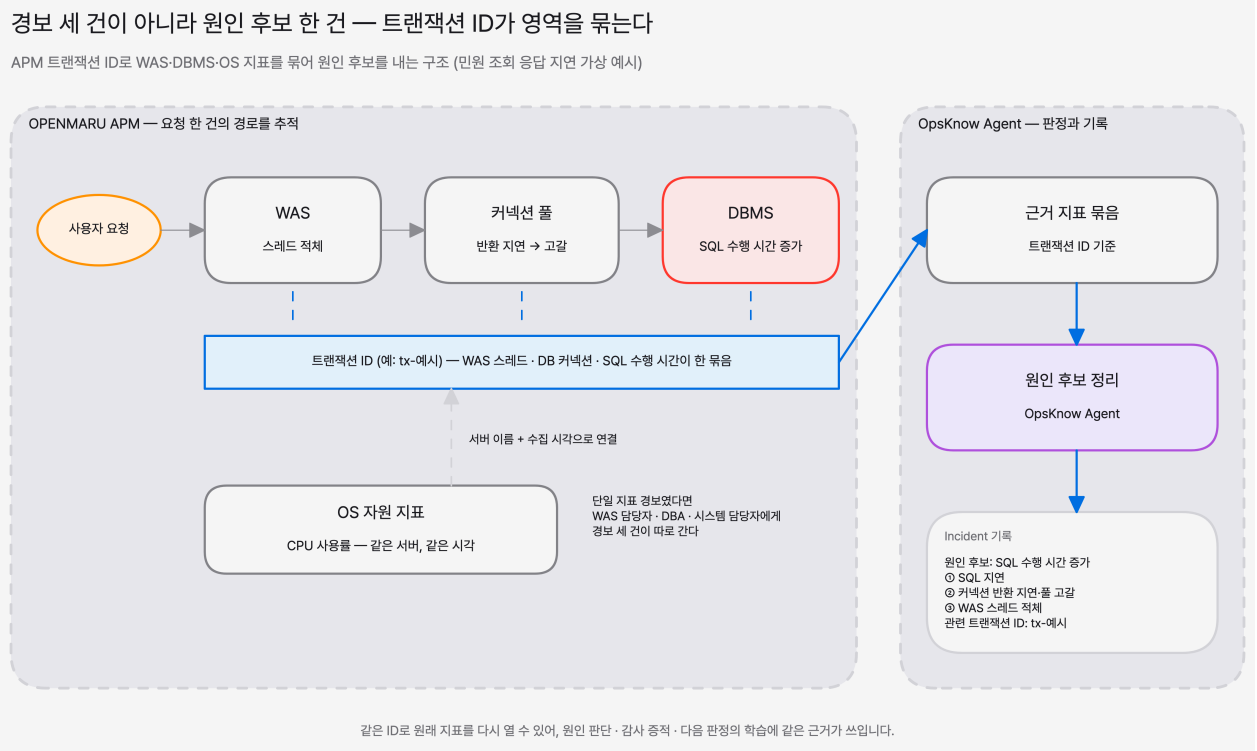


그림. APM 트랜잭션 ID로 WAS·DBMS·OS 지표를 묶어 원인 후보를 내는 구조

연결 정보가 증적과 학습으로 이어지는 경로

트랜잭션 ID는 원인 후보를 고를 때만 쓰고 버리는 값이 아닙니다. 조치가 끝난 뒤 Incident 기록에는 관련 트랜잭션 ID가 함께 저장되므로, 나중에 같은 SQL이나 같은 서비스 경로에서 지연이 다시 생기면 과거 기록을 ID와 경로로 찾아낼 수 있습니다. 감사 때 「이 원인 판단의 근거가 무엇인가」라는 질문에는 해당 트랜잭션의 원래 지표를 열어 답할 수 있습니다. 즉 연관 단계의 연결 정보는 원인 판단, 증적, 학습 세 단계에 같은 근거를 공급합니다.

3.4 조치에서 학습까지 — 승인 기록과 운영 지식 축적이 남기는 증적

조치·보고·학습 단계는 판정 결과가 실제 변경과 보고서, 다음 판정의 근거로 이어지는 마지막 단계입니다. 단순 자동화에서는 이 단계의 기록이 결재 문서, 작업 메모, 보고서 파일로 흩어지기 때문에, 하나의 점검 결과에서 시작해 무엇이 바뀌었는지까지 따라가기 어렵습니다. 지능화 단계에서는 점검, 판정, 승인, 조치, 재점검, 저장이 하나의 기록으로 이어지고, 그 기록이 OpsKnow에 쌓여 다음 판정에 쓰입니다. 고시 제2026-23호가 예방점검과 함께 장애 관리를 안정성 기준에 넣은 만큼[S02], 이 연결된 기록이 곧 장애 예방과 장애 관리의 공통 증적입니다.

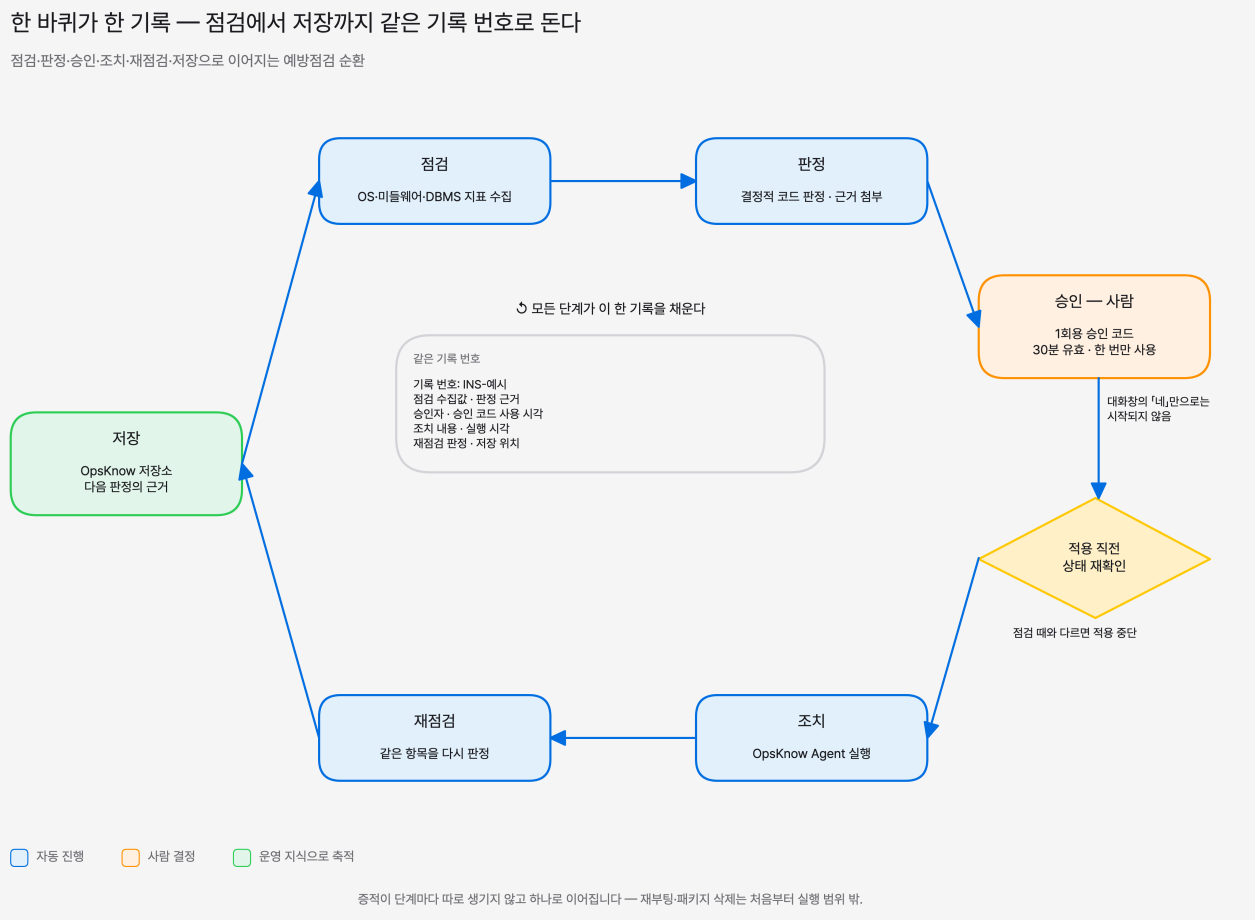


그림. 점검·판정·승인·조치·재점검·저장으로 이어지는 예방점검 순환

3.4.1 사람 승인 뒤 실행되고 재점검까지 한 기록으로 남는 조치

자동 실행과 사람 결재의 경계

지능화 단계에서도 모든 조치를 자동으로 실행하지는 않습니다. OpsKnow 운영 구조에서 알림 발송, OpsKnow 저장, 임계치 보고서 생성은 서버 상태를 바꾸지 않으므로 자동으로 실행합니다. 반면 임계치 조정, WAS 재시작, 인스턴스 증설과 감축, 세션 강제 종료, 외부 시스템 재시작, DB 풀 크기 변경은 서비스에 직접 영향을 주므로 사람의 결재를 거칩니다. 이 경계는 기관의 운영 정책에 따라 정하되, 기준은 「서버나 서비스의 상태를 바꾸는가」 하나로 두는 편이 감사 설명에 유리합니다.

1회용 승인 코드로 실행되는 OpsKnow Agent의 조치

OpsKnow Agent는 서버를 바꾸는 조치를 1회용 승인 코드가 있을 때만 실행합니다. 승인 코드는 30분 동안만 유효하고 한 번만 쓸 수 있으며, 대화창에서 「네」라고만 답해서는 적용이 시작되지 않습니다. 적용 직전에는 서버 상태를 다시 확인하고, 점검 때와 상태가 다르면 적용을 멈춥니다. 점검과 승인 사이에 다른 작업자가 같은 서버를 바꾼 경우에도 승인 당시와 다른 상태에 조치가 들어가는 일을 막을 수 있습니다. 재부팅, 설정 변경 실행, 압축 설치본 형태의 제품 패치, 패키지 삭제와 다운그레이드는 처음부터 OpsKnow Agent의 실행 범위에 넣지 않았습니다.

조치 기록에 남는 항목

감사에 쓸 조치 기록에는 원래 점검 결과, 판정 상태, 승인자와 승인 시각, 승인 코드 사용 여부, 적용 직전 상태 확인 결과, 적용 내용, 적용 뒤 재점검 결과가 순서대로 이어져 있어야 합니다. 이 순서가 한 기록으로 묶여 있으면 감사에서 특정 보안 취약점 조치를 확인할 때 담당자는 별도 결재 문서를 찾지 않고 이 기록 하나를 제시하면 됩니다. 재점검 결과가 기대와 다르면 같은 기록에 실패와 그 사유가 남고, 다음 조치는 새 승인으로 이어집니다. 승인과 실행, 재점검이 한 기록에 묶여 있어야 「조치했다」는 보고가 확인 가능한 사실이 됩니다.

3.4.2 원자료에서 다시 생성되는 보고서와 손으로 고친 보고서의 차이

양식 보고서를 손으로 고칠 때 생기는 불일치

단순 자동화의 보고서는 스크립트가 양식의 칸을 채운 뒤 담당자가 문서를 열어 설명을 덧붙이거나 숫자를 고치는 방식으로 완성됩니다. 이 과정에서 보고서의 숫자와 원자료가 어긋나기 시작합니다. 예를 들어 판정 보류 건을 보고서에서 정상으로 바꿔 적거나, 재점검 결과를 반영하려고 칸을 손으로 수정하면, 원자료를 다시 열었을 때 같은 숫자가 나오지 않습니다. 이런 불일치는 보고서 자체의 신뢰를 떨어뜨리고, 감사에서 원자료 제출을 요구받으면 설명할 근거가 사라집니다.

원자료 재생성 방식이 보고서에 주는 성질

지능화 단계의 보고서는 저장된 판정 기록과 조치 기록에서 매번 다시 생성합니다. 숫자와 표는 원자료에서 계산하고, AI는 그 숫자를 설명하는 문장만 씁니다. 같은 기간의 보고서를 다시 만들면 언제나 같은 숫자가 나오고, 설명 문장이 달라지더라도 근거 숫자는 바뀌지 않습니다. OpsKnow는 일간, 주간, 월간, 분기, 연간 운영 보고를 쌓고 월간 보고를 분기·연간 보고의 기준 데이터로 쓰므로, 상위 보고서의 숫자를 하위 보고서까지 거슬러 올라가 확인할 수 있습니다.

비교 항목	손으로 고친 보고서	원자료에서 다시 생성한 보고서
숫자의 출처	스크립트 출력 위에 사람이 수정	저장된 판정·조치 기록에서 계산
설명 문장	담당자가 직접 작성	AI가 숫자를 근거로 작성하고 사람이 검토
같은 기간 재작성	수정 이력이 없으면 재현 불가	언제나 같은 숫자
감사 대응	보고서와 원자료를 따로 맞춰야 함	보고서의 숫자에서 원자료로 바로 이동

보고서 설명문을 맡는 모델과 데이터 위치

보고서 설명문을 쓰는 모델은 기관 안의 폐쇄망에서 GPU 1대로 실행되는 30B 수준 내부 sLLM(소형 언어 모델)입니다. 이 구성에서는 판정 기록과 조치 기록이 외부 AI 서비스로 나가지 않고, 이는 6장에서 다루는 소버린 AI Level 3 구성의 기본 조건입니다. 국가정보원이 2025년 9월 공개한 N2SF(국가 망 보안체계) 가이드라인 1.0이 업무 정보를 기밀·민감·공개로 나누고 등급별로 통제를 달리하므로[S08][S09], 운영 기록처럼 기관 내부 구성 정보가 담긴 자료는 내부 모델로 처리하는 구성이 등급 판단을 단순하게 만듭니다.

3.4.3 Incident 기록이 다음 판정의 근거로 돌아오는 학습 고리

OpsKnow Incident 기록의 7개 필드

OpsKnow는 장애 처리가 끝날 때마다 Incident 기록을 원인, 조치, 재발방지, 관련 트랜잭션 ID, 사이트 태그, 운영자 ID, 자동 근본 원인 분석(RCA, Root Cause Analysis) 시간의 7개 필드로 남깁니다. 이 가운데 관련 트랜잭션 ID와 사이트 태그는 다음 장애를 찾을 때 검색 기준이 되고, 원인·조치·재발방지는 다음 판정에서 참고할 내용이 됩니다. 운영자 ID와 자동 RCA 시간은 누가 처리했고 원인 분석에 얼마나 걸렸는지를 남겨, 정기 보고와 운영 개선의 측정 항목으로 쓰입니다.

다음 판정에서 과거 기록을 조회하는 방식

같은 서비스 경로에서 비슷한 지표 묶음이 다시 나타나면, OpsKnow에 쌓인 과거 Incident 기록을 검색해 비슷한 사례를 원인 후보와 함께 보여 줍니다. 예를 들어 3.3.1의 가상 예시에서 같은 SQL이 다시 느려지면, 지난번에 어떤 조치로 해결했고 재발방지 항목에 무엇을 적었는지가 판정 결과 옆에 붙습니다. 다만 과거 기록은 판정을 바꾸지 않고 참고 자료로만 붙습니다. 판정은 여전히 결정적 코드가 내리고, AI는 과거 기록과 새 판정의 관계를 설명하며, 조치 결정은 사람이 내립니다.

기록이 쌓일수록 판정 근거가 늘어나는 구조

학습 고리의 효과는 모델을 다시 학습시키는 데서 오지 않고, 판정 옆에 붙는 근거 기록이 늘어나는 데서 옵니다. Incident 기록, 예방점검 결과, 정기 운영 보고가 한 저장소에 쌓이므로, 담당자가 바뀌어도 이전 담당자의 판단 근거가 기록으로 남습니다. 공공 정보시스템 15,033개가 등급별로 관리 대상이 된 상황에서[S01], 담당자 기억에 의존하던 운영 지식을 기록으로 옮기는 일은 인력 교체와 위탁 운영 사업자 변경에도 운영 수준을 유지하는 방법입니다. 따라서 학습 단계는 3.1 비교표에서 단순 자동화와 지능화 사이의 차이가 가장 오래 누적되는 단계이며, 차년도 사업에서 가장 먼저 요구사항으로 적어야 할 항목 가운데 하나입니다.

4장: 해법의 구조 — APM 수집, OpsKnow Agent 판정, OpsKnow 축적으로 나눈 예방점검 역할 분담

OPENMARU OpsKnow는 예방점검 결과, 장애 기록, 정기 운영 보고를 한 저장소에 쌓아 운영 지식으로 남기는 제품이며, 이 장은 그 저장소를 중심으로 수집·판정·축적·가시화를 다섯 구성요소에 나눠 맡기는 구조를 설명합니다. 행정안전부가 2026년 8월 27일 공공 정보시스템 15,033개의 등급을 A1 73개, A2 157개, A3 671개, A4 14,132개로 정해 발표하면서, 등급에 맞는 예방점검을 실제로 했다는 증적을 시스템마다 남기는 일이 기관의 공통 과제가 됐습니다[S01]. 그런데 점검 지표를 모으는 도구, 결과를 판정하는 사람, 보고서를 쓰는 사람, 기록을 보관하는 폴더가 제각각이면 같은 점검을 하고도 판정 근거와 이력이 흩어집니다. 이 장은 OS·미들웨어·DBMS 예방점검을 기준으로 각 구성요소가 맡는 일을 나누고, 그 분담을 SW 사업 내역서의 품목과 맞춰 볼 수 있도록 정리했습니다.

다섯 구성요소의 역할은 다음과 같이 나뉩니다. OPENMARU APM은 OS·미들웨어(WEB/WAS)·DBMS 지표를 상시 수집하고, OpsKnow Agent는 그 지표를 임계치에 비추어 판정한 뒤 조치 제안과 보고서 초안을 만듭니다. 이때 과거 기록을 찾아 근거로 붙이는 근거 검색(RAG)과 설명 문장을 만드는 30B 수준 내부 sLLM의 서빙은 OPENMARU CogentAI가 기관 서버 안에서 맡으므로, 판정 근거와 설명이 기관 밖으로 나가지 않습니다. OpsKnow는 판정 결과·Incident 기록·정기 보고를 쌓아 다음 판정의 근거로 돌려주며, Wallboard·전국 Dashboard는 운영실과 상급 기관이 같은 기록을 화면으로 보게 합니다. 지표 수집, 판정, 기록 축적을 서로 다른 구성요소에 맡겨 어느 한 곳이 바뀌어도 나머지가 그대로 동작하게 만든 구조입니다.

구성요소	맡는 일	산출물	사람이 개입하는 일
OPENMARU APM	OS·미들웨어·DBMS 지표 상시 수집	시계열 지표, 트랜잭션 추적 기록	수집 대상 등록
OpsKnow Agent	임계치 판정, 조치 제안, 보고서 초안	판정 결과, 조치안, 보고 초안	조치 승인, 보고서 검토
OpsKnow	판정·Incident·정기 보고 축적과 검색	점검 이력, 장애 기록, 운영 보고	재발방지 내용 입력
Wallboard	운영실 상시 화면	시스템 상태와 점검 이행 현황	이상 표시 확인
전국 Dashboard	다기관 통합 화면	등급별 시스템 현황, 미이행 점검	상급 기관 보고 결재

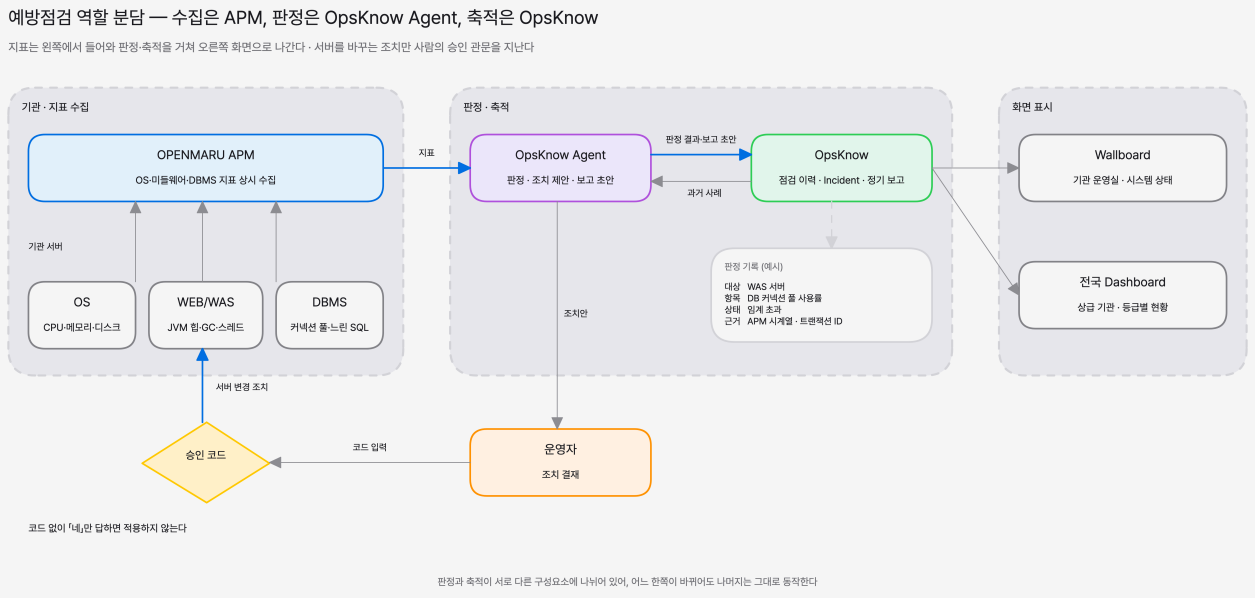


그림. APM 수집에서 OpsKnow Agent 판정, OpsKnow 축적, Wallboard·전국 Dashboard 표시로 이어지는 예방점검 흐름

4.1 OPENMARU APM이 상시 수집하는 OS와 미들웨어(WEB/WAS), DBMS 지표의 범위

OPENMARU APM은 WAS 안에서 실행되는 트랜잭션을 요청 단위로 추적하고, 그 트랜잭션이 거쳐 간 OS 자원과 DB 호출까지 함께 기록하는 국산 APM 제품입니다. 예방점검 관점에서 APM이 중요한 이유는 점검 항목 상당수가 이미 APM이 상시 수집하는 지표와 겹치기 때문입니다. 2024년 마련된 범정부 예방점검체계는 일상점검·특별점검·구조진단 3개 분야, 8개 점검항목으로 나뉘며, 일상점검에는 CPU·메모리·디스크 상태와 메인 화면 접속 여부, 접속 시간, 인증서 유효기간, 도메인 만료일 같은 항목이 들어갑니다[S07]. 따라서 이 항목들을 점검 당일에 별도 스크립트로 다시 재는 대신, APM이 쌓아 둔 시계열 지표에서 해당 시점 값을 꺼내 쓰면 점검 결과와 증적이 같은 원천에서 나옵니다.

4.1.1 영역별 일상·특별 점검 항목과 APM 수집 지표의 대응표

OS 영역 점검과 APM 서버 지표

OS 영역의 일상점검은 CPU 사용률, 메모리 사용률, 디스크 사용률처럼 서버 자원의 여유를 보는 항목이 중심이며, OPENMARU APM 에이전트는 이 값을 짧은 주기로 수집해 시계열로 보관합니다[S07]. 점검표에 적는 값은 점검 시각 한 번의 측정값인데, APM 기록에는 그 전후 하루치 추이가 함께 남기 때문에 순간 수치가 정상이어도 증가 추세가 이어지는 디스크를 미리 가려낼 수 있습니다. 예를 들어 로그 파일이 매일 일정량씩 쌓여 디스크 사용률이 꾸준히 오르는 서버는 점검 당일 수치만 보면 정상으로 판정되지만, 추이를 같이 보면 며칠 뒤 임계치를 넘는다는 판단이 가능합니다. OS 영역에서 APM이 더하는 정보는 측정값 자체보다 그 값이 어디로 움직이는지에 관한 기록입니다.

미들웨어 영역 점검과 WAS 내부 지표

미들웨어(WEB/WAS) 영역은 OPENMARU APM이 가장 깊게 들여다보는 영역으로, JVM 힙 메모리 사용량, GC(Garbage Collection, 쓰지 않는 메모리를 회수하는 작업) 빈도와 소요 시간, 스레드 풀 사용률, 활성 서비스 수, 세션 수, 트랜잭션 응답 시간과 처리량을 수집합니다. 일상점검의 메인 화면 접속 여부와 접속 시간 항목

은 APM이 실제 사용자 요청의 응답 시간과 오류율로 상시 기록하므로, 점검자가 브라우저로 한 번 접속해 보는 확인보다 넓은 근거를 제공합니다[S07]. 반면 특별 성능점검처럼 부하가 몰리는 시간대의 응답 지연을 따져야 하는 항목은 느린 트랜잭션 목록과 그 트랜잭션이 오래 머문 메서드·SQL 정보가 있어야 원인까지 적을 수 있습니다. 미들웨어 영역은 APM 기록만으로 점검 결과와 원인 설명을 함께 채울 수 있는 비중이 가장 큽니다.

DBMS 영역 점검과 DB 호출 지표

DBMS 영역에서 OPENMARU APM은 WAS가 DB를 호출한 기록을 기준으로 DB 커넥션 풀 사용률, SQL 수행 시간과 호출 빈도, 오래 걸린 SQL 목록을 남깁니다. 이 기록은 특정 화면이 느려졌을 때 원인이 애플리케이션 로직인지, 커넥션 부족인지, 특정 SQL의 수행 계획인지를 가르는 근거가 됩니다. 다만 DBMS 내부의 세션·대기 상태, 백업 수행 여부, 데이터 파일의 물리 저장 상태처럼 WAS의 DB 호출 기록에 나타나지 않는 항목은 이 장에서 APM 수집 범위로 보지 않으며, 이 차이는 4.1.2에서 따로 나눕니다. 다음 표는 세 영역의 대표 점검 항목과 APM 수집 지표를 대응시킨 것이며, 영역별 항목 수는 공개 원문 대조 전이므로 적지 않았습니다.

점검 영역	대표 점검 항목	OPENMARU APM 수집 지표	증적으로 남는 형태
OS	CPU·메모리·디스크 상태	서버 자원 사용률 시계열	점검 시각 값과 전후 추이
WEB/WAS	메인 화면 접속 여부와 접속 시간	실제 요청의 응답 시간·오류율	시간대별 응답 분포
WEB/WAS	성능 저하 원인	JVM 힙, GC, 스레드, 느린 트랜잭션	트랜잭션 ID별 추적 기록
DBMS	커넥션 상태	WAS 쪽 커넥션 풀 사용률·대기	시계열과 임계 초과 이력
DBMS	느린 SQL	SQL 수행 시간·호출 빈도	SQL별 상위 목록

4.1.2 추가 에이전트 설치 없이 확보되는 미들웨어·DBMS 증적의 범위

APM 기록으로 채워지는 증적 범위

이미 OPENMARU APM을 운영하는 기관은 예방점검을 위해 서버마다 점검용 에이전트를 새로 깔 필요 없이, 수집 중인 지표를 판정과 보고에 다시 쓰는 방식으로 미들웨어·DBMS 증적 상당 부분을 채울 수 있습니다. 예를 들어 WAS 응답 시간, GC, 커넥션 풀, 느린 SQL 기록은 점검 주기와 무관하게 상시 쌓이므로, 일상점검 보고서에는 점검일 값과 함께 직전 기간의 추이 그래프를 근거로 붙일 수 있습니다. 트랜잭션 ID 단위로 남는 추적 기록은 점검 중 발견한 이상을 장애 기록과 연결할 때 같은 식별자로 묶을 수 있어, 점검 결과와 Incident 기록 사이에 근거가 끊기지 않습니다. APM이 있는 환경에서 예방점검 체계를 갖추는 비용은 수집 장치를 새로 사는 비용보다 판정 규칙과 보고 양식을 정하는 비용에 가깝습니다.

사람이 직접 확인해야 하는 점검 범위

반면 네트워크 장비의 물리 상태, 백업 매체의 실물 보관, 이중화 설비의 전환 동작처럼 애플리케이션 요청 경로 밖에 있는 항목은 APM 기록으로 증명할 수 없습니다. 2023년 11월 행정전산망 장애는 정부 발표 기준으로 라우터의 케이블 연결 모듈에 있는 포트 일부 이상이 원인이었고, 해킹 흔적은 없었으며, 11월 18일 오전 9시 서비스를 재개한 뒤 11월 19일 포트를 교체해 정상화됐습니다[S04][S05]. 이런 장비 결함은 WAS 지표에 응답 지

연으로 나타날 수는 있어도 원인 자체는 네트워크 장비 점검과 사람의 현장 확인으로만 밝혀집니다. 행정예고된 「행정기관 및 공공기관 정보시스템 안정성 고시」가 요구하는 것으로 알려진 주기 백업, 원거리 소산 백업, 실전환 훈련은 수행 기록·결과 확인서가 증적이 되는 항목입니다[S02].

구분	해당 항목	증적 원천	OpsKnow 처리
APM 기록으로 채움	WAS 응답·GC·스레드, 커넥션 풀, 느린 SQL, 서버 자원	OPENMARU APM 시계열·추적 기록	자동 판정 후 저장
사람 확인 후 입력	네트워크 장비 상태, 백업 실물, 이중화 전환 훈련	현장 확인서, 훈련 결과 기록	확인 결과를 첨부로 저장
혼합	성능 저하 원인 규명, 재발 방지 조치	APM 기록과 담당자 판단	담당자가 재발방지 입력

이 구분이 의미를 가지는 이유는 OpsKnow가 두 종류의 증적을 같은 점검 회차 아래에 묶어 보관하기 때문입니다. 자동으로 채워진 항목과 사람이 확인한 항목이 하나의 점검 기록 안에서 출처 표시와 함께 남으므로, 감사 대응 때 어떤 판정이 지표에서 나왔고 어떤 판정이 사람의 확인에서 나왔는지 따로 설명할 필요가 줄어듭니다.

APM 기록이 예방점검 증적으로 쓸모가 큰 이유는 지표 값 하나가 아니라 그 값이 나온 요청 경로까지 함께 남기 때문입니다. 서버 자원이나 응답 시간 같은 수치는 지표로, 요청 하나가 WAS를 지나 커넥션 풀과 SQL 수행까지 이어진 경로는 트랜잭션 추적으로 남으므로, 「응답이 느렸다」는 판정 옆에 「어느 구간에서 느렸는지」를 근거로 붙일 수 있습니다. 판정 결과에 이 경로가 함께 붙어 있으면 감사인이나 후임 담당자가 같은 판정을 다시 따라가 볼 수 있고, 같은 증상이 다시 나타났을 때도 과거 기록과 경로 단위로 비교할 수 있습니다.

4.2 점검 결과를 판정하는 OpsKnow Agent와 9단계로 자동 적재하는 OpsKnow

OpsKnow Agent는 AAS 위에서 동작하는 운영 에이전트이며, OPENMARU APM이 모은 지표를 판정하고 조치를 제안하고 보고서 초안을 만든 뒤 그 결과를 OpsKnow에 넘깁니다. AAS 위 에이전트 구조가 공공 운영에 맞는 이유는 5장에서 설계 원칙으로 다룹니다. 점검 결과를 판정하는 일이 중요해진 배경은 일상점검 항목이 장비마다 매일 반복된다는 데 있습니다. 2024년 체계의 일상점검이 수십에서 수백 대 서버에 매일 적용되면, 판정 근거를 남기지 않은 「이상 없음」 표시만으로는 행정예고된 안정성 기준 46개 항목에 대응하기 어렵습니다[S07][S02]. OpsKnow Agent는 판정에 쓴 지표와 임계치 비교 결과를 남기고, OpsKnow는 그 결과를 정해진 순서에 따라 자동으로 저장합니다.

4.2.1 OpsKnow Agent가 맡는 판정과 조치 제안, 보고서 초안 작성

OpsKnow Agent 판정은 코드가 수행

OpsKnow Agent의 판정은 결정적 코드가 수행하며, 언어 모델은 사용자 요청을 해석해 넘기고 판정 결과를 사람이 읽을 문장으로 설명하는 일만 맡습니다. 같은 지표와 같은 임계치가 주어지면 판정 결과는 언제나 같고, 어떤 지표가 어떤 임계치를 넘었는지가 판정 기록에 그대로 남습니다. 예를 들어 패키지 보안 점검에서는 상태를 「이미 조치됨」 「설치돼 있지 않음」 「저장소에 수정본 없음」 「보안 정보 없음」 「지원 종료 OS」로 나누고, 근거가 확실하지 않으면 「최신」이라고 판정하지 않습니다. 판정을 코드에 맡기면 언어 모델이 그럴듯한 문장을 만들어 판정을 대신하는 일을 막을 수 있으므로, 판정 기록이 그대로 감사 증적이 됩니다.

조치 제안과 승인 코드의 관계

OpsKnow Agent는 판정에서 이상이 나오면 조치안을 제안하지만, 서버 상태를 바꾸는 조치는 1회용 승인 코드가 있어야 실행합니다. 이 승인 코드 절차는 OS 패치·보안 조치와 함께 임계치 조정, WAS 재시작, 인스턴스 증설·감축, 세션 강제 종료, 외부 시스템 재시작, DB 풀 크기 변경의 결재 6종에 똑같이 적용됩니다. 승인 코드는 30분 동안 한 번만 쓸 수 있고, 적용 직전에 서버 상태를 다시 확인해 점검 때와 달라졌으면 실행을 멈추며, 코드 없이 「네」라고만 답해도 적용하지 않습니다. 재부팅, 설정 변경 실행, 설치본(zip·tar) 형태 제품의 패치, 패키지 삭제·다운그레이드는 OpsKnow Agent가 직접 하지 않는 일로 정해 두었습니다. 미들웨어·DBMS 취약점도 OpsKnow Agent가 점검해 보고하며, 그 조치는 사람의 승인을 거친 뒤에 적용됩니다. 다만 KISA 보호나라·KrCERT/CC가 공지하는 보안 권고나 KNVD에 올라오는 CVE 정보처럼 외부 권고가 나왔을 때 대상 서버를 고르고 조치안을 만드는 일은 OpsKnow Agent가 맡으며, 이 흐름은 7장 보안 취약점 점검에서 자세히 다룹니다[S10].

보고서 초안과 사람의 검토

보고서 초안은 판정 결과, 근거 지표, 과거 유사 사례를 묶어 OpsKnow Agent가 작성하고, 담당자는 내용을 검토해 승인합니다. 초안에 들어가는 수치는 모두 OPENMARU APM 기록과 OpsKnow에 쌓인 판정 기록에서 가져오고, 언어 모델은 문장 연결·요약만 맡기 때문에 보고서의 숫자와 원천 기록이 어긋나지 않습니다. 보고서는 요청에 따라 HWP 파일로도 만들어지고, 수치와 문장마다 근거가 된 원 기록으로 가는 출처 링크가 붙습니다(9장). 따라서 담당자는 보고서를 처음부터 쓰는 대신 판정 결과가 맞는지, 재발방지 내용이 충분한지를 확인하는 일에 시간을 씁니다. 다음 표는 OpsKnow Agent가 맡는 일과 사람이 맡는 일을 나눈 것입니다.

기능	OpsKnow Agent가 하는 일	사람이 하는 일
판정	지표와 임계치 비교, 상태 분류, 근거 기록	임계치 정책 결정
조치 제안	대상 선별, 조치안 작성, 적용 직전 재확인	승인 코드 발급 여부 결정
보고서 초안	판정·근거·유사 사례 묶음, 문장 작성	내용 검토와 확정
직접 하지 않는 일	재부팅, 설정 변경 실행, 설치본 패치, 패키지 삭제	별도 작업 절차로 수행

4.2.2 Incident·예방점검 결과·정기 운영 보고를 쌓는 OpsKnow 3대 자산

Incident 케이스 자산의 일곱 필드

OpsKnow의 첫 번째 자산인 Incident 케이스는 실제로 처리한 장애를 원인, 조치, 재발방지, 관련 트랜잭션 ID, 사이트 태그, 운영자 ID, 자동 원인 분석 소요 시간의 일곱 필드로 남깁니다. 이 가운데 원인과 조치, 관련 트랜잭션 ID는 원인 분석이 끝나는 시점에 자동으로 채워지고, 담당자는 재발방지 필드만 추가로 입력합니다. 장애를 처리한 담당자가 순환보직으로 자리를 옮겨도 그 장애의 원인과 조치 기록은 OpsKnow에 그대로 남고, 다른 기관에서 비슷한 패턴이 나타나면 과거 기록이 판정 근거로 인용됩니다. 행정안전부가 2024년 10월 마련한 표준 운영절차가 장애관리를 핵심 절차로 두고 있다는 점을 고려하면, 장애 기록을 정해진 필드로 남기는 일은 절차 준수의 증적과도 직결됩니다[S06][S07].

