



OpsKnow AI 기반 IT 인프라 운영 지식 관리 시스템

운영 조직이 문서를 적게 만드는 경우는

운영 조직이 문서를 적게 만드는 경우는 드뭅니다. 정기점검이 끝나면 점검보고서가 나오고, 시스템을 새로 올리면 설치계획서와 설치보고서가 짝을 이루며, 변경 작업 하나에도 작업계획서와 작업보고서가 따라붙습니다. 부족한 것은 문서의 수가 아니라, 그 문서들이 필요한 순간에 다시 손에 잡히는 것입니다.

목차

OpsKnow AI 기반 IT 인프라 운영 지식 관리 시스템

- 1장: 운영 지식이 흩어질 때 생기는 세 가지 공백
 - 1.1 점검 기록이 이력으로 이어지지 않는 공백
 - 1.2 장애 시각에 드러나는 공백
- 2장: 사람만 읽던 문서를 AI 도 읽기 시작하면서 달라진 것
 - 2.1 독자가 둘이 되면서 생긴 요건
 - 2.2 사람과 기계가 같은 문서를 쓰는 상태
- 3장: 한글 문서가 아니라 마크다운이어야 하는 이유
 - 3.1 마크다운을 고르는 근거
 - 3.2 운영 문서와 증적에 한정한 형식 선택 기준
- 4장: OpsKnow 의 정의와 제품 경계
 - 4.1 제품 정의
 - 4.2 제품 경계 — 이 백서가 하지 않는 주장
- 5장: 운영 문서 열한 종이 한 책장에 서는 모습
 - 5.1 열한 종 컬렉션의 구성
 - 5.2 첫 적재 대상을 고르는 기준
- 6장: 매일 쌓이는 장애예방보고서
 - 6.1 날짜 트리 적재 규칙
 - 6.2 판정에서 문서 고정까지
- 7장: AI 에게 물은 질문이 그대로 운영 일지가 되는 흐름
 - 7.1 질의가 기록으로 남는 절차
 - 7.2 운영 일지와와의 접합
- 8장: 사람과 에이전트가 같은 책장을 보는 접점
 - 8.1 사람이 쓰는 화면
 - 8.2 에이전트가 쓰는 접점
- 9장: 관계형 데이터베이스와 벡터 데이터베이스를 함께 쓰는 이유
 - 9.1 역할 분담
 - 9.2 순위 병합
- 10장: GPU 48GB 에서 도는 온프레미스 구성과 그 규모
 - 10.1 GPU 48GB 구성의 조건
 - 10.2 그 규모가 감당하는 범위
- 11장: 답이 흔들리지 않게 만드는 장치와 남는 한계
 - 11.1 인용 추적
 - 11.2 도입 후 지속 부담
- 12장: 도입 판단에서 보아야 할 것
 - 12.1 세 가지 판정 기준

12.2 기술적 우위 요약

OpsKnow AI 기반 IT 인프라 운영 지식 관리 시스템

운영 조직은 문서를 많이 만듭니다. 정기점검이 끝나면 점검보고서가 나오고, 시스템을 새로 올리면 설치계획서와 설치보고서가 짝을 이루며, 변경 작업 하나에도 작업계획서와 작업보고서가 따라붙습니다. 부족한 것은 문서의 수가 아니라, 그 문서들이 필요한 순간에 다시 손에 잡히는 것입니다.

OpsKnow 는 IT 인프라 운영 문서를 한 저장소에 마크다운으로 모아, 점검·장애·작업 기록이 검색과 답변으로 되돌아오게 하는 운영 지식 관리 시스템입니다. 열린 종 컬렉션이 한 저장소에 서고, 장애예방보고서가 날짜 트리를 따라 매일 쌓이며, 사람과 코드 에이전트가 같은 문서를 검색합니다. 답변에는 참조한 문서가 따라 붙어 원문까지 되짚을 수 있습니다.

이 백서를 읽고 나면 세 가지를 판단하실 수 있습니다. 우리 조직의 어떤 문서가 먼저 들어가야 하는가, 우리 환경에서 돌릴 수 있는가, AI 가 준 답을 어디까지 믿을 것인가. 제품이 책임지는 사양과 도입 조직이 자기 환경에서 확인할 조건을 끝까지 구분해 적었습니다.

1장: 운영 지식이 흩어질 때 생기는 세 가지 공백

운영 조직은 문서를 많이 만듭니다. 정기점검이 끝나면 점검보고서가 나오고, 시스템을 새로 올릴 때는 설치계획서와 설치보고서가 짝을 이루며, 변경 작업 하나에도 작업계획서와 작업보고서가 따라붙습니다. 장애가 지나가면 장애보고서가 남고, 제품 매뉴얼과 작업가이드도 버전이 바뀔 때마다 손을 봅니다. 부족한 것은 문서의 수가 아닙니다.

부족한 것은 이미 쓴 문서가 운영 판단으로 되돌아오는 경로입니다. 문서는 만들어진 자리에 그대로 머무르고, 나중에 같은 질문이 생기면 그 자리를 기억하는 사람만 문서를 꺼냅니다. 기록은 성실하게 쌓이는데 조회는 사람의 기억을 거쳐야 하니, 쌓는 방향과 꺼내는 방향이 어긋나 있습니다. 이 장은 그 어긋남에서 생기는 손실을 세 갈래로 나눠 봅니다.

첫째는 이력의 공백입니다. 점검이 회차 단위로 끝나면 같은 항목이 언제부터 나빠졌는지 답할 근거가 남지 않습니다. 둘째는 사람의 공백입니다. 어느 문서에 무엇이 있는지가 담당자 개인의 기억에만 있으면 담당자 교체가 곧 지식 손실이 됩니다. 셋째는 시간의 공백입니다. 장애가 나면 원인을 조사하기 전에 근거 문서를 다시 모으는 절차부터 밟게 됩니다.