예방점검 결과와 정기 운영 보고 자산

두 번째 자산인 예방점검 결과에는 점검 항목 목록, 사이트별 임계치, 일일·주간 점검 결과 이력이 들어가고, 정기 스케줄과 담당자의 수동 요청 양쪽으로 채워집니다. 세 번째 자산인 정기 운영 보고는 일·주·월·분기·연 다섯 단위의 보고서를 보관하며, 상위 단위 보고서는 하위 단위 보고서를 기준 데이터로 삼아 만들어집니다. 예를 들어 분기 보고서는 해당 분기의 월간 보고와 Incident 기록을 묶어 전분기 대비 악화된 지표와 권장 조치를 정리하고, 연간 보고서는 한 해 동안 쌓인 Incident를 원인별로 분류해 반복된 장애 패턴을 뽑아냅니다. 서비스 수준 협약(SLA) 이행 실적처럼 기간 단위로 정리해야 하는 자료도 정기 운영 보고 자산의 기록을 근거로 작성할 수 있습니다[S02].

자산	담는 내용	채워지는 계기	담당자 입력
Incident 케이스	원인·조치·재발방지 등 일곱 필드	원인 분석 완료 시 자동 등록	재발방지 내용
예방점검 결과	점검 항목, 임계치, 일일·주간 결과 이력	정기 스케줄, 수동 점검 요청	사람 확인 항목 결과
정기 운영 보고	일·주·월·분기·연 보고서	정기 스케줄, 자연어 보고 요청	보고서 검토

세 자산은 서로 참조하는 관계로 묶여 있습니다. 다시 말해 예방점검에서 나온 이상이 장애로 이어지면 Incident 케이스가 같은 트랜잭션 ID로 연결되고, 그 Incident는 다음 정기 보고의 악화 지표 설명에 인용되며, 보고서에서 나온 재발방지 조치는 다시 예방점검의 임계치 조정으로 돌아갑니다. 예방점검 증적을 한 저장소에 모으는 도입 기준은 예방점검 증적 저장소, AI 기반 운영 문서화 도입 기준[R1]에서 함께 읽을 수 있습니다.

예방점검 결과 자산이 실제로 어떤 모양으로 쌓이는지는 점검 날짜 하나를 펼쳐 보면 알 수 있습니다. 아래 그림은 한 날짜 아래에 OS, WEB, WAS, DB, 이벤트 다섯 영역의 점검 문서가 같은 형식으로 놓이고, 영역마다 먼저 열어 볼 담당과 요약이 향하는 판정이 정해져 있는 구성을 보여 줍니다.



그림. 한 점검 날짜 아래 OS·WEB·WAS·DB·이벤트 다섯 영역 문서가 같은 형식으로 쌓이는 OpsKnow 예방점검 결과 구성

다섯 문서가 같은 뼈대를 쓰고 결론만 다르기 때문에, 담당자는 요약 문단만 이어 읽어 그날 먼저 들여다볼 영역을 정할 수 있습니다. 이벤트 영역 문서는 알림이 몰린 호스트를 원인 후보로 지목하고 다른 네 영역 문서와 대조하게 해, 3.3절에서 설명한 영역 간 연관 판정이 저장소 안의 문서 구성으로도 이어지게 합니다.

4.2.3 수집에서 보고서 생성까지 이어지는 9단계 자동 적재 순서

정기 점검 1회가 거치는 9단계

OpsKnow의 자동 적재는 정기 점검 1회를 기준으로 아홉 단계를 거치며, 수집에서 보고서 생성까지 사람이 개입하는 단계는 마지막 알림 이후뿐입니다. 일상점검이 장비마다 매일 반복되는 항목이라는 점을 고려하면, 이 순서가 매일 같은 방식으로 실행돼야 점검 결과를 날짜별로 비교할 수 있습니다[S07]. 아홉 단계는 다음과 같습니다.

1. 정해진 시각에 AAS의 예약(주기) 실행이 정기 점검을 시작합니다.
2. OpsKnow Agent가 그날의 점검 항목 목록과 임계치를 불러옵니다.
3. 기관별 OPENMARU APM에 지표 조회를 동시에 요청합니다.
4. 각 APM이 전날의 동시접속자 추이, GC 패턴, 느린 트랜잭션 상위 목록을 돌려줍니다.
5. 판정 코드가 지표를 기관별 임계치에 비추어 초과 항목을 뽑습니다.
6. 요일과 시간대에 따른 정상 변동 폭으로 결과를 보정합니다.
7. OpsKnow에서 과거 Incident 기록을 검색해 비슷한 사례를 근거로 붙입니다.
8. 점검 결과는 예방점검 자산에, 요약은 일일 운영 보고 자산에 저장합니다.
9. 임계치를 넘은 기관이 있으면 담당자에게 알리고 승인 절차로 넘깁니다.

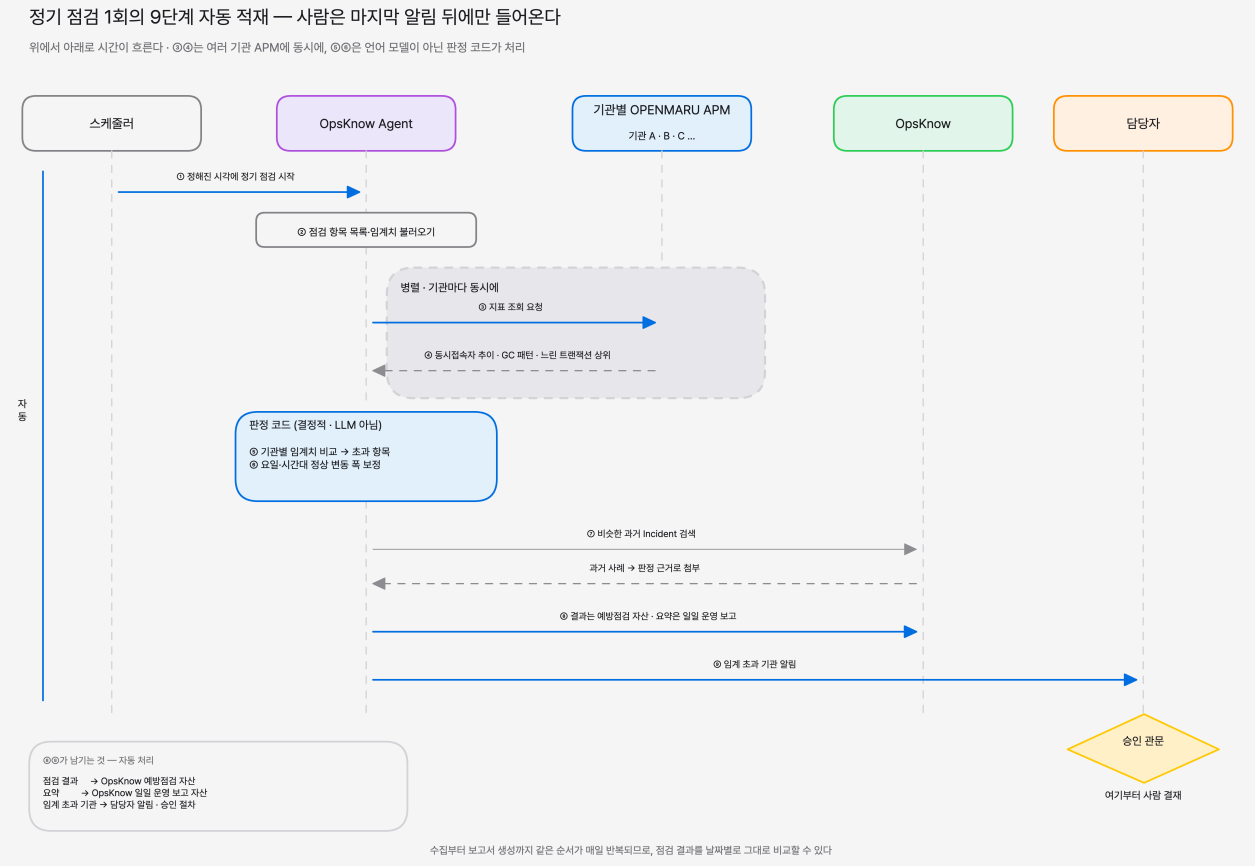


그림. OpsKnow 정기 점검 1회의 9단계 자동 적재 순서 — 수집, 판정, 보정, 사례 검색, 적재, 알림

자동 처리와 결재 처리의 경계

9단계 가운데 알림 발송, OpsKnow 저장, 임계치 보고서 생성은 자동으로 처리되고, 임계치 조정, WAS 재시작, 인스턴스 증설·감축, 세션 강제 종료, 외부 시스템 재시작, DB 커넥션 풀 크기 변경은 사람의 결재를 거칩니다. 결재는 1회용 승인 코드로 받고, 적용 직전 재확인과 적용 뒤 재점검까지 같은 절차로 이어집니다. 기준은 되돌릴 수 있는지 여부이며, 알림이나 기록처럼 잘못돼도 바로잡을 수 있는 일은 즉시 처리하고 서비스 상태를 바꾸는 일은 결재 뒤에만 실행합니다. 경계를 이렇게 나눠 두면 장애 회고 때 AI가 자동으로 한 일과 사람이 결재한 일이 기록에서 바로 구분되므로, 책임 추적과 감사 보고서 작성이 단순해집니다. 결재 경계의 세부 항목은 8장에서 장애·작업 관리와 함께 다시 다룹니다.

4.3 기관 상황판 Wallboard와 범정부 단위 전국 Dashboard의 Hub & Spoke 구성

Wallboard는 기관 운영실 벽면에 운영 현황을 상시 띄우는 화면이고, 전국 Dashboard는 OPENMARU Dashboard의 다기관 화면으로 여러 기관의 현황을 상급 기관이 한 화면에서 보게 합니다. 두 화면이 필요한 이유는 보는 사람과 판단의 단위가 다르기 때문입니다. 기관 운영실은 지금 어느 서버가 이상한지를 분 단위로 봐야 하고, 상급 기관은 등급별로 몇 개 시스템이 점검을 이행했는지를 일·월 단위로 봐야 합니다. 2026년 8월 발표에서 등급 기준이 기관 규모에서 국민이 받는 서비스의 중요도로 바뀌었다는 점을 고려하면, 상급 기관 화면은 기관별 목록보다 A1~A4 등급별 현황을 먼저 보여 주는 편이 판단에 맞습니다[S01].

4.3.1 운영실 벽 화면에 상시 표시되는 시스템 상태와 점검 이행 현황

Wallboard 화면의 표시 항목

Wallboard는 운영실 벽면의 큰 화면에 운영 현황을 상시 띄워, 운영실 인원이 자리에서 고개만 들어도 이상 여부를 알 수 있게 하는 화면입니다. Wallboard 화면은 예방점검 결과, 지연 트랜잭션, 이상 그룹, 승인 대기의 네 영역으로 짜여 있어, 응답 시간 같은 실시간 지표와 당일 예방점검 판정, 결재를 기다리는 조치가 한 화면에 함께 올라옵니다. 경보 화면이 그날의 점검이 끝났는지를 보여 주는 이행 현황판 역할까지 겸하는 구성입니다. 예를 들어 예방점검 결과 영역은 A1 등급 시스템을 앞쪽에 두고 당일 일상점검이 끝나지 않은 시스템을 색으로 구분하므로, 국민 영향이 큰 시스템의 점검 누락을 먼저 알아볼 수 있습니다[S01].

의사결정권자가 보는 결재 근거 화면

운영실 화면을 의사결정권자의 결재 근거로 쓰려면 화면의 점검 이행 현황이 OpsKnow의 판정 기록과 같은 원천에서 나와야 합니다. 그래야 담당자가 별도로 정리해 올리는 보고서와 화면의 수치가 어긋나지 않고, 일상점검의 메인 화면 접속 시간이나 인증서 유효기간 같은 항목도 점검표를 따로 열지 않고 판정 결과로 확인할 수 있습니다[S07]. 따라서 차년도 사업 요구사항에는 화면의 모양과 함께 각 영역이 어느 기록에서 값을 가져오는지를 적는 편이 검수에 유리합니다. 다음 표는 Wallboard 화면의 영역별 배치와 각 영역의 데이터 원천입니다.

화면 영역	배치	보여 주는 내용	데이터 원천
예방점검 결과	화면 왼쪽, 등급 순서	시스템별 당일 판정(정상·주의·이상)과 점검 완료·미완료	OpsKnow 예방점검 결과

화면 영역	배치	보여 주는 내용	데이터 원천
지연 트랜잭션	화면 가운데 위	응답이 느린 트랜잭션 목록과 오래 머문 구간(WAS·커넥션 풀·SQL)	OPENMARU APM 트랜잭션 추적
이상 그룹	화면 가운데 아래	같은 시각에 함께 이상을 보인 서버·WAS·DB 묶음	OpsKnow Agent 연관 판정
승인 대기	화면 오른쪽	결재를 기다리는 조치 건과 판정 근거	OpsKnow Agent 조치 제안

4.3.2 기관별 APM(Spoke)과 전국 Dashboard(Hub)를 잇는 통합 현황 구조

Hub & Spoke 구성의 역할 분담

Hub & Spoke는 중앙 한 곳(Hub)이 여러 지점(Spoke)의 정보를 모아 처리하는 구성으로, 여기서는 기관별 OPENMARU APM이 Spoke, 전국 Dashboard가 Hub입니다. Spoke는 지표를 수집해 표준 인터페이스로 내놓는 일만 맡고, 판정과 축적, 언어 모델 처리는 Hub 쪽 OpsKnow Agent와 OpsKnow가 맡습니다. 따라서 기관마다 언어 모델이나 검색 저장소를 따로 둘 필요가 없고, 판정 규칙과 임계치, 보고 양식이 Hub 한 곳에서 관리되므로 기관 사이에 판정 기준이 달라지지 않습니다. Hub 쪽에서는 OPENMARU CogentAI가 OpsKnow에 쌓인 과거 Incident 기록을 RAG(Retrieval-Augmented Generation, 저장된 기록을 검색해 답변 근거로 붙이는 방식)로 검색해 판정 결과에 비슷한 사례를 붙이고, 설명 문장을 만드는 30B 수준 내부 sLLM도 함께 서비스합니다. 근거 검색과 모델 서빙을 CogentAI 한 곳이 맡으므로, 판정 근거를 찾는 일과 그 근거로 문장을 쓰는 일이 같은 기관 서버 안에서 끝납니다. 이 언어 모델을 외부 연결 없이 30B 수준 내부 sLLM으로 운영하는 폐쇄망 구성은 6장 소버린 AI Level 3 구성에서 다룹니다.

등급별 현황과 상급 기관 보고

전국 Dashboard는 A1 73개, A2 157개, A3 671개, A4 14,132개로 나뉜 등급 체계를 기준축으로 삼아, 등급별로 점검 이행 시스템 수와 미이행 시스템, 임계 초과 건수를 보여 줍니다[S01]. A4 시스템이 전체의 대부분을 차지하므로, 수가 적은 A1·A2 시스템은 개별 목록으로, 수가 많은 A3·A4 시스템은 기관별 집계로 나누어 보여 주는 구성이 판단에 맞습니다. 행정예고안에는 중요 시스템의 장애를 행정안전부 디지털안전상황실에 즉시 보고하는 내용이 들어 있는 것으로 알려져 있습니다[S02]. 따라서 상급 기관 보고도 OpsKnow의 Incident 기록을 근거로 삼도록 요구사항을 정해 두면, 기관 담당자가 같은 내용을 다시 입력하는 일을 줄일 수 있습니다.

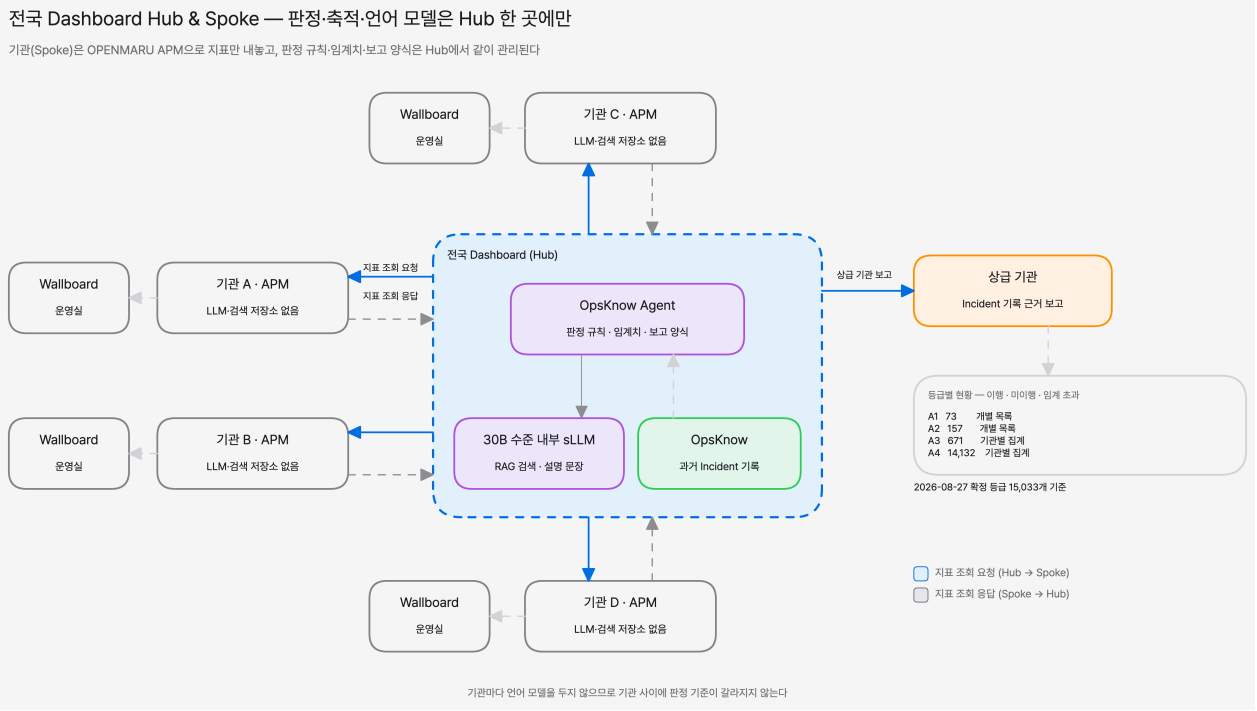


그림. 기관별 APM(Spoke)과 전국 Dashboard(Hub)의 Hub & Spoke 구성

조달 품목과의 대응

이 구성은 SW 사업 내역서의 품목과 그대로 대응합니다. 기관별 수집은 OPENMARU APM, 판정과 조치 제안은 OpsKnow Agent와 그 실행 환경인 AAS, 축적과 검색은 OpsKnow, 가시화는 Wallboard와 전국 Dashboard로 나뉘므로, 기관은 이미 가진 구성요소를 빼고 필요한 품목만 고를 수 있습니다. 예를 들어 OPENMARU APM을 운영 중인 기관은 OpsKnow와 OpsKnow Agent만 더해 예방점검 판정과 증적 축적을 시작하고, 상급 기관은 전국 Dashboard를 더해 소속 기관의 현황을 모을 수 있습니다. 결국 단순 자동화가 점검표를 빨리 채우는 데서 멈추는 반면, 이 구조는 판정 근거와 증적, 장애 기록을 한 저장소에 이어 붙여 지능화로 넘어가는 바탕을 만들며, 그 차이는 고시 제2026-23호가 요구하는 안정성 기준에 대응할 때 드러납니다 [S02].

5장: AAS 위 OpsKnow Agent가 공공 운영에 맞는 이유 — 판정은 코드, 결정은 사람, 설명은 AI

OpsKnow Agent는 OPENMARU OpsKnow에 들어 있는 AI 에이전트로, 정보시스템 예방점검의 판정과 조치 승인, 보고를 한 흐름으로 잇는 역할을 맡습니다. 이 에이전트는 오픈마루의 에이전트 운영 플랫폼인 AAS 위에서 동작하고, 에이전트별 권한과 실행 기록은 AAS가 관리합니다. 행정안전부는 2026년 8월 27일 공공 정보시스템 15,033개의 등급을 국민이 받는 서비스의 중요도에 따라 A1부터 A4까지 다시 정했고, 그중 가장 높은 A1 등급은 73개입니다[S01]. 등급마다 복구 목표를 달리 두는 방식도 함께 알려졌기 때문에, 높은 등급 시스템일수록 점검 결과를 누가 어떤 근거로 판정했는지 설명할 수 있어야 합니다[S03].

1장에서 살펴본 대로 고시 제2026-23호는 행정예고 단계에서 예방점검과 장애 관리를 포함한 안정성 기준 46개, 그리고 서비스 수준 협약(SLA) 체결 의무를 담은 것으로 알려졌습니다[S02]. 따라서 감사 대응의 중심은 점검을 몇 번 했는지보다 점검 결과와 조치가 기준에 맞았음을 보여 주는 증적으로 옮겨 갑니다. AI 에이전트를 운영에 쓰려면 「AI가 서버를 임의로 바꾸지 않는가」와 「AI의 판단이 실행할 때마다 달라지지 않는가」라는 두 질문에 먼저 답해야 합니다. 생성형 AI는 같은 질문에도 답의 표현과 결론이 조금씩 달라지는 특성이 있어서, 판정까지 맡기면 같은 서버 상태를 두고도 결과가 바뀔 수 있기 때문입니다.

OpsKnow Agent는 이 두 질문에 역할 분리로 답합니다. LLM은 관리자의 문장을 받아 넘기고 결과를 설명하는 일만 맡고, 점검 결과의 판정은 같은 입력에 늘 같은 결과를 내는 결정적 코드가 맡습니다. 서버를 바꾸는 조치는 사람이 승인 코드를 입력해야만 실행되며, 재부팅이나 설정 변경처럼 영향이 큰 작업은 처음부터 에이전트 범위에 넣지 않았습니다. 「판정은 코드, 결정은 사람, 설명은 AI」라는 원칙은 에이전트의 구조 안에 들어가 있으므로, 운영 지침을 따로 지키는지에 기대지 않고 제품 동작으로 확인할 수 있습니다.

의사결정권자가 이 장에서 판단할 것은 「LLM이 서버를 바꾸지 않는다」는 구조가 감사와 책임 소재 요건을 채우는지입니다. 판단 근거로 역할 3분할의 구성, 승인 코드와 적용 직전 재확인의 동작, 에이전트가 하지 않는 작업 목록, AAS가 남기는 권한·실행 기록을 차례로 제시합니다. 각 항목에서 점검표를 자동으로 채우는 단순 자동화와 무엇이 다른지를 함께 비교합니다. 다만 지능화는 사람의 결재를 없애는 방향이 아니며, 사람이 결정해야 할 일만 남기고 나머지 반복 작업을 코드와 AI가 나누어 맡는 구성입니다.

5.1 LLM은 문장 해석과 설명만, 판정은 결정적 코드가 맡는 설계

행정안전부가 2024년에 마련한 예방점검체계는 일상점검·특별점검·구조진단 3개 분야와 8개 예방점검 항목으로 이루어져 있습니다. 일상점검만 해도 CPU·메모리·디스크 상태, 메인 화면 접속 여부, 인증서 유효기간 같은 항목을 장비마다 매일 확인하므로[S07], 판정 결과가 쌓이는 속도는 사람이 하나씩 검토할 수 있는 양을 넘습니다. OpsKnow Agent는 반복되는 판정을 코드에 맡기고, LLM에는 사람이 읽고 쓰는 문장을 다루는 일만 맡깁니다. 여기서 LLM의 문장 해석은 관리자의 요청을 받아 결정적 코드에 넘기는 단계까지이며, 어떤 점검 명령을 실행할지와 결과가 기준에 맞는지는 코드가 정해진 규칙으로 정합니다.

판정의 입력이 되는 지표는 수집 계층에서 들어옵니다. 미들웨어·DBMS 지표는 OPENMARU APM이 수집하며, OPENMARU APM을 함께 언급하는 이유는 판정 코드가 다루는 미들웨어·DBMS 수치의 출처가 이 제품이기 때문입니다. 반면 OpsKnow Agent는 그 지표를 받아 판정하고, 판정 결과를 보고와 조치 승인으로 잇는 일을 맡습니다. 세 주체가 맡는 일과 맡지 않는 일을 나누면 다음 표와 같습니다.

주체	맡는 일	맡지 않는 일	남는 결과
LLM(30B 수준 내부 sLLM)	관리자 문장 전달, 판정 결과 설명·요약	점검 명령 선택, 판정, 서버 변경	설명 문장
결정적 코드	요청 해석, 점검 명령 선택, 기준 대비 판정	서버 변경 결정	판정값과 근거 수치
사람(운영 책임자)	조치 승인, 승인 코드 입력	반복 판정	승인 기록

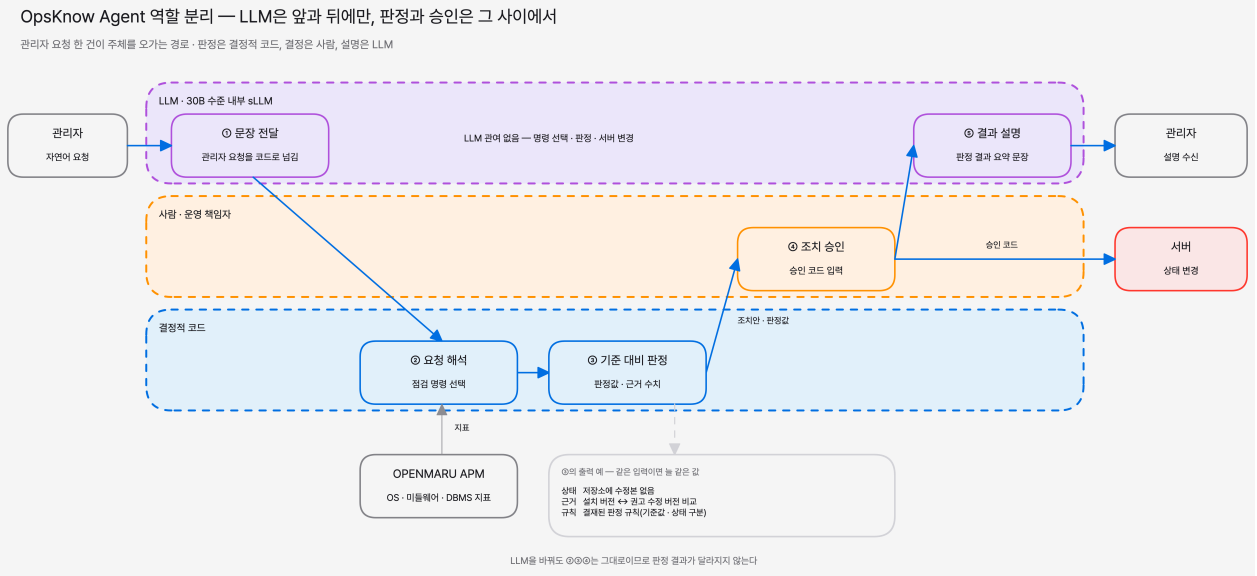


그림. 관리자 요청이 LLM(전달) → 결정적 코드(해석·판정) → 사람(승인) → LLM(설명)으로 이어지는 OpsKnow Agent 역할 분리

5.1.1 같은 입력에 같은 판정이 나오는 결정적 코드와 감사 재현성

결정적 코드의 판정 재현성

결정적 코드는 같은 입력에 늘 같은 결과를 내는 프로그램 코드입니다. 예를 들어 디스크 사용률이 기준값을 넘는지, 설치된 패키지 버전이 보안 권고의 수정 버전보다 낮은지 같은 판정은 비교 규칙으로 표현할 수 있으므로, 코드가 맡으면 언제 다시 실행해도 결과가 같습니다. 반면 LLM이 판정까지 맡으면 같은 서버 상태를 두고도 실행마다 결론이 달라질 수 있고, 감사 시점에 당시의 판정을 다시 만들어 보이기 어렵습니다. OpsKnow Agent는 판정값과 그 근거 수치를 코드의 출력으로 남기고, LLM은 이 출력을 사람이 읽기 쉬운 문장으로 옮기는 일만 합니다.

보안 패치 판정의 상태 구분

판정 결과는 정해진 상태값으로만 나오므로, 보고서를 받은 담당자가 결과를 해석하는 방법도 하나로 고정됩니다. 보안 패치 점검의 경우 판정 상태는 다음 다섯 가지로 나뉩니다.

- 이미 조치됨
- 설치돼 있지 않음
- 저장소에 수정본 없음

- 보안 정보 없음
- 지원 종료 OS

근거가 확실하지 않은 서버는 「최신」으로 표시하지 않습니다. 근거가 불충분한 상태를 정상으로 보고하는 일을 판정 규칙 단계에서 막아 두었기 때문에, 보고서의 「이상 없음」이 실제 확인 결과를 뜻한다는 점을 감사 때 그대로 설명할 수 있습니다.

2023년 행정전산망 장애와 원인 판단의 재검증

판정 근거를 나중에 다시 검증할 수 있어야 한다는 요구는 실제 장애 대응에서도 드러났습니다. 2023년 11월 행정전산망 장애 때 초기에는 L4 스위치 운영체제 업데이트 오류가 원인으로 지목됐지만, 정부는 같은 달 25일 브리핑에서 라우터 포트 불량을 원인으로 발표하고 해킹 흔적은 없다고 밝혔습니다[S04][S05]. 이처럼 원인 판단은 조사 과정에서 바뀔 수 있으므로, 점검 단계의 판정이 수치와 규칙으로 남아 있어야 어느 판단이 맞았는지 나중에 대조할 수 있습니다. 결정적 코드의 판정은 장애 뒤 원인 분석에서도 같은 입력으로 다시 실행해 볼 수 있는 기준 기록 역할을 합니다.

두 방식의 차이를 감사 대응 관점에서 나누면 다음과 같습니다.

비교 항목	LLM이 판정까지 맡는 방식	결정적 코드가 판정하는 방식
같은 입력의 결과	실행마다 표현과 결론이 달라질 수 있음	늘 같은 판정값
감사 때 재현	당시 응답을 다시 만들기 어려움	같은 입력으로 다시 실행해 같은 결과 확인
판정 근거	설명 문장 속에 섞여 있음	판정값과 근거 수치가 따로 남음
기준 변경 관리	프롬프트 수정에 의존	판정 규칙의 변경 이력으로 관리

5.1.2 모델이 바뀌어도 흔들리지 않는 판정 기준

판정 기준과 언어 모델의 분리

판정 기준이 코드에 들어 있으면 언어 모델을 바꾸어도 판정 결과는 달라지지 않습니다. AI 모델은 성능 개선이나 보안 정책 변화에 따라 교체 주기가 짧고, 공공기관은 국가정보원의 국가 망 보안체계(N2SF) 가이드라인처럼 업무 정보의 중요도에 따라 생성형 AI 사용 조건을 달리하는 정책을 따라야 합니다[S08]. 모델 교체가 판정 기준 변경으로 이어지는 구조라면, 모델을 바꿀 때마다 점검 결과 전체를 다시 검증해야 합니다. 반면 OpsKnow Agent에서는 모델이 맡는 일이 설명 문장으로 한정되므로, 모델을 바꾼 뒤 확인할 범위도 설명 문장의 정확성으로 줄어듭니다.

30B 수준 내부 sLLM과 CogentAI의 역할

OpsKnow Agent가 설명 문장을 만들 때 쓰는 모델은 기관 내부에서 동작하는 30B 수준 내부 sLLM입니다. OPENMARU CogentAI는 OpsKnow 구성에서 근거 검색(RAG)과 이 30B 수준 내부 sLLM의 서빙을 함께 맡습니다. CogentAI를 함께 언급하는 이유는 설명 문장을 만드는 모델과 그 모델이 근거로 붙이는 과거 기록 검색이 모두 외부로 나가지 않고 기관 안에서 처리되기 때문입니다. 판정은 모델 밖의 코드가 하므로, 모델 규모가

대형 상용 모델보다 작더라도 판정의 정확도는 모델 성능에 좌우되지 않습니다. 이 구성은 GPU 1대로 구성한 폐쇄망(외부 연결 없음)에서 동작하는 소버린 AI Level 3 구성을 전제로 합니다(6장).

구성 요소마다 바꿀 때 판정에 주는 영향은 다음과 같이 다릅니다.

바뀌는 구성 요소	판정 결과에 주는 영향	바꾼 뒤 확인할 것
언어 모델(30B 수준 내부 sLLM) 교체	없음	설명 문장의 정확성과 표현
판정 규칙(기준값·상태 구분) 변경	있음	변경 전후 판정 비교, 결재 기록
수집 대상(OS·미들웨어·DBMS) 추가	판정 대상만 늘어남	새 대상의 판정 결과 표본 검토
보고서 양식 변경	없음	보고서 표기

판정 규칙 변경의 결재 관리

판정 기준을 바꾸는 일은 모델 교체와 달리 운영 책임자가 결정할 사항입니다. 기준값이나 상태 구분을 고치면 같은 서버 상태에 대한 판정이 달라지므로, 규칙 변경 자체를 변경관리 절차의 대상으로 두어야 감사 때 판정 차이를 설명할 수 있습니다. 행정안전부가 2024년 예방점검체계와 함께 마련한 표준운영절차(SOP)에는 장애관리와 변경관리가 들어 있어[S06][S07], 판정 규칙 변경을 기존 변경관리 기록에 함께 남기면 별도 절차를 새로 만들 필요가 없습니다. 모델은 운영팀이 필요에 따라 바꾸고 판정 규칙은 결재를 거쳐 바꾸는 이원 관리가 OpsKnow Agent 운영의 기본 형태입니다.

5.2 30분 유효 1회용 승인 코드와 적용 직전 상태 재확인으로 보장하는 사람 결재

한국인터넷진흥원(KISA)은 보호나라&KrCERT/CC와 취약점 정보포털(KNVD)에서 보안공지와 CVE 정보를 제공합니다[S10]. 이런 보안 취약점 권고가 나오면 대상 서버에 패치를 적용해야 하는데, 패치 적용은 서버 상태를 바꾸는 작업이므로 판정과 달리 사람의 결정이 필요합니다. 그래서 OpsKnow Agent는 이 결정을 승인 코드라는 별도 증표로만 받고, 대화 속의 동의 표현은 결정으로 인정하지 않습니다. 이 승인 코드 절차는 OS 패치·보안 조치와 8장 경계표의 결재 6종(임계치 조정, WAS 재시작, 인스턴스 증설·감축, 세션 강제 종료, 외부 시스템 재시작, DB 풀 크기 변경)에 똑같이 적용됩니다. 승인을 받은 뒤에도 적용 직전에 서버 상태를 다시 확인해, 점검 때와 다르면 적용을 멈춥니다.

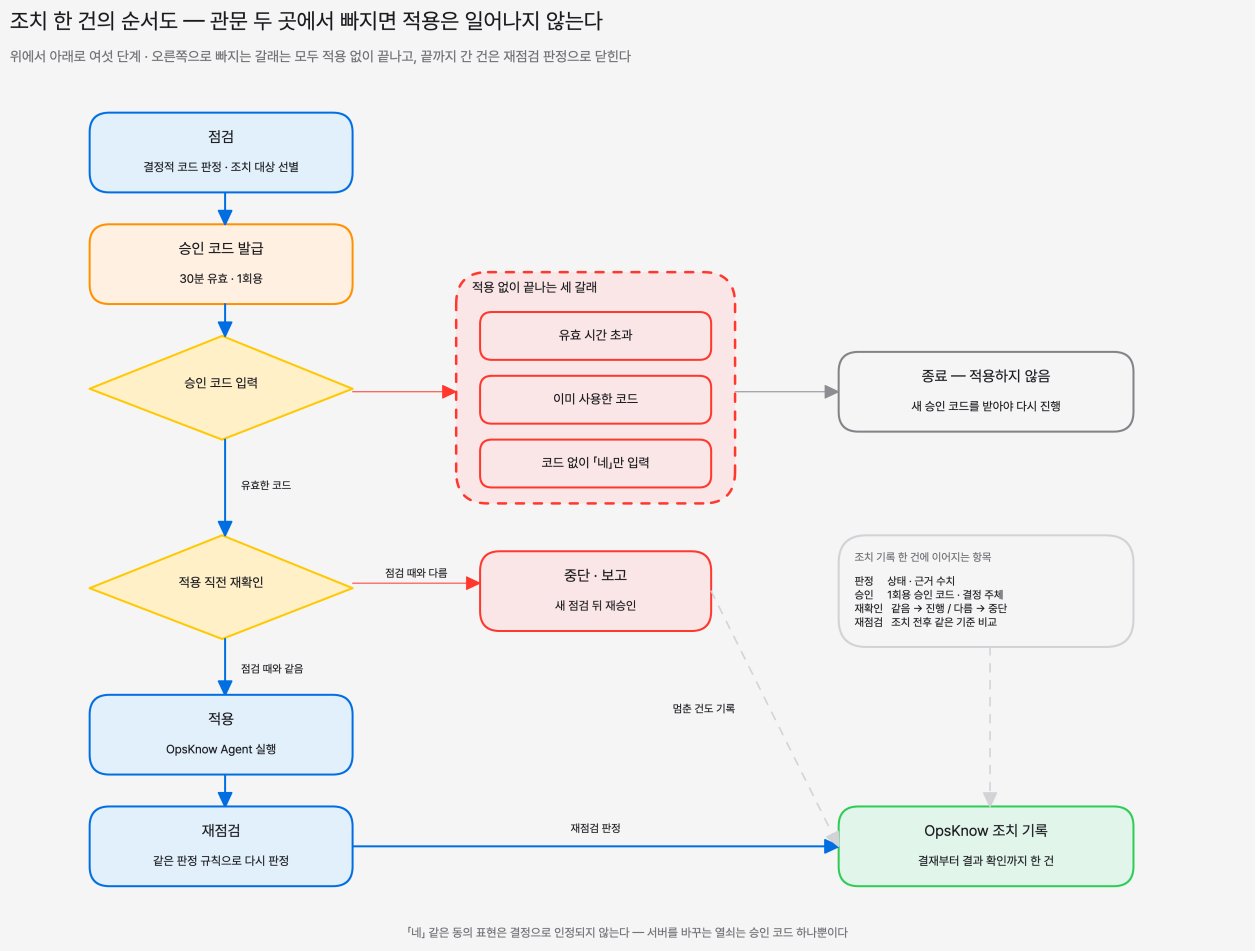


그림. 점검 → 승인 코드 발급 → 승인 코드 입력 → 적용 직전 재확인 → 적용 → 재점검으로 이어지는 조치 순서도

5.2.1 승인 코드 없이 「네」라는 답만으로는 적용하지 않는 규칙

승인 코드의 발급과 유효 범위

OpsKnow Agent는 조치가 필요한 점검 결과를 보고할 때 조치 대상과 내용을 함께 제시하고, 적용하려면 1회용 승인 코드를 입력하도록 안내합니다. 승인 코드는 발급 뒤 30분 동안만 유효하고 한 번 쓰면 다시 쓸 수 없으므로, 시간이 지난 승인이나 다른 조치에 같은 승인을 다시 쓰는 일이 생기지 않습니다. 예를 들어 유효 시간이 지난 코드나 이미 사용한 코드를 입력하면 에이전트는 적용하지 않으며, 새 승인 코드를 받아야 적용할 수 있습니다. 한 번 쓴 승인은 그 조치에만 묶이고, 승인과 적용이 한 쌍으로 이어집니다.

「네」 답변과 승인 코드의 구분

대화형 AI에서는 관리자가 「네」나 「진행해」처럼 짧게 답하는 일이 흔하지만, OpsKnow Agent는 이런 답만으로 서버를 바꾸지 않습니다. 짧은 동의 표현은 앞선 질문을 잘못 읽었거나 다른 대화에 답한 것일 수 있고, LLM이 문맥을 오해해 동의로 해석할 위험도 있기 때문입니다. 그래서 적용 여부는 LLM의 문장 해석과 관계없이 결정적 코드가 승인 코드의 일치 여부로만 판단합니다. 서버 변경의 결정권은 승인 코드를 입력한 사람에게만 있고, 입력별 에이전트 동작은 다음과 같습니다.

관리자 입력	에이전트 동작
유효한 승인 코드	적용 직전 재확인 뒤 적용
「네」 같은 동의 표현만 입력	적용하지 않음(승인 코드 필요)
유효 시간이 지난 승인 코드	적용하지 않음
이미 사용한 승인 코드	적용하지 않음

5.2.2 점검 시점과 서버 상태가 다르면 중단하는 적용 직전 재확인

점검 시점과 적용 시점의 상태 차이

점검과 승인 사이에는 시간이 흐르고, 그동안 다른 운영자가 같은 서버에 패치를 먼저 적용하거나 패키지 구성을 바꿀 수 있습니다. 점검 때의 정보로 그대로 적용하면 이미 조치된 서버에 같은 작업을 반복하거나, 바뀐 구성 위에 맞지 않는 패치를 얹게 됩니다. OpsKnow Agent는 승인 코드를 받은 뒤 적용 직전에 대상 서버의 상태를 다시 읽고, 점검 때 기록한 상태와 비교합니다. 두 상태가 다르면 적용을 멈추며, 관리자는 다시 점검한 결과를 보고 새로 승인할지 정합니다.

이 재확인 규칙은 패치와 결재 6종에 똑같이 적용되며, 재확인 결과에 따른 에이전트 동작은 다음과 같습니다.

적용 직전 재확인 결과	에이전트 동작
점검 때와 같은 상태	적용 진행
점검 때와 다른 상태	적용 중단, 새 점검과 재승인 필요

적용 중단 결과의 기록

적용을 멈춘 건도 승인 뒤 적용되지 않은 조치로 남으므로, 운영 책임자는 승인한 조치가 실행되지 않았다는 사실과 그 계기가 상태 변화였다는 점을 나중에 확인할 수 있습니다. 예를 들어 다른 운영자가 먼저 같은 패치를 적용했거나 승인받은 DB 풀 크기를 먼저 바꿨다면, 재확인 단계에서 상태 차이가 드러나 같은 작업이 두 번 실행되지 않습니다. 반면 점검 시점의 정보로 곧바로 실행하는 단순 자동화 스크립트는 그 사이의 상태 변화를 알아채지 못하므로, 이미 바뀐 서버에 작업을 다시 실행할 수 있습니다. 적용 직전 재확인은 잘못된 적용을 막는 장치이면서, 조치가 멈춘 이유를 증적으로 설명할 수 있게 하는 장치입니다.

5.2.3 결재 기록이 그대로 증적이 되는 승인 이력

승인 이력에 남는 항목

승인 코드 방식은 결재 행위가 시스템 흐름 안에 기록으로 남는다는 점에서 감사 대응에 유리합니다. 조치 한 건은 점검 판정, 승인 코드 입력, 적용 직전 재확인, 적용, 재점검의 순서를 거치고, 각 단계의 결과가 같은 조치 건으로 이어집니다. 단계별로 남는 결과는 다음과 같으며, 승인 단계의 승인자 계정과 승인 시각은 OpsKnow Agent가 동작하는 AAS가 기록합니다.

단계	남는 결과	감사 때 쓰임
점검 판정	결정적 코드의 판정 상태와 근거 수치	조치가 필요했던 근거 확인

단계	남는 결과	감사 때 쓰임
승인	1회용 승인 코드로 받은 사람의 결정, AAS가 남긴 승인자 계정과 승인 시각	결정 주체와 결정 시각 확인
적용 직전 재확인	점검 때와 같은지 여부, 다르면 중단	중단 사유 확인
재점검	적용 뒤 같은 규칙으로 다시 낸 판정	취약점 해소 확인

SLA 보고와 승인 이력의 연결

행정예고된 안정성 고시는 민간 클라우드와 위탁 운영을 포함해 SLA 체결을 요구하고, 중요 시스템 장애는 행정안전부 디지털안전상황실에 즉시 보고하도록 한 것으로 알려졌습니다[S02]. 이런 요구에 대응하려면 조치가 언제 누구의 결정으로 이루어졌는지를 보고서에 바로 옮길 수 있어야 합니다. 승인 이력이 OpsKnow에 쌓이면 정기 운영 보고서와 장애 보고서가 같은 기록을 근거로 만들어지므로, 결재 문서를 따로 모아 증적을 맞추는 수작업이 줄어듭니다. 결재 기록이 곧 증적이 되고, 운영팀은 보고서를 위해 기록을 다시 만들 필요가 없습니다.

재점검 결과로 끝나는 조치 기록

조치 기록은 재점검 결과까지 남아야 끝난 것으로 볼 수 있습니다. 패치를 적용했다는 기록만 있으면 실제로 취약점이 해소됐는지는 다음 정기 점검 때까지 알 수 없고, 그 사이에 문제가 남아 있어도 보고서에는 「조치 완료」로 표시됩니다. OpsKnow Agent는 적용 뒤 같은 판정 규칙으로 다시 점검하고, 그 판정으로 조치를 마무리합니다. 재점검도 결정적 코드가 판정하므로, 조치 전후의 판정을 같은 기준으로 비교한 결과가 증적으로 남습니다.

5.3 재부팅과 설정 변경, 제품 설치본 패치를 에이전트 범위 밖에 둔 작업 경계

OpsKnow Agent가 하지 않는 작업은 재부팅, 설정 변경 실행, 제품 설치본 패치, 패키지 삭제와 다운그레이드입니다. 이 작업들은 승인 코드가 있어도 에이전트가 실행하지 않으며, 운영자가 기존 변경 절차에 따라 직접 수행합니다. 여기서 말하는 설정 변경 실행은 결재 6종에 들지 않는 미들웨어·DBMS 설정 변경을 뜻하고, WAS 재시작이나 DB 풀 크기 변경 같은 결재 6종은 범위 밖 작업이 아니라 승인 코드 절차를 거쳐 실행되는 조치입니다. 서비스 영향이 크거나 되돌리기 어려운 작업을 처음부터 범위 밖에 두면, 에이전트에 주는 권한의 상한을 문서 한 장으로 설명할 수 있습니다. 행정예고된 고시가 위탁 운영에도 SLA 체결을 요구하는 만큼[S02], 이 상한은 위탁 운영 계약과 보안성 검토에서 그대로 쓸 수 있는 근거가 됩니다.

5.3.1 에이전트가 하지 않는 작업 목록과 서비스 영향·롤백 위험

제외 작업 네 가지의 공통 기준

에이전트 범위에서 뺀 작업의 공통점은 실행 결과가 서비스 중단으로 바로 이어지거나, 문제가 생겼을 때 되돌리는 롤백(rollback, 변경 전 상태로 되돌리기) 절차가 시스템마다 달라 사람의 판단이 필요하다는 점입니다. 운영체제 패키지 관리자로 설치한 보안 패치는 수정 버전과 의존 관계가 기록되므로 적용 전후를 코드로 비교할 수 있습니다. 반면 재부팅이나 미들웨어·DBMS 설정 변경은 결과가 서비스 동작 전체에 영향을 주고, 이중화 전환이나 업무 시간 같은 운영 조건을 함께 따져야 합니다. OpsKnow Agent는 적용 전후 상태를 코드로 비교할 수 있는 작업만 맡고, 나머지는 운영자에게 넘깁니다.

제외 작업	서비스 영향	롤백 위험	처리 주체
재부팅	서비스 중단이 바로 발생	기동 실패 시 복구 판단이 필요	운영자, 작업 계획 결재
설정 변경 실행	미들웨어·DBMS 동작 전체에 영향	변경 전 값 복원 방법이 시스템마다 다름	운영자
제품 설치본(zip/tar) 패치	패키지 관리자 밖이라 버전 추적이 어려움	원래 파일 보관과 복원을 사람이 관리	공급사 지침에 따른 운영자
패키지 삭제·다운그레이드	의존 패키지에 연쇄 영향	이미 적용한 보안 수정이 되돌려질 수 있음	운영자

자동 처리와 사람 결재의 운영 조치 구분

OpsKnow의 운영 조치도 같은 기준으로 나뉩니다. 알림 발송, OpsKnow 기록 저장, 임계치 보고서 생성은 자동으로 처리되지만, 서비스에 직접 영향을 주는 다음 조치는 사람의 결재를 거쳐야 합니다. 이 결재는 OS 패치·보안 조치와 같은 승인 코드 절차(점검, 승인 코드, 적용 직전 재확인, 재점검)로 받습니다.

- 임계치 조정
- WAS 재시작
- 인스턴스 증설과 감축
- 세션 강제 종료
- 외부 시스템 재시작
- DB 풀 크기 변경

이처럼 자동과 결재의 경계가 작업 종류별로 정해져 있으므로, 운영자는 어떤 알림이 자동으로 끝나고 어떤 알림이 결재 요청으로 오는지를 미리 알 수 있습니다. 경계가 분명하면 결재 요청의 양도 예측할 수 있고, 결재자가 반복 판정에 시간을 쓰지 않고 서비스 영향이 있는 결정에만 집중할 수 있습니다.

5.3.2 권한 범위 문서화로 대응하는 위탁 운영 계약과 보안성 검토

위탁 운영 계약의 권한 조항

위탁 운영 사업자가 운영하는 정보시스템에 AI 에이전트를 도입하면, 계약서에 에이전트의 권한 범위를 적어야 합니다. 행정예고된 고시가 민간 클라우드와 위탁 운영까지 SLA 체결 대상으로 넓힌 것으로 알려진 만큼[S02], 에이전트가 무엇을 하고 무엇을 하지 않는지는 사업자의 책임 범위와 직접 연결됩니다. OpsKnow Agent는 하지 않는 작업이 제품 동작으로 정해져 있으므로, 계약서에 적는 권한 조항과 실제 동작이 어긋나지 않습니다. 예를 들어 다음과 같은 문안을 계약의 권한 조항에 옮겨 쓸 수 있습니다.

조항	문안 예
판정	예방점검 판정은 사전에 결재된 판정 규칙에 따라 에이전트가 수행하고, 판정 규칙 변경은 발주기관 승인을 받는다
조치	OS 패치·보안 조치와 결재 6종 등 서버 상태를 바꾸는 조치는 운영 책임자가 1회용 승인 코드를 입력한 경우에만

조항	문안 예
	수행한다
제외 작업	재부팅, 설정 변경 실행, 제품 설치본 패치, 패키지 삭제와 다운그레이드는 에이전트가 수행하지 않는다
기록	판정, 승인, 적용, 재점검 결과를 보관하고 정기 보고에 포함한다

보안성 검토에 제출할 권한 범위 자료

보안성 검토에서는 새로 들이는 소프트웨어가 어떤 권한으로 무엇에 접근하는지를 설명해야 합니다. AI 에이전트는 「AI가 판단해 서버를 바꿀 수 있는가」라는 질문을 가장 먼저 받으며, OpsKnow Agent는 판정·결정·설명의 역할 분리와 제외 작업 목록으로 이 질문에 답할 수 있습니다. 또한 설명 문장을 만드는 30B 수준 내부 sLLM이 폐쇄망 안에서 동작하므로, 점검 데이터가 외부 AI 서비스로 나가지 않는다는 점도 같은 자료에 담을 수 있습니다. 권한 범위 문서는 계약서와 보안성 검토 자료에 같은 내용으로 들어가고, 담당자가 바뀌어도 같은 문서를 기준으로 운영을 이어 갈 수 있습니다.

5.4 AAS(AI Agent Studio)에서 정하는 에이전트별 권한 범위와 실행 기록이 남기는 책임 소재

AAS는 오픈마루의 AI 에이전트 운영 플랫폼이며, OpsKnow Agent는 이 플랫폼 위에서 실행됩니다. AAS를 함께 다루는 이유는 OpsKnow Agent의 권한 상한과 실행 기록이 에이전트 코드 바깥의 플랫폼 수준에서 한 번 더 통제되기 때문입니다. 정보시스템 등급이 국민 영향 기준의 A1~A4 체계로 바뀌면서[S01], 높은 등급 시스템을 맡는 에이전트와 낮은 등급 시스템을 맡는 에이전트의 권한을 다르게 두어야 할 필요도 커졌습니다. 그래서 OpsKnow Agent는 판정과 승인 규칙을 에이전트 안에 두고, 어떤 에이전트가 어떤 도구를 쓸 수 있는지는 AAS에서 관리하는 두 겹 구조를 씁니다. AAS는 에이전트의 예약(주기) 실행, 도구별 승인 요청, 승인자 계정·시각 기록을 제공하므로, 점검을 언제 돌리고 누가 언제 조치를 승인했는지가 플랫폼 기록으로 남습니다.

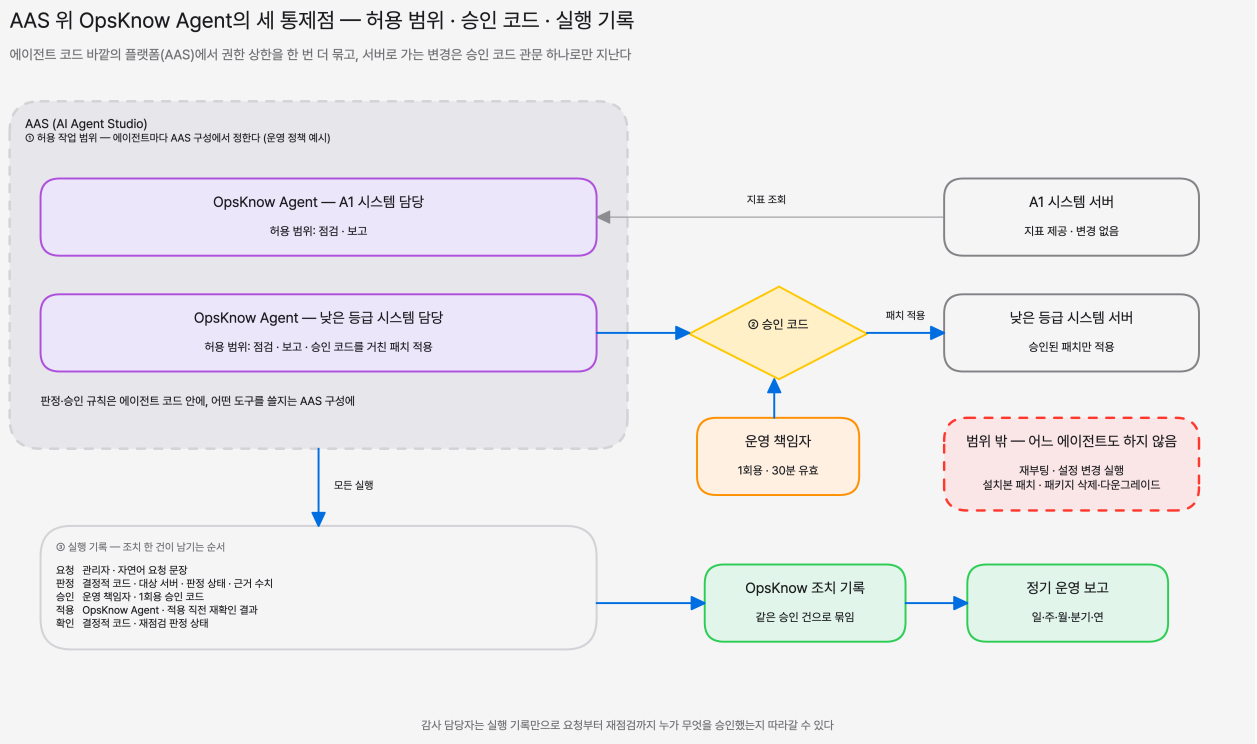


그림. AAS 위에서 OpsKnow Agent가 동작할 때 권한·승인·기록이 통제되는 지점

5.4.1 에이전트별 허용 작업 범위와 실행 이력 보존

AAS의 에이전트별 기능·도구 범위

에이전트 단위로 허용 범위를 나누면, 같은 OpsKnow Agent라도 맡는 시스템에 따라 허용 범위를 다르게 둘 수 있습니다. 예를 들어 A1 등급 시스템을 맡는 에이전트에는 점검과 보고만 허용하고, 낮은 등급 시스템을 맡는 에이전트에는 승인 코드를 거친 패치 적용까지 허용하는 식의 운영 정책을 기관이 정할 수 있습니다. 국가정보원의 N2SF 가이드라인도 업무 정보를 기밀·민감·공개로 나누고 등급별로 보안 통제를 달리하므로[S09], 시스템 등급에 맞춰 에이전트 권한을 나누는 방식은 공공 보안 정책의 구조와 같은 방향입니다. 권한 범위를 에이전트 단위로 나누면, 등급 체계가 바뀌어도 에이전트 설정만 고쳐 대응할 수 있습니다. AAS의 예약(주기) 실행으로 에이전트를 정해진 시각에 돌리므로 일상점검처럼 매일 반복되는 점검을 사람이 시작하지 않아도 되고, 도구마다 승인 요청 여부를 정해 서버 상태를 바꾸는 도구는 호출 전에 승인을 요청하게 할 수 있습니다.

AAS가 남기는 통제점별 기록

AAS가 에이전트의 권한과 기록을 통제하는 지점과 감사 때 확인할 수 있는 것은 다음과 같습니다.

통제점	AAS의 동작	감사 때 확인할 수 있는 것
에이전트별 허용 범위	판정·승인 규칙은 에이전트 코드에, 허용 작업 범위는 AAS 구성에 둠	에이전트가 할 수 있는 작업 목록
예약(주기) 실행	정해진 주기로 에이전트를 실행하고 실행마다 기록을 남김	정해진 주기대로 점검이 돌았는지
도구별 승인 요청	서버 상태를 바꾸는 도구는 호출 전에 승인을 요청하고, 1회용 승인 코드 없	조치마다 사람의 승인이 있었는지

통제점	AAS의 동작	감사 때 확인할 수 있는 것
	이는 적용하지 않음	
승인자 기록	승인자 계정과 승인 시각을 기록	누가 언제 승인했는지
실행 기록	요청, 판정, 적용, 재점검 결과를 기록으로 남김	요청부터 결과까지의 순서

감사 담당자는 특정 점검 결과가 어떤 요청에서 시작됐고 어떤 판정과 조치로 이어졌는지를 기록으로 따라갈 수 있습니다.

5.4.2 누가 무엇을 승인했는지 남는 결재 라인 기록

결재 라인 기록의 구성

공공 운영에서 책임 소재가 분명하려면 판정, 승인, 적용, 확인 각 단계의 주체가 기록에 따로 남아야 합니다. OpsKnow Agent의 조치 한 건은 결정적 코드의 판정, 운영 책임자의 승인 코드 입력, 에이전트의 적용, 재점검 판정 순서로 이어지며, 각 단계의 주체와 결과가 같은 승인 건에 묶입니다. 예를 들어 보안 취약점 패치 한 건의 결재 라인 기록은 다음과 같은 형태입니다.

단계	주체	남는 결과
요청	관리자	자연어 요청 문장
판정	결정적 코드	대상 서버, 판정 상태, 근거 수치
승인	운영 책임자	1회용 승인 코드 입력, AAS의 승인자 계정·승인 시각 기록
적용	OpsKnow Agent	적용 직전 재확인 결과, 적용 결과
확인	결정적 코드	재점검 판정 상태
설명	30B 수준 내부 sLLM	결과 요약 문장

담당자 교체와 무관한 운영 절차

결재 라인이 기록으로 남으면 운영 절차가 특정 담당자의 경험에 묶이지 않습니다. 공공 정보시스템은 인사 이동이나 위탁 사업자 교체로 담당자가 자주 바뀌는데, 판정 규칙과 승인 절차가 코드와 AAS 설정에 들어 있으면 새 담당자도 같은 절차를 그대로 이어받습니다. 이전 담당자가 승인한 조치의 근거와 결과가 OpsKnow에 운영 지식으로 쌓여 있으므로, 같은 유형의 보안 취약점이 다시 나왔을 때 과거 조치를 찾아 참고할 수 있습니다. 따라서 AAS 위의 OpsKnow Agent는 판정은 코드, 결정은 사람, 설명은 AI로 나눈 구조를 기록으로 고정해, 감사 대응과 책임 소재 요건을 담당자 교체와 관계없이 유지합니다.

6장: 소버린 AI Level 3로 구축하는 이유 — GPU 1대 폐쇄망에서 실행하는 30B 수준 내부 sLLM

소버린 AI(Sovereign AI)는 AI 모델과 그 모델이 읽는 데이터를 기관이 직접 소유하고 통제하는 운영 방식입니다. 예방점검에 AI 에이전트를 붙이면 서버 설정, 패치 상태, 장애 이력 같은 운영 데이터가 모델의 입력이 되므로, 그 데이터가 기관 밖으로 나가는지를 가장 먼저 검토해야 합니다. 국가정보원은 2025년 9월 30일 국가 망 보안체계(N2SF, National Network Security Framework) 보안가이드라인 1.0을 공개하면서 업무 정보를 등급으로 나누고 등급마다 통제를 달리하는 원칙을 세웠습니다[S08]. 공공기관이 생성형 AI를 운영 업무에 쓰려면 모델 성능보다 데이터가 머무는 위치와 그 위치를 입증하는 방법을 먼저 정해야 합니다.

OPENMARU OpsKnow는 이 요구에 맞춰 외부 LLM API를 한 번도 부르지 않는 폐쇄망 구성을 기본으로 삼습니다. OpsKnow Agent가 쓰는 언어 모델은 기관 내부 GPU 서버 1대에서 실행되는 30B 수준 내부 sLLM입니다. 30B는 매개변수가 약 300억 개 규모라는 뜻이고, 이 정도 크기는 GPU 장비 1대에 올려 운영할 수 있습니다. 모델이 작아도 예방점검 판정의 정확도가 유지되는 이유는 기준 대비 판정을 코드가 맡고 모델은 설명만 맡기 때문입니다(6.3절).

이 장은 보안성 검토 문서와 예산 요구서에 그대로 옮길 수 있는 세 가지 판단 근거를 제시합니다. 첫째는 운영 데이터가 외부로 나가면 안 되는 근거이고, 둘째는 「외부 LLM API 호출 0건」을 시험으로 판정하는 소버린 AI Level 3 구성 기준이며, 셋째는 GPU 서버 1대를 단위로 하는 비용과 운영 안정성입니다. 차년도 사업 요구사항에 「외부 LLM API 호출 없이 기관 내부 GPU 서버에서 동작」이라는 문장을 넣을지는 이 세 근거로 정할 수 있습니다.

6.1 운영 데이터의 외부 LLM 전송 위험과 국가정보원 N2SF 등급별 보안 통제

N2SF 보안가이드라인 1.0은 공공기관이 생성형 AI와 외부 클라우드를 쓸 수 있는 조건을 처음으로 정식 문서에 담은 기준입니다[S08]. 그동안 공공 업무망은 망분리 원칙 때문에 외부 AI 서비스를 사실상 쓸 수 없었고, 새 가이드라인은 조건을 갖추면 업무 PC에서도 생성형 AI와 외부 클라우드를 쓸 수 있도록 정보 등급에 따라 허용 범위를 나누었습니다[S09]. 다만 허용 조건이 생겼다고 해서 모든 정보를 외부 모델로 보내도 되는 것은 아닙니다. 예방점검 데이터가 어느 등급에 해당하는지, 그 등급에서 외부 LLM 전송이 허용되는지를 먼저 따져야 합니다.

6.1.1 기밀·민감·공개 3등급 분류와 약 280개 보안 통제를 담은 N2SF 가이드라인 1.0

N2SF 등급 분류의 세 단계

N2SF는 업무 정보를 기밀(Classified, C), 민감(Sensitive, S), 공개(Open, O) 세 등급으로 나누고 등급마다 다른 보안 통제를 적용합니다[S08]. 공개된 보도에 따르면 정식판은 약 280개 보안 통제 항목과 11종 정보 서비스 모델을 제시하며, 조건을 갖춘 업무 PC에서 생성형 AI와 외부 클라우드를 쓰는 환경도 그 범위에 들어갔습니다[S08][S09]. N2SF는 등급에 맞는 통제를 갖추면 AI를 쓸 수 있게 한 조건부 허용 체계이며, 금지 목록과는 성격이 다릅니다.

예방점검 데이터에 적용할 등급 판단 기준

어느 정보를 어느 등급으로 둘지는 각 기관이 정하므로, 예방점검 데이터의 등급도 기관이 판단해야 합니다. 서버 목록, 미들웨어와 DBMS의 설정값, 패치 적용 현황은 그 자체로 시스템 구조를 드러내기 때문에 공개 등급으로 두기 어렵습니다. 아래 표는 N2SF 원문의 판정 기준이 아니라, 기관이 예방점검 데이터의 등급을 정할 때 참고할 수 있는 판단 예시입니다.

등급	예방점검 데이터 예	외부 LLM 전송 판단(예시)
기밀(C)	핵심 시스템 망 구성도, 접근 통제 정책	내부 처리
민감(S)	서버 설정, 패치 미적용 목록, 장애 원인 기록	내부 모델 처리 권장
공개(O)	공개된 CVE 설명, 제품 매뉴얼	기관이 정한 조건을 갖추면 전송 검토

이 예시대로라면 예방점검이 다루는 데이터 대부분은 민감 등급 이상에 해당할 가능성이 높습니다. 운영 데이터를 읽는 AI 에이전트를 처음부터 내부 모델로 처리하도록 구성해 두면, 데이터마다 등급을 다시 따지지 않고도 보안성 검토 자료를 준비할 수 있습니다.

6.1.2 서버 설정과 취약점, 장애 이력이 외부로 나가면 안 되는 운영 정보인 이유

취약점 정보와 미적용 목록의 결합 위험

보안 취약점 정보는 한국인터넷진흥원이 운영하는 보호나라&KrCERT/CC와 KNVD(취약점 정보포털)에서 누구나 볼 수 있습니다[S10]. 공개된 CVE 설명만으로는 공격 대상을 특정할 수 없지만, 「어느 기관의 어느 서버가 어느 CVE를 아직 조치하지 않았는가」라는 목록과 결합하면 사정이 달라집니다. 이 결합 정보는 공격자에게 바로 쓸 수 있는 공격 대상 목록이 되기 때문입니다. AI 에이전트가 보안 취약점 점검 결과를 요약하려고 외부 LLM에 미적용 목록을 보내는 순간, 기관은 가장 민감한 운영 정보를 기관 밖에 남기게 됩니다.

외부 LLM 의존이 만드는 세 가지 위험 유형

외부 LLM API에 운영 업무를 맡기면 정보 노출 말고도 서비스 중단과 정책 변경이라는 두 위험이 함께 따라옵니다. 외부 서비스가 장애로 멈추면 점검 보고서 작성이 함께 멈추고, 제공사가 약관이나 데이터 보관 정책을 바꾸면 기관이 이미 보낸 데이터의 처리 방식도 바뀝니다. 아래 표는 세 위험을 내부 sLLM 구성과 비교한 것입니다.

위험 유형	외부 LLM API 사용 시	내부 sLLM 사용 시
정보 노출	설정·취약점·장애 기록이 제공사 서버를 거쳐 감	데이터가 기관 망 안에서만 처리됨
서비스 중단	제공사 장애·회선 단절 때 점검 보고가 멈춤	기관 서버가 살아 있으면 동작
정책 변경	약관·보관 기간·단가가 제공사 결정으로 바뀜	모델 교체 시점을 기관이 정함

증적의 무결성과 보관 위치

예방점검 결과는 감사 때 제출하는 증적이므로, 증적을 만든 과정까지 기관이 설명할 수 있어야 합니다. 증적의 일부가 외부 모델의 응답으로 만들어졌다면 그 응답이 어떤 입력을 받았고 어디에 기록됐는지를 기관이 직접 보여 주기 어렵습니다. 반면 모델 입력과 출력이 모두 기관 서버에 남으면 감사인이 요청한 기록을 그대로 꺼낼 수 있습니다. 외부 전송을 막는 일은 보안 통제이면서 동시에 증적 관리 요건이기도 합니다.

6.2 외부 LLM API 호출 0건을 판정 기준으로 삼는 소버린 AI Level 3 구성

소버린 AI Level 3 구성은 OPENMARU가 자사 제품의 AI 구성 수준을 나누려고 정한 3단계 분류 가운데 가장 엄격한 단계입니다. 「소버린 AI를 지원한다」는 문장만으로는 보안성 검토 담당자가 무엇을 확인해야 하는지 알 수 없으므로, 각 단계를 시험으로 판정할 수 있는 조건으로만 정의했습니다. Level 3는 인터넷을 막은 상태에서 모든 AI 기능이 동작하고 외부 LLM API 호출이 0건이어야 한다는, 측정 가능한 기준 하나로 요약됩니다. N2SF 아래에서 민감 등급 이상으로 분류할 운영 데이터를 AI로 처리한다면[S08], 이 수준의 구성이 가장 보수적인 선택입니다.

6.2.1 Level 1부터 Level 3까지 단계 정의와 외부 송신 전면 차단 상태의 동작 요건

AI 구성 요소 위치에 따른 단계 구분

세 단계는 LLM, 임베딩 모델, 벡터DB, 가드레일이 어디에 있느냐로 갈립니다. 임베딩 모델은 문서를 검색용 숫자 벡터로 바꾸는 모델이고, 벡터DB는 그 벡터를 저장해 비슷한 문서를 찾아 주는 저장소이며, 가드레일은 모델의 입력과 출력에서 위험한 내용을 걸러 내는 장치입니다. 네 요소 가운데 하나라도 외부 서비스에 있으면 운영 데이터 일부가 기관 밖으로 나갑니다. 아래 표는 OPENMARU가 정한 단계 정의입니다.

단계	표기	필수 기준	판정 방법
Level 1	BUILT-IN(제품 내장)	제품 화면과 API 안에 AI 기능이 들어 있음	기능 시연과 기능 목록
Level 2	ON-PREM(온프레미스)	Level 1에 더해 LLM·임베딩·벡터DB·가드레일을 모두 기관 서버에 설치	설치 구성 확인
Level 3	AIR-GAPPED(폐쇄망)	Level 2에 더해 인터넷 차단 상태에서 모든 AI 기능 동작, 외부 LLM API 호출 0건, 지정 GPU 장비 1대에서 동작	외부 송신 전면 차단 시험과 외부 연결 로그 0건 확인

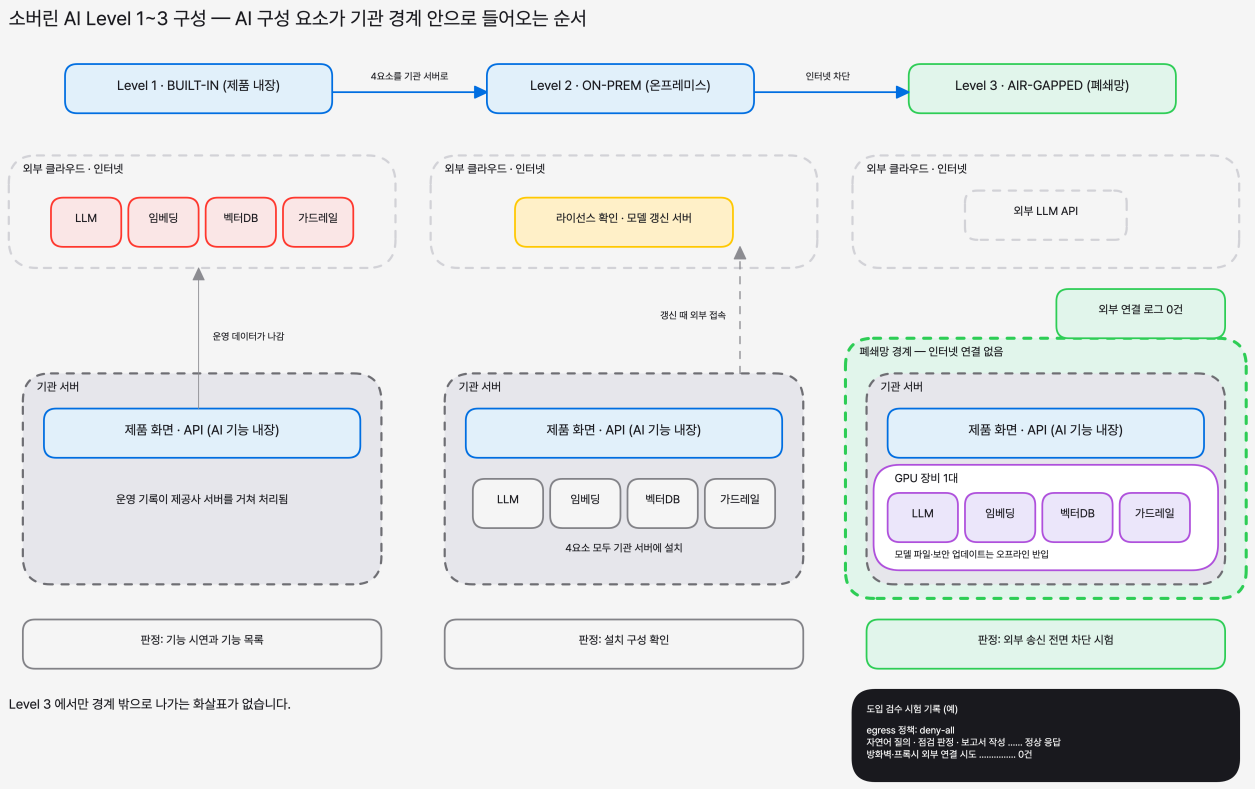


그림. 소버린 AI Level 1~3 구성 비교 — AI 구성 요소가 기관 경계 안으로 들어오는 순서

Level 2와 Level 3를 가르는 동작 요건

Level 2와 Level 3를 가르는 기준은 인터넷이 끊긴 상태에서도 모든 AI 기능이 동작하느냐입니다. 두 단계 모두 구성 요소는 기관 서버에 있습니다. 예를 들어 모든 모델을 기관 서버에 설치했더라도 라이선스 확인이나 모델 갱신 때 외부 서버에 접속해야 한다면 Level 3 요건을 채우지 못합니다. 따라서 Level 3 구성은 모델 파일 반입, 보안 업데이트, 설치 절차까지 오프라인 반입 방식으로 갖춰야 합니다. 이 요건을 충족하면 업무 정보 등급별로 통제를 달리하는 N2SF 환경에서도 AI 기능을 기관 망 안에서만 쓰는 구성이 됩니다[S08][S09].