세 공백은 서로 다른 사건이 아니라 한 구조에서 나옵니다. 평시에 이력이 끊기고 위치가 개인에게 묶여 있으면 그 대가는 장애 시각에 시간으로 청구됩니다. 그래서 이 장은 평소 운영에서 출발해 장애 시각으로 이어 있도록 배열했습니다. 각 항의 표를 읽으실 때는 자기 조직의 현재 상태가 왼쪽 열과 오른쪽 열 가운데 어디에 놓이는지 짚어 보시기 바랍니다.

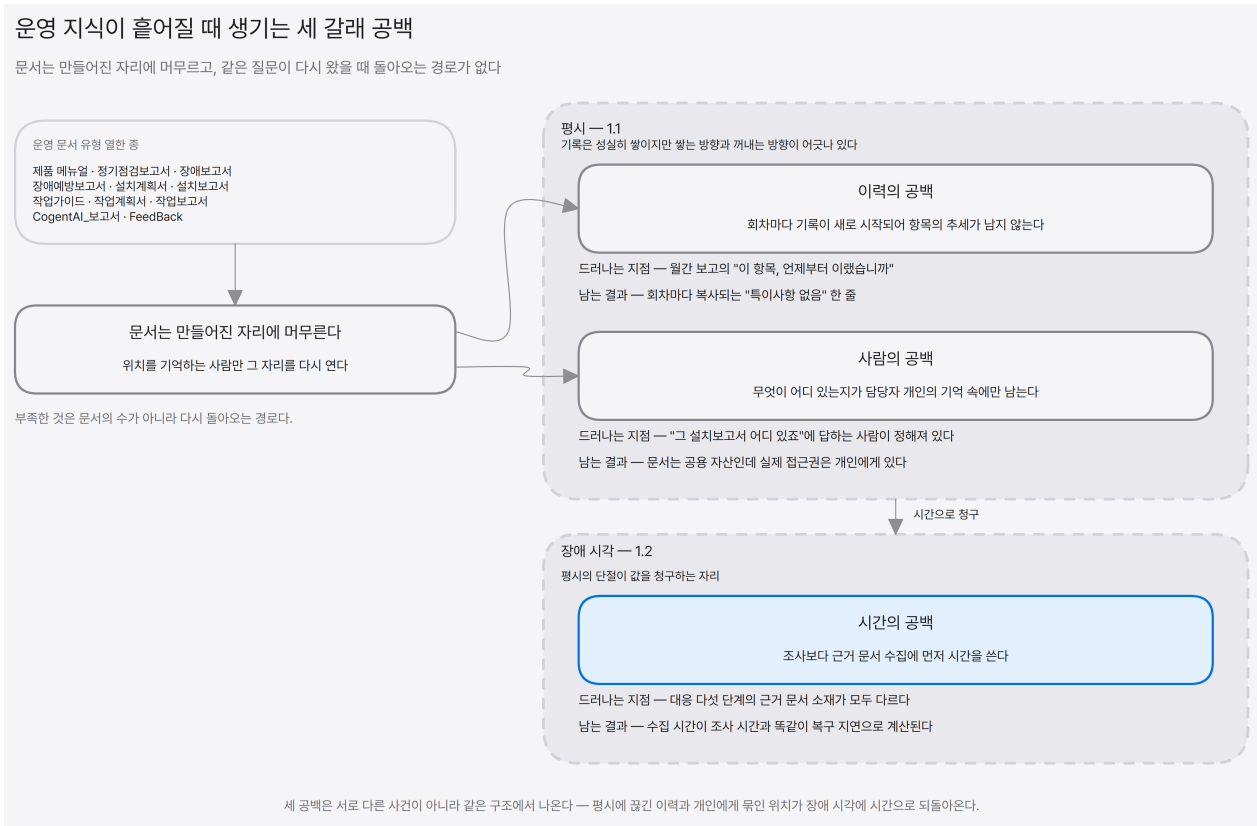


그림 1. 운영 지식이 흩어질 때 생기는 세 갈래 공백

1.1 점검 기록이 이력으로 이어지지 않는 공백

평시 운영의 기록은 대체로 성실합니다. 점검 주기는 지켜지고 보고서 서식도 갖춰져 있으며 결재선도 살아 있습니다. 그런데도 기록이 다음 판단의 근거가 되지 못하는 지점이 두 군데 있습니다. 하나는 회차 단위로 끊기는 점검 보고서이고, 다른 하나는 위치가 사람에게 묶인 문서입니다.

1.1.1 회차마다 새로 쓰는 점검 보고서

월간 보고 자리에서 운영팀장이 가장 자주 받는 질문은 단순합니다. 이 항목이 언제부터 이랬느냐는 물음입니다. 이번 달 점검보고서에는 이번 달 수치가 있고 지난달 보고서에는 지난달 수치가 있으므로, 답을 만들려면 담당자가 여러 회차 파일을 차례로 열어 눈으로 맞춰야 합니다. 회차가 늘어날수록 이 대조는 뒤로 밀리고, 결국 특이사항 없음이라는 한 줄만 회차마다 되풀이됩니다.

문제는 회차별 판정이 모두 정상이어도 항목의 값은 정상 범위 안에서 계속 나빠질 수 있다는 점입니다. 자원 사용률이 회차마다 조금씩 올라가는 동안 어느 회차의 판정도 이상을 가리키지 않습니다. 이런 악화는 회차를 가로질러 볼 때만 드러나는데, 기록이 회차 안에 갇혀 있으면 가로로 볼 방법이 없습니다. 점검 항목을 체크 목록으로 남기고 문서마다 리비전 이력을 붙여 두어도, 그 이력이 문서 한 건의 안쪽에서만 유지되는 한 회차와 회차 사이는 여전히 비어 있습니다.