6.2.2 외부 송신 차단 시험과 외부 연결 로그 0건으로 확인하는 판정 방법

외부 송신 전면 차단 시험의 절차

Level 3 판정은 외부 송신 전면 차단(deny-all egress, 기관 망에서 밖으로 나가는 모든 연결을 막는 방화벽 정책) 상태에서 AI 기능을 실제로 돌려 보는 시험으로 합니다. 시험 중에는 방화벽과 프록시에서 외부로 나가려 한 연결 시도를 모두 기록하고, 그 기록이 0건인지 확인합니다. 「외부로 보내지 않는다」는 제공사의 설명을 믿는 대신, 막아 놓은 상태에서 기능이 동작하는지와 나가려는 시도가 없었는지를 로그로 확인하는 방식입니다. 아래 표는 기관이 도입 검수 때 그대로 쓸 수 있는 판정 순서입니다.

순서	확인 항목	통과 조건
1	방화벽에 외부 송신 전면 차단 정책 적용	기관 망 밖으로 나가는 연결이 모두 막힘
2	자연어 질의, 점검 판정, 보고서 작성 기능 실행	모든 기능이 오류 없이 응답

순서	확인 항목	통과 조건
3	시험 중 방화벽·프록시 연결 로그 수집	외부로 나가려 한 연결 시도 0건
4	모델 입력·출력 기록 위치 확인	기록이 기관 서버에만 남음

RFP 문안으로 옮기는 방법

이 판정 방법은 차년도 사업의 제안요청서(RFP)에 요구사항 문장으로 그대로 옮길 수 있습니다. 예를 들어 「AI 기능은 외부 LLM API 호출 없이 기관 내부 GPU 서버에서 동작해야 하며, 검수 시 외부 송신 전면 차단 상태에서 시험해 외부 연결 로그 0건을 입증한다」라고 쓰면 됩니다. 이렇게 쓰면 「소버린 AI 지원」 같은 모호한 문구로 제안사를 고를 때보다 평가 기준이 분명해집니다. 같은 시험을 운영 중에도 주기적으로 반복하면 구성 변경 뒤에도 Level 3 요건이 유지되는지 확인할 수 있습니다.

6.2.3 나가지 않고 사람이 결정하며 끌 수 있는 소버린 AI 5원칙

전 단계에 공통으로 적용하는 다섯 가지 원칙

OPENMARU는 Level 1~3 단계와 별도로 모든 단계에 공통으로 적용하는 다섯 가지 원칙을 둡니다. 단계 구분이 AI 구성 요소의 위치를 정한다면, 다섯 원칙은 그 AI를 누가 통제하는지를 정합니다. 아래 표의 뜻풀이와 확인 방법은 기관이 보안성 검토와 도입 검수 때 쓸 수 있도록 정리한 예시입니다.

원칙	뜻(예시)	확인 방법(예시)
나가지 않는다	추론 중 외부 송신 0건	망 차단 상태의 연결 로그
사람이 결정한다	되돌릴 수 없는 동작은 사람 승인 뒤에만 실행	승인자와 승인 기록
끌 수 있다	AI 기능을 꺼도 제품 본래 기능은 동작	AI 비활성 상태 운영 시험
들여다볼 수 있다	모델 버전·프롬프트·입출력 기록이 기관 장비에 남음	기록 열람
감당할 수 있다	구성 사양을 GPU 1대로 명시하고 그 사양에서 재현	지정 장비 동작 확인

OpsKnow Agent에서 원칙이 동작하는 방식

OpsKnow Agent는 다섯 원칙 가운데 「사람이 결정한다」를 승인 코드로 구현합니다. 서버를 바꾸는 조치는 한 번만 쓸 수 있는 승인 코드가 있어야 실행되고, 적용 직전에 서버 상태를 다시 확인해 점검 때와 다르면 실행을 멈춥니다. 승인 코드 없이 「네」라고만 답하면 조치는 실행되지 않습니다. 결정권을 사람에게 두는 구조는 AI가 운영 서버를 직접 바꾸는 위험을 줄이고, 누가 언제 승인했는지를 증적으로 남깁니다.

6.3 GPU 1대에서 30B 수준 내부 sLLM으로 충분한 이유 — 판정을 코드가 맡는 구조

30B 수준 내부 sLLM이 예방점검 업무에 충분한 이유는 모델이 맡는 일이 요청 해석과 결과 설명으로 좁혀져 있기 때문입니다. 대형 상용 모델은 복잡한 추론을 모델 혼자 끝내야 하는 업무에서 강점을 보이지만, 예방점검의 판정은 임계값과 규칙으로 정해지는 결정적 계산입니다. 따라서 판정을 코드에 맡기면 모델 크기가 정확도를 좌우하지 않고, 필요한 GPU 자원도 1대 단위로 줄어듭니다. 공공기관은 N2SF 등급 체계[S09]에서 민감 등급으로 볼 운영 데이터를 내부에서 처리하면서도 장비 예산을 작게 잡을 수 있습니다.

6.3.1 모델이 해석과 설명만 맡기 때문에 대형 모델이 필요 없는 구조

코드와 모델의 역할 분담

OpsKnow Agent에서 언어 모델은 사용자의 문장을 받아 넘기고, 코드가 만든 결과를 사람이 읽기 쉬운 문장으로 전하는 일만 합니다. 문장 해석, 실행할 명령 선택, 점검 결과 판정은 결정적으로 동작하는 코드가 맡습니다. 예를 들어 「지난주 DBMS 점검에서 위험 판정이 나온 서버를 보여 줘」라는 요청이 들어오면, 코드가 대상 기간과 판정 조건을 정해 조회하고 모델은 그 결과를 문장으로 정리합니다. 즉, 같은 입력에는 늘 같은 판정이 나오므로 모델이 바뀌어도 점검 결과가 흔들리지 않습니다.

역할에 따라 달라지는 자원 요구

모델이 판정까지 맡으면 오판 위험을 줄이려고 더 큰 모델을 써야 하고, 그만큼 GPU가 늘어납니다. 반면 해석과 설명만 맡기면 중간 규모 모델로 충분하고, 오판 가능성은 코드 검증으로 관리합니다. 아래 표는 역할별로 필요한 자원을 비교한 것입니다.

역할	맡는 주체	필요한 자원
지표 수집	OPENMARU APM 수집기	대상 서버의 수집 에이전트, GPU 불필요
기준 대비 판정	결정적 코드	일반 CPU 서버, GPU 불필요
자연어 요청 해석, 결과 설명	30B 수준 내부 sLLM	GPU 장비 1대
조치 승인	사람(승인 코드)	별도 자원 불필요

표에서 GPU가 필요한 역할은 하나뿐입니다. 예방점검 AI의 GPU 규모는 점검 대상 서버 대수보다 동시에 들어오는 요청과 설명 문장의 양에 더 크게 좌우되며, OPENMARU는 GPU 장비 1대를 기준 구성으로 둡니다. 이 분담은 5장에서 설명한 「판정은 코드, 결정은 사람, 설명은 AI」 원칙을 자원 산정에 그대로 옮긴 결과입니다.

6.3.2 LLM과 임베딩, 벡터DB, 가드레일을 모두 기관 서버 안에 두는 폐쇄망 구성

폐쇄망 구성 요소의 배치

폐쇄망 구성은 수집, 판정, AI 처리, 저장, 화면 다섯 계층을 모두 기관 망 안에 둡니다. OPENMARU APM은 OS, 미들웨어, DBMS 지표를 기관 서버에서 수집하는 계층이므로, 원천 데이터가 처음부터 기관 망을 벗어나지 않는다는 점에서 이 구성의 출발점이 됩니다. OPENMARU CogentAI는 LLM·임베딩·벡터DB·가드레일을 GPU 장비 1대 안에서 함께 실행하는 AI 엔진이므로, 근거 검색(RAG)과 30B 수준 내부 sLLM 서빙을 외부 서비스 없이 제공한다는 점에서 이 장의 핵심 구성 요소입니다. AAS는 OpsKnow Agent를 실행하는 에이전트 실행 기반이므로, 연결된 모델이 내부 sLLM이면 에이전트의 모든 대화가 기관 망 안에서 끝난다는 점에서 이 구성에 포함됩니다.

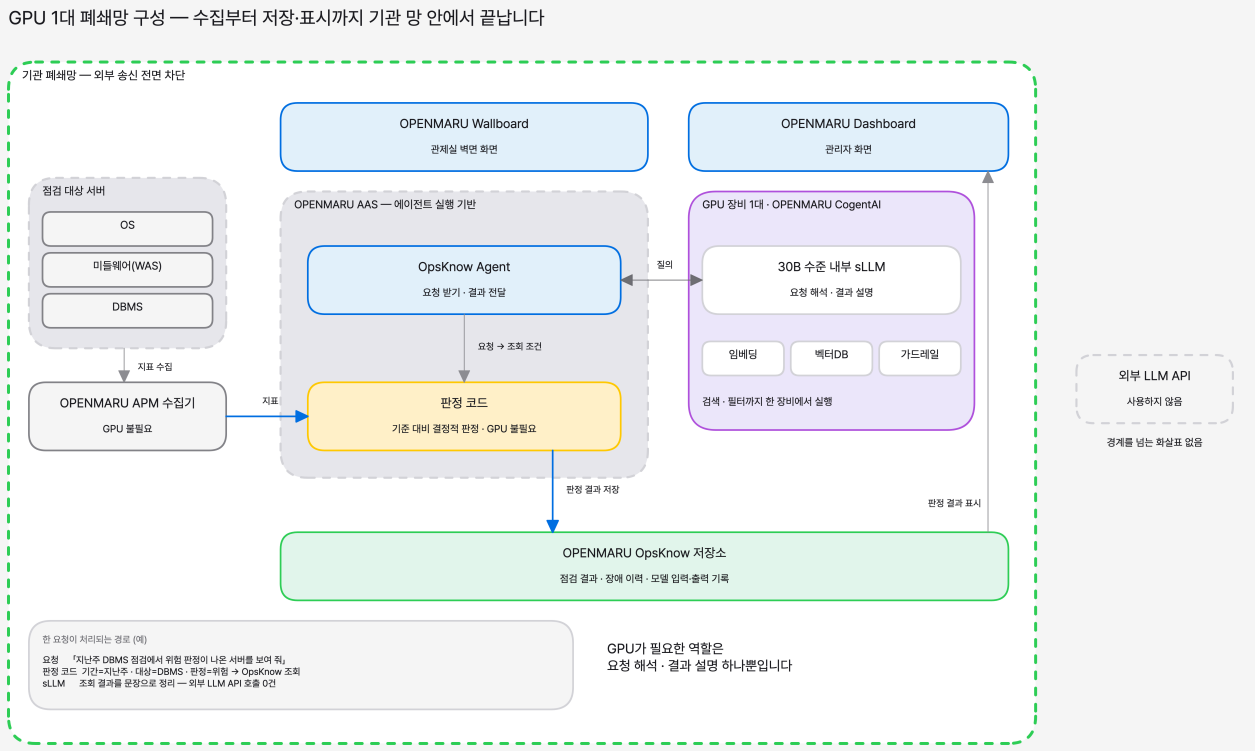


그림. GPU 1대 폐쇄망 구성도 — 지표 수집부터 OpsKnow 저장까지 기관 망 안에서 끝나는 흐름

화면과 저장소까지 내부에 두는 이유

AI 처리만 내부에 두고 결과 화면이나 저장소를 외부 서비스에 두면 증적이 다시 기관 밖으로 나갑니다. 그래서 점검 결과를 보여 주는 Wallboard와 OPENMARU Dashboard도 기관 망 안에서 동작해야 하며, 두 화면은 OpsKnow에 쌓인 판정 결과를 관제실과 관리자에게 보여 주는 출구라는 점에서 같은 경계 안에 둡니다. 수집부터 판정, 설명, 저장, 표시까지 한 경계 안에서 끝나므로 6.2.2의 외부 송신 차단 시험을 구성 전체에 한 번에 적용할 수 있습니다. 이 방식은 N2SF 환경의 보안성 검토에서 AI 기능까지 포함한 외부 송신 여부를 한 번에 확인하는 방법이 됩니다[S08].

6.4 GPU 서버 1대 단위의 도입 비용과 외부 서비스 정책 변경에서 분리된 운영 안정성

GPU 서버 1대 단위 구성은 예산 요구서에 산정 단위를 고정할 수 있다는 점에서 외부 LLM API 방식과 다릅니다. 외부 API는 사용량에 따라 비용이 바뀌는 변동비이고, 단가와 약관도 제공사가 정하므로 연간 예산을 미리 묶기 어렵습니다. 반면 내부 GPU 서버는 장비 도입 뒤 운영비가 전력, 유지보수, 모델 갱신 작업으로 정해집니다. 2026년 8월 행정안전부가 공공 정보시스템 15,033개의 등급을 새로 정했듯이 운영 대상이 많은 공공 환경에서는 예측 가능한 비용 구조가 특히 중요합니다[S01].

6.4.1 예산 산정 단위가 되는 GPU 서버 1대 구성

예산 요구서에 넣을 산정 항목

GPU 서버 1대 구성의 예산은 장비 자체와 그 장비를 폐쇄망에서 운영하는 데 드는 항목으로 나뉩니다. 금액은 장비 사양과 조달 시점에 따라 달라지므로, 예산 요구서에는 아래 항목을 빠짐없이 적고 견적으로 금액을 채우는 방식이 안전합니다. 폐쇄망에서는 모델과 보안 업데이트를 오프라인으로 반입해야 하므로 반입 절차 비용을 따로 잡아야 합니다.

산정 항목	내용	비용 성격
GPU 장비	30B 수준 내부 sLLM과 임베딩·벡터 DB·가드레일을 함께 실행하는 장비 1대	일회성 도입비
상면·전력	전산실 랙 공간, 전력, 냉각	연간 고정비
유지보수	장비 하드웨어 유지보수, AI 엔진 기술지원	연간 고정비
오프라인 반입	모델·보안 업데이트 파일의 반입 검사와 설치	연간 고정비
운영 인력	판정 규칙 관리, 승인 업무	기존 운영 조직의 업무로 배정

장비 1대로 시작하는 확장 방식

GPU 장비 1대 구성은 대상 서버가 늘어도 GPU를 그만큼 늘리지 않아도 된다는 점에서 확장 비용이 작습니다. 6.3.1에서 본 것처럼 지표 수집과 판정은 GPU를 쓰지 않으므로, 점검 대상이 늘면 수집기와 판정 서버만 늘리면 됩니다. 다만 동시에 질문하는 관리자가 크게 늘거나 여러 기관이 한 장비를 함께 쓰는 경우에는 장비를 추가해야 하며, 이때도 같은 구성을 한 대씩 더하는 방식이라 예산 산정 근거가 바뀌지 않습니다. 따라서 의사결정권자는 첫해 예산을 1대 기준으로 잡고, 운영 지표를 본 뒤 증설 여부를 판단하면 됩니다.

6.4.2 외부 API 단가와 약관 변경, 서비스 중단 위험의 제거

외부 서비스 의존이 운영 안정성에 주는 영향

정보시스템은 장비 하나의 결함으로도 멈출 수 있습니다. 실제로 2023년 11월 17일 행정전산망 장애는 라우터의 케이블 연결 모듈에 있는 포트 일부 이상이 원인이었고, 해킹 흔적은 없었으며, 서비스는 11월 18일 09시에 재개되고 11월 19일 포트 교체 뒤 정상화됐습니다[S04][S05]. 이 사례는 기관이 직접 관리하는 장비에서도 단일 결함이 서비스 중단으로 이어진다는 점을 보여 줍니다. 기관이 통제할 수 없는 외부 AI 서비스와 그 회선까지 운영 경로에 넣으면, 장애 원인을 찾고 복구하는 범위가 기관 밖으로 넓어집니다.

안정성 고시와 서비스 수준 관리

고시 제2026-23호는 행정예고 단계에서 46개 안정성 기준과 서비스 수준 협약(SLA) 체결 의무를 담은 것으로 알려졌습니다[S02]. SLA는 운영 사업자가 약속한 가용성과 복구 수준을 계약으로 묶는 장치이므로, 운영 업무의 일부가 외부 AI 서비스에 기대면 그 서비스의 가용성까지 기관이 책임져야 합니다. 해외 AI 제공사의 서비스 수준은 기관이 운영 사업자와 맺는 SLA에 직접 담기 어렵습니다. 반면 내부 GPU 서버는 다른 정보시스템과 같은 SLA 체계 안에서 관리할 수 있습니다.

외부 API 방식과 내부 sLLM 방식의 위험 비교

아래 표는 운영 독립성 관점에서 두 방식을 비교한 것입니다. 단순 자동화 단계의 점검 도구는 AI를 쓰지 않으므로 이런 위험이 없었지만, 지능화 단계로 넘어가면서 AI 공급 경로가 새로운 운영 위험이 됩니다. 따라서 지능화의 효과를 얻으면서 이 위험을 없애려면 AI 공급 경로를 기관 안으로 들여야 합니다.

위험	외부 LLM API 방식	내부 sLLM 방식(GPU 1대)
단가 변경	제공사가 정하며 연중 바뀔 수 있음	장비 도입 뒤 고정비 중심
약관·데이터 정책 변경	기관 동의 없이 바뀔 수 있음	기관 정책이 그대로 적용
서비스 중단	제공사 장애와 국외 회선 상태에 좌우	기관 서버와 SLA로 관리
모델 교체	제공사가 구형 모델을 종료하면 강제 교체	기관이 검증 뒤 교체 시점 결정

소버린 AI Level 3 구성은 보안 통제, 증적 관리, 비용 예측, 운영 안정성을 한 번의 구성 결정으로 함께 다룹니다. 그래서 차년도 사업 요구사항에 「외부 송신 0건·GPU 1대」 구성을 넣으면, 보안성 검토와 예산 산정이 같은 근거 위에서 진행됩니다.

7장: 보안 취약점 점검의 지능화 — KISA·CVE 권고 접수부터 승인 코드 적용과 재점검까지

보안 취약점 점검은 한국인터넷진흥원(KISA)이 운영하는 보호나라&KrCERT/CC 보안공지와 CVE 발표를 받아 우리 기관 서버 가운데 영향을 받는 서버를 가려내고 고치는 일입니다[S10]. 이 일은 예방점검을 맡는 운영 조직이 함께 수행하는 경우가 많지만, 시작 신호가 기관 밖에서 온다는 점에서 기관이 주기를 정하는 일상점검과 성격이 다릅니다. 권고가 나온 날부터 조치가 끝난 날까지 서버는 이미 공개된 취약점을 가진 채로 운영되므로, 그 기간을 얼마나 짧게 관리하느냐가 보안 운영 수준을 결정합니다. 이 장은 5장에서 다룬 승인 구조를 보안 권고 대응에 적용해, OPENMARU OpsKnow의 실행 계층인 OpsKnow Agent가 권고 접수, OS 패키지와 미들웨어·DBMS의 영향 서버 선별, 승인 코드 기반 패치 적용, 재점검과 증적 기록을 한 흐름으로 잇는 방식을 설명하고, 권고 발표에서 조치 완료까지의 기간을 관리 지표로 삼을지 판단할 근거를 제시합니다.

7.1 KISA 보안공지와 CVE 발표 뒤 영향 서버를 가려내는 패키지 버전 대조

영향 서버 선별은 보안 권고에 적힌 패키지 이름과 취약 버전을 각 서버에 실제로 설치된 패키지 목록과 맞대어 보는 작업입니다. 보호나라는 보안공지를 게시하고 KNVD(취약점 정보포털)는 국내 취약점 정보와 CVE 통합 검색을 제공하므로, 권고의 원천 정보는 이미 공개돼 있습니다[S10]. 어려운 부분은 권고 한 건을 서버 수백 대의 설치 목록과 대조하는 수작업이며, 단순 자동화 단계의 기관은 스크립트로 버전 문자열을 뽑아 엑셀에 모은 뒤 담당자가 눈으로 비교해 조치 대상을 고르는 방식으로 이 작업을 처리합니다. 이 방식에서는 권고가 몰리는 시기에 대조 작업이 밀리고, 비교 기준이 담당자마다 달라 같은 서버가 권고마다 다르게 판정되기도 합니다.

OpsKnow Agent는 이 대조를 결정적 코드로 처리하고, 결과를 다섯 가지 상태 가운데 하나로 판정합니다. 같은 입력에는 언제나 같은 판정이 나오며 언어 모델은 판정 결과를 사람이 읽기 쉬운 문장으로 설명하는 역할만 맡습니다. 담당자의 시간은 대조표를 만드는 작업에서 판정된 목록을 보고 조치 순서를 결정하는 작업으로 옮겨 갑니다. 공공 정보시스템 15,033개가 A1부터 A4까지 등급을 받은 상황에서는 이 조치 순서를 시스템 등급과 함께 정해야 하므로, 판정 목록에 등급 정보가 붙어 있어야 합니다[S01].

7.1.1 보호나라 보안공지와 KNVD 취약점 정보를 설치 패키지 목록과 대조하는 흐름

권고 대조는 권고 정보 확인, 서버별 설치 목록 수집, 저장소 수정본 확인의 세 단계를 거칩니다. 첫 단계에서는 보호나라 보안공지나 KNVD에 올라온 권고에서 CVE 번호, 영향 받는 패키지, 고쳐진 버전을 확인합니다[S10]. 서버별 설치 패키지와 버전은 OS 패키지 관리자(RHEL 계열의 RPM 등)에서 읽어 오고, 고쳐진 버전의 존재 여부는 서버가 연결된 패키지 저장소에서 확인합니다. 세 단계의 결과를 한 서버 단위로 묶으면, 그 서버가 권고에 해당하는지와 지금 고칠 수 있는지가 함께 드러납니다.

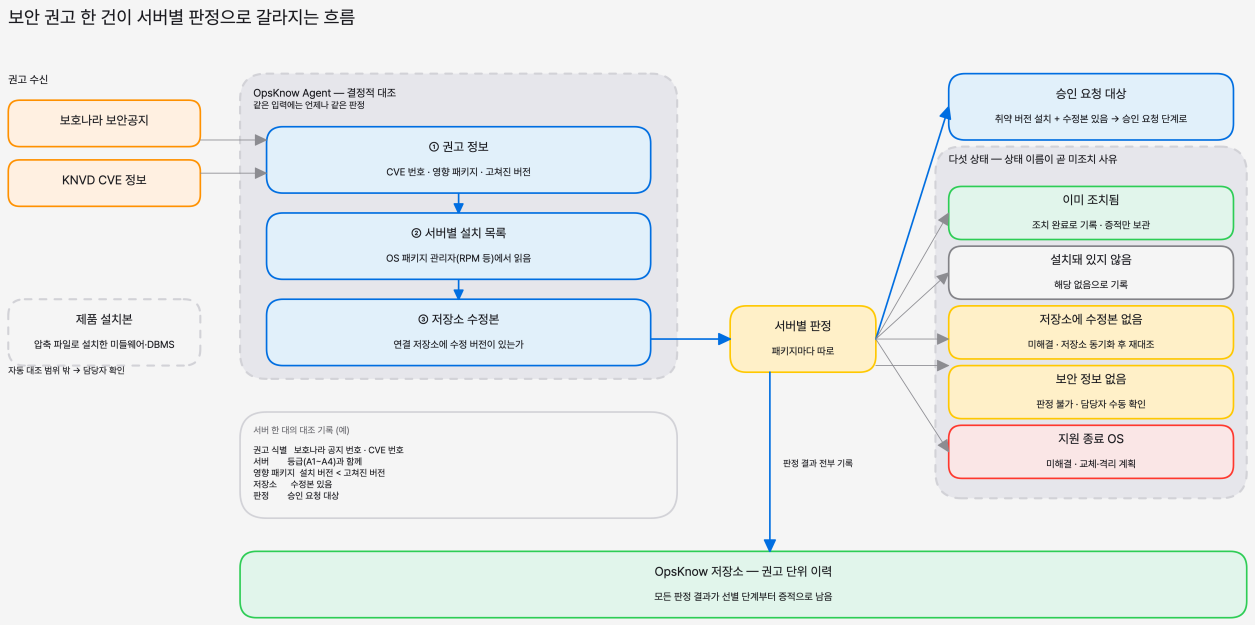


그림. 보안 권고 수신부터 영향 서버 선별과 상태 판정까지의 흐름

대조 결과 목록의 쓰임 대조 결과는 「이 권고에 해당하는 서버가 몇 대이고 각각 어떤 상태인가」라는 질문에 바로 답하는 목록입니다. 예를 들어 권고 한 건에 영향 패키지가 두 개이면 서버마다 두 패키지의 판정이 따로 남습니다. 이렇게 정리된 목록이 있으면 담당자는 서버에 접속하지 않고도 조치 대상과 예외 대상을 구분할 수 있습니다. 미들웨어·DBMS 취약점도 OpsKnow Agent가 점검하고 보고합니다. 압축 파일로 설치한 미들웨어·DBMS 제품 설치본은 패키지 관리자에 등록되지 않으므로, OpsKnow Agent는 설치된 제품이 권고에 해당하는지를 따로 점검해 영향 서버와 조치안을 같은 목록에 올립니다. 제품 설치본 패치의 적용은 사람의 승인을 거친 뒤 운영자가 공급사 지침에 따라 진행합니다.

7.1.2 이미 조치됨부터 지원 종료 OS까지 5가지 상태 구분

다섯 상태는 「조치가 필요한가」와 「지금 조치할 수 있는가」라는 두 질문에 동시에 답하도록 나뉩니다. 단순 자동화 도구는 보통 「취약」과 「양호」 두 값만 내놓는데, 그렇게 두 값으로만 나누면 고칠 수 없는 서버와 판단 근거가 없는 서버가 모두 「양호」 쪽으로 흘러가는 문제가 생깁니다. 반면 다섯 상태로 나누면 각 상태가 다음 조치와 하나씩 이어지므로, 담당자가 상태 이름만 보고도 다음에 할 일을 알 수 있습니다. 상태별 판정 근거와 다음 조치는 다음과 같습니다.

상태	판정 근거	보고 방식	다음 조치
이미 조치됨	설치 버전이 권고의 수정 버전 이상	조치 완료로 기록	없음(증적만 보관)
설치돼 있지 않음	영향 패키지가 서버에 없음	해당 없음으로 기록	없음
저장소에 수정본 없음	취약 버전 설치, 연결 저장소에 수정 버전 미등록	미해결로 기록	저장소 동기화 확인 후 재대조
보안 정보 없음	저장소가 해당 패키지의 보안 권고 정보를 제공하지 않음	판정 불가로 기록	담당자 수동 확인

상태	판정 근거	보고 방식	다음 조치
지원 종료 OS	공급사 지원이 끝나 수정본이 나오지 않는 OS	미해결로 별도 관리	교체·격리 계획 수립

다섯 상태 가운데 사람의 판단이 먼저 필요한 상태는 「저장소에 수정본 없음」, 「보안 정보 없음」, 「지원 종료 OS」의 세 가지입니다. 승인 요청으로 바로 넘어가는 대상은 취약 버전이 설치돼 있고 저장소에 수정본이 있는 서버뿐이며, 이 서버들은 다섯 상태와 별도로 승인 요청 대상으로 모입니다. 이 구분이 감사 대응에서 중요한 이유는 조치하지 않은 서버마다 조치하지 않은 사유가 상태 이름으로 함께 남기 때문입니다. 결과적으로 보안 감사에서 미조치 사유를 서버마다 따로 소명하는 작업이 줄어듭니다.

7.2 승인 코드 기반 OS 패키지 패치 적용과 재점검 결과의 증적화

승인 코드 기반 패치는 서버를 바꾸는 모든 작업에 사람이 발급한 1회용 승인 코드를 요구하는 방식입니다. OpsKnow Agent는 OPENMARU AAS 위에서 실행되는데, AAS가 판정은 코드에, 결정은 사람에게, 설명은 AI에게 나눠 맡기는 실행 구조를 제공하기 때문에 패치 적용도 같은 원칙을 따릅니다. 승인 코드는 짧은 유효 시간 안에 한 번만 쓸 수 있고, 코드 없이 대화창에 「네」라고 답해도 적용은 일어나지 않습니다. 언어 모델이 사용자의 말을 잘못 해석하더라도 그 오류가 서버 변경으로 이어지지 않도록 승인 코드가 막는 구조입니다.

OpsKnow Agent가 승인 코드를 받아 직접 적용하는 패치는 OS 패키지 패치이고, 미들웨어·DBMS 취약점은 점검해 보고한 뒤 사람의 승인을 거쳐 조치합니다. 재부팅·설정 변경 실행·제품 설치본 패치·패키지 삭제와 다운그레이드는 에이전트가 직접 실행하지 않으며, 승인 코드 절차는 패치뿐 아니라 8장 경계표의 결재 6종에도 같은 순서로 적용됩니다. 실행하는 작업에 대해서는 승인, 적용 결과, 재점검 결과가 권고 한 건 아래 하나의 이력으로 OpsKnow에 쌓이므로, 패치 작업을 수행하는 과정이 곧 증적을 만드는 과정이 됩니다. 고시 제2026-23호는 행정예고 단계에서 예방점검과 장애 관리를 포함한 46개 안정성 기준을 담은 것으로 알려져 있습니다 [S02]. 그러한 기준 아래에서는 조치 기록을 작업이 끝난 뒤에 모으는 방식으로 기준 충족을 설명하기 어렵기 때문에, 작업과 기록이 함께 생기는 구조가 필요합니다.

7.2.1 승인과 적용 직전 재확인, 적용, 재점검으로 이어지는 조치 순서

조치 순서는 점검, 승인 요청, 적용 직전 재확인, 적용, 재점검의 다섯 단계이며 단계마다 남는 기록 항목이 정해져 있습니다. 점검 단계에서는 7.1의 대조 결과가 남고, 승인 요청 단계에서는 대상 서버·패키지·적용할 버전이 담긴 요청서가 남습니다. 담당자가 승인 코드를 발급하면 OpsKnow Agent는 적용 직전에 서버 상태를 다시 읽어 점검 때의 설치 버전과 저장소 상태를 비교합니다. 이때 점검 결과와 다른 점이 하나라도 있으면 적용을 멈추고 그 차이를 담당자에게 보고합니다.

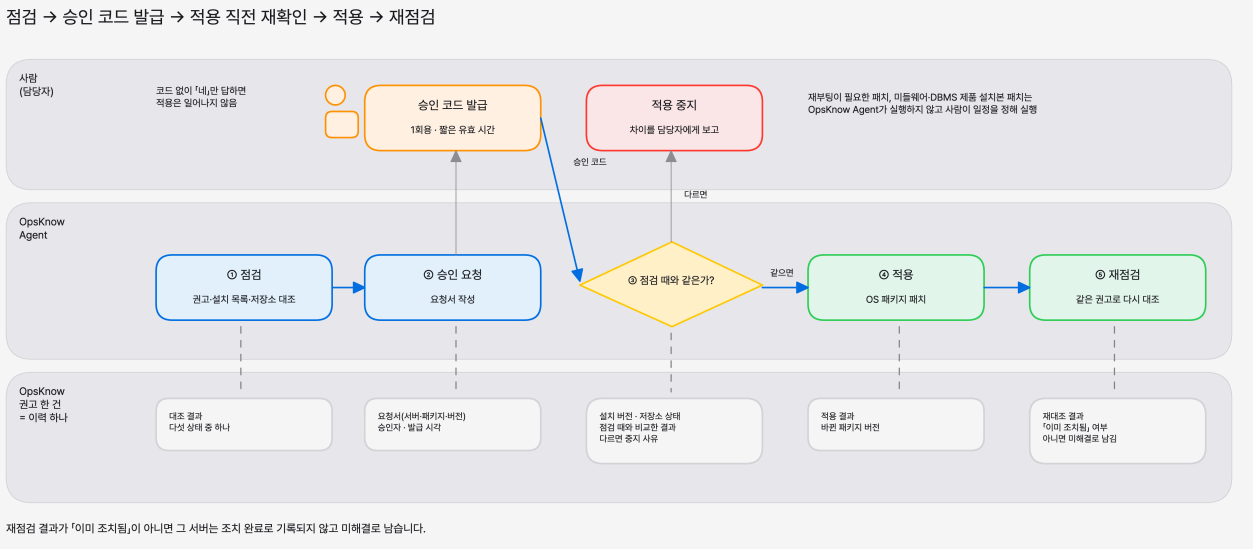


그림. 점검 → 승인 코드 발급 → 적용 직전 재확인 → 적용 → 재점검 순서도

적용과 재점검의 기록 항목 적용 단계에서는 적용 결과와 바뀐 패키지 버전이 기록됩니다. 그다음 재점검 단계에서 같은 권고로 대조를 다시 실행해 상태가 「이미 조치됨」으로 바뀌었는지 확인하고, 그 결과를 같은 이력에 붙입니다. 이렇게 쌓인 이력에서는 권고 한 건에 대해 「누가 언제 승인했고, 무엇이 바뀌었고, 바뀐 뒤 무엇이 확인됐는가」가 한 묶음으로 조회됩니다. 재점검 결과가 「이미 조치됨」이 아니면 그 서버는 조치 완료로 기록되지 않고 미해결 상태로 남습니다.

7.2.2 권고별 조치 현황표로 바로 산출되는 보안 감사 증적

권고별 조치 현황표는 OpsKnow에 쌓인 이력을 권고 단위로 묶어 출력한 감사 제출 자료입니다. 보안 감사의 질문은 대개 「이 권고가 나온 뒤 해당 서버를 언제 어떻게 조치했는가」이므로, 서버 단위 점검표보다 권고 단위 현황표가 그 질문에 더 바로 답합니다. 대조·승인·적용·재점검 기록이 권고 한 건 아래 이미 묶여 있으므로, 현황표는 그 기록을 권고 단위로 모아 만들 수 있습니다. 다만 현황표의 각 칸은 실행 기록에서 직접 나온 값이어야 증적으로 인정받을 수 있으며, 사람이 옮겨 적은 값은 원천 기록과 대조하는 확인 작업이 다시 필요합니다.

열	담는 내용	원천 기록
권고 식별	보호나라 공지 번호, CVE 번호	권고 수신 기록
영향 서버 수	상태별 서버 수(다섯 상태와 승인 요청 대상)	대조 결과
승인 이력	승인자, 승인 시각, 대상 서버	승인 코드 발급 기록
적용 결과	적용 결과, 바뀐 패키지 버전	적용 실행 기록
재점검 결과	조치 후 상태	재점검 대조 결과
미해결 사유	수정본 없음, 지원 종료 OS, 판정 불가	상태 판정 기록

현황표의 마지막 열은 감사 대응에서 가장 자주 쓰입니다. 예를 들어 감사관이 미조치 서버를 지적하면 담당자는 그 서버가 「저장소에 수정본 없음」인지 「지원 종료 OS」인지를 표에서 바로 보여 줄 수 있습니다. 예방점검 결과

와 보안 조치 이력이 같은 OpsKnow 저장소에 있으므로, CPU·메모리·디스크 상태처럼 일상점검에서 본 서버 상태와 보안 조치 이력을 한 화면에 함께 제시할 수도 있습니다[S07]. 보안 감사 대응은 흩어진 자료를 모으는 작업에서 이미 쌓인 기록을 뽑아내는 작업으로 바뀝니다.