구분	회차 단위로 끝나는 기록	이력으로 이어지는 기록
조회 단위	이번 회차 문서 한 건	항목 하나의 회차별 값 전체

구분	회차 단위로 끝나는 기록	이력으로 이어지는 기록
언제부터라는 질문	담당자가 과거 파일을 손으로 대조	같은 항목을 회차 순으로 나란히 조 회
정상 범위 안의 악화	회차 판정이 정상이면 드러나지 않 음	값이 움직인 방향으로 드러남
담당자 교체 시	어느 회차가 중요한지 함께 사라짐	문서가 그대로 근거로 남음

1.1.2 문서가 사람 기억에 묶이는 지점

그 설치보고서가 어디 있느냐는 질문에 답하는 사람은 대개 정해져 있습니다. 그 담당자는 문서가 어느 공유 폴더의 몇 번째 하위 경로에 있는지, 파일명이 사업명으로 시작하는지 날짜로 시작하는지, 중간에 이름이 한 번 바뀐 적이 있는지를 기억합니다. 이 기억은 문서 안에 적혀 있지 않고 사람 안에만 있습니다. 조직도상으로는 문서가 공용 자산이지만 실제 접근권은 개인에게 있는 셈입니다.

이 구조의 비용은 인수인계 기간에 드러납니다. 후임자가 업무를 넘겨받을 때 실제로 넘어오는 것은 문서 목 록이 아니라 문서를 찾는 요령이고, 요령은 말로만 전달되므로 빠짐없이 옮겨지지 않습니다. 선임이 자리를 비운 하루가 곧 탐색이 멈추는 하루가 되며, 신규 인력이 들어올 때마다 같은 질문이 반복되어 선임의 시간이 다시 들어갑니다. 문서가 제품과 컴포넌트와 문서 종류를 따라 내려가는 계층 위에 놓여 있으면, 위치를 기억 하는 대신 계층을 따라가면 됩니다.

상황	위치가 사람에게 묶인 경우	위치가 계층에 남은 경우
문서 찾기	담당자에게 묻고 답을 기다림	상위 구분에서 하위 구분으로 내려 감
담당자 부재	탐색이 사실상 멈춤	영향 없음
인수인계	문서 위치를 말로 전달	계층 자체가 인계 대상
신규 인력 투입	질문 횟수만큼 선임 시간 소모	스스로 찾아 들어감

1.2 장애 시각에 드러나는 공백

평시의 공백은 평시에 잘 보이지 않습니다. 점검은 끝났고 보고서는 결재되었으며 시스템은 돌고 있기 때문 입니다. 공백이 청구서로 바뀌는 시점은 장애가 났을 때입니다. 이 절은 그 청구서의 두 항목, 곧 근거 문서를 다시 모으는 시간과 원인 후보를 좁히는 시간을 봅니다.



그림 2. 장애 대응 절차와 근거 문서를 다시 모으는 접점

1.2.1 근거 문서 재수집에 쓰는 시간

장애 대응은 조사로 시작하지 않습니다. 무엇이 언제 어떻게 바뀌었는지를 알아야 조사를 시작할 수 있고, 그 답은 사람 머릿속이 아니라 문서에 있습니다. 그래서 대응 초반은 대체로 문서를 다시 모으는 절차로 채워집니다. 걸리는 시간은 조직마다 다르지만, 거쳐야 하는 절차의 수는 문서가 흩어져 있는 정도에 따라 정해집니다.

절차를 늘어놓아 보면 조사에 닿기까지 몇 개의 관문을 지나야 하는지가 드러납니다. 대상 시스템의 구성을 확인하려면 설치계획서와 설치보고서를 찾아야 하고, 최근에 무엇이 바뀌었는지 확인하려면 작업계획서와 작업보고서를 찾아야 하며, 같은 증상이 전에 있었는지 확인하려면 과거 장애보고서를 찾아야 합니다. 문서마다 보관 위치와 파일명 규칙이 다르면 관문마다 담당자를 부르는 단계가 하나씩 더 붙습니다.

대응 절차	필요한 문서	문서가 흩어져 있을 때 붙는 단계
대상 구성 확인	설치계획서·설치보고서	보관 위치를 아는 담당자부터 확인
최근 변경 확인	작업계획서·작업보고서	기간별 파일을 하나씩 열람
유사 사례 확인	과거 장애보고서	찾을 단서가 없어 목록을 훑음
조치 근거 확인	제품 매뉴얼·작업가이드	버전이 맞는 문서인지 되확인
사후 기록	장애보고서	대응 중 메모를 다시 옮겨 적음

1.2.2 원인 선별이 늦어지는 이유

문서를 모아 놓아도 다음 관문이 남습니다. 지금 보이는 증상이 처음 보는 것인지 전에도 있었던 것인지 가려야 하는데, 이 판단은 과거 사례를 얼마나 빨리 꺼내 볼 수 있느냐에 달려 있습니다. 과거 장애보고서가 연도

별 폴더에 파일로만 쌓여 있으면 증상 문구로 찾을 방법이 없고, 결국 그 사건을 겪은 사람이 기억해 내야 탐색이 시작됩니다. 그 사람이 자리에 없으면 처음 보는 장애로 취급되어 조사 범위가 넓어집니다.

같은 원인이 여러 번 돌아오는데도 조직이 그것을 알아채지 못하는 상태가 여기서 만들어집니다. 재발은 개별 사건 기록에는 남지만 사건과 사건을 가로지르는 자리에는 남지 않으므로, 재발 건수를 세는 사람이 없으면 재발이라는 사실 자체가 드러나지 않습니다. 알려진 오류와 그 조치가 찾을 수 있는 형태로 있으면 원인 후보를 빠르게 줄일 수 있고, 없으면 매번 전수 조사에 가까운 절차를 반복하게 됩니다. 조직이 학습하느냐 마느냐는 담당자의 성실성이 아니라 기록을 꺼내 볼 수 있느냐에서 갈립니다.

기록이 남는 위치	담기는 내용	다음 장애에서 참조되는 조건
개별 장애보고서 파일	사건 하나의 경위와 조치	그 사건을 기억하는 사람이 대응에 참여할 때
담당자 개인 메모	실제 원인과 우회 방법	담당자가 자리에 있을 때
사건을 가로지르는 조회	같은 증상의 과거 발생 이력	증상 문구만으로 참조 가능
알려진 오류 목록	원인과 확정된 조치	원인 후보를 좁히는 첫 단계로 사용

세 공백은 모두 문서가 없어서 생긴 것이 아니라 문서가 돌아오지 않아서 생긴 것입니다. 그렇다면 이미 쓰고 있는 문서를 어떤 형태로 두어야 되돌아오게 만들 수 있는지가 다음 질문이 됩니다. 이 질문은 저장 공간의 문제가 아니라 문서 자체의 요건과 맞닿아 있습니다. 2장은 그 요건이 최근에 어떻게 달라졌는지부터 짚습니다.

2장: 사람만 읽던 문서를 AI 도 읽기 시작하면서 달라진 것

운영 문서의 독자는 오랫동안 사람 하나였습니다. 읽는 쪽이 사람이므로 잘 만든 문서란 사람 눈에 잘 들어오는 문서였고, 표 선을 굵게 하고 항목마다 기호를 달고 중요한 문장에 색을 넣는 편집이 곧 품질이었습니다. 문서 도구가 제공하는 기능은 대부분 이 목적에 맞춰져 있었고, 조직의 문서 규정도 같은 방향으로 쌓였습니다. 이 전제가 최근 몇 해 사이에 바뀌었습니다.

바뀐 것은 독자의 수입입니다. 같은 문서를 사람과 기계가 함께 읽기 시작했습니다. 기계가 문서를 읽는다는 말은 화면에 그려진 모습을 본다는 뜻이 아니라, 파일에서 글자를 뽑아내고 그 글자를 일정한 크기의 조각으로 나눈 뒤 조각 단위로 찾아 쓴다는 뜻입니다. 이 과정에서 남는 것은 글자와 글자의 순서, 그리고 문서 안에 표시로 남아 있는 구조뿐입니다.

이 변화는 개별 조직의 선택을 넘어선 자리에서도 확인됩니다. 국가인공지능전략위원회는 2026년 3월 5일 보도자료에서 분과별 회의와 토론 결과를 마크다운으로 작성하고 관리하며 누리집에 공개한다고 밝혔습니다. 같은 자료는 회의 결과 문서를 공식 의사결정이 축적된 고품질 데이터로 규정했습니다. 같은 날 브릿지경제도 이 발표를 보도했습니다. 문서가 읽히는 자리에서 쓰이는 자리로 옮겨 가고 있다는 신호입니다.

이 장은 그 전환을 유행이 아니라 문서 요건의 변화로 다룹니다. 먼저 독자가 둘이 되면서 새로 생긴 요건을 보고, 이어서 두 독자를 한 벌의 문서로 감당할 수 있는지 봅니다. 형식의 이름을 고르는 일은 그다음 문제이므로 이 장에서는 아직 꺼내지 않습니다.



그림 3. 한 벌의 운영 문서를 함께 읽는 두 독자

2.1 독자가 돌이 되면서 생긴 요건

사람이 읽기 좋은 문서와 기계가 읽기 좋은 문서는 요건이 다릅니다. 사람은 모양에서 구조를 유추하지만 기계는 표시된 구조만 봅니다. 이 절은 그 차이가 실제 운영 문서에서 어떤 형태로 나타나는지, 그리고 이미 쌓아 둔 문서가 어떤 조건에서 쓸모 있는 데이터가 되는지를 봅니다.

2.1.1 편집이 겹친 문서에서 구조가 사라지는 현상

오래 유지된 운영 문서에는 편집이 겹겹이 얹혀 있습니다. 제목은 글꼴 크기와 굵기로만 표시되어 있고, 목록은 기호표와 들여쓰기로 흉내 내어져 있으며, 문단 사이의 간격은 빈 줄이 아니라 자간과 행간 설정으로 만들어져 있습니다. 표 안에 표가 들어가 있거나 칸을 합쳐 제목처럼 쓴 자리도 흔합니다. 사람 눈에는 이 모든 것이 구조로 보이지만, 문서 안에 이것이 제목이라거나 이것이 목록 항목이라는 표시는 남아 있지 않습니다.

글자만 뽑아내면 그 차이가 곧바로 드러납니다. 제목이 본문과 같은 평문 한 줄로 나오고, 목록 항목은 앞머리 기호를 잃은 채 문장들과 섞이며, 표는 칸 구분이 사라진 문자열로 이어집니다. 조각으로 나누는 단계에서 경계로 삼을 표시가 없으니 조각이 문장 한가운데에서 끊기고, 앞뒤 맥락이 잘린 조각은 나중에 찾아 써도 무슨 항목의 내용인지 알 수 없습니다. 문서를 어떻게 꾸몄느냐의 문제가 아니라 문서에서 무엇을 알아볼 수 있는냐의 문제입니다.