7.3 지원 종료 OS와 수정본 없는 취약점을 「최신」으로 오판정하지 않는 별도 관리

오판정 방지는 판정 근거가 없는 서버를 「양호」나 「최신」으로 보고하지 않는 원칙입니다. 보안 점검에서 가장 위험한 결과는 취약한 서버를 안전하다고 보고하는 경우이며, 이런 오판정은 대개 판정 근거가 비어 있을 때 생깁니다. 예를 들어 지원 종료 OS에서는 패키지 관리자가 「설치 가능한 업데이트 없음」이라고 답하는데, 이 응답을 「최신 상태」로 읽으면 수정본이 더 나오지 않는 서버가 안전한 서버로 집계됩니다. OpsKnow Agent는 이런 서버를 별도 상태로 판정하고, 미해결 취약점 목록에 따로 모읍니다.

미해결 취약점 목록은 보안 운영 자료이면서 동시에 예산 자료입니다. 지원 종료 OS와 수정본 없는 패키지는 운영 조직이 패치로 해결할 수 없고 교체나 고도화 사업으로만 해결되기 때문입니다. 공공 정보시스템 등급이 국민이 받는 서비스의 중요도를 기준으로 다시 매겨진 상황에서는, 이 목록에 시스템 등급을 함께 붙이면 교체 순서를 정하는 근거가 됩니다[S01]. 판정 원칙이 지켜져야 목록이 정확해지고, 목록이 정확해야 예산 요구의 근거가 될 수 있습니다.

7.3.1 확실하지 않으면 최신으로 보고하지 않는 판정 원칙

판정 원칙은 「확인된 근거가 있을 때만 조치 완료나 최신으로 보고한다」는 한 줄로 정리됩니다. 다섯 상태 가운데 「보안 정보 없음」과 「지원 종료 OS」를 독립 상태로 둔 이유도 이 원칙에 있습니다. 판정 근거가 없으면 판정을 미루고 그 사실을 보고하며, 근거가 없는 상태를 좋은 쪽으로 해석하지 않습니다. 언어 모델은 판정 결과를 설명하는 문장을 만들 뿐이고, 판정 자체를 바꾸는 권한은 갖지 않습니다.

불확실 상태의 보고 방식 불확실 상태는 보고서에서 조치 완료 집계와 섞이지 않고 별도 항목으로 표시됩니다. 예를 들어 분기 보고서의 보안 항목을 조치 완료, 승인 요청 대상, 미해결(수정본 없음), 미해결(지원 종료 OS), 판정 불가의 다섯 칸으로 나누면, 의사결정권자는 「판정 불가」 칸의 크기만 보고도 보안 정보가 부족한 서버 규모를 알 수 있습니다. 이 판정과 설명은 30B 수준 내부 sLLM을 GPU 1대로 폐쇄망(외부 연결 없음) 안에서 실행하는 소버린 AI Level 3 구성에서 이뤄지므로, 취약점 목록과 서버 구성 정보가 기관 밖으로 나가지 않습니다. 국가정보원 N2SF 가이드라인이 업무 정보를 기밀·민감·공개 등급으로 나눠 통제하도록 한 점에 비추어 보면, 미조치 취약점 목록은 외부 AI 서비스로 보내기 어려운 정보에 해당합니다[S08][S09].

7.3.2 교체·고도화 예산의 근거 목록이 되는 미해결 취약점

미해결 취약점 목록은 패치로 해결되지 않는 위험을 서버와 권고 단위로 정리한 자료이므로, 차년도 교체·고도화 예산 요구의 근거로 쓸 수 있습니다. 예산 요구서에 「노후 OS 교체 필요」라고만 쓰면 우선순위를 설명하기 어렵지만, 권고 번호와 미조치 기간이 붙은 목록이 있으면 같은 요구를 공개 권고와 노출 기간으로 설명할 수 있습니다[S10]. 목록의 정확성은 7.3.1의 판정 원칙에 달려 있습니다. 지원 종료 OS가 「최신」으로 집계되면 그 서버는 예산 근거 목록에서도 빠집니다.

목록 항목	담는 내용	예산 요구서에서 쓰는 방법
대상 서버	서버 이름, 소속 정보시스템, 시스템 등급	등급이 높은 시스템부터 교체 순서 제시

목록 항목	담는 내용	예산 요구서에서 쓰는 방법
미해결 사유	지원 종료 OS 또는 저장소에 수정본 없음	패치로 해결할 수 없음을 소명
관련 권고	보호나라 공지 번호, CVE 번호	위험의 공개 근거 제시
미조치 기간	권고 발표일부터 현재까지	위험 노출 기간 제시
임시 통제	접근 제한·격리 등 사람이 결정한 대응	교체 전까지의 관리 방안 제시

이 목록은 시스템 등급과 결합될 때 쓰임이 커집니다. A1·A2 등급은 국민이 받는 서비스의 중요도가 높은 시스템에 매겨지므로[S01], 이런 시스템에 지원 종료 OS가 남아 있으면 장애 위험과 보안 위험이 겹쳐 교체 우선순위가 올라갑니다. 이처럼 미해결 취약점 목록은 보안 담당자의 업무 자료로 끝나지 않고, 의사결정권자가 사업 우선순위를 정하는 자료로 이어집니다.

7.4 분기 일괄 패치에서 권고 단위 처리로 — 공격 가능 기간을 줄이는 패치 일정

권고 단위 처리는 패치를 분기마다 모아서 적용하는 대신, 권고가 나올 때마다 영향 서버를 선별하고 승인된 서버부터 적용하는 일정 운영 방식입니다. 분기 일괄 패치는 작업 일정을 미리 알 수 있다는 장점이 있습니다. 하지만 권고가 분기 초에 나오면 다음 일괄 작업까지 서버는 알려진 취약점을 가진 채 운영되고, 그 기간 전체가 공격 가능 기간이 됩니다. 권고 단위 처리가 현실적인 선택지가 된 것은 7.1의 선별과 7.2의 승인·재점검이 자동으로 이어지면서 권고 한 건을 처리하는 데 드는 수작업이 크게 줄었기 때문입니다.

모든 패치를 권고 단위로 바꿀 수는 없습니다. 재부팅이 필요한 커널 패치와 미들웨어·DBMS 제품 설치본 패치는 서비스 중단 일정과 맞물리므로 사람이 작업 시간대를 정해야 합니다. 그래서 권고 단위 처리는 기간을 재는 지표와, 에이전트 작업과 사람 작업을 나누는 운영 절차를 함께 갖춰야 운영할 수 있습니다. 보호나라 보안공지의 게시 시각은 누구나 확인할 수 있는 공개 기록이므로, 기간 측정의 시작 시각으로 쓰기에 적합합니다[S10].

7.4.1 권고 발표에서 조치 완료까지 걸린 기간을 의사결정 지표로 보는 관점

권고 조치 기간은 보안 권고가 공개된 시각부터 영향 서버 전체가 「이미 조치됨」 또는 사유가 기록된 미해결 상태로 확정된 시각까지 걸린 시간입니다. 점검 횟수나 패치 건수는 공격 가능 기간을 나타내지 못하지만, 이 지표는 그 기간을 직접 나타내므로 의사결정권자가 보안 운영 수준을 하나의 숫자로 볼 수 있게 합니다. 시작 시각과 종료 시각이 모두 실행 기록에서 나오기 때문에 측정에 사람의 주관이 들어가지 않으며, 같은 지표를 기관 안의 정보시스템끼리 그리고 분기끼리 비교할 수 있습니다. 권고 조치 기간과 함께 볼 보조 지표는 다음과 같습니다.

지표	정의	시작 시각 원천	종료 시각 원천
권고 조치 기간	권고 공개부터 영향 서버 전체 상태 확정까지	보호나라 공지·CVE 공개 시각	마지막 재점검 기록 시각
선별 소요 시간	권고 공개부터 영향 서버 목록 확정까지	보호나라 공지·CVE 공개 시각	대조 결과 기록 시각
승인 대기 시간	승인 요청부터 승인 코드 발급까지	승인 요청 기록 시각	승인 코드 발급 시각

지표	정의	시작 시각 원천	종료 시각 원천
미해결 잔존 기간	미해결 상태로 남은 서버의 노출 기간	보호나라 공지·CVE 공개 시각	현재 또는 교체 완료 시각

네 지표를 함께 보면 기간이 어느 단계에서 길어지는지 알 수 있습니다. 예를 들어 선별 소요 시간은 짧은데 승인 대기 시간이 길면 먼저 손볼 대상은 승인 절차이고, 미해결 잔존 기간이 길면 7.3.2의 교체 예산 문제로 넘어갑니다. 「권고 발표에서 조치 완료까지의 기간을 관리 지표로 삼을 것인가」라는 판단에서는, 측정 원천이 이미 OpsKnow에 쌓이므로 지표를 새로 도입하는 부담이 작다는 점이 근거가 됩니다. 행정예고된 안정성 고시가 서비스 수준 협약(SLA) 체결을 의무화한 것으로 알려진 만큼, 권고 조치 기간은 위탁 운영 사업자와 맺는 협약의 보고 항목으로 넣기에도 적합한 지표입니다[S02].

7.4.2 재부팅과 제품 설치본 패치를 사람 작업으로 나누는 운영 절차

작업 분리 절차는 서비스 영향이 없는 OS 패키지 교체는 OpsKnow Agent가 승인 코드를 받아 실행하고, 서비스 중단이 따르는 작업은 사람이 일정을 정해 실행하도록 나누는 규칙입니다. 이 구분은 결정 책임을 사람에게 두기 위한 설계 선택입니다. 재부팅과 제품 설치본 패치는 서비스 중단 시각, 사용자 공지, 원복 계획을 함께 정해야 하는 작업이어서 사람의 판단이 필요합니다. OpsKnow Agent는 이런 작업을 직접 실행하지 않으므로, 담당자가 대상 서버와 관련 권고를 보고 작업 일정을 정합니다.

작업	OpsKnow Agent	사람
권고 수신과 영향 서버 선별	실행	결과 확인
OS 패키지 패치 적용	승인 코드를 받아 실행	승인 코드 발급
적용 뒤 재점검과 이력 기록	실행	결과 확인
재부팅이 필요한 패치	실행하지 않음	작업 시간대 결정과 실행
미들웨어·DBMS 취약점 점검	권고 해당 여부 점검과 보고, 조치안 제시	결과 확인과 조치 승인
미들웨어·DBMS 제품 설치본 패치	직접 실행하지 않음	승인 뒤 패치 실행
패키지 삭제·다운그레이드	실행하지 않음	필요 시 직접 판단
지원 종료 OS 대응	미해결 목록 관리	교체·격리 결정

사람 작업으로 넘어간 항목도 같은 권고 이력 아래 두는 것이 운영 절차의 핵심입니다. 사람이 재부팅이나 제품 패치를 마친 뒤 같은 권고로 재점검을 실행하면 상태가 확정되므로, 권고 조치 기간의 종료 시각이 사람 작업에서도 같은 기준으로 기록됩니다. 자동으로 처리되는 작업과 결재가 필요한 작업의 경계가 표 한 장으로 공개돼 있기 때문에, 감사에서도 그 경계를 그대로 설명할 수 있습니다. 보안 취약점 점검의 지능화는 결국 사람이 결정해야 하는 항목을 골라 사람 앞에 올리고, 나머지 선별·적용·재점검·기록을 코드가 빠짐없이 처리하도록 하는 운영 구조입니다.

8장: 예방점검을 넘어 장애 대응과 정기 보고까지 — 한 저장소에 쌓는 운영 기록과 자동 처리·결재의 경계

1장에서 다룬 고시 제2026-23호는 예방점검만 요구하는 규정이 아닙니다. 행정예고된 안에는 예방점검과 장애 관리를 묶은 46개 안정성 기준, 장애관리 기본계획과 연간 시행계획, 서비스 수준 협약(SLA) 체결 의무가 함께 들어 있고, 중요 시스템 장애는 행정안전부 디지털안전상황실에 즉시 보고하도록 했습니다[S02]. 기관이 남겨야 하는 증적은 점검표 한 장으로 끝나지 않고, 장애가 났을 때의 원인과 조치, 그 뒤의 재발방지, 그리고 주기별 운영 보고까지 이어집니다. 2026년 8월 등급이 확정된 정보시스템이 15,033개이므로[S01], 이 기록을 담당자 개인의 파일과 메일함에 흩어 두면 감사 때 같은 사실을 여러 곳에서 다시 모아야 합니다.

OPENMARU OpsKnow는 이 기록들을 한 저장소에 쌓는 운영 지식 저장소입니다. Incident 케이스, 예방점검 결과, 일·주·월·분기·연 단위 정기 운영 보고가 같은 시스템 식별자로 묶이고, 장애 이력은 다음 예방점검의 판정 근거로 다시 쓰입니다. 또한 OpsKnow Agent가 판정하고 처리하는 일 가운데 사람 결재가 필요한 조치와 자동으로 끝나는 조치를 미리 나누어 두었기 때문에, 위탁 운영 계약에서 운영자 권한을 정할 때 이 경계를 그대로 옮겨 쓸 수 있습니다. 이 장에서는 세 자산의 연결 구조, Incident 기록의 필드 구성, 자동 처리와 결재의 경계, 정기 보고의 누적 방식을 차례로 살펴보고, 기관이 차년도 사업 요구사항에 무엇을 적어야 하는지를 정리했습니다.

8.1 Incident 케이스와 예방점검 결과, 정기 운영 보고를 한 저장소에서 연결하는 OpsKnow

예방점검 결과와 장애 기록이 서로 다른 시스템에 남으면, 장애가 난 뒤에 직전 점검에서 무엇이 정상으로 판정됐는지 확인하려면 두 기록을 사람이 맞춰 봐야 합니다. 행정안전부가 2024년 10월 마련한 예방점검체계도 장애의 예방, 대응, 사후관리를 한 흐름으로 점검·관리하는 것을 목적으로 둡니다[S06]. 따라서 운영 기록 저장소는 점검 결과만 보관하는 창고로는 부족하고, 장애 기록과 보고서가 같은 대상과 같은 시점을 가리키도록 설계돼야 합니다. OpsKnow는 Incident 케이스, 예방점검 결과, 정기 운영 보고 세 가지를 운영 자산으로 정의하고, 세 자산이 서로를 참조하도록 저장합니다.

이 구조에서 지표의 수집은 OPENMARU APM이 맡습니다. OPENMARU APM은 OS, 미들웨어, DBMS의 성능 지표와 트랜잭션을 수집하는 제품이며, OpsKnow에 쌓이는 판정 결과의 원자료가 여기에서 나오므로 이 장의 연결 구조와 직접 관련됩니다. OpsKnow Agent는 APM이 모은 지표를 읽어 예방점검 항목을 판정하고, 그 결과와 근거 지표를 OpsKnow에 기록합니다. 담당자가 바뀌어도 어떤 시스템이 어떤 상태였고 누가 무엇을 조치했는지는 저장소에 남으므로, 운영 지식이 사람과 함께 빠져나가지 않습니다.

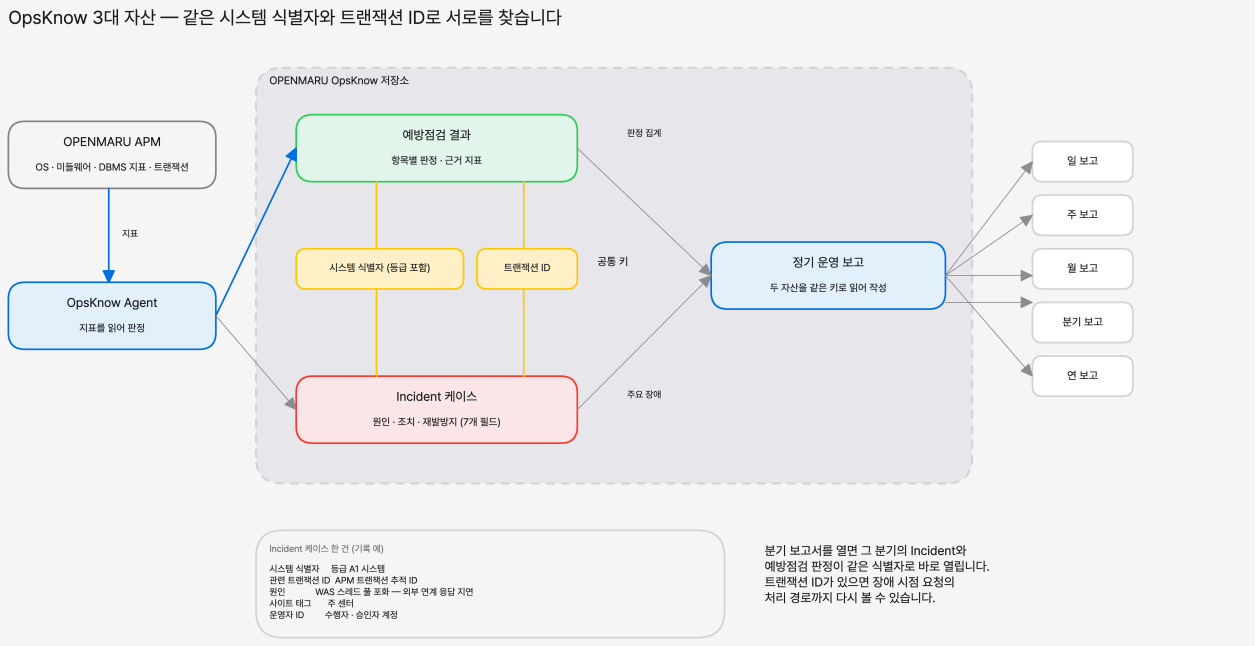


그림. OpsKnow 3대 자산(Incident 케이스 · 예방점검 결과 · 정기 운영 보고)의 연결 구조

8.1.1 같은 시스템과 트랜잭션 ID로 서로 참조되는 3대 자산

시스템 식별자 중심의 연결 키 세 자산을 묶는 첫 번째 키는 시스템 식별자입니다. 예방점검 결과는 어떤 정보시스템의 어떤 서버에서 어떤 항목을 판정했는지를 남기고, Incident 케이스는 장애가 난 시스템과 사이트 태그를 남기며, 정기 운영 보고는 같은 식별자로 기간별 결과를 모읍니다. 그래서 등급이 A1로 지정된 시스템의 분기 보고서를 열면, 그 분기에 있었던 Incident와 예방점검 판정을 같은 식별자로 바로 찾을 수 있습니다. 행정안전부가 확정한 등급별 시스템 수는 A1 73개, A2 157개, A3 671개, A4 14,132개이므로[S01], 등급을 식별자에 함께 붙여 두면 기관이 상위 등급 시스템만 골라 증적을 모으는 일도 단순해집니다.

트랜잭션 ID로 장애와 지표를 잇는 방식 두 번째 키는 트랜잭션 ID입니다. 미들웨어나 DBMS 성능 저하로 장애가 생기면 원인은 특정 요청의 처리 경로에 남는데, OPENMARU APM은 요청 단위로 트랜잭션을 추적하므로 Incident 기록에 그 ID를 적어 두면 장애 시점 요청의 처리 경로와 응답 시간을 다시 열어 볼 수 있습니다. 반면 트랜잭션 ID 없이 「오후 2시경 응답 지연」이라고만 적은 기록은, 원자료 보관 기간이 지나면 어떤 요청이 느렸는지 증명할 수 없습니다. 아래 표는 세 자산이 공통으로 쓰는 연결 키와 각 자산에서의 쓰임을 정리한 것입니다.

연결 키	예방점검 결과에서의 쓰임	Incident 케이스에서의 쓰임	정기 운영 보고에서의 쓰임
시스템 식별자(등급 포함)	점검 대상 시스템·서버 지정	장애 발생 시스템 지정	시스템별·등급별 집계
트랜잭션 ID	성능 판정의 근거 요청	장애 원인 요청 추적	주요 장애의 근거 링크
사이트 태그	점검 대상 센터·위치 구분	장애 발생 위치 구분	사이트별 묶음 보고
운영자 ID	판정 확인자	조치 수행자·승인자	기간별 조치 이력

연결 키를 사업 요구사항에 적는 방법 연결 키는 저장소를 들이는 시점에 정해야 합니다. 운영 중에 키 체계를 바꾸면 이미 쌓인 기록과 새 기록이 서로를 찾지 못하기 때문입니다. 예를 들어 차년도 운영 사업의 과업지시서에

「예방점검 결과, 장애 기록, 정기 보고는 시스템 식별자와 트랜잭션 ID로 상호 조회가 가능해야 한다」는 문장을 넣어 두면, 수행사가 바뀌어도 기록의 연결 방식이 유지됩니다. 연결 키는 기술 설계이면서 동시에 감사 대응을 위한 계약 조건입니다.

8.1.2 장애 이력이 다음 예방점검 항목과 판정 근거로 돌아가는 순환

장애 기록이 점검 기준으로 바뀌는 흐름 예방점검의 일상점검은 CPU·메모리·디스크 상태, 메인 화면 접속 여부와 접속 시간, 인증서 유효기간과 도메인 만료일 같은 항목을 봅니다[S07]. 이 항목들은 모든 시스템에 공통으로 적용되는 기준이지만, 실제 장애는 시스템마다 다른 원인에서 반복됩니다. 그래서 OpsKnow는 Incident의 원인과 재발방지 필드를 다음 예방점검 판정에 참고 근거로 연결합니다. 예를 들어 DB 커넥션 풀 고갈로 장애가 났던 시스템이라면, 다음 점검에서 커넥션 사용률을 판정할 때 과거 장애 기록을 함께 참조할 수 있어 같은 징후를 더 일찍 알아볼 수 있습니다.

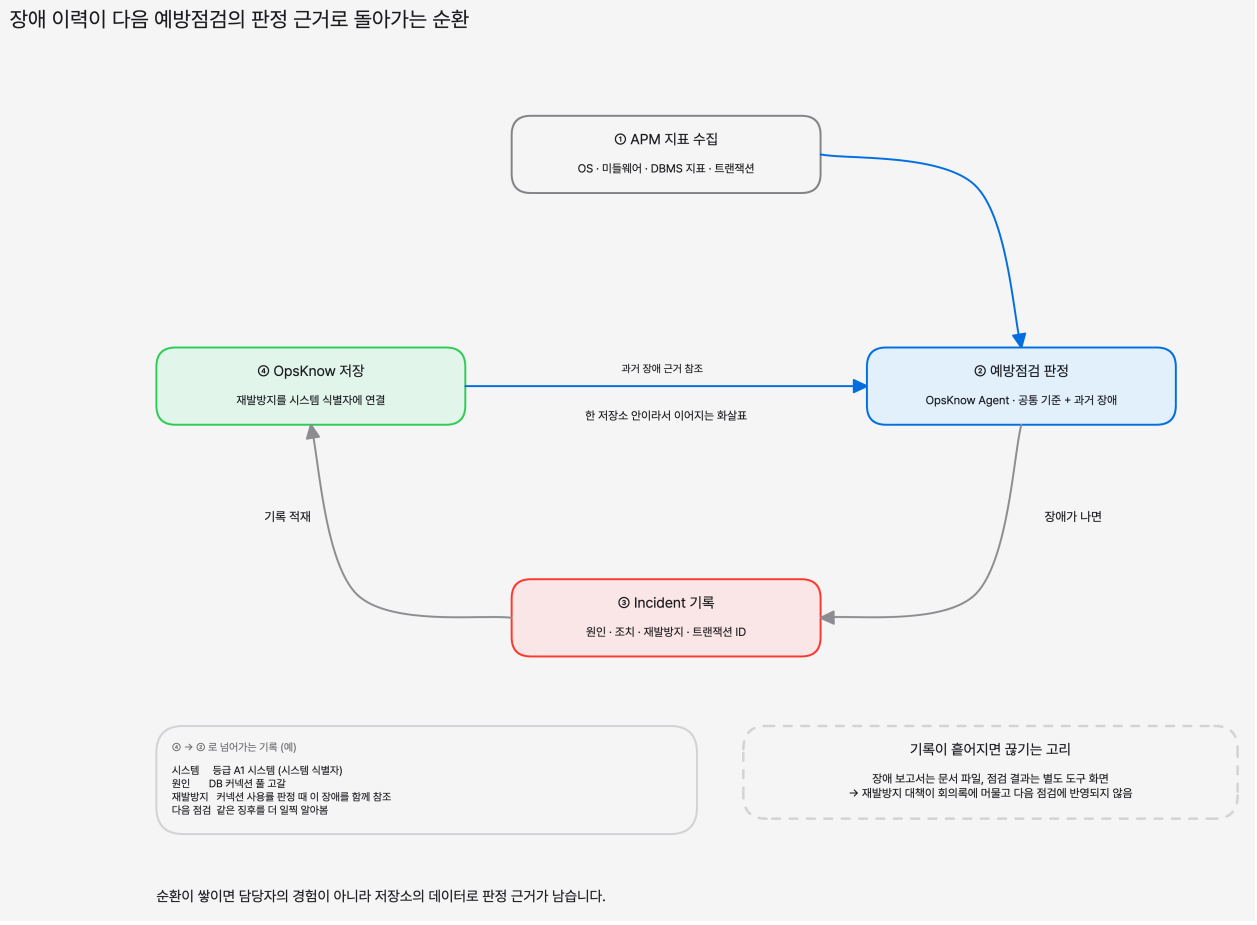


그림. 장애 이력이 다음 예방점검의 판정 근거로 돌아가는 순환

순환이 끊기는 운영 방식 이 순환은 기록이 한 저장소에 있어야만 성립합니다. 장애 보고서가 문서 파일로, 점검 결과가 별도 도구의 화면으로만 남으면 재발방지 대책은 회의록에 적힌 채 다음 점검에 반영되지 않습니다. 2023년 11월 행정전산망 장애도 처음에는 L4 스위치가 원인으로 지목됐다가, 정부 발표에서 라우터의 케이블 연결 모듈에 있는 포트 일부 이상으로 결론이 났습니다[S04][S05]. 따라서 원인이 확정되기까지 오간 판단과 최종 결론이 같은 기록에 남아 있어야, 다음 점검에서 어떤 장비 항목을 더 자세히 볼지 정할 수 있습니다.

순환이 남기는 운영 지식 순환이 몇 년 이어지면 OpsKnow에는 시스템별로 어떤 징후가 어떤 장애로 이어졌는지에 대한 기록이 쌓입니다. 이 기록은 특정 담당자의 경험이 아니라 저장소의 데이터이므로, 신규 담당자도 같

은 근거로 판정을 이해할 수 있습니다. 예방점검이 매번 같은 항목을 반복 확인하는 단순 자동화에 머물지 않고, 과거 장애를 근거로 판정을 조정하는 지능화 단계로 넘어가는 데 필요한 재료가 바로 이 순환 기록입니다.

8.2 원인부터 자동 RCA 시간까지 7개 필드로 남기는 Incident 기록

장애 기록의 품질은 필드 구성에서 정해집니다. 자유 서술 한 칸에 경위를 적는 방식은 작성자마다 내용이 달라 나중에 같은 유형의 장애를 검색하기 어렵습니다. 행정예고된 안정성 고시가 장애관리 기본계획과 연간 시행계획을 요구하고 중요 장애를 디지털안전상황실에 즉시 보고하도록 한 만큼[S02], 기관은 장애 한 건의 원인과 조치를 정해진 형식으로 남겨야 합니다. 그래서 OpsKnow는 Incident를 원인, 조치, 재발방지, 관련 트랜잭션 ID, 사이트 태그, 운영자 ID, 자동 RCA 시간의 7개 필드로 기록합니다.

여기서 RCA는 장애의 직접 원인과 함께 그 원인을 만든 조건까지 찾아내는 분석을 말합니다. OpsKnow Agent가 APM 지표와 과거 Incident를 대조해 원인 후보를 자동으로 좁히고, 그 분석에 걸린 시간을 자동 RCA 시간 필드에 남깁니다. 이 필드는 장애 대응 속도를 기간별로 비교하는 기준이 되므로, 도입 효과를 측정할 때 그대로 측정 항목으로 쓸 수 있습니다.

8.2.1 원인·조치·재발방지에서 운영자 ID까지 Incident 필드 구성

Incident 7개 필드의 정의 7개 필드는 장애 한 건을 다시 열었을 때 답해야 할 질문에 하나씩 대응합니다. 원인·조치·재발방지는 무엇이 일어났고 어떻게 대응했으며 같은 일이 생기지 않도록 무엇을 바꿨는지를 남기고, 나머지 네 필드는 그 서술을 원자료와 사람에게 연결합니다. 아래 표는 각 필드의 정의와 기록 예를 정리한 것입니다.

필드	정의	기록 예
원인	장애를 일으킨 직접 원인과 근본 원인	WAS 스레드 풀 포화, 원인은 외부 연계 응답 지연
조치	장애를 해소하려고 실제로 한 일	결재 후 WAS 재시작, 연계 타임아웃 값 확인
재발방지	같은 장애를 막으려고 바꾼 기준·설정·절차	연계 응답 시간 임계치 추가, 다음 점검 항목 반영
관련 트랜잭션 ID	원인을 보여 주는 요청의 추적 ID	APM 트랜잭션 추적 ID
사이트 태그	장애가 난 센터·위치·서비스 구분	주 센터 / 재해복구 센터
운영자 ID	조치를 수행하고 승인한 사람	수행자·승인자 계정
자동 RCA 시간	원인 후보를 좁히는 데 걸린 시간	분석 시작·종료 시각

필드가 감사 대응에 쓰이는 방식 감사에서 묻는 내용은 대체로 세 가지입니다. 장애가 언제 무엇 때문에 났는지, 누가 어떤 권한으로 조치했는지, 같은 장애를 막기 위해 무엇을 바꿨는지입니다. 7개 필드는 이 세 질문에 각각 대응하므로, 기관은 장애 경위서를 새로 쓰지 않고 Incident 기록을 그대로 증적으로 제출할 수 있습니다. 운영자 ID 필드는 위탁 운영 환경에서 조치를 한 사람이 수행사 인력인지 기관 담당자인지를 구분해 주므로, SLA 위반 여부를 판단할 때 책임 소재를 가르는 근거가 됩니다[S02].

기록 작성 부담을 줄이는 자동 입력 필드가 많으면 기록이 부실해진다는 우려가 있습니다. 다만 OpsKnow에서는 트랜잭션 ID, 사이트 태그, 자동 RCA 시간처럼 시스템이 아는 값은 OpsKnow Agent가 장애 판단 시점에

자동으로 채우고, 담당자는 원인 확정과 재발방지처럼 판단이 필요한 필드만 검토합니다. 그 결과 사람이 적어야 하는 칸은 사람만 판단할 수 있는 내용으로 줄어듭니다.

8.2.2 장애 이력 보관 기간을 사업 요구사항에 적어야 하는 이유

장애 이력 보관 기간과 재발 판단의 관계 장애 이력의 가치는 보관 기간이 길수록 커집니다. 계절성 업무가 있는 시스템은 연말 정산이나 신학기 접수처럼 해마다 같은 시기에 몰리는 부하에서 장애가 반복되는데, 보관 기간이 짧으면 작년의 같은 시기 기록이 이미 지워져 있습니다. 예를 들어 비교 대상인 A사 자료는 이력 보관 기간에 제한을 둡니다. 이런 방식에서는 한 해 전 같은 시기의 장애와 올해의 징후를 대조하는 일이 보관 기간의 경계에 걸리기 쉽습니다.

보관 기간과 계획 문서의 연결 행정예고된 안정성 고시는 장애관리 기본계획과 연간 시행계획을 요구합니다[S02]. 기본계획은 여러 해에 걸친 장애 추이를 근거로 세워야 하므로, 이력이 짧은 기간만 남아 있으면 기본계획의 근거 자료를 매번 다른 곳에서 다시 모아야 합니다. 따라서 장애 이력 보관 기간은 계획 수립 주기에 맞춘 운영 요건으로 정하는 것이 맞습니다.

사업 요구사항 문안 예 차년도 운영 사업이나 운영 도구 도입 사업의 요구사항에는 다음과 같은 문장을 넣을 수 있습니다.

- 장애 기록은 원인·조치·재발방지·관련 트랜잭션 ID·사이트 태그·운영자 ID·분석 소요 시간을 필드로 구분해 저장한다.
- 장애 기록과 예방점검 결과는 장애관리 기본계획의 수립 주기 이상 보관하며, 보관 기간 안에서 시스템 식별자로 조회할 수 있어야 한다.
- 수행사 교체 시 장애 기록 전체를 같은 필드 구조로 이관한다.

이 세 문장은 기록의 형태와 보관 기간만 요구하므로, 특정 제품에 유리한 조건을 두지 않으면서도 증적의 연속성을 보장합니다.

보관 기간 요구사항은 수행사가 바뀌는 시점에 특히 효과를 냅니다. 장애 이력이 수행사의 도구 안에만 있으면 계약이 끝날 때 기록도 함께 빠져나가고, 새 수행사는 같은 시스템의 과거 장애를 모른 채 첫해를 시작하게 됩니다. 기록의 필드 구조와 보관 위치를 기관 쪽에 두면, 담당 인력과 수행사가 바뀌어도 다음 예방점검의 판정 근거는 그대로 이어집니다.

8.3 알림과 적재는 자동, 미들웨어(WAS) 재시작과 DB 풀 크기 변경은 결재로 나눈 경계 매트릭스

AI 에이전트가 운영에 들어오면 가장 먼저 정해야 할 것은 에이전트가 혼자 끝낼 수 있는 일과 사람이 결재해야 하는 일의 경계입니다. 행정예고된 안정성 고시는 민간 클라우드와 위탁 운영을 포함한 모든 운영 형태에 SLA 체결을 요구하므로[S02], 누가 어떤 조치를 할 권한이 있는지는 계약서에 적혀야 합니다. OpsKnow Agent는 알림 발송, OpsKnow 적재, 임계치 보고서 생성 세 가지를 자동으로 처리하고, 서비스 상태를 바꾸는 여섯 가지 조치는 사람 결재 없이 실행하지 않습니다. 이 구분은 OpsKnow Agent가 동작하는 AAS에서 승인 절차로 강제되며, AAS는 에이전트의 실행 권한과 승인 기록을 관리하는 실행 환경이라 이 경계의 실제 적용 수단이 됩니다.

경계를 이렇게 나눈 기준은 되돌릴 수 있는지와 서비스에 영향을 주는지입니다. 알림을 보내거나 기록을 쌓는 일은 잘못되더라도 서비스가 멈추지 않지만, WAS 재시작이나 DB 풀 크기 변경은 처리 중인 요청을 끊거나 성능을 바꿉니다. 그래서 OpsKnow Agent는 앞의 일은 즉시 처리하고, 뒤의 일은 근거를 붙인 결재 요청까지만 만든 뒤 승인 코드를 받아야 실행합니다.

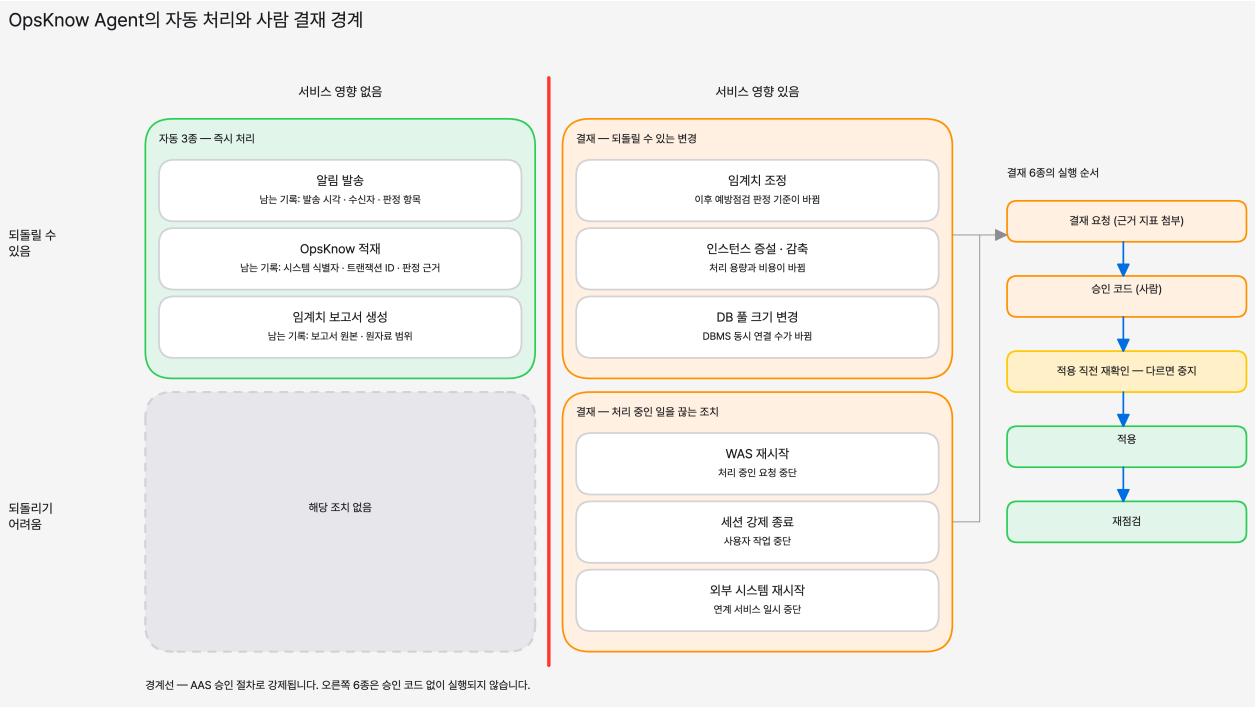


그림. OpsKnow Agent의 자동 처리와 사람 결재 경계 매트릭스

8.3.1 자동으로 처리되는 알림 발송과 OpsKnow 적재, 임계치 보고서 생성

자동 처리 3종의 정의 자동 처리 대상은 서비스 상태를 바꾸지 않고 정보만 만들거나 옮기는 일입니다. 판정 결과를 담당자에게 알리고, 판정과 근거를 저장소에 남기고, 임계치를 넘은 항목을 보고서로 묶는 일이 여기에 속합니다. 아래 표는 세 가지 자동 처리의 내용과 남는 기록을 정리한 것입니다.

자동 처리	내용	남는 기록
알림 발송	예방점검 판정이 주의·위험이면 담당자에게 즉시 알림	발송 시각, 수신자, 판정 항목
OpsKnow 적재	판정 결과·근거 지표·Incident 초안을 저장소에 기록	시스템 식별자, 트랜잭션 ID, 판정 근거
임계치 보고서 생성	임계치를 넘은 항목을 시스템·기간별로 묶어 보고서 작성	보고서 원본과 생성에 쓴 원자료 범위

자동 처리가 증적의 누락을 막는 방식 이 세 가지를 자동으로 두는 이유는 기록 누락을 막기 위해서입니다. 사람이 판정 결과를 옮겨 적는 방식에서는 바쁜 날일수록 기록이 빠지고, 빠진 날짜는 감사에서 점검을 하지 않은 날로 읽힙니다. 반면 OpsKnow Agent가 판정과 동시에 기록까지 끝내면, 점검이 수행된 모든 날짜에 판정 근거가 남습니다. 알림은 OPENMARU Wallboard 화면과 OPENMARU Dashboard에도 함께 표시됩니다. Wallboard는 운영실 벽면 화면에 등급별 시스템 상태를 띄우는 제품이고 OPENMARU Dashboard는 여러

기관·센터의 상태를 한 화면에 모으는 제품이므로, 자동 처리 결과를 사람이 즉시 보는 수단으로 이 장과 연결됩니다.

8.3.2 결재가 필요한 임계치 조정, WAS 재시작, 인스턴스 증설과 DB 풀 크기 변경

결재 대상 6종의 정의 결재 대상은 서비스의 동작이나 판정 기준을 바꾸는 조치입니다. WAS 재시작은 처리 중인 요청을 끊고, DB 풀 크기 변경은 DBMS에 걸리는 동시 연결 수를 바꾸며, 임계치 조정은 이후 모든 예방점검의 판정 결과를 바꿉니다. 아래 표는 여섯 가지 결재 대상과 각각이 서비스에 주는 영향을 정리한 것입니다.

결재 대상	서비스에 주는 영향	결재 때 확인할 지표
임계치 조정	이후 예방점검 판정 기준이 바뀜	최근 지표 분포, 과거 Incident
WAS 재시작	처리 중인 요청 중단, 일시적 응답 지연	스레드·힙 메모리 지표, 관련 트랜잭션 ID
인스턴스 증설·감축	처리 용량과 비용 변화	사용률 추이, 예상 부하
세션 강제 종료	사용자 작업 중단	장기 점유 세션 목록
외부 시스템 재시작	연계 서비스 일시 중단	연계 응답 시간 지표
DB 풀 크기 변경	DBMS 동시 연결 수와 메모리 사용 변화	커넥션 사용률, 대기 시간

승인 코드로 묶인 결재 절차 결재는 형식적인 「확인」 버튼으로 끝나지 않습니다. 결재 6종은 모두 OS 패치·보안 조치와 같은 승인 코드 절차(점검, 승인 코드, 적용 직전 재확인, 재점검)를 거칩니다. 서버를 바꾸는 조치는 담당자가 결재 요청의 근거를 확인한 뒤 발급하는 1회용 승인 코드가 있어야 실행되며, 승인 코드에는 유효 시간이 있고 한 번만 쓸 수 있습니다. OpsKnow Agent는 적용 직전에도 대상 서버의 상태를 다시 확인해, 결재 요청을 만들 때와 상태가 달라졌으면 적용을 멈춥니다. 대화 중에 사람이 「네」라고 답해도 승인 코드가 없으면 적용하지 않으므로, 모호한 응답이 조치 승인으로 처리되는 일이 생기지 않습니다.

판정은 코드, 결정은 사람인 이유 이 절차에서 판정과 명령 선택은 결정적으로 동작하는 코드가 하고, 언어 모델은 사용자의 요청을 받아 전달하고 결과를 설명하는 역할만 맡습니다. 따라서 같은 지표에는 항상 같은 판정이 나오고, 결재 대상 조치는 사람의 승인 없이는 시작되지 않습니다. 의사결정권자 관점에서 이 구조는 AI가 운영을 맡더라도 서비스 변경의 책임이 기관과 승인권자에게 남는다는 것을 뜻하며, 위탁 운영 계약의 운영자 권한 조항에 자동 3종과 결재 6종을 그대로 옮겨 적을 수 있는 이유이기도 합니다.

8.3.3 변경 작업 요청과 결재 이력을 같은 저장소에 남기는 작업 관리 범위

OpsKnow가 기록하는 변경 작업의 범위 행정안전부 예방점검체계에는 장애관리와 함께 변경관리 표준운영절차가 마련돼 있습니다[S07]. 기관의 변경관리는 대개 기존 서비스 관리 도구와 결재 체계에서 운영되므로, OpsKnow는 그 절차 전체를 대신하지 않고 OpsKnow Agent를 거친 변경 작업의 요청과 결재, 적용 결과를 저장소에 남기는 데 범위를 둡니다. 결재 6종에 해당하는 조치가 OpsKnow Agent를 통해 요청되고 실행되면, 그 이력은 관련 Incident나 예방점검 결과와 같은 시스템 식별자로 연결됩니다.

변경 작업 기록 항목 변경 작업 한 건에는 다음 항목이 남습니다.

- 요청 내용과 요청 근거(판정 결과, 근거 지표, 관련 Incident)

- 결재자와 승인 코드 발급 기록
- 적용 직전 상태 재확인 결과
- 적용 결과와 적용 뒤 재점검 판정

기존 변경관리 절차와의 연결 이 기록은 기관의 변경관리 절차에서 요구하는 변경 요청서와 결과 보고의 근거 자료로 쓰입니다. 예를 들어 DB 풀 크기를 바꾼 경우, 변경 전후의 커넥션 사용률과 재점검 판정이 함께 남으므로 변경 결과 보고서를 따로 작성하지 않고 기록을 첨부할 수 있습니다. 다만 OpsKnow Agent를 거치지 않은 수작업 변경은 이 기록에 자동으로 남지 않으므로, 기관은 수작업 변경도 같은 저장소에 등록하도록 운영 절차를 정해 두어야 기록의 빈틈을 막을 수 있습니다.

8.4 일일 보고부터 연간 보고까지 누적되는 정기 운영 보고와 SLA 산출 근거

정기 운영 보고는 예방점검과 장애 대응의 결과를 기간 단위로 묶어 보여 주는 증적입니다. 행정예고된 안정성 고시가 SLA 체결을 의무로 두었으므로[S02], 기관과 수행사는 서비스 수준을 무엇으로 쟀는지 기간마다 같은 방식으로 보여 줄 수 있어야 합니다. OpsKnow는 일간, 주간, 월간, 분기, 연간 보고서 5종을 저장소의 원자료로 만들고, 상위 주기 보고서는 하위 주기의 결과를 다시 집계해 만듭니다. 연말에는 한 해 동안의 일간 보고부터 연간 보고까지가 같은 원자료에서 나온 한 묶음의 증적으로 남습니다.

보고서 문장은 GPU 1대로 폐쇄망(외부 연결 없음)에서 동작하는 30B 수준 내부 sLLM이 작성합니다. 다만 수치는 언어 모델이 만들지 않고 저장소의 원자료에서 코드가 계산하며, 언어 모델은 계산된 결과를 문장으로 설명하는 일만 맡습니다. 이 구성은 CogentAI 위에서 동작하는데, CogentAI는 내부 sLLM과 검색 기능을 기관 서버 안에서 제공하는 제품이므로 운영 기록이 외부로 나가지 않는 소버린 AI Level 3 구성을 보고 단계에서도 유지하는 수단이 됩니다.

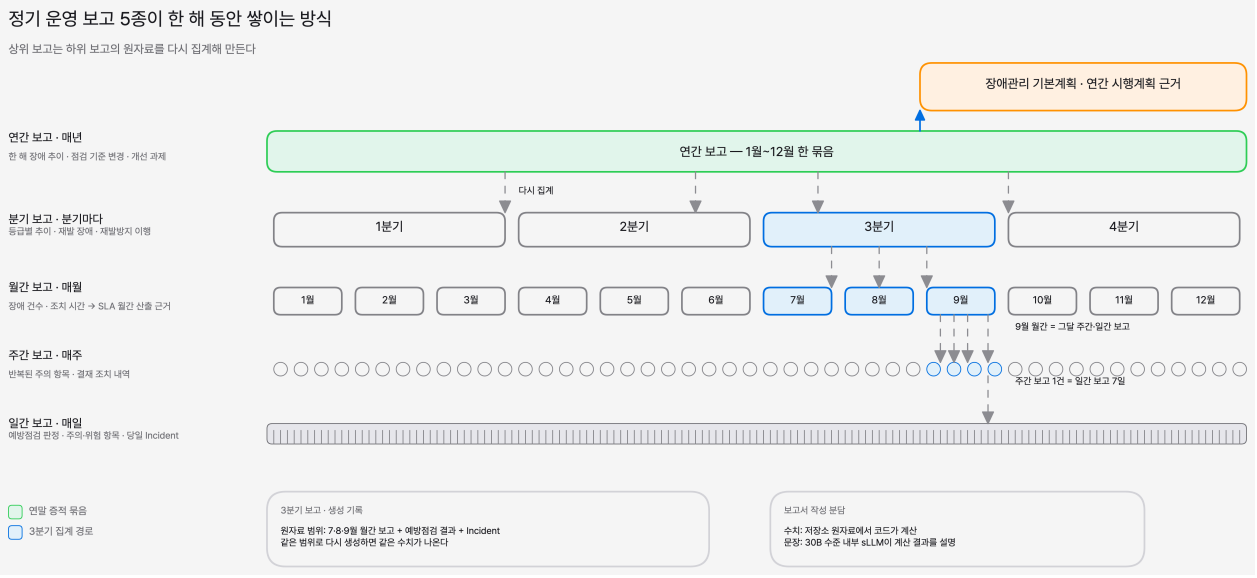


그림. 일·주·월·분기·연 정기 운영 보고의 누적 달력

8.4.1 일·주·월·분기·연 보고서 5종과 원자료 재생성 원칙

보고서 5종의 내용과 생성 주기 보고서 5종은 주기가 길어질수록 개별 판정보다 추이와 판단에 무게를 둡니다. 일간 보고는 그날의 예방점검 판정과 알림을, 연간 보고는 한 해의 장애 추이와 재발방지 이행 결과를 담습니다. 아래 표는 각 보고서의 주된 내용과 쓰임을 정리한 것입니다.

보고서	생성 주기	주된 내용	주된 쓰임
일간 보고	매일	예방점검 판정, 주의·위험 항목, 당일 Incident	일상점검 수행 증적
주간 보고	매주	반복된 주의 항목, 결재 조치 내역	운영 회의 자료
월간 보고	매월	시스템별 장애 건수와 조치 시간, 임계치 변경 이력	SLA 월간 산출 근거
분기 보고	분기마다	등급별 추이, 재발 장애, 재발방지 이행	기관 보고·위탁 운영 평가
연간 보고	매년	한 해 장애 추이, 점검 기준 변경, 개선 과제	장애관리 기본계획·연간 시행계획 근거

원자료 재생성 원칙 모든 보고서는 저장소의 원자료에서 다시 만들 수 있어야 합니다. 보고서 파일만 남기고 원자료를 지우면, 감사에서 보고서의 수치가 어떻게 나왔는지 물을 때 답할 수 없기 때문입니다. 따라서 OpsKnow는 보고서마다 생성에 쓴 원자료의 범위를 함께 기록하고, 같은 범위로 보고서를 다시 생성하면 같은 수치가 나오도록 계산을 코드로 고정합니다. 이 원칙 덕분에 SLA 산출값에 이견이 생기면 기관과 수행사가 같은 원자료로 수치를 다시 계산해 확인할 수 있습니다.

감사 대응 관점에서 보면 이 원칙은 답변의 형식까지 바꿉니다. 감사인의 질문에 「점검을 했다」라고 답하는 대신, 해당 수치와 그 수치가 저장된 위치, 그리고 그 기록이 만들어진 시점을 함께 제시할 수 있기 때문입니다. 점검 결과는 성능 관리 도구에, 조치는 업무 처리 시스템에, 구성 변경은 별도 대장에 흩어져 있으면 이 세 가지를 한 번에 내놓기 어렵습니다. 보고서와 원자료가 같은 저장소에 있으면 감사 초기의 자료 제출 단계를 정해진 형식의 조회로 대신할 수 있고, 담당자가 바뀐 기관에서도 같은 답이 나옵니다.

상급 기관 보고와의 관계 중요 시스템 장애를 디지털안전상황실에 즉시 보고하는 일은 정기 보고와 별개로 장애 발생 시점에 이뤄집니다[S02]. 이때 OpsKnow에 쌓인 Incident 기록의 원인과 조치 필드는 보고 내용의 초안이 되므로, 담당자가 장애 대응 중에 경위서를 처음부터 쓰지 않아도 됩니다. 기관마다 쓰는 보고 양식은 다를 수 있으므로, 도입 시 기관 양식에 맞춘 출력 형식을 요구사항에 함께 적어 두는 것이 좋습니다.

의무화 이후의 장애 보고는 사고가 끝난 뒤 결재를 거쳐 올리는 보고보다, 대응하는 동안 상황을 실시간으로 공유하는 쪽에 가까워집니다. 그만큼 보고가 늦어지는 원인은 대개 보고서 작성 속도보다 원인과 조치 경과를 한곳에서 확인할 수 없다는 데 있습니다. 장애가 진행되는 동안 OpsKnow Agent가 판정 결과와 조치 경과를 Incident 기록에 이어 붙이면, 상황 공유에 필요한 최신 내용이 이미 한 기록 안에 모여 있으므로 담당자는 그 기록을 확인하고 보내는 일에 집중할 수 있습니다.

8.4.2 장애관리 기본계획과 연간 시행계획에 다시 쓰이는 연간 보고 묶음

연간 보고 묶음의 재사용 방식 행정예고된 안정성 고시는 기관이 장애관리 기본계획을 세우고 해마다 연간 시행계획을 만들도록 했습니다[S02]. 이 두 계획은 지난 운영 결과를 근거로 다음 해의 점검 기준과 개선 과제를 정하는 문서이므로, 연간 보고 묶음이 그대로 입력 자료가 됩니다. 아래 표는 연간 보고 묶음의 각 내용이 계획 문서의 어느 항목에 다시 쓰이는지를 정리한 것입니다.

연간 보고 묶음의 내용	다시 쓰이는 계획 문서 항목
시스템별·등급별 장애 추이	기본계획의 중점 관리 대상 선정
재발 장애와 재발방지 이행 결과	연간 시행계획의 개선 과제
임계치·점검 기준 변경 이력	차년도 예방점검 기준 조정
결재 조치 내역과 조치 시간	SLA 목표와 위탁 운영 평가 기준

감사 제출용 연간 증적 묶음 연간 보고 묶음은 감사 때도 그대로 제출할 수 있는 형태입니다. 일간 보고가 일상점검의 수행을, Incident 기록이 장애 대응을, 결재 기록이 변경 조치의 권한을, 연간 보고가 그 모두의 추이를 보여 주므로, 기관은 한 저장소에서 같은 식별자로 묶인 기록을 꺼내 제출하면 됩니다. 2024년 예방점검체계가 일상점검·특별점검·구조진단 3개 분야로 점검을 나눈 만큼[S07], 연간 묶음에도 분야별로 무엇을 언제 점검했는지가 구분돼 있어야 합니다.

의사결정권자가 확인할 판단 기준 이 장에서 의사결정권자가 확인할 것은 두 가지입니다. 첫째, 우리 기관의 예방점검 결과, 장애 기록, 정기 보고가 같은 식별자로 한 저장소에서 조회되는지입니다. 둘째, 자동 처리 3종과 결재 6종의 경계가 위탁 운영 계약의 운영자 권한 조항에 적혀 있는지입니다. 두 질문 가운데 하나라도 답이 「아니오」라면, 그 기관의 예방점검은 점검표를 자동으로 채우는 단순 자동화 단계에 있으며, 차년도 사업 요구사항에 연결 키, 보관 기간, 경계 매트릭스를 적는 것부터 시작해야 합니다.

9장: 관리자가 실제로 쓰는 장면 — 자연어 요청 한 줄에서 점검 보고서와 조치 승인까지

OpsKnow Agent는 관리자가 한국어 문장으로 요청하면 OS·미들웨어·DBMS 예방점검 결과를 조회하고 판정해서, 근거가 붙은 보고서와 조치 제안으로 답하는 AI 에이전트입니다. 2026년 3월 행정예고된 「행정기관 및 공공기관 정보시스템 안정성 고시」는 예방점검과 장애 관리를 포함한 46개 안정성 기준, 정보시스템 표준운영절차(SOP), 서비스 수준 협약(SLA) 체결 의무를 함께 담고 있는 것으로 알려져 있습니다[S02]. 따라서 고시 제2026-23호 체계에서 운영 조직이 내야 하는 결과물은 점검을 했다는 사실 하나로 끝나지 않고, 무엇을 어떤 기준으로 판정했는지 보여 주는 증적까지 이어집니다. 관리자가 그 증적을 매번 손수 모아 문서로 엮으면 예방점검 업무의 대부분이 점검이 아닌 문서 작성 업무로 바뀝니다.

이 문제가 지금 커지는 이유는 점검 대상의 규모에 있습니다. 행정안전부는 2026년 8월 27일 공공 정보시스템 15,033개의 등급을 확정했고, 그 분포는 A1 73개, A2 157개, A3 671개, A4 14,132개입니다[S01]. 2024년에 마련된 예방점검체계는 일상점검·특별점검·구조진단 3개 분야와 8개 예방점검 항목으로 짜여 있어서, 등급이 정해진 시스템마다 같은 형식의 점검 기록이 주기적으로 쌓입니다[S07]. 장비마다 매일 생기는 점검 결과를 사람이 읽고 판정하고 보고서로 옮기는 방식은 시스템 수가 늘수록 운영 인력을 작성 업무에 묶어 둡니다.

관리자의 업무 달력에서 자연어 요청이 가장 자주 생기는 경우는 네 가지입니다. 아래 표는 이 장에서 다루는 네 장면을 요청 문장, 응답 형태, 사람이 결정하는 단계로 나누어 정리한 것입니다. 네 장면 모두 판정은 코드가 하고, 결정은 사람이 하며, 설명 문장은 AI가 쓰는 같은 원칙을 따릅니다. 보고 장면에서는 결과를 문서 파일로 받는 요청도 자주 나옵니다. 예를 들어 관리자가 「지난주 시스템 작업과 결과 보고서 찾아서 hwp로 보고서 만들어 줘」라고 요청하면, OpsKnow Agent는 OpsKnow에서 지난주 작업 기록과 결과 보고서를 찾아 HWP 보고서 파일로 만들고 각 내용에 원 기록으로 가는 출처 링크를 붙입니다(9.2.2).

장면	요청 문장 예	OpsKnow Agent 응답	사람이 결정하는 단계
월간 점검 보고(9.1)	「이번 달 OS와 미들웨어, DBMS 점검 결과 보여 줘」	영역별 판정 요약, 주의·이상 항목 근거, 증적 첨부	보고서 확정
분기 감사 준비(9.2)	「이번 분기 미이행 점검 항목 알려 줘」	미이행 목록과 근거, 분기 운영 보고서 초안	초안 검토와 확정
보안공지 발표일(9.3)	「오늘 나온 보안공지 영향 서버 찾아 줘」	대상 서버 수, 상태별 분류, 조치 제안	승인 코드 입력
장애 직후(9.4)	「방금 장애와 비슷한 과거 사례 찾아 줘」	유사 Incident, 원인 후보, 재발방지 조치 제안	결재가 필요한 조치의 실행

이 처리 구조는 외부로 데이터를 보내지 않는 소버린 AI Level 3 구성 위에서 동작합니다. 응답 문장을 만드는 모델은 30B 수준 내부 sLLM 하나이고, GPU 1대를 갖춘 폐쇄망(외부 연결 없음) 서버 한 대가 이 모델을 실행합니다. OPENMARU CogentAI는 근거 검색(RAG)과 이 내부 sLLM의 서빙을 기관 서버 안에서 맡는 AI 플랫폼이므로, 점검 결과와 장애 이력을 기관 밖으로 보내지 않고 자연어 응답을 만듭니다. 그래서 의사결정권자는 네 장면에서 담당자 업무가 작성에서 검토로 바뀌는 단계를 확인하고, 차년도 사업 요구사항에 「자연어 요청」「증

적 첨부」「승인 코드」 같은 조건을 넣을 근거를 얻을 수 있습니다. 화면 예시는 모두 서버 이름과 IP 주소를 가린 상태이고, 개수는 「○」로 표기했습니다.

9.1 「이번 달 OS와 미들웨어, DBMS 점검 결과」 요청에 대한 영역별 판정 요약 보고

월간 점검 결과 요청은 네 장면 가운데 가장 자주 반복되는 요청이고, 예방점검 의무를 지키고 있는지를 의사결정권자가 확인하는 기본 보고입니다. 행정안전부는 2024년 10월 정보시스템 예방점검체계와 표준운영절차를 마련해 공공 정보시스템의 장애 예방·대응·사후관리를 점검하고 있습니다[S06]. 따라서 월간 보고에는 영역별 판정 결과와 함께 그 판정을 뒷받침하는 수집 자료가 증적으로 붙어야 합니다. OpsKnow Agent는 한 줄 요청을 받아 이 두 가지를 한 번에 만들어 냅니다.

9.1.1 요청 문장에서 결정적 조회와 판정, 근거가 붙은 응답까지의 흐름

월간 요청의 처리 단계와 주체

「이번 달 OS와 미들웨어, DBMS 점검 결과 보여 줘」라는 요청은 다섯 단계를 거쳐 응답이 됩니다. LLM이 맡는 일은 사용자 문장을 받아 넘기는 첫 단계와 판정 결과를 설명 문장으로 옮기는 마지막 단계뿐입니다. 반면 문장 해석, 실행할 조회 명령 선택, 정상·주의·이상 판정은 같은 입력에 늘 같은 결과를 내는 결정적 코드가 맡습니다. 아래 표는 단계마다 처리 주체와 남는 기록을 정리한 것입니다.

단계	처리 주체	하는 일	남는 기록
1. 요청 접수	LLM	사용자 문장을 그대로 코드에 전달	요청 원문과 요청 시각
2. 의도 해석	코드	영역(OS·미들웨어·DBMS)과 기간(이번 달) 추출, 조회 명령 선택	선택한 명령과 조건
3. 조회	코드	OS 지표와 OPENMARU APM이 수집한 미들웨어·DBMS 지표 조회	원값과 수집 시각
4. 판정	코드	기준값과 비교해 정상·주의·이상 분류	판정 결과와 적용 기준
5. 설명	LLM	판정 결과만으로 요약 문장 작성, 증적 첨부	응답 원문

미들웨어·DBMS 지표의 수집 출처

WAS 스레드, 커넥션 풀, DBMS 세션과 잠금 같은 미들웨어·DBMS 지표는 OPENMARU APM이 수집한 자료를 기준으로 합니다. OPENMARU APM은 미들웨어와 DBMS의 상태를 상시 수집하는 기능을 제공하는 애플리케이션 성능 관리(APM, Application Performance Management) 제품이므로, OpsKnow Agent는 별도의 수집 장치를 더 설치하지 않고 그 지표를 판정과 보고로 잇습니다. 일상점검은 CPU·메모리·디스크 상태, 메인 화면 접속 여부와 접속 시간, 인증서 유효기간과 도메인 만료일 같은 항목을 봅니다[S07]. 월간 요청에 대한 응답은 OS 지표, APM 지표, 서비스 접속 결과를 한 판정표로 모은 결과입니다.

판정을 코드에 두는 이유와 감사 대응

판정을 코드에 두면 같은 달의 같은 자료를 다시 조회했을 때 같은 판정이 나옵니다. 반면 언어 모델은 같은 질문에도 매번 조금씩 다른 문장을 만들기 때문에, 판정까지 모델에 맡기면 감사 때 지난달 결과를 재현해 보일 수 없습니다. OpsKnow Agent의 응답에서 「주의」나 「이상」이라는 낱말은 항상 코드가 낸 판정 결과에서 오고, 언어 모델은 그 판정을 읽기 쉬운 문장으로 옮기는 일만 합니다. 이 구조 덕분에 단순 자동화 도구가 채우는 점검표와 달리, 응답마다 어떤 기준으로 판정했는지가 증적으로 함께 남습니다.

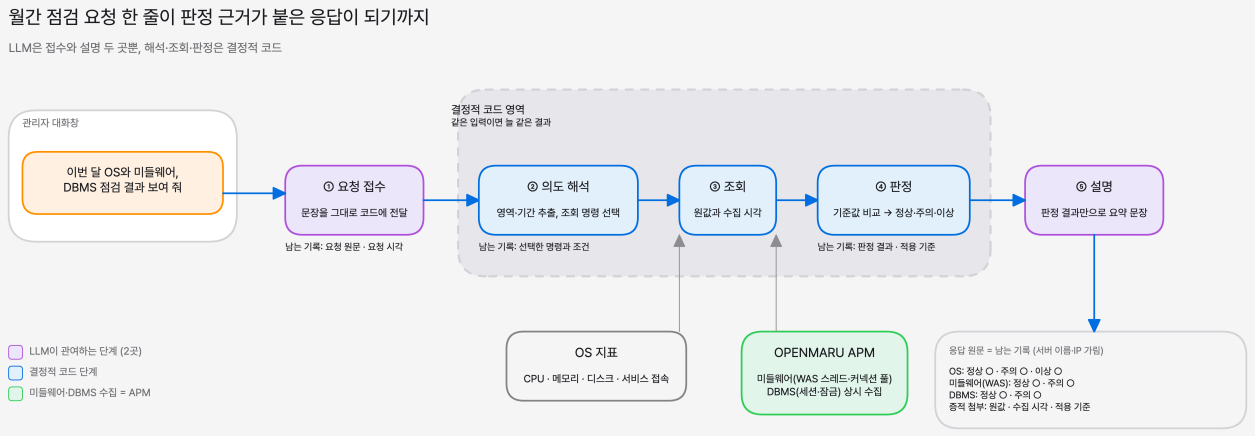


그림. 월간 점검 요청이 LLM 접수 → 코드 해석·조회·판정 → LLM 설명으로 이어지는 흐름

9.1.2 주의·이상 항목만 추려 보여 주는 의사결정권자용 응답 형태

주의·이상 항목 중심의 응답 구성

의사결정권자에게 보내는 응답은 영역별 집계를 먼저 보이고, 그 아래에 주의와 이상으로 판정된 항목만 근거와 함께 나열합니다. 정상으로 판정된 항목은 본문에서 빼고 첨부된 증적 파일에만 남기므로, 응답 길이는 점검 대상 수가 아니라 조치가 필요한 항목 수에 따라 정해집니다. 다음은 그 형태를 보여 주는 응답 예시이며, 서버 이름은 가렸고 개수는 「○」로 표기했습니다.

> ***이번 달 예방점검 결과 요약*** *OS ○대: 정상 ○ · 주의 ○ · 이상 ○ / 미들웨어 ○개: 정상 ○ · 주의 ○ / DBMS ○개: 정상 ○ · 이상 ○*
>
> ***이상*** *업무서버 A의 데이터 디스크 사용률이 기준값을 넘었습니다. 근거: 수집 시각, 사용률 원값, 적용 기준값. 조치 제안: 로그 보관 정책 점검(담당자 작업).* ***이상*** *DBMS B의 장시간 잠금이 반복되었습니다. 근거: OPENMARU APM 잠금 대기 기록. 조치 제안: 해당 트랜잭션 확인(결재 불필요, 조회만).* ***주의*** *WAS C의 커넥션 풀 사용률이 기준값에 근접했습니다. 근거: 일별 최대치 추이. 조치 제안: DB 풀 크기 변경 검토(결재 필요).*
>
> *첨부: 영역별 판정표 전체, 항목별 원값과 수집 시각, 적용 기준 목록*

항목 하나에 붙는 네 가지 정보

응답의 각 항목에는 대상, 판정, 근거, 조치 제안 네 가지가 들어갑니다. 조치 제안에는 그 조치를 자동으로 처리하는지, 결재가 필요한지를 함께 표시하므로 의사결정권자가 승인할 일과 확인만 할 일을 바로 나눌 수 있습니다. 다만 IT 담당자는 같은 자료에서 더 깊은 근거가 필요하므로 「WAS 영역 주의 항목 근거 자세히」처럼 이어서

요청하고, OpsKnow Agent는 같은 판정표에서 항목별 원값과 추이를 펼쳐 보여 줍니다. 의사결정권자와 담당자는 같은 판정 결과를 서로 다른 깊이로 읽습니다.

자료가 없는 항목의 표시 방법

OpsKnow Agent는 확실하지 않은 상태를 정상으로 보고하지 않는다는 판정 원칙을 따르므로, 수집 자료가 비어 있거나 판정에 필요한 정보가 부족한 항목은 「정상」으로 합산하지 않고 「수집 자료 없음」으로 따로 표시합니다. 확실하지 않은 상태를 정상으로 보고하면 다음 감사에서 증적과 보고서가 서로 맞지 않게 되기 때문입니다. 행정예고된 안정성 고시가 중요 시스템 장애를 행정안전부 디지털안전상황실에 즉시 보고하도록 요구하는 것도 같은 이유로, 보고 내용이 실제 상태와 어긋나지 않아야 합니다[S02]. 응답의 집계 숫자에는 판정된 항목만 들어가고, 판정하지 못한 항목은 별도 줄로 남아 담당자의 확인 대상이 됩니다.

9.2 분기 감사 준비 장면 — 미이행 점검 항목 조회와 분기 운영 보고서 초안 검토

분기 감사 준비는 석 달 동안 쌓인 점검 결과에서 빠진 항목을 찾고, 그 기간의 운영 내용을 보고서 한 건으로 묶는 일입니다. 행정예고된 안정성 고시는 장애관리 기본계획과 연간 시행계획, 그리고 SLA 체결을 함께 요구하는 것으로 알려져 있어서, 분기 단위 이행 기록은 연간 계획과 SLA 이행을 확인하는 중간 증적이 됩니다[S02]. 반면 이 기록을 사람이 여러 시스템의 점검표에서 다시 모으면, 감사 준비 기간 대부분이 자료 수집에 쓰입니다. OpsKnow에 쌓인 운영 지식은 이 수집 과정을 조회 요청 한 번으로 줄이고, 담당자가 검토하는 일에 시간을 쓰게 합니다.

9.2.1 「이번 분기 미이행 점검 항목」 질의에 대한 판정 결과와 근거 목록

미이행 항목을 가르는 두 가지 기준

「이번 분기 미이행 점검 항목 알려 줘」라는 요청에 대해 OpsKnow Agent는 OpsKnow에 저장된 예방점검 결과를 기간과 영역으로 조회하고, 두 종류의 항목을 나누어 보여 줍니다. 첫째는 기관이 정한 점검 주기 안에 결과 기록이 비어 있는 항목이고, 둘째는 이상 판정을 받았는데 조치 기록이 이어지지 않은 항목입니다. 점검을 하지 않은 경우와 점검은 했지만 후속 조치가 끝나지 않은 경우를 구분합니다. 예방점검체계가 일상점검·특별점검·구조진단 3개 분야로 나뉘어 있으므로, 응답도 이 분야별로 미이행 항목을 묶어 보여 줍니다[S07].

> ***이번 분기 미이행 점검 항목*** *일상점검: 결과 없음 ○건(대상 시스템 ○개) · 이상 후 조치 미완료 ○건 특별점검: 결과 없음 ○건*
>
> *감사 제출용 근거 목록: 항목별 점검 주기, 마지막 점검 결과와 시각, 이상 판정 근거, 연결된 조치 기록 유무*

감사 제출용 증적 목록의 구성

미이행 목록과 함께 나오는 근거 목록은 감사에서 확인하는 질문에 바로 답하는 형태입니다. 아래 표는 OpsKnow에 쌓이는 자산을 감사 관점으로 나눈 것이며, 표에 적은 네 종류가 분기 응답의 첨부 자료가 됩니다. Incident 케이스는 원인, 조치, 재발방지, 관련 트랜잭션 ID, 사이트 태그, 운영자 ID, 자동 RCA 소요 시간 7개 필드로 남기 때문에, 장애가 있었던 분기에는 그 기록이 함께 제출됩니다.

증적 종류	담긴 내용	감사에서 확인하는 것
예방점검 결과	항목별 원값, 수집 시각, 판정, 적용 기준	주기대로 점검했는가
Incident 케이스	7개 필드(원인·조치·재발방지 등)	장애 뒤 원인과 재발방지를 기록했는가
일정별 운영 보고	일·주·월·분기·연 보고서	운영 상태를 정기적으로 보고했는가
조치 승인 기록	승인자, 승인 코드 사용 시각, 적용 전후 상태	변경이 승인 절차를 거쳤는가

판정이 필요한 질의와 단순 자동화의 차이

점검표를 자동으로 채우는 단순 자동화 도구도 분기 점검 결과를 한곳에 모을 수는 있습니다. 하지만 「미이행」을 찾으려면 점검 주기와 결과를 대조하고, 이상 판정 뒤 조치 기록이 있는지를 확인하는 판정이 필요합니다. 그래서 이 질의는 기록을 모으는 자동화와 기록을 읽고 판정하는 지능화가 실제로 갈라지는 경우이고, 15,033개로 확정된 공공 정보시스템처럼 대상이 많을수록 그 차이가 커집니다[S01].

9.2.2 축적 자료로 만든 분기 운영 보고서 초안과 담당자 검토 단계

분기 운영 보고서 초안의 재료

OpsKnow는 일·주·월·분기·연 다섯 종류의 일정별 운영 보고를 자동으로 생성해 쌓아 두므로, 분기 보고서 초안은 이미 쌓인 월간 보고와 예방점검 결과, Incident 케이스를 묶어서 만듭니다. 알림 발송, OpsKnow 저장, 임계치 보고서 생성은 사람 결재 없이 OpsKnow Agent가 자동으로 처리하는 작업입니다. 따라서 분기 초안을 만드는 데 새 자료를 수집할 필요가 없고, 초안에 들어가는 모든 수치는 OpsKnow의 원 기록으로 되짚어 갈 수 있습니다. 행정예고된 고시가 정보시스템 표준운영절차를 요구하는 만큼, 담당자는 초안을 확정하기 전에 기관 표준운영절차의 보고 항목과 초안 내용을 대조합니다[S02].