편집 방식	사람이 읽을 때	글자만 뽑아냈을 때 남는 것
글꼴 크기·굵기로 제목 표현	제목으로 보임	본문과 구분되지 않는 한 줄
기호표와 들여쓰기로 목록 표현	항목으로 보임	앞머리 기호를 잃은 문장 나열
칸 합치기로 표 안에 제목 배치	구획으로 보임	칸 경계가 사라진 긴 문자열
행간·자간으로 문단 구분	문단으로 보임	문단 경계 없음
색과 음영으로 강조	중요 표시로 보임	아무 표시도 남지 않음

2.1.2 운영 문서가 고품질 데이터가 되는 조건

새로 만들어야 할 것이 있다는 이야기가 아닙니다. 점검보고서, 장애보고서, 설치보고서, 작업보고서는 이미 조직 안에 쌓여 있고, 그 안에는 어떤 판단을 어떤 근거로 내렸는지가 사건 단위로 들어 있습니다. 국가인공지능전략위원회가 2026년 3월 5일 회의 결과 문서를 공적 의사결정이 축적된 고품질 데이터로 규정한 것과 같은 성격입니다. 운영 문서 역시 결정과 그 근거가 시간 순으로 쌓인 기록입니다.

다만 쌓여 있다는 사실만으로 데이터가 되지는 않습니다. 문서 안에 구조가 표시되어 있어야 조각으로 나눌 때 의미 단위가 지켜지고, 같은 종류의 문서가 같은 형태로 반복되어야 여러 회차를 나란히 놓고 볼 수 있으며, 답을 받은 뒤 원문을 되짚어 열 수 있어야 그 답을 근거로 쓸 수 있습니다. 이 세 조건이 갖춰지지 않으면 같은 문서가 자산이 아니라 잡음으로 남습니다. 조건이 갖춰지면 이미 쓰고 있는 문서가 그대로 판단의 재료가 됩니다.

문서 유형	담고 있는 결정	데이터로서의 쓰임
정기점검보고서	회차마다의 상태 판정	항목별 값의 이동을 회차 순으로 확인

문서 유형	담고 있는 결정	데이터로서의 쓰임
장애보고서	증상과 원인, 그리고 조치	유사 증상의 과거 사례를 원인 후보로 사용
설치계획서·설치보고서	구성과 설정의 확정 내용	대상 시스템의 현재 구성 확인
작업계획서·작업보고서	변경 범위와 실제 수행 결과	최근 변경과 증상 발생 시점의 대조
제품 매뉴얼·작업가이드	표준 절차와 권장 설정	조치 근거의 확인

2.2 사람과 기계가 같은 문서를 쓰는 상태

요건이 다르다고 해서 문서를 두 벌 만들어야 하는 것은 아닙니다. 두 요건이 충돌하는 지점은 생각보다 좁고, 겹치는 지점은 넓습니다. 이 절은 한 벌의 문서로 두 독자를 감당하려면 무엇이 필요한지, 그리고 그 조건이 어떤 형식을 가리키는지까지 봅니다.



그림 4. 단일 원본에서 갈라지는 사람용 열람과 기계용 검색

2.2.1 한 벌의 문서로 두 독자를 만족시키는 조건

사람용 문서와 기계용 문서를 따로 만들면 관리 대상이 두 배가 됩니다. 문제는 분량이 아니라 어긋남입니다. 한쪽만 고쳐진 상태로 시간이 지나면 두 벌의 내용이 달라지고, 그때부터는 어느 쪽이 맞는지 확인하는 일이 새로운 업무로 추가됩니다. 증적으로 제출해야 하는 문서라면 두 벌 가운데 어느 것이 원본인지도 답할 수 있어야 합니다.

한 벌로 유지하려면 원본이 두 가지 성질을 함께 가져야 합니다. 첫째, 구조가 문서 안에 글자로 표시되어 있어야 합니다. 제목인지 목록인지 표인지가 모양이 아니라 표기로 남아 있으면 기계는 그 표기를 그대로 경계로 씁니다. 둘째, 그 표기가 사람이 읽는 데 방해가 되지 않아야 합니다. 표기가 몇 개 되지 않고 눈에 익으면 원본을 그대로 읽어도 무리가 없고, 보여 줄 자리에서는 화면이나 문서 형태로 다시 그려 내면 됩니다. 원본은 하나로 두고 사람에게 보여 줄 모습과 기계가 읽을 재료를 같은 파일에서 뽑아내는 구성입니다.

항목	사람용과 기계용을 따로 둘 때	원본 한 벌만 둘 때
갱신 절차	같은 내용을 두 곳에 반영	한 곳만 수정
내용 어긋남	시간이 지나면 발생	발생하지 않음
원본 확인	어느 쪽이 원본인지 별도 판단	원본이 하나뿐
증적 제출	제출본과 내부본의 대조 필요	원본에서 제출 형태로 변환
담당자 부담	문서 종류만큼 두 배	기존 작성 부담 그대로

2.2.2 3장으로 넘기는 질문

여기까지의 요건은 특정 제품이나 도구를 가리키지 않습니다. 구조가 글자로 표시되어 있을 것, 사람이 원본을 그대로 읽어도 부담이 없을 것, 보여 줄 형태로 다시 그려 낼 수 있을 것. 이 세 가지를 모두 만족하는 형식이 무엇인가가 남은 질문입니다.

답은 마크다운입니다. 다만 이 선택은 담당자의 취향이나 개발 조직의 관습으로 정당화될 수 있는 성격이 아니고, 공개된 자료로 근거를 댈 수 있어야 하는 성격입니다. 3장은 그 근거를 세 갈래로 나누어 살펴보고, 동시에 어떤 문서까지를 대상으로 삼아야 하는지도 함께 줍니다.

3장: 한글 문서가 아니라 마크다운이어야 하는 이유

한글 문서를 마크다운으로 바꾸자고 하면 대개 취향 다툼이 됩니다. 쓰던 도구가 편하다는 쪽과 새 형식이 낯
다는 쪽이 각자의 경험을 근거로 내세우면 결론이 나지 않습니다. 그래서 이 장은 경험 대신 공개된 자료를 놓
고 이야기합니다. 근거를 대는 방식이 분명하면 형식 선택은 논쟁이 아니라 판단이 됩니다.

다만 공개 자료의 수치를 그대로 옮겨 오지는 않습니다. 형식별 비교는 측정 대상 문서, 사용한 모델, 조각을
나누는 방식에 따라 결과가 달라지므로, 어떤 자료도 모든 환경에 그대로 적용되는 값을 주지 못합니다. 이 장
에서 인용하는 수치는 어느 쪽이 유리한가라는 방향까지만 받아들이고, 얼마나 유리한가는 각자의 문서로 확
인할 몫으로 남깁니다. 방향만으로도 형식 선택에는 충분합니다.

장의 뒤쪽 절반은 범위를 좁히는 데 씁니다. 모든 문서를 바꾸라는 주장은 실행되지 않고, 실행되지 않는 제안
은 판단에 도움이 되지 않기 때문입니다. 조직이 당장 고를 수 있는 대상이 어디까지인지, 그리고 그 전환에
실제로 무엇이 드는지를 함께 적었습니다.

3.1 마크다운을 고르는 근거

마크다운을 고르는 근거는 세 갈래에서 나옵니다. 같은 내용을 담을 때 소비하는 토큰의 양, 조각을 나눌 때
지켜지는 구조, 그리고 표기 형식 자체가 결과에 미치는 영향입니다. 세 갈래는 성격이 다르고 근거의 무게도
다르므로, 어디까지 단정할 수 있는지를 함께 적습니다.

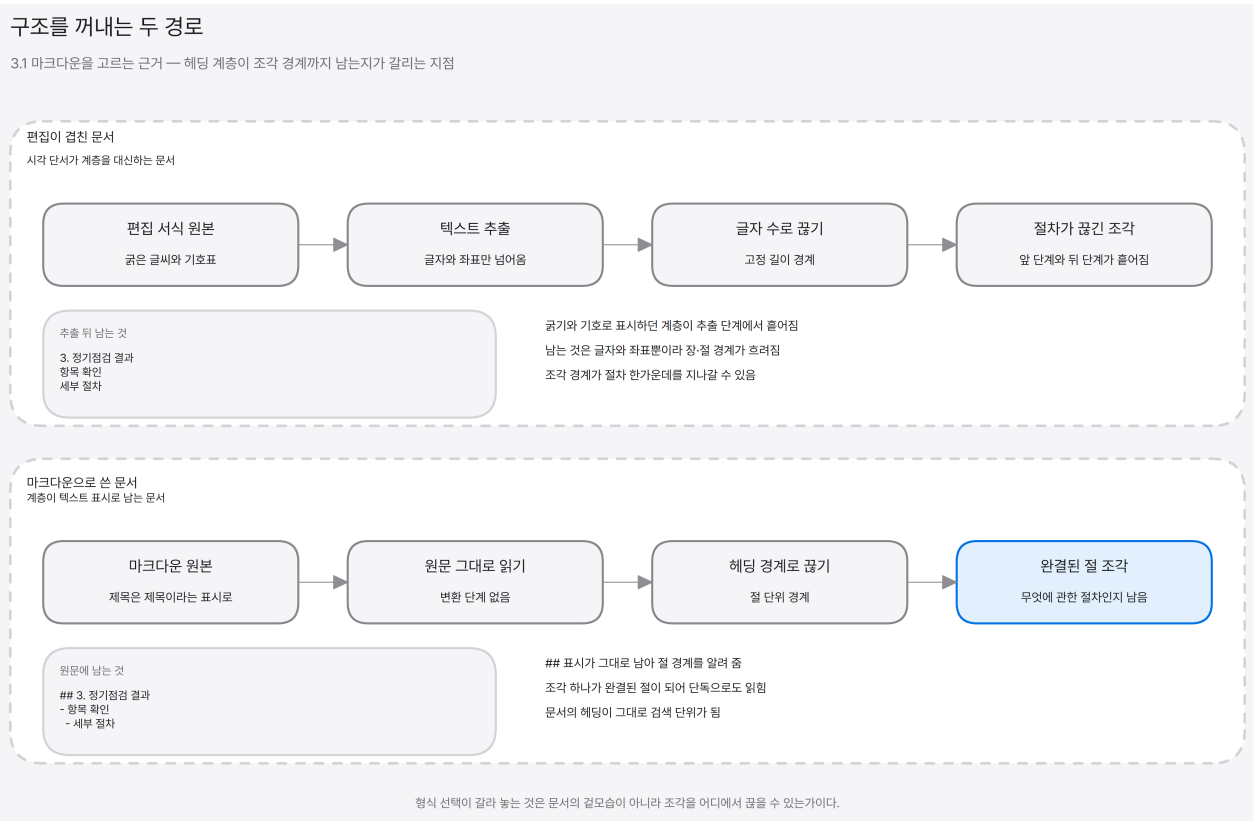


그림 5. 편집이 곁친 문서와 마크다운의 구조 추출 경로 차이

3.1.1 토큰 효율과 그 근거의 등급

같은 목록과 헤딩 블록을 여러 형식으로 적어 놓고 소비 토큰을 비교한 자료들이 공개되어 있습니다. 보고의 결론은 대체로 한 방향입니다. 태그가 겹겹이 붙는 형식보다 마크다운 쪽이 같은 내용을 더 적은 토큰으로 표현한다는 것입니다. 목록과 제목이 많은 문서일수록 차이가 커지고, 서술형 문장만 이어지는 문서에서는 차이가 줄어드는 경향도 함께 보고됩니다.