HWP 보고서 파일 요청

관리자가 「지난주 시스템 작업과 결과 보고서 찾아서 hwp로 보고서 만들어줘」처럼 파일 형식을 정해 요청하면, OpsKnow Agent는 OpsKnow에서 해당 기간의 작업 기록과 결과 보고서를 찾아 HWP 보고서 파일로 만듭니다. 보고서의 수치와 문장에는 근거가 된 원 기록으로 가는 출처 링크가 붙으므로, 담당자는 파일을 연 채로 링크를 따라가 원 기록과 대조할 수 있습니다. 공공기관의 내부 보고는 대개 HWP 문서로 결재를 올리므로, 다른 형식의 초안을 HWP 양식으로 다시 옮겨 적는 일이 없어집니다.

초안 생성부터 확정까지의 검토 단계

초안이 나온 뒤의 일은 작성이 아니라 검토와 확정입니다. 아래 표는 그 단계와 처리 주체를 나눈 것이며, 사람이 맡는 단계는 근거 대조, 해석 문장 수정, 확정 세 가지입니다.

근거 대조 단계에서 담당자가 확인하는 것은 문장이 매끄러운지가 아니라, 초안의 수치마다 그 수치를 만든 원 기록의 위치와 기록 시점이 붙어 있는지입니다. 분기 보고서의 한 줄이 「어느 수치가, 어디에 저장된 기록에서, 언제 만들어졌는지」의 세 요소로 답할 수 있으면, 감사에서 같은 질문이 나와도 보고서를 다시 열어 그 위치를 짚

는 것으로 답이 끝납니다. 세 요소 가운데 하나라도 빠진 수치는 확정 전에 원 기록과 다시 맞추거나 해석 문장에서 빼는 것이 안전합니다.

단계	처리 주체	확인 내용	결과물
초안 생성	OpsKnow Agent	분기 점검 결과·Incident·월간 보고 집계	분기 운영 보고서 초안
근거 대조	IT 담당자	초안 수치와 OpsKnow 원 기록의 일치	대조 확인 표시
해석 문장 수정	IT 담당자	원인 서술과 개선 계획 문장	수정본
확정	관리자	미이행 항목의 후속 계획	확정 보고서
보관	자동	확정본을 증적으로 저장	분기 운영 보고 자산

작성 시간의 측정 방법

초안 작성에 드는 시간은 기관이 보유한 시스템 수와 점검 항목 구성에 따라 크게 달라집니다. 보고서 작성 시간은 도입 전후에 같은 분기 보고서를 만드는 데 걸린 시간으로 재며, 이 값은 10.2절의 도입 효과 측정 항목에 들어갑니다. 다만 업무의 형태가 바뀌는 방향은 분명합니다. 담당자는 여러 시스템의 점검표를 모아 문장을 처음부터 쓰는 대신, 초안의 수치가 원 기록과 맞는지 확인하고 해석 문장을 고치는 일을 합니다.

9.2.3 Wallboard에서 확인하는 분기 점검 이행 현황

상황실 화면으로 보는 이행 현황

OPENMARU Wallboard는 운영 상황실의 벽면 화면에 시스템 상태와 점검 현황을 크게 띄우는 제품입니다. 분기 감사 준비 회의에서는 여러 사람이 같은 숫자를 함께 보아야 하므로, OpsKnow Agent가 판정한 분기 이행 현황을 Wallboard에 띄우면 보고서를 나누어 읽지 않고도 미이행 항목을 바로 논의할 수 있습니다. OPENMARU Dashboard는 여러 기관이나 센터의 상태를 한 화면에 모으는 상위 화면이어서, 상급 기관이 소속 기관들의 점검 이행 현황을 같은 기준으로 비교할 때 쓰입니다. 등급이 A1부터 A4까지 나뉘어 있으므로, 화면도 등급별로 이행 현황을 따로 보여 주어 상위 등급 시스템의 미이행이 먼저 보이게 합니다[S01].

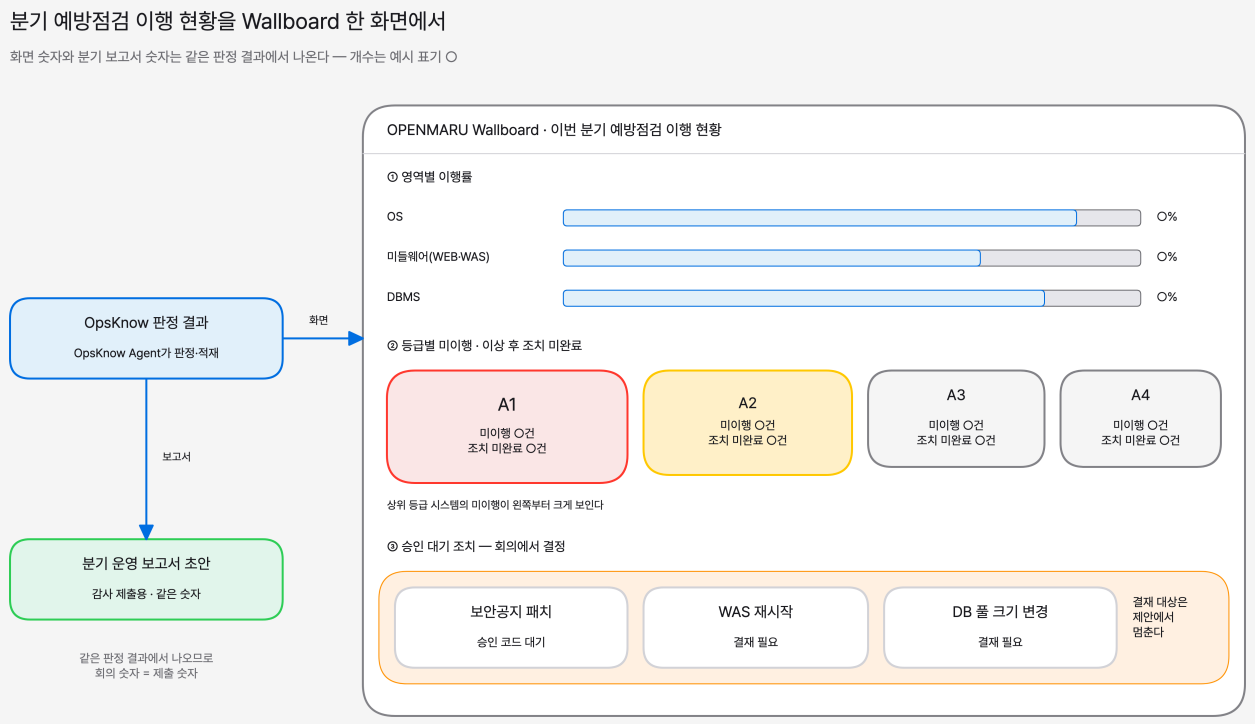


그림. Wallboard에 띄운 분기 예방점검 이행 현황 화면 예

화면에서 확인할 세 가지 항목

Wallboard 화면은 예방점검 결과, 지연 트랜잭션, 이상 그룹, 승인 대기 네 영역으로 나뉘며(4.3.1), 분기 감사 준비 회의에서는 예방점검 결과 영역을 분기 누적으로 바꿔 띄웁니다. 이때 의사결정권자가 확인할 항목은 영역별 이행률, 이상 판정 뒤 조치가 끝나지 않은 건수, 승인을 기다리는 조치 목록 세 가지입니다. 이 가운데 승인 대기 목록은 9.3절과 9.4절의 조치 제안이 모이는 곳이므로, 회의 결과가 곧 승인 결정으로 이어집니다. 미이행 건수가 특정 영역에 몰려 있으면, 그 영역 담당자에게 후속 계획을 요청하는 것이 회의의 결론이 됩니다.

화면 숫자와 보고서 숫자의 일치

Wallboard 화면의 숫자와 분기 보고서 초안의 숫자는 모두 OpsKnow에 쌓인 같은 판정 결과에서 나옵니다. 회의에서 본 숫자와 감사에 제출하는 보고서의 숫자가 서로 다른 일이 생기지 않습니다. 반면 화면과 보고서를 서로 다른 도구로 만들면 집계 시점과 기준이 어긋나 감사 때 어느 쪽이 맞는지를 다시 설명해야 합니다.

9.3 보안공지 발표일 장면 — 영향 서버 조회부터 승인 코드 발급과 재점검 회신까지

보안공지가 발표된 날의 일은 영향받는 서버를 빨리 가려내고, 조치할 서버와 조치할 수 없는 서버를 나누어 승인을 받는 것입니다. 한국인터넷진흥원(KISA)이 운영하는 보호나라&KrCERT/CC는 보안공지를 게시하고, KNVD(취약점 정보포털)는 보안공지, 국내 취약점 정보, CVE 통합검색을 제공합니다[S10]. 그래서 보안 취약점 대응의 출발점은 공지 내용과 기관 서버의 실제 패키지 상태를 대조하는 일이 됩니다. OpsKnow Agent는 이 대조와 분류를 코드로 처리하고, 서버를 바꾸는 조치는 관리자의 승인 코드가 있어야만 적용합니다.

9.3.1 대상 서버 수와 상태별 분류가 담긴 조치 제안 응답

보안공지 대조 응답의 구성

「오늘 나온 보안공지 영향 서버 찾아 줘」라는 요청에 OpsKnow Agent는 공지에 적힌 패키지와 수정 버전을 기관 서버의 OS 패키지 설치 상태와 대조하고, 대상 서버 수와 상태별 분류를 담은 응답을 냅니다. 대조와 분류는 결정적 코드가 하므로, 같은 공지에 대해 다시 조회해도 같은 분류가 나옵니다. 다음은 서버 이름을 가린 응답 예시입니다.

> ***보안공지 영향 서버 조회 결과*** *대상 패키지가 설치된 서버 O대 조치 필요 O대 · 이미 조치됨 O대 · 저장소에 수정본 없음 O대 · 지원 종료 OS O대 대상 패키지가 설치돼 있지 않은 서버 O대 · 보안 정보 없음 O대*
>
> *조치 제안: 「조치 필요」 서버 O대에 수정 버전 패키지 적용. 적용하려면 승인 코드가 필요합니다.*

서버 상태 분류 여섯 가지

응답의 분류는 서버마다 다음 조치가 무엇인지 알 수 있도록 나뉘어 있습니다. 「저장소에 수정본 없음」과 「지원 종료 OS」는 패키지 적용으로 해결되지 않으므로, 의사결정권자가 별도 대응 계획을 정해야 하는 서버입니다. 아래 표는 분류별 의미와 다음 조치를 정리한 것입니다.

분류	의미	다음 조치
조치 필요	취약한 버전이 설치돼 있고 수정본이 있음	승인 코드 입력 후 적용
이미 조치됨	수정 버전 이상이 설치돼 있음	없음(증적으로 기록)
설치돼 있지 않음	대상 패키지가 서버에 없음	없음(증적으로 기록)
저장소에 수정본 없음	기관 패키지 저장소에 수정 버전이 없음	저장소 갱신 요청
보안 정보 없음	판정에 필요한 보안 정보가 없음	담당자 확인
지원 종료 OS	공급사 지원이 끝난 OS	전환 또는 보완 대책 결정

확인되지 않은 상태의 표기 원칙

판정에 필요한 정보가 부족하면 OpsKnow Agent는 서버를 「최신」으로 표시하지 않고 「보안 정보 없음」으로 남깁니다. 보안 취약점 대응에서 가장 위험한 보고는 상태가 불분명한 서버를 안전하다고 적는 것이기 때문입니다. OpsKnow Agent가 직접 적용하는 패치는 OS 패키지 업데이트이고, 재부팅, 설정 변경 실행, zip·tar 형태로 설치한 제품의 패치, 패키지 삭제나 다운그레이드는 직접 하지 않습니다. 미들웨어·DBMS 취약점은 같은 응답에서 해당 여부를 점검해 보고하고, 제품 설치본 패치처럼 사람이 실행할 조치는 승인 뒤 담당자 작업으로 따로 표시합니다.

9.3.2 관리자가 승인 코드를 입력하는 지점과 재점검 결과 회신

1회용 승인 코드의 조건

조치 제안을 받은 관리자가 적용을 결정하면 OpsKnow Agent는 1회용 승인 코드를 발급하고, 관리자는 그 코드를 대화창에 입력해야 적용이 시작됩니다. 승인 코드는 발급 후 30분 동안만 유효하고 한 번만 쓸 수 있으며, 코드 없이 「네」나 「진행해」라고만 답하면 적용하지 않습니다. 대화 중에 무심코 남긴 동의나 다른 질문에 대한 답이 서버 변경으로 이어지는 일을 막습니다. 이 구조에서 서버를 바꿀지 결정하는 주체는 항상 사람이고, AI는 그

결정을 요청하고 설명하는 역할만 맡습니다. 같은 승인 코드 절차가 보안 패치뿐 아니라 9.4.2의 결재 6종에도 그대로 쓰입니다.

적용 직전 재확인과 중단 조건

승인 코드가 입력되면 OpsKnow Agent는 적용 직전에 대상 서버의 상태를 다시 확인합니다. 점검했을 때와 상태가 다르면, 예를 들어 그사이 다른 담당자가 이미 패키지를 바꾸었으면 적용을 멈추고 달라진 내용을 보고합니다. 그래서 관리자가 승인한 내용과 실제로 적용되는 내용이 어긋나지 않고, 승인 시점의 판단 근거가 적용 시점에도 유효함이 보장됩니다.

재점검 결과 회신과 증적 저장

적용이 끝나면 OpsKnow Agent는 같은 서버를 다시 점검해 수정 버전이 설치되었는지 확인하고, 적용 전후 버전과 재점검 판정을 관리자에게 회신합니다. 이 회신과 함께 승인자, 승인 코드 사용 시각, 적용 전후 상태가 OpsKnow에 조치 승인 기록으로 쌓이므로, 9.2.1의 감사 증적 목록에 그대로 들어갑니다. 보안공지 대조 요청부터 재점검 회신까지가 한 대화 기록으로 이어지기 때문에, 공지별 대응 이력을 나중에 추적하기도 쉽습니다.

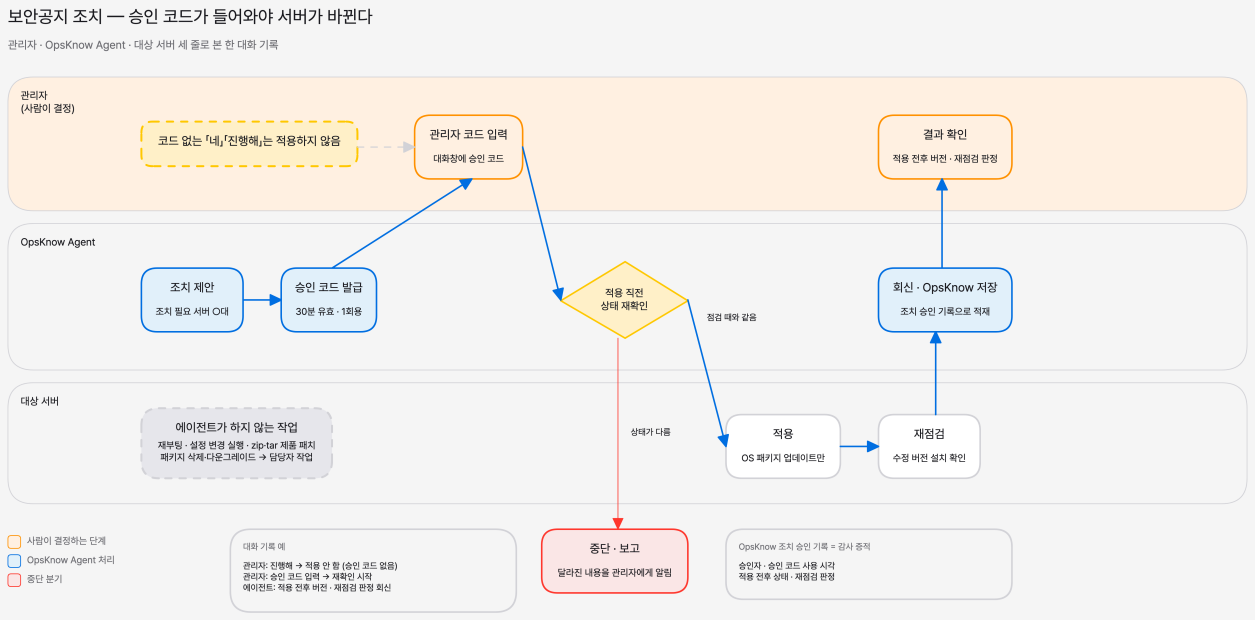


그림. 조치 제안 → 승인 코드 입력 → 적용 직전 재확인 → 적용 → 재점검 회신 순서도

9.4 장애 직후 장면 — 과거 유사 Incident 검색과 결재가 필요한 조치 제안

장애가 수습된 직후에는 원인을 좁히고 재발방지 조치를 정해 기록으로 남겨야 합니다. 2023년 11월 17일 행정전산망 장애의 원인은 라우터의 케이블 연결 모듈에 있는 포트 일부 이상으로 발표되었고, 해킹 흔적은 없었으며, 11월 18일 9시에 서비스가 재개된 뒤 11월 19일 포트를 교체해 정상화되었습니다[S04]. 장애 원인이 장비 부품 단위로 좁혀지기까지 시간이 걸렸고, 초기에는 다른 장비가 원인으로 거론되기도 했습니다[S05]. 그래서 과거 장애의 원인과 조치가 구조화된 기록으로 남아 있으면, 비슷한 증상이 다시 나왔을 때 원인 후보를 빨리 좁힐 수 있습니다. 행정예고된 안정성 고시는 중요 시스템 장애를 행정안전부 디지털안전상황실에 즉시 보고하도록 요구하는 것으로 알려져 있어서, 장애 직후의 원인과 조치 기록은 보고 의무와도 연결됩니다[S02].

9.4.1 OpsKnow에 쌓인 장애 이력에서 찾은 유사 사례와 재발방지 조치

유사 Incident 검색 응답의 구성

「방금 장애와 비슷한 과거 사례 찾아 줘」라는 요청에 OpsKnow Agent는 이번 장애의 증상, 영역, 사이트 태그를 조건으로 OpsKnow의 Incident 케이스를 검색합니다. 응답에는 유사 사례마다 원인, 당시 조치, 재발방지 조치, 관련 트랜잭션 ID가 함께 나오므로, 담당자는 과거 사례의 근거 기록까지 바로 열어 볼 수 있습니다. 관련 트랜잭션 ID는 OPENMARU APM이 수집한 트랜잭션 기록과 연결되어 있어서, 과거 장애 당시 어느 요청이 느려졌는지를 이번 장애와 나란히 비교할 수 있습니다. 다음은 서버 이름을 가린 응답 예시입니다.

> *유사 Incident O건*
>
> 1. *WAS C 응답 지연 — 원인: DB 커넥션 풀 고갈. 조치: 풀 크기 조정(결재 후 실행). 재발방지: 커넥션 풀 사용률 임계치 보고서 추가.*
> 2. *WAS C 응답 지연 — 원인: 특정 배치 작업과 업무 시간대 겹침. 조치: 배치 시각 변경. 재발방지: 배치 일정 점검 항목 추가.*
>
> *원인 후보: 1번 사례와 증상이 가장 가깝습니다. 근거: 이번 장애의 커넥션 풀 사용률 추이(OPENMARU APM). 제안 조치: DB 풀 크기 변경 — **결재 필요**. 자동 처리 항목: 담당자 알림 발송, Incident 케이스 저장.*

원인 후보와 확정 원인의 구분

유사 사례 검색은 원인 후보를 좁히는 도구이고, 원인을 확정하는 일은 담당자가 합니다. 증상이 같아도 원인이 다를 수 있다는 점은 2023년 행정전산망 장애에서 초기 지목된 원인과 최종 발표된 원인이 달랐던 사례에서도 확인됩니다[S05]. OpsKnow Agent의 응답은 「원인 후보」라는 표현을 쓰고, 증상이 가장 가까운 사례와 그 근거를 함께 제시합니다. 담당자가 원인을 확정하면 그 결과가 원인, 조치, 재발방지를 포함한 7개 필드의 새 Incident 케이스로 OpsKnow에 쌓이고, 다음 장애의 검색 대상이 됩니다.

장애 이력이 운영 지식으로 쌓이는 과정

이 순환이 반복되면 기관의 장애 대응 경험이 담당자 개인의 기억에 머물지 않고 검색 가능한 운영 지식이 됩니다. 반면 장애 보고서를 문서 파일로만 보관하면 같은 증상이 다시 나왔을 때 담당자가 과거 파일을 일일이 찾아 읽어야 합니다. 의사결정권자가 볼 지표는 장애 처리 건수보다, 확정 원인과 재발방지 조치가 빠짐없이 Incident 케이스로 남았는지입니다.

9.4.2 결재가 필요한 조치는 제안에서 멈추는 응답 경계

자동 처리와 결재 대상의 구분 기준

장애 직후 응답에서 OpsKnow Agent가 스스로 처리하는 일과 제안에서 멈추는 일은 미리 정해진 경계를 따릅니다. 알림 발송, OpsKnow 저장, 임계치 보고서 생성은 서비스 상태를 바꾸지 않으므로 OpsKnow Agent가 바로 처리합니다. 반면 임계치 조정, WAS 재시작, 인스턴스 증설·감축, 세션 강제 종료, 외부 시스템 재시작, DB 풀 크기 변경은 서비스에 직접 영향을 주므로 사람의 결재가 필요합니다. 아래 표는 8.3절의 경계 구분이 장애 직후 응답에서 어떻게 표시되는지를 대응시킨 것입니다.

조치	구분	장애 직후 응답에서의 처리
담당자 알림 발송	자동	응답과 함께 즉시 처리, 처리 결과 표시
Incident 케이스 저장	자동	응답과 함께 즉시 처리, 저장 결과 표시
임계치 보고서 생성	자동	보고서 생성 후 첨부
임계치 조정	결재	조정안과 근거만 제시하고 멈춤
WAS 재시작	결재	영향 범위와 근거만 제시하고 멈춤
인스턴스 증설·감축	결재	필요 규모 근거만 제시하고 멈춤
세션 강제 종료	결재	대상 세션 목록만 제시하고 멈춤
외부 시스템 재시작	결재	영향 범위와 근거만 제시하고 멈춤
DB 풀 크기 변경	결재	변경안과 근거만 제시하고 멈춤

결재 표시가 붙은 제안의 처리 흐름

결재가 필요한 조치는 응답에 「결재 필요」라는 표시와 근거를 달고 제안 단계에서 멈춥니다. 이 경계는 응답 문장을 만드는 모델의 판단에 맡기지 않고 코드에 정해 두었기 때문에, 질문을 어떻게 바꾸어 표현해도 결재 대상 조치가 자동으로 실행되지 않습니다. 따라서 기관은 표준운영절차의 변경관리 규칙을 그대로 유지한 채 AI 응답을 운영에 쓸 수 있고, 행정예고된 안정성 고시가 요구하는 장애 관리 기준과도 충돌하지 않습니다[S02]. 결재 6종은 9.3.2와 같은 승인 코드 절차, 곧 승인 코드 입력, 적용 직전 재확인, 적용, 재점검을 거쳐 실행되고, 그 결과는 9.3.2의 승인 기록과 같은 형식으로 OpsKnow에 남습니다.

AAS 위 OpsKnow Agent의 판정·결정·설명 분리

OpsKnow Agent는 OPENMARU AAS 위에서 동작하는 AI 에이전트이고, 문장 해석·명령 선택·판정과 결재 경계를 모두 결정적 코드에 두었습니다. 그래서 네 장면 어디에서나 판정은 코드가 하고 결정은 사람이 하며 설명은 AI가 하는 분리가 같은 규칙으로 유지되고, 응답 문장을 만드는 모델이 그 경계를 바꾸지 못합니다. 의사결정권자가 차년도 사업 요구사항을 정할 때 확인할 조건은 세 가지로 모입니다. 자연어 요청에 대한 응답이 판정 근거와 증적을 함께 내는지, 서버를 바꾸는 조치가 1회용 승인 코드와 적용 직전 재확인을 거치는지, 결재 대상 조치가 제안에서 멈추고 그 결과가 운영 지식으로 쌓이는지입니다. 이 세 조건을 모두 갖춘 구성이 담당자 업무를 작성에서 검토로 바꾸며, 같은 기준으로 다른 방식과 비교한 결과는 10장의 비교표에 있습니다.

10장: 경쟁 방식 비교와 도입 효과 — A사·B사 방식과 다른 판정·증적 범위, 도입 뒤 측정할 지표

행정안전부는 2026년 8월 27일 공공 정보시스템 15,033개의 등급을 A1 73개, A2 157개, A3 671개, A4 14,132개로 확정했고, 등급 기준을 기관 규모에서 국민이 받는 서비스의 중요도로 바꾸었습니다[S01]. 등급이 정해지면 기관은 고시 제2026-23호가 요구하는 예방점검과 장애 관리를 차년도 사업에 담아야 하는데, 행정예고 단계에서 알려진 안정성 기준만 46개이고 서비스 수준 협약(SLA) 체결도 의무로 들어 있습니다[S02]. 같은 의무를 두고 여러 업체가 서로 다른 방식의 예방점검 솔루션을 제안하고 있으며, 발주 기관은 그 방식들이 점검 결과의 판정과 증적까지 책임지는지를 가려서 제안요청서에 적어야 합니다. 이 장에서는 일상점검 스크립트와 양식 보고에 초점을 둔 A사·B사 방식과 OPENMARU OpsKnow 기반 지능화 방식을 일곱 가지 기준으로 비교하고, 도입 전후 효과를 확인할 측정 지표와 차년도 사업 준비 순서를 결재 문서에 옮길 수 있는 형태로 정리했습니다.

10.1 일상점검 스크립트와 양식 보고 중심의 A사·B사 방식과 지능화 방식의 7개 기준 비교

2024년 10월 마련된 범정부 예방점검체계는 일상점검·특별점검·구조진단 3개 분야, 8개 점검항목으로 나뉘고 장애관리·변경관리 같은 표준운영절차(SOP)가 함께 정해졌습니다[S07]. 이 체계에서 일상점검은 CPU·메모리·디스크 상태와 메인 화면 접속 여부, 인증서 유효기간처럼 장비마다 매일 확인하는 항목이어서, 업체들이 내놓은 공공 예방점검 자료도 대개 이 영역의 자동화를 중심으로 다룹니다. 하지만 점검표를 빠짐없이 채우는 일과 그 결과를 감사 때 근거로 설명하는 일은 서로 다른 기능을 요구하므로, 비교 기준도 수집 범위보다 판정과 증적의 범위를 먼저 봐야 합니다. 아래 비교는 A사와 B사가 2026년 하반기에 배포한 공공 예방점검 자료와 OPENMARU 제품 구성을 같은 기준으로 놓고 본 결과입니다.

10.1.1 AI 판정 유무부터 폐쇄망 AI까지 7개 비교 기준

7개 비교 기준의 출발점

비교 기준은 감사에서 실제로 받는 질문에서 출발해야 제안요청서의 평가 항목으로 그대로 쓸 수 있습니다. 고시 제2026-23호 행정예고안은 예방점검과 함께 장애 관리 기본계획, 표준운영절차, 서비스 수준 협약, 주기 백업과 원거리 소산 백업, 연 1회 이상의 실전환 훈련까지 요구하므로 점검 기능 하나만 떼어 비교하면 범위를 좁게 보게 됩니다[S02]. 아래 일곱 기준은 점검 결과의 판정 방식, 점검 분야의 범위, 장애·변경 관리, 보안 취약점 조치, 이력 보관, 배포 형태, 폐쇄망에서의 AI 사용 가능성을 차례로 다룹니다. 기관은 이 기준으로 차년도 사업에서 어느 범위까지 한 체계로 묶을지를 정할 수 있습니다.

기준	확인하는 내용	제안요청서 평가 질문 예
AI 판정·설명	점검 수치를 정상·주의·이상으로 판정한 근거를 문장으로 남기는가	판정 근거를 원자료와 함께 다시 보여줄 수 있는가
특별점검·구조진단 범위	일상점검 외 분야를 제품 안에서 다루는가	특별점검 결과와 구조진단 기록을 같은 저장소에 두는가
장애·변경 관리	장애와 변경 작업을 점검 이력과 연결하는가	장애 기록에서 직전 점검 결과로 이동할 수 있는가

기준	확인하는 내용	제안요청서 평가 질문 예
보안 취약점 조치	KISA·CVE 권고 대응을 대상 선별부터 재점검까지 다루는가	조치 승인과 적용 결과가 기록으로 남는가
이력 보관 기간	점검·장애 이력을 몇 해에 걸쳐 비교할 수 있는가	감사 대상 기간 전체의 이력을 조회할 수 있는가
배포 형태	설치형·구독형 등 기관 환경에 맞는 형태를 고를 수 있는가	기관 보안 정책에 맞는 배포 형태를 제시하는가
폐쇄망 AI	외부 연결 없이 AI 기능이 동작하는가	외부 LLM 호출 없이 판정·보고가 가능한가

판정과 폐쇄망 기준의 비중

일곱 기준 가운데 AI 판정·설명과 폐쇄망 AI는 일상점검 자동화 범위만 비교해서는 드러나지 않는 항목입니다. 특히 국가정보원이 2025년 9월 공개한 N2SF(국가 망 보안체계) 보안가이드라인 정식판은 업무 정보를 기밀·민감·공개로 나누고 등급별 통제를 달리하도록 했으므로, 운영 데이터를 다루는 AI 기능은 이제 망 분류와 함께 평가해야 합니다[S08]. 예를 들어 서버 구성 정보와 장애 로그가 외부 LLM으로 나가는 구조라면 판정 기능이 좋아도 기관 보안 정책에서 막힐 수 있습니다. 두 기준은 기능 비교 항목이면서 동시에 도입 가능 여부를 가르는 선행 조건으로 다루는 편이 맞습니다.

10.1.2 특별점검·구조진단 범위와 이력 보관 기간에서 드러나는 A사·B사의 차이

A사·B사 공통의 일상점검 중심 범위

A사와 B사의 공공 예방점검 자료는 모두 일상점검 항목을 매일 스크립트로 점검하고 정해진 양식의 보고서를 남기는 데 초점을 둡니다. 두 자료 모두 AI나 LLM을 이용한 판정·설명 기능을 언급하지 않으며, 장애·변경·문제 관리는 외부 ITSM 도구와 연동하는 방식으로 넘기고 있습니다. 보안 취약점 진단과 패치 적용은 두 회사 모두 제품 범위 밖으로 두고 있어, 보안공지 대응은 별도 사업이나 담당자의 수작업으로 남게 됩니다. 다만 두 자료 모두 「임계치 설정만으로는 감사 대응이 어렵다」는 점을 강조하므로, 판정 근거와 증적이 필요하다는 문제의식 자체는 업계가 공유하고 있다고 볼 수 있습니다.

특별점검·구조진단과 이력 보관의 차이

두 회사가 갈리는 부분은 일상점검 밖의 분야입니다. A사 자료는 특별점검과 구조진단을 범위 밖으로 두고, 점검 이력의 보관 기간을 제한하며, 설치형으로만 제공합니다. 반면 B사 자료는 특별점검 결과를 「기록 지원」 수준으로 받아 두고, 구조진단은 외부 전문 기관에 맡기는 방식을 제시합니다. 어느 방식을 고르더라도 특별점검·구조진단 결과와 일상점검 이력이 한 저장소에서 이어지지 않으며, 감사 때는 담당자가 여러 출처의 기록을 손으로 맞춰야 합니다.

구분	A사 방식	B사 방식
일상점검	스크립트 점검 + 양식 보고	스크립트 점검 + 양식 보고
AI·LLM 판정	언급 없음	언급 없음
특별점검	범위 밖	결과 기록 지원

구분	A사 방식	B사 방식
구조진단	범위 밖	외부 전문 기관 위탁
장애·변경 관리	ITSM 연동에 맡김	ITSM 연동에 맡김
보안 취약점 조치	범위 밖	범위 밖
이력 보관	보관 기간 제한	자료에 명시 없음
배포 형태	설치형 전용	자료 기준 확인 필요

발주 기관이 떠안는 남은 일

이 차이는 기관이 사업을 마친 뒤 스스로 떠안는 업무의 양으로 이어집니다. 예를 들어 2023년 11월 행정전산망 장애의 원인은 라우터의 케이블 연결 모듈에 있는 포트 일부 이상으로 발표됐고, 11월 18일 오전 9시 서비스를 재개한 뒤 11월 19일 포트를 교체하고서야 정상화됐습니다[S04]. 원인 장비를 특정하기까지 걸린 시간은 점검 이력과 장애 기록이 얼마나 이어져 있는지에 좌우될 수 있으며, 당시에도 초기에는 다른 장비를 원인으로 지목했습니다[S05]. 일상점검만 자동화한 체계에서는 수집 기록은 많아도 장애 시점의 판단을 뒷받침할 연결 기록이 부족할 수 있습니다.

10.1.3 판정·증적·보안 취약점 조치·운영 지식 축적을 한 구조로 묶는 OPENMARU 방식

수집·판정·축적을 나눈 제품 구성

OPENMARU 방식은 수집, 판정과 조치, 축적을 서로 다른 구성 요소에 맡기고 그 결과를 한 저장소로 모읍니다. 먼저 OPENMARU APM이 OS·미들웨어·DBMS 지표와 트랜잭션 기록을 모으는데, 판정에 쓰는 원자료가 이미 이 제품에 쌓이기 때문에 예방점검의 출발점이 됩니다. 이어 OpsKnow Agent가 그 지표를 정상·주의·이상으로 판정하고 근거 문장을 붙여 보고하며, 서버를 바꾸는 조치는 사람의 승인을 거쳐서만 실행합니다. 마지막으로 OPENMARU OpsKnow가 Incident 케이스, 예방점검 결과, 일·주·월·분기·연 단위 운영 보고를 운영 지식으로 쌓아 다음 판정과 감사 대응에 다시 씁니다.

판정 근거와 승인 코드로 남는 증적

OpsKnow Agent에서 판정과 명령 선택은 정해진 규칙을 따르는 코드가 맡고, LLM은 사용자의 요청 문장을 넘기고 결과를 설명하는 역할만 합니다. 이 구조에서는 같은 입력에 같은 판정이 나오므로 감사 때 판정 근거를 원자료와 함께 다시 보여 줄 수 있습니다. OS 패치·보안 조치와 결재 6종처럼 서버를 바꾸는 조치는 한 번만 쓸 수 있고 유효 시간이 짧은 승인 코드가 있어야 실행되고, 적용 직전에 서버 상태를 다시 확인해 점검 때와 다르면 멈춥니다. 결과적으로 「누가 언제 무엇을 승인했고 적용 뒤 결과가 어떠했는가」라는 감사 질문에 승인 코드와 재점검 결과가 그대로 증적이 됩니다.

보안 취약점 조치와 폐쇄망 구성

보안 취약점 대응도 같은 흐름에 들어갑니다. 한국인터넷진흥원(KISA)이 운영하는 보호나라&KrCERT/CC와 KNVD 취약점 정보포털에는 보안공지와 CVE 검색 메뉴가 있으며, OpsKnow Agent는 이런 권고를 받아 대상 서버를 고르고 승인 뒤 적용과 재점검까지 잇습니다[S10]. 점검 범위는 OS 패키지에 그치지 않고 미들웨어·DBMS 취약점까지 포함하며, 어느 쪽이든 조치 적용은 사람의 승인 뒤에 이뤄집니다. OpsKnow Agent는 확실하지 않은 상태를 「최신」으로 보고하지 않고 이미 조치됨, 설치돼 있지 않음, 지원 종료 OS처럼 상태를 나눠

기록합니다. 근거 검색(RAG)과 30B 수준 내부 sLLM 서빙은 OPENMARU CogentAI가 맡으며, CogentAI는 GPU 1대 폐쇄망에서 외부 연결 없이 모델을 돌리므로 운영 데이터가 기관 밖으로 나가지 않는 소버린 AI Level 3 구성의 기반이 됩니다. 아울러 OpsKnow Agent는 예약(주기) 실행, 도구별 승인 요청, 승인자 계정·시각 기록을 제공하는 OPENMARU AAS 위에서 실행되며, 자동으로 하는 일과 결재가 필요한 일의 경계를 결정적 코드에 정해 두어 응답 문장을 만드는 모델이 그 경계를 넘지 못하게 합니다.