여기서 주의할 점은 보고된 절감 폭의 성격입니다. 개인이 자신의 문서로 측정해 공개한 값과 연구 설정을 갖 추고 비교한 값은 무게가 다르고, 어느 쪽이든 측정에 쓴 문서와 모델이 달라지면 값도 달라집니다. 문서 표기 형식이 언어 모델의 응답 품질에 유의하게 작용한다는 연구도 함께 공개되어 있으나, 그 연구 역시 특정 과제와 특정 모델 설정 아래에서 얻은 결과입니다. 그러므로 절감 폭을 도입 근거의 숫자로 쓰지 않고, 형식을 고를 때 어느 쪽이 유리한가를 가리키는 방향으로만 받아들이는 편이 안전합니다.

자료 성격	무엇을 비교했는가	보고되는 경향	받아들일 범위
개인이 공개한 형식별 토큰 비교	같은 목록·헤딩 블록의 형식별 토큰 수	마크다운 쪽이 더 적음	방향까지만
표기 형식과 응답 품질 연구	표기 형식을 바꿨을 때의 응답 품질	표기 선택이 유의하게 작용	연구 설정과 함께
조직 내부 문서로 직접 측정	자기 조직의 실제 문서	조직마다 다름	도입 판단의 실제 근거

3.1.2 구조 인식 청킹과 검색 정확도

문서를 찾아 쓰려면 먼저 조각으로 나뉘야 합니다. 나누는 방식은 크게 둘입니다. 글자 수를 정해 놓고 기계적으로 끊는 방식과, 헤딩과 절 경계 같은 문서 구조를 살려 끊는 방식입니다. 두 방식을 같은 문서 집합에 적용해 검색 정확도를 비교한 자료가 공개되어 있고, 구조를 살려 끊는 쪽이 앞선다는 방향으로 보고됩니다.

이유는 조각의 내용을 보면 짐작할 수 있습니다. 글자 수로 끊으면 한 절의 앞부분과 다음 절의 뒷부분이 한 조각에 섞이거나, 표의 머리 행과 값이 갈라져 서로 다른 조각에 들어갑니다. 헤딩을 경계로 삼으면 조각 하나가 온전한 항 하나가 되고, 조각마다 어느 문서의 어느 항인지가 함께 남습니다. 다만 비교 자료는 특정 문서 집합과 특정 질의 묶음 위에서 얻은 결과이므로 정확도 항상 폭을 그대로 기대하기는 어렵습니다. 방향이 중요한 이유는 검색이 틀리면 그 뒤의 답변도 함께 틀리기 때문입니다. 잘못 찾아온 조각 위에서 만들어진 답은 문장만 매끄럽고 근거는 엉뚱한 곳을 가리킵니다.

조각 나누기 방식	조각의 경계	조각 하나에 담기는 것	함께 밝혀야 할 조건
글자 수로 끊기	정해진 길이	절의 일부와 다음 절의 일부가 섞임	길이 설정값에 따라 결과가 달라짐
구조를 살려 끊기	헤딩과 절 경계	항 하나의 온전한 내용	문서에 구조 표기가 있어야 성립
비교 실험의 조건	문서 집합·질의 묶음·모델	해당 조건에서의 정확도	조건이 바뀌면 항상 폭도 바뀜

3.2 운영 문서와 증적에 한정한 형식 선택 기준

앞 절의 근거는 모든 문서를 바꾸라는 결론으로 이어지지 않습니다. 형식을 바꿔서 얻는 것이 큰 문서가 있고, 바꿔도 얻는 것이 없거나 오히려 곤란해지는 문서가 있습니다. 첫 결정은 전면 전환의 여부가 아니라 대상 선별입니다.



그림 6. 형식을 바꿀 문서와 그대로 둘 문서의 경계

3.2.1 형식을 바꿀 문서와 그대로 둘 문서

기준은 단순합니다. 나중에 다시 찾아 읽을 문서, 여러 회차를 나란히 놓고 볼 문서, 답변의 근거로 쓸 문서가 전환 대상입니다. 정기점검보고서와 장애보고서, 설치·작업 관련 문서, 제품 매뉴얼과 작업가이드가 여기에 들어갑니다. 이 문서들은 조직 안에서 반복 생산되고 반복 조회되므로, 구조를 표기로 남겨 두면 그만큼 되돌아옵니다.

반대로 대외 제출본은 지금 형식을 유지하는 편이 낫습니다. 제출 서식이 정해져 있거나 결재와 날인 흔적이 문서의 일부인 경우, 형식을 바꾸면 얻는 것보다 잃는 것이 큼니다. 계약 문서와 공식 공문도 마찬가지입니다. 다만 두 갈래는 배타적이지 않습니다. 원본을 마크다운으로 두고 제출이 필요할 때 기존 형식으로 변환해 내 보내면, 조회와 제출을 한 원본으로 함께 감당할 수 있습니다.