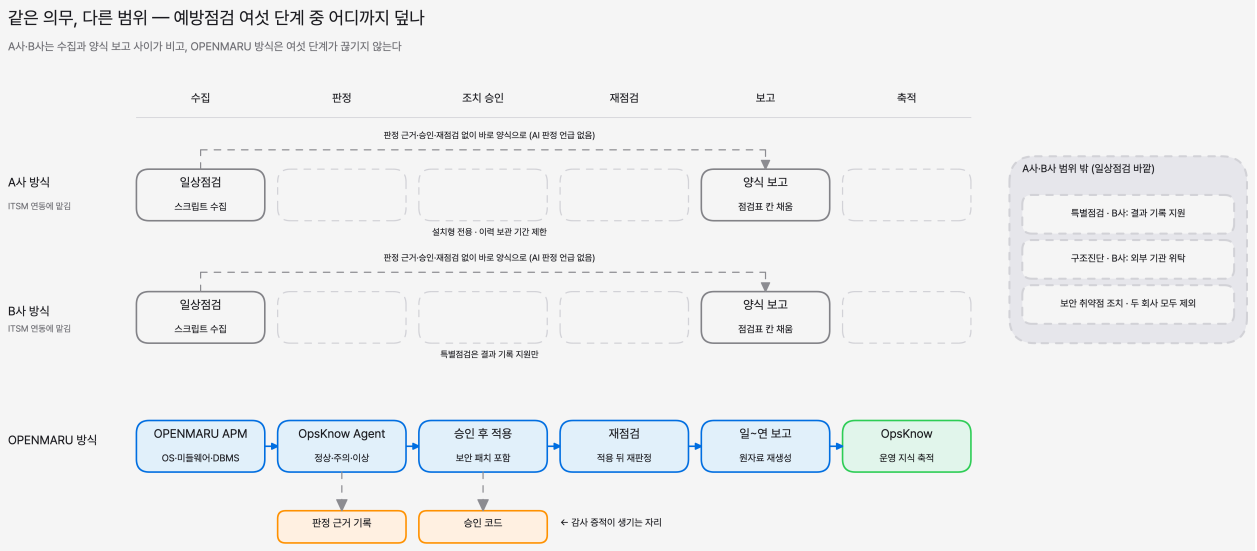


그림. A사·B사 방식과 OPENMARU 방식이 덮는 예방점검 범위 비교

운영 현황을 보여 주는 화면 구성

축적한 결과는 화면으로도 확인할 수 있어야 의사결정권자가 상황을 판단할 수 있습니다. Wallboard는 운영실 벽 화면에 예방점검 결과, 지연 트랜잭션, 이상 그룹, 승인 대기를 띄우는 상황판이어서, 주의·이상 판정이 나왔을 때 운영 인력이 같은 화면을 보고 대응 순서를 정할 수 있습니다. OPENMARU Dashboard는 여러 기관이나 여러 센터의 APM 현황을 한 화면에 모으므로, 등급이 높은 A1·A2 시스템이 흩어져 있는 기관에서도 전체 점검 현황을 한 번에 확인할 수 있습니다[S01]. OPENMARU 방식의 차이는 개별 기능 수보다 판정, 증적, 보안 취약점 조치, 운영 지식 축적이 끊기지 않고 이어진다는 구조에 있습니다.

10.2 도입 전후를 비교하는 측정 지표 — 점검 소요 시간부터 증적 누락 건수까지

예방점검 지능화의 효과는 업체가 약속한 수치보다 기관이 도입 전에 직접 잰 기준선과 도입 뒤 같은 방법으로 잰 값을 비교할 때 설득력을 가집니다. 고시 제2026-23호 행정예고안이 서비스 수준 협약 체결을 의무로 넣었기 때문에, 운영 사업의 성과도 앞으로는 협약에 적힌 지표로 확인하게 됩니다[S02]. 아래 지표는 효과 수치를 미리 적지 않고, 무엇을 어떻게 재는지와 데이터를 어디서 가져오는지만 정했습니다. 지표 정의가 같아야 도입 전후 값이 비교 가능하므로, 사업 성과관리 계획서에는 정의와 측정 방법을 함께 옮기는 편이 좋습니다.

10.2.1 도입 전 기준선 측정 절차와 지표별 데이터 출처

기준선 측정 시점과 기간

기준선은 차년도 사업 착수 전에 현재 방식 그대로 일정 기간을 재서 정합니다. 기간은 최소 한 번의 분기 보고 주기를 포함해야 보고서 작성 시간까지 잴 수 있고, 그 기간 안에 보안공지 대응과 장애 처리가 한 건 이상 들어 있어야 권고 조치 기간과 원인 규명 시간도 값을 얻을 수 있습니다. 예를 들어 기존 점검 대장, 결재 문서, 장애 보

고서, 보안공지 처리 기록을 모아 각 지표의 시작 시각과 종료 시각을 뽑으면 별도 도구 없이도 기준선을 만들 수 있습니다. 점검 분야는 2024년 예방점검체계의 일상점검·특별점검·구조진단 구분을 그대로 써야 도입 뒤 기록과 같은 분류로 비교할 수 있습니다[S07].

지표별 데이터 출처

도입 전과 도입 뒤의 데이터 출처는 다르지만, 같은 지표를 같은 시각 정의로 뽑는다는 원칙은 같습니다. 도입 전에는 수작업 점검 대장과 결재 시스템, 장애 보고서가 주된 출처이고, 도입 뒤에는 OPENMARU APM 수집 기록과 OpsKnow에 저장된 Incident 케이스, 예방점검 결과, 승인 기록이 출처가 됩니다. OPENMARU APM은 OS·미들웨어·DBMS 지표를 수집 시각과 함께 남기는 기능을 제공하므로, 점검 소요 시간의 시작과 종료 시각을 기록에서 바로 얻을 수 있습니다. Incident 케이스에는 원인·조치·재발방지와 관련 트랜잭션 ID, 자동 원인 분석에 걸린 시간이 함께 남으므로, 원인 규명 시간은 사람이 따로 적지 않아도 기록에서 바로 뽑을 수 있습니다. 따라서 도입 뒤 측정은 대부분 별도 조사 업무 없이 운영 기록을 조회하는 것으로 처리할 수 있습니다.

단계	하는 일	데이터 출처(도입 전)	데이터 출처(도입 뒤)
1. 범위 확정	측정 대상 시스템과 등급, 점검 분야 확정	등급 통보 자료	등급 통보 자료
2. 기록 수집	측정 기간의 점검·장애·보안공지·보고 기록 모음	점검 대장, 결재 문서, 장애 보고서	APM 수집 기록, OpsKnow 저장 기록
3. 시각 추출	지표별 시작·종료 시각 추출	문서 작성·결재 일시	판정·승인·재점검 기록 시각
4. 값 산출	지표 산식에 따라 값 계산	담당자 수작업	기록 조회
5. 기준선 확정	결과를 성과관리 계획서에 기재	사업 담당 부서	사업 담당 부서

기준선을 먼저 재야 하는 이유

기준선을 재지 않고 도입하면 사업 종료 때 효과를 숫자로 보여 줄 방법이 없고, 결국 업체 자료의 예시값을 성과로 옮겨 적게 됩니다. 반면 도입 전에 재 둔 값이 있으면 도입 뒤 같은 정의로 잰 값과 바로 비교할 수 있어 차년도 예산 심의에서도 근거로 쓸 수 있습니다. 등급 확정으로 A1·A2 시스템의 관리 수준이 높아진 만큼, 등급별로 기준선을 나눠 두면 중요 시스템에서 생긴 변화를 따로 보여 줄 수 있습니다[S01].

10.2.2 점검 소요 시간과 권고 조치 기간, 원인 규명 시간, 보고서 작성 시간의 정의

다섯 가지 측정 지표의 정의

측정 지표는 점검 소요 시간, 권고 조치 기간, 원인 규명 시간, 보고서 작성 시간, 증적 누락 건수 다섯 가지로 정합니다. 앞의 네 지표는 업무 속도를 보여 주고, 마지막 증적 누락 건수는 감사 대응 품질을 보여 줍니다. 다만 속도 지표만 보면 점검을 빨리 끝내고 기록을 빠뜨리는 경우를 구분하지 못하므로, 증적 누락 건수를 함께 두어야 단순 자동화와 지능화의 차이가 드러납니다. 다섯 지표는 한 묶음으로 보고하고 하나만 떼어 성과로 내세우지 않는 편이 좋습니다.

지표	정의	산식	시작 시각	종료 시각
점검 소요 시간	한 주기의 일상점검을 판정까지 마치는 데 걸린 시간	종료 − 시작, 주기별 평균	점검 시작	마지막 항목 판정 기록
권고 조치 기간	보안공지 발표부터 대상 서버 조치·재점검 완료까지	종료 − 시작, 권고별	보안공지 게시일	재점검 결과 기록
원인 규명 시간	장애 발생부터 원인이 기록될 때까지	종료 − 시작, 장애별	장애 인지	원인 필드 기록
보고서 작성 시간	보고 주기 마감부터 결재 상신까지	종료 − 시작, 보고서별	주기 마감	결재 상신
증적 누락 건수	점검·조치 기록 중 판정 근거·승인·재점검 기록이 빠진 건수	누락 건수 ÷ 전체 건수	측정 기간 시작	측정 기간 종료

증적 누락 건수의 판정 기준

증적 누락 건수는 다섯 지표 가운데 정의가 가장 엄격해야 합니다. 한 건의 점검 기록은 판정 결과, 판정 근거, 조치가 있었다면 승인 기록과 재점검 결과까지 갖춰야 완전한 기록으로 셉니다. 예를 들어 「디스크 사용률 주의」라는 판정만 있고 어떤 수치를 어떤 기준과 비교했는지가 빠졌다면 판정 근거 누락으로 세고, 패치를 적용했는데 승인자와 승인 시각이 없으면 승인 기록 누락으로 셉니다. 이 기준은 2.4절 자가진단의 다섯 문항과 같은 내용을 수치로 바꾼 것이므로, 자가진단 결과와 기준선 값을 함께 보면 기관이 어느 단계에 있는지를 더 분명하게 설명할 수 있습니다.

시각 정의를 고정해야 하는 이유

지표 값은 시작 시각과 종료 시각을 어디로 잡느냐에 따라 크게 달라집니다. 예를 들어 권고 조치 기간을 「담당자가 공지를 확인한 날」부터 재면, 공지를 늦게 확인할수록 기간이 짧게 나오는 왜곡이 생깁니다. 권고 조치 기간의 시작은 한국인터넷진흥원 보호나라에 보안공지가 게시된 날로 고정하고, 종료는 대상 서버의 재점검 결과가 기록된 시각으로 고정합니다[S10]. 시각 정의를 표에 적어 두면 사업자가 바뀌어도 같은 방법으로 다시 잴 수 있습니다.

10.2.3 사업 성과관리 계획서에 옮기는 측정 지표 표

계획서 양식에 맞춘 지표 표

사업 성과관리 계획서는 보통 지표명, 측정 방법, 데이터 출처, 측정 주기, 목표값을 칸으로 받습니다. 여기서 목표값은 기준선을 잰 뒤에 정해야 하므로 계획서 첫 판에서는 「기준선 대비 개선」으로 적고, 기준선 측정이 끝나면 발주 기관과 사업자가 협의해 채우는 편이 안전합니다. 측정 주기는 고시 제2026-23호 행정예고안이 요구하는 장애 관리 연간 시행계획과 맞추면 연간 보고에 그대로 쓸 수 있습니다[S02]. 아래 표는 그 양식에 맞춰 다섯 지표를 옮긴 예시입니다.

지표명	측정 방법	데이터 출처	측정 주기	목표값 기재 방식
점검 소요 시간	주기별 점검 시작~마지막 판정 기록 평균	APM 수집 기록, OpsKnow 예방점검 결과	월	기준선 측정 뒤 협의
권고 조치 기간	보안공지 게시일~재점검 결과 기록	보호나라 공지, OpsKnow 조치 기록	권고별·분기 집계	기준선 측정 뒤 협의
원인 규명 시간	장애 인지~원인 기록	OpsKnow Incident 케이스	장애별·분기 집계	기준선 측정 뒤 협의
보고서 작성 시간	주기 마감~결재 상신	OpsKnow 운영 보고, 결재 기록	보고서별	기준선 측정 뒤 협의
증적 누락 건수	누락 건수 ÷ 전체 점검·조치 건수	OpsKnow 예방점검 결과, 승인 기록	분기	기준선 측정 뒤 협의

지표를 보고하는 방법

지표는 분기마다 다섯 개를 한 표로 보고하고, 등급별로 나눈 값을 함께 붙이면 의사결정권자가 중요 시스템의 변화를 따로 볼 수 있습니다. OpsKnow는 운영 보고와 함께 그 원자료인 예방점검 결과와 Incident 케이스를 같은 저장소에 두므로, 감사 때 특정 분기의 지표를 다시 계산해 보여 달라는 요청에도 같은 원자료에서 값을 다시 낼 수 있습니다. 지표가 나빠진 분기는 해당 Incident 케이스와 점검 기록으로 바로 이어서 원인을 설명할 수 있어야 하며, 이 연결이 바로 단순 자동화 체계에서 얻기 어려운 증적입니다. 성과관리 계획서에는 지표 표와 함께 「지표 값은 원자료에서 다시 산출할 수 있어야 한다」는 조건을 적어 두는 편이 좋습니다.

10.3 차년도 사업 준비 순서 — 단계 자가 판정, 요구사항 선택, OpsKnow Agent 시연 요청

차년도 예방점검 사업은 기관의 현재 단계를 판정하고, 그 단계에 맞는 요구사항을 고르고, 제안요청서에 넣기 전에 실제 동작을 시연으로 확인하는 세 단계로 준비합니다. 등급 확정으로 A1 73개, A2 157개 시스템이 높은 관리 수준을 요구받게 됐으므로, 같은 기관 안에서도 시스템 등급에 따라 고를 요구사항의 범위가 달라질 수 있습니다[S01]. 단계 판정은 기관 전체가 아니라 등급별 대상 시스템 묶음 단위로 하는 편이 정확합니다. 아래 순서는 일정표보다 판단 순서를 정한 것이며, 각 단계의 결과가 다음 단계의 입력이 됩니다.

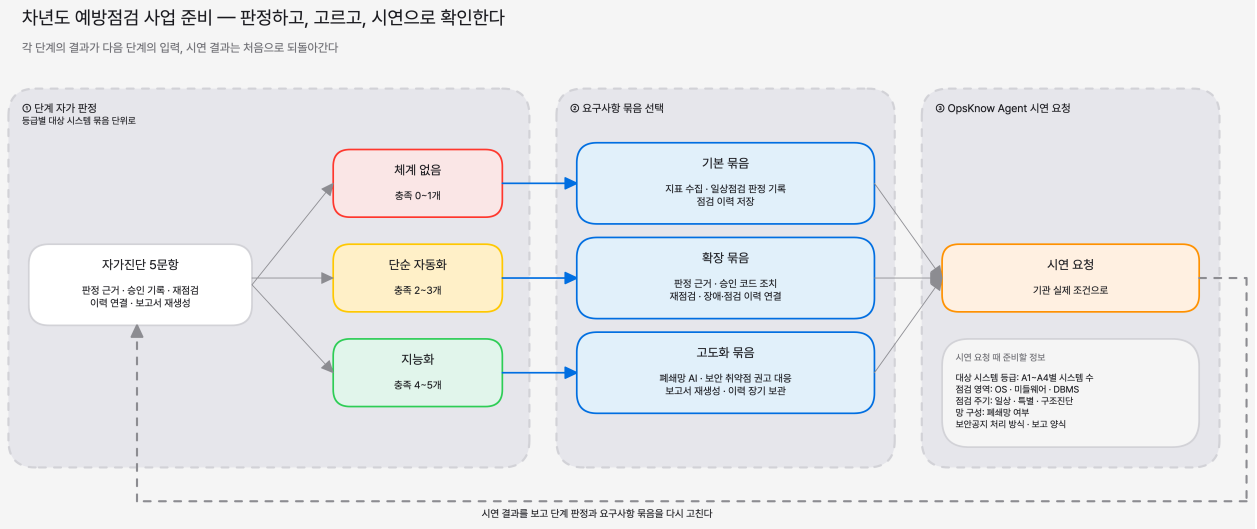


그림. 자가진단 단계 판정에서 요구사항 묶음 선택, OpsKnow Agent 시연 요청으로 이어지는 차년도 사업 준비 흐름

10.3.1 2.4 자가진단 결과별로 고르는 차년도 요구사항 묶음

자가진단 단계별 요구사항 묶음

2.4절 자가진단은 판정 근거, 조치 승인 기록, 재점검 결과, 장애와 점검 이력의 연결, 원자료에서의 보고서 재생성을 묻는 다섯 문항으로 기관 단계를 체계 없음, 단순 자동화, 지능화로 나눕니다. 각 단계는 부족한 기능이 다르므로 요구사항도 단계에 맞춰 묶어야 사업 범위가 지나치게 넓어지거나 좁아지지 않습니다. 예를 들어 체계 없음 단계에서 바로 보안 취약점 자동 조치까지 요구하면 기초 수집 체계가 없어 판정 근거를 만들 수 없고, 반대로 단순 자동화 단계에서 수집만 다시 발주하면 증적 문제는 그대로 남습니다. 아래 표처럼 단계별로 먼저 채울 요구사항을 정하고, 남는 예산 범위에서 다음 단계 항목을 더하는 순서가 합리적입니다.

자가진단 단계	주된 부족 사항	먼저 고를 요구사항 묶음	함께 검토할 구성
체계 없음	지표 수집과 점검 기록 자체가 없음	OS·미들웨어·DBMS 지표 수집, 일상점검 판정 기록, 점검 이력 저장	OPENMARU APM, OpsKnow
단순 자동화	점검표는 채워지나 판정 근거·승인·재점검 기록이 없음	판정 근거 기록, 승인 코드 기반 조치, 재점검 결과 기록, 장애·점검 이력 연결	OpsKnow Agent, AAS
지능화	기록은 있으나 폐쇄망 AI·보안 취약점 조치·보고 재생성이 부분적임	폐쇄망 AI 판정·보고, 보안 취약점 권고 대응, 원자료 기반 보고서 재생성, 이력 장기 보관	CogentAI, Wallboard, OPENMARU Dashboard

등급에 따른 요구사항 조정

같은 단계에 있는 기관이라도 대상 시스템의 등급에 따라 요구사항의 깊이를 조정해야 합니다. A1·A2 시스템은 장애가 국민 서비스에 바로 영향을 주는 시스템이므로 원인 규명 시간과 권고 조치 기간을 줄이는 요구사항을 우선하고, A4 시스템처럼 수가 많은 묶음은 수집과 판정 기록을 빠짐없이 남기는 요구사항을 우선하는 편이 맞습

니다[S01]. 행정예고안이 중요 시스템 장애를 행정안전부 디지털안전상황실에 즉시 보고하도록 하고 있으므로, 높은 등급 시스템에는 장애 기록과 보고 양식을 바로 이어 주는 요구사항이 필요합니다[S02]. 등급별로 묶음을 나누면 한정된 예산을 중요 시스템에 먼저 배분할 수 있습니다.

요구사항을 고르기 전에 대상 시스템의 등급이 확정 통보된 등급인지부터 확인해야 합니다. 등급이 바뀌면 의무로 수행해야 하는 점검의 범위가 함께 바뀌므로, 등급이 정해지지 않은 상태에서 요구사항을 넣으면 의무가 아닌 일에 예산을 쓰거나 반대로 의무 항목을 빠뜨릴 수 있습니다. 시스템이 여럿인 기관은 시스템별로 확정 등급을 표에 먼저 적고, 그 표를 요구사항 묶음 선택의 첫 열로 두는 방식이 실무에서 다루기 쉽습니다.

10.3.2 판정 근거 기록과 승인 코드, 폐쇄망 AI를 담은 제안요청서 문안 예시

제안요청서 기능 요구사항 문안

요구사항 묶음을 고른 뒤에는 제안요청서의 기능 요구사항 형식으로 옮겨야 평가 위원이 제안서를 같은 기준으로 비교할 수 있습니다. 문안은 「무엇을 제공해야 한다」와 함께 「어떻게 확인한다」를 적어야 제안사가 기능 이름만 나열하는 답변을 막을 수 있습니다. 아래 문안은 판정 근거 기록, 승인 기록, 폐쇄망 AI, 보고서 재생성, 이력 보관 기간을 다섯 개 요구사항으로 나눈 예시이며, 기관 상황에 맞게 대상 범위와 보관 기간을 채워 쓰면 됩니다.

- **판정 근거 기록**: 예방점검 결과는 항목별로 정상·주의·이상 판정과 판정에 쓴 수치, 비교 기준을 함께 기록해야 하며, 같은 원자료로 다시 판정하면 같은 결과가 나와야 한다. 확인 방법은 임의 점검 기록 재판정 시연으로 한다.
- **조치 승인 기록**: 서버 설정·패키지를 바꾸는 조치는 담당자가 발급받은 1회용 승인 코드를 입력해야 실행돼야 하며, 적용 직전 서버 상태를 다시 확인해 점검 때와 다르면 실행을 멈춰야 한다. 승인자·승인 시각·적용 결과·재점검 결과를 기록해야 한다.
- **폐쇄망 AI**: AI 기반 판정 설명과 보고서 작성 기능은 외부 LLM 호출 없이 기관 내부 GPU 장비에서 동작해야 하며, 운영 데이터가 외부로 송신되지 않음을 시험으로 보여야 한다.
- **보고서 재생성**: 일·주·월·분기·연 운영 보고서는 저장된 원자료에서 다시 생성할 수 있어야 하며, 감사 요청 시 지정한 기간의 보고서를 다시 만들어 원본과 대조할 수 있어야 한다.
- **이력 보관 기간**: 예방점검·장애·조치 이력은 기관이 정한 감사 대상 기간 이상 보관하고 같은 화면에서 조회할 수 있어야 한다.

문안에 담은 판단 기준

다섯 문안은 모두 「감사 때 다시 보여 줄 수 있는가」라는 하나의 판단 기준을 기능 요구사항으로 바꾼 것입니다. 예를 들어 폐쇄망 AI 요구사항은 N2SF 보안가이드라인이 업무 정보를 등급별로 나눠 통제하도록 한 점을 반영해, 운영 데이터가 외부로 나가지 않는다는 사실을 시험으로 증명하게 했습니다[S09]. 조치 승인 기록 요구사항은 승인 코드와 적용 직전 재확인을 함께 요구하므로, 자동 실행의 편의와 결재 통제를 동시에 갖춘 제안만 통과할 수 있습니다. 이 문안을 쓰면 일상점검 스크립트와 양식 보고만 제공하는 방식은 판정 근거 기록과 폐쇄망 AI 항목에서 차이가 바로 드러납니다.

위탁 운영이나 유지보수 계약의 과업 문안에도 같은 기준을 적용할 수 있습니다. 「모니터링 도구를 도입한다」라고만 적으면 도구가 설치된 것으로 과업이 끝난 것처럼 읽히므로, 점검 항목별 임계치를 설정하고 그 관리 결과와 판정 근거를 정해진 주기로 제출하는 일을 과업으로 적어야 합니다. 이렇게 적어 두면 수행사가 바뀌어도 임계치와 판정 기록이 기관 쪽 저장소에 남고, 다음 사업의 요구사항을 정할 때 그 기록을 기준선으로 다시 쓸 수 있습니다.

10.3.3 시연 요청 때 준비할 대상 시스템 등급과 점검 영역, 점검 주기

시연 요청 때 준비할 정보

OpsKnow Agent 시연은 기관의 실제 조건을 반영할수록 결과를 제안요청서 작성에 바로 쓸 수 있습니다. 시연을 요청할 때는 대상 시스템의 등급, 점검 영역, 점검 주기, 망 구성, 보안공지 처리 방식, 보고 양식을 미리 정리해 전달하는 편이 좋습니다. 점검 영역은 OS, 웹 서버·WAS 같은 미들웨어, DBMS로 나눠 적어야 OPENMARU APM이 어떤 지표를 모으고 OpsKnow Agent가 어떤 판정과 보고로 잇는지를 영역별로 확인할 수 있습니다. 점검 주기는 2024년 예방점검체계의 일상점검·특별점검·구조진단 구분을 따라 적으면 됩니다 [S07].

준비 항목	적을 내용	시연에서 확인할 것
대상 시스템 등급	대상 시스템별 A1~A4 등급과 시스템 수	등급별로 판정·보고 화면이 나뉘는가
점검 영역	OS·미들웨어·DBMS 구성과 제품 종류	영역별 지표 수집과 판정 근거 기록
점검 주기	일상점검·특별점검·구조진단 수행 주기	주기별 점검 결과가 한 저장소에 쌓이는가
망 구성	폐쇄망 여부, 외부 연결 허용 범위	외부 연결 없이 AI 판정·보고가 되는가
보안공지 처리 방식	현재 KISA 보안공지 확인·조치 절차	대상 선별, 승인 코드, 적용, 재점검 흐름
보고 양식	현재 일·주·월·분기·연 보고서 양식	원자료에서 보고서를 다시 만들 수 있는가

시연 결과를 사업 준비에 반영하는 방법

시연이 끝나면 그 결과를 10.3.1의 요구사항 묶음과 10.3.2의 문안에 다시 반영합니다. 예를 들어 시연에서 기관의 보고 양식을 원자료에서 그대로 다시 만들 수 있음을 확인했다면 보고서 재생성 요구사항의 확인 방법을 「기관 양식 재생성 시연」으로 구체화할 수 있습니다. 반대로 기관 망 구성이 예상보다 엄격하다면 폐쇄망 AI 요구사항에 GPU 장비 위치와 외부 연결 차단 시험 방법을 더 자세히 적어야 합니다. 시연을 요구사항 확정 앞에 두면 제안요청서가 실제 운영 조건과 어긋나는 일을 줄일 수 있습니다.

OPENMARU 방식의 기술적 차이

같은 고시 제2026-23호 의무에 대응하더라도 일상점검 스크립트와 양식 보고에 머무는 방식은 점검표를 채우는 단순 자동화 단계에서 멈춥니다. 반면 OPENMARU 방식은 OPENMARU APM이 OS·미들웨어·DBMS 지표를 모으고, OpsKnow Agent가 그 지표를 판정하고 승인 코드를 거쳐 조치와 재점검까지 이으며, OpsKnow가 그 결과를 운영 지식으로 쌓습니다. 이 흐름 전체가 30B 수준 내부 sLLM으로 GPU 1대 폐쇄망에서 동작하는 소버린 AI Level 3 구성 위에 있으므로, 운영 데이터를 기관 밖으로 보내지 않고도 지능화된 판정과 보고를 쓸 수 있습니다. 따라서 차년도 사업을 준비하는 기관은 2.4절 자가진단으로 현재 단계를 판정하고, 단계에 맞는

요구사항 묶음과 제안요청서 문안을 고른 뒤, 그 요구사항이 실제로 동작하는지를 OpsKnow Agent 시연으로 확인하는 순서로 진행하면 됩니다.

Appendix

이 부록은 본문에 인용한 공개 출처와 본문에 나오는 주요 용어의 뜻을 모은 참조 자료입니다.

References

- [S01] 행정안전부 정보자원관리혁신과 (2026). 공공 정보시스템 1만 5천여 개 등급 전면 재조정, 국민 영향 중심으로 개편(보도자료, KDI 경제정보센터 전재). https://eiec.kdi.re.kr/policy/materialView.do?num=285894
- [S02] 보안뉴스 (2026). 「재난에도 멈추지 않는 디지털 행정」 행안부 「정보시스템 안정성 고시」 제정. http://www.boannews.com/news/articleView.html?idxno=142641
- [S03] 디지털데일리 (2026). 행안부, 국가 정보시스템 1만5000여개 등급 재분류…「A1은 재난 시 1시간 내 복구」. https://www.ddaily.co.kr/page/view/2026100215444741131
- [S04] 뉴시스 (2023). 행안부 「지방행정전산서비스 장애원인, 네트워크 영역 장비 불량」. https://www.newsis.com/view/NISX20231125_0002534888
- [S05] 디지털데일리 (2023). 「라우터 포트 불량」 결론 난 행정 전산망 마비. https://www.ddaily.co.kr/page/view/2023112615421232385
- [S06] 보안뉴스 (2025). 행안부, 공공 정보시스템 「응용프로그램(AP) 표준운영절차」 배포. https://m.boannews.com/html/detail.html?idx=136871
- [S07] 보안뉴스 (2025). [2025 장애진단·모니터링·복구 솔루션 리포트] 전자정부법 개정안. https://m.boannews.com/html/detail.html?idx=136314
- [S08] 바이라인네트워크 (2025). 국정원, N²SF 가이드라인 1.0 정식 버전 공개. https://byline.network/2025/09/30-445/
- [S09] 보안뉴스 (2025). 「업무 PC로 AI·클라우드 안전하게」 국정원 N2SF 가이드라인 정식판. https://m.boannews.com/html/detail.html?idx=139553
- [S10] 한국인터넷진흥원 (2026 접속). 보호나라&KrCERT/CC 소개 · KNVD 취약점 정보포털. https://www.kisa.or.kr/1020605 · https://knvd.krcert.or.kr

함께 읽을 자료

- [R1] MSAP.ai (2026). 예방점검 증적 저장소, AI 기반 운영 문서화 도입 기준. https://www.msap.ai/blog-home/blog/preventive-inspection-evidence-repository/

Glossary

용어	정의
30B 수준 모델	모델이 학습으로 정한 값(매개변수)이 약 300억 개인 언어 모델 규모. 본문에서는 GPU 1대로 기관 안에서 돌릴 수 있는 내부 sLLM의 크기를 가리킨다
고시 제2026-23호	본문에서 예방점검 의무의 근거로 다루는 행정안전부 「행정기관 및 공공기관 정보시스템 안정성 고시」. 등급별 안정성 기준과 표준운영절차, 서비스 수준 협약 의무를 담는다
구조진단	정보시스템의 구성과 설계가 장애에 견딜 수 있는지를 긴 주기로 깊게 살피는 점검. 일상점검·특별점검과 함께 예방점검의 3개 분야를 이룬다
근본 원인 분석(RCA, Root Cause Analysis)	장애가 겉으로 드러난 증상이 아니라 그것을 일으킨 처음 원인을 찾아내는 분석
단순 자동화	점검 지표를 모아 점검표와 보고서 양식을 자동으로 채우는 단계. 판정 근거를 남기고 조치까지 잇는 일은 여전히 사람이 따로 챙겨야 한다
등급(A1~A4)	공공 정보시스템을 국민이 받는 서비스의 중요도에 따라 네 단계로 나눈 등급. A1이 가장 중요하며 목표복구시간이 가장 짧다
목표복구시간(RTO, Recovery Time Objective)	장애가 난 뒤 서비스를 다시 정상으로 돌려놓아야 하는 최대 허용 시간
미들웨어	운영체제와 업무 응용프로그램 사이에서 웹 요청 처리, 데이터베이스 연결 같은 공통 기능을 맡는 소프트웨어. 본문에서는 웹 서버(WEB)와 WAS를 가리킨다
서비스 수준 협약(SLA, Service Level Agreement)	운영 주체가 지켜야 할 가용률, 복구 시간 같은 서비스 수준을 수치로 정해 약속하는 협약
소버린 AI	AI 모델과 데이터, 운영 통제권을 기관이 직접 갖고 외부 AI 서비스에 기대지 않는 AI 구성
소버린 AI Level 3	외부 인터넷을 막은 폐쇄망에서 모든 AI 기능이 동작하고, 지정한 GPU 장비 1대에서 기준 성능을 내는 구성 단계. 시험 중 외부 연결 기록이 0건이어야 한다
승인 코드	서버를 바꾸는 조치 전에 관리자가 입력해야 하는 1회용 코드. 정해진 시간 안에 한 번만 쓸 수 있어 승인과 조치가 같은 기록으로 이어진다

용어	정의
일상점검	CPU·메모리·디스크 상태, 서비스 접속 여부와 응답 시간, 인증서 유효기간처럼 매일 확인하는 기본 점검
임계치	지표가 정상인지 이상인지를 가르는 기준값
재점검	조치를 적용한 뒤 같은 항목을 다시 점검해 조치가 제대로 반영됐는지 확인하는 단계
전국 Dashboard	여러 기관의 OPENMARU APM이 보낸 지표와 판정 결과를 한 화면에 모아 상급 기관이 보는 화면. Hub&Spoke 구조에서 Hub를 맡는다
제안요청서(RFP, Request for Proposal)	발주 기관이 사업 범위와 요구사항을 적어 사업자에게 제안을 요청하는 문서
증적	점검을 실제로 했고 어떤 근거로 판정했는지 나중에 보여줄 수 있는 기록
지능화	수집한 지표를 AI가 판정하고 다른 기록과 연관 지어 조치 제안과 보고까지 이어 주며, 그 결과를 다음 판정의 근거로 쌓는 단계
트랜잭션 ID	하나의 요청이 여러 서버를 거쳐 처리되는 경로를 끝까지 따라가도록 붙이는 고유 식별 번호
특별점검	성능·용량처럼 일상점검보다 깊은 항목을 정해진 주기나 특정 시기에 확인하는 점검
폐쇄망	외부 인터넷과의 연결을 끊은 기관 내부 전용 네트워크
표준운영절차(SOP, Standard Operating Procedure)	장애 관리, 변경 관리처럼 반복되는 운영 업무를 누가 하더라도 같은 순서로 처리하도록 정한 절차
한국인터넷진흥원(KISA)	보호나라&KrCERT/CC와 KNVD를 운영하며 보안공지와 취약점 정보를 내는 공공 기관
AAS(AI Agent Studio)	OPENMARU의 AI 에이전트 실행 환경. OpsKnow Agent가 이 위에서 동작하며, 예약(주기) 실행, 도구별 승인 요청, 승인자 계정·시각 기록을 제공한다
APM(Application Performance Management)	응용프로그램의 응답 시간, 처리량, 오류와 서버 자원 지표를 실시간으로 모으고 추적하는 성능 관리 도구
CogentAI	OPENMARU의 AI 플랫폼. 기관 서버 안에서 근거 검색(RAG)과 30B 수준 내부 sLLM을 서비스한다
CVE(Common Vulnerabilities and Exposures)	공개된 보안 취약점마다 붙이는 국제 공통 식별 번호
DBMS	데이터를 저장하고 조회·변경하는 일을 맡는 데이터베이스 관리 시스템

용어	정의
GC(Garbage Collection)	응용프로그램이 더 이상 쓰지 않는 메모리를 자동으로 회수하는 작업. 자주 일어나거나 오래 걸리면 응답이 멈칫한다
Hub&Spoke	가운데 한 곳(Hub)이 판정 규칙과 화면을 관리하고 여러 지점(Spoke)은 지표 수집만 맡는 구조
Incident	서비스에 영향을 준 장애 한 건을 원인·조치·재발방지와 함께 남긴 기록 단위
ITSM(IT Service Management)	장애·변경·문제 관리 같은 IT 서비스 운영 업무를 접수부터 종료까지 관리하는 체계와 도구
JVM(Java Virtual Machine)	자바 응용프로그램을 실행하는 가상 실행 환경. WAS가 그 위에서 동작한다
KNVD	한국인터넷진흥원이 운영하는 취약점 정보포털. 보안공지, 국내 취약점 정보, CVE 통합검색을 제공한다
LLM(Large Language Model)	많은 양의 글을 학습해 문장을 이해하고 만들어 내는 대규모 언어 모델
N2SF(국가 망 보안체계)	국가정보원이 업무 정보를 기밀·민감·공개로 나누고 등급별로 보안 통제를 달리하도록 정한 보안 체계
OpsKnow	예방점검 결과, Incident, 정기 보고를 한 저장소에 쌓아 다음 판정의 근거로 돌려주는 OPENMARU의 운영 지식 저장소
OpsKnow Agent	수집된 지표를 판정하고 승인 코드를 거친 조치, 재점검, 보고서 작성까지 잇는 OPENMARU의 AI 에이전트
RAG(Retrieval-Augmented Generation)	저장된 기록을 먼저 검색해 그 내용을 답변 근거로 붙여 주는 생성 방식
sLLM	기관 안 서버나 GPU 1대에서도 돌릴 수 있도록 규모를 줄인 소형 언어 모델
Wallboard	운영실 벽면 화면에 점검 이행 현황과 이상 상태를 크게 띄워 여럿이 함께 보도록 하는 상황판
WAS(Web Application Server)	웹 요청을 받아 업무 로직을 실행하고 데이터베이스와 주고받는 미들웨어 서버



2026 예방점검 의무화, 자동화를 넘어 지능화로

CONTACT

WEB
openmaru.io

EMAIL
hello@openmaru.io

TEL
+82-2-1670-1010
+82216701010

YOUTUBE
@openmaru
www.youtube.com/@openmaru

LINKEDIN
linkedin.com/company...
www.linkedin.com/company/opennaru

FACEBOOK
facebook.com/opennaru
www.facebook.com/opennaru

SCAN

openmaru

openmaru.io

© 2026 OPENMARU Marketing, OPENMARU. All rights reserved.