문서 유형	전환 대상 여부	판단 근거
정기점검보고서·장애보고서	전환	회차를 나란히 놓고 조회하는 문서
설치계획서·설치보고서	전환	구성 확인의 근거로 반복 조회
작업계획서·작업보고서	전환	변경 이력 추적의 재료
제품 매뉴얼·작업가이드	전환	조치 근거로 자주 인용

문서 유형	전환 대상 여부	판단 근거
대외 제출본·계약 문서	유지	서식과 날인이 문서의 일부
발표 자료·홍보물	유지	조회 대상이 아니라 전달물

3.2.2 전환에 드는 실제 부담

전환은 파일 확장자를 바꾸는 일이 아닙니다. 기존 문서에서 글자를 뽑아내고, 제목과 목록과 표를 표기로 다시 표시하고, 문서에 딸린 그림과 첨부를 옮겨 붙이고, 옮긴 결과가 원본과 같은 내용인지 확인하는 절차가 이어집니다. 특히 칸을 합쳐 만든 복잡한 표와 문서 안에 그림으로 들어간 도해는 자동 변환만으로 끝나지 않아 사람이 손을 대야 합니다. 이 부담을 감추면 도입 초기에 신뢰를 잃습니다.

들어가는 작업량은 기존 문서의 형식과 수량에 따라 달라지므로 도입 전에 표본을 정해 재 보시기를 권합니다. 문서 종류별로 몇 건씩 실제로 옮겨 보면 종류마다 걸리는 시간과 손이 가는 지점이 드러나고, 그 값으로 전체 규모를 가늠할 수 있습니다. 한 번에 전부 옮길 필요도 없습니다. 앞으로 새로 만드는 문서부터 새 형식으로 쓰고 과거 문서는 조회 빈도가 높은 것부터 차례로 옮기면, 전환이 진행되는 동안에도 조회는 계속 가능합니다.

전환 작업 항목	내용	손이 가는 지점
본문 추출	기존 문서에서 글자를 뽑아냄	머리말·꼬리말과 본문의 구분
구조 표기	제목·목록·표를 표기로 표시	모양으로만 표현된 제목의 판별
표 정리	칸을 합친 표를 단순한 표로 다시 작성	합친 칸이 많은 표
그림과 첨부 이관	문서에 딸린 이미지와 파일을 옮김	문서 안에 그림으로 들어간 도해
대조 확인	옮긴 결과와 원본의 내용 일치 확인	수치와 항목명
규칙 정리	파일명과 보관 위치 규칙 통일	종류마다 다르게 쌓인 기존 규칙

4장: OpsKnow 의 정의와 제품 경계

제품을 도입할지 판단하려면 그 제품이 무엇인지와 무엇이 아닌지를 같은 무게로 알아야 합니다. 앞 장까지는 운영 지식이 흩어질 때 생기는 손실과 문서 형식을 마크다운으로 고르는 근거를 다뤘습니다. 이 장은 그 자리에 놓이는 물건의 윤곽을 그립니다. 정의를 한 문장으로 세우고, 그 문장을 떠받치는 구성 요소를 나눈 다음, 제품이 말지 않는 일을 이어서 적습니다.

경계를 함께 적는 까닭은 도입 계획이 대개 빠진 자리에서 어긋나기 때문입니다. 제품이 하는 일만 늘어놓으면 독자는 나머지도 따라오리라 기대하게 되고, 그 기대는 운영 조직이 직접 말아야 할 일을 계획에서 빠뜨리게 만듭니다. 그래서 4.1 은 정의를, 4.2 는 경계를 비슷한 분량으로 다룹니다. 두 절을 이어 읽으면 도입 범위와 조직이 준비할 몫이 함께 드러납니다.

4.1 제품 정의

정의는 한 문장에서 출발합니다. 그 뒤에 문장의 각 마디가 어떤 구성 요소로 실현되는지를 붙이고, 마지막으로 이 물건이 조직의 도구 지형에서 어느 자리에 서는지를 밝힙니다. 순서를 이렇게 잡은 이유는 구성 요소를 먼저 늘어놓으면 독자가 부품 목록을 읽게 되고, 부품 목록만으로는 도입 여부를 판단할 수 없기 때문입니다.

4.1.1 한 문장 정의

OpsKnow 는 IT 인프라 운영 문서를 한 저장소에 마크다운으로 모아, 사람과 코드 에이전트가 같은 본문을 검색하고 답변에서 근거 문서로 되짚어 갈 수 있게 하는 운영 지식 저장소입니다. 이 문장에는 네 마디가 들어 있습니다. 대상은 운영 문서이고, 형식은 마크다운이며, 이용자는 사람과 에이전트 양쪽이고, 답변에는 원문으로 돌아가는 길이 달려 나옵니다. 네 마디 가운데 하나라도 빠지면 남는 것은 문서를 모아 둔 폴더이거나 근거를 밝히지 않는 답변 화면입니다.

각 마디는 구체적인 구성 요소로 실현됩니다. 운영 문서는 열한 종 컬렉션으로 나뉜 자리에 적재되고, 제품 메뉴얼은 제품과 컴포넌트를 거쳐 장 단위 문서까지 계층으로 내려갑니다. 검색은 관계형 조회와 벡터 검색을 함께 돌린 뒤 결과를 하나의 순위로 병합해 내놓습니다. 답변마다 참조한 문서를 되짚어 원문을 열 수 있고, 코드 에이전트는 MCP 접점으로 같은 제품 메뉴얼을 검색해 답을 만듭니다.

정의 문장의 마디	제품이 그 마디를 실현하는 방식
대상 — 운영 문서	열한 종 컬렉션으로 나뉜 적재 자리
형식 — 마크다운	헤딩 계층이 그대로 검색 단위가 되는 평문 문서
이용자 — 사람과 에이전트	화면 검색과 코드 에이전트의 MCP 접점
되짚기 — 인용 추적	답변마다 참조 문서를 열어 원문을 확인
배치 — 조직 경계 안	GPU 48GB 급 단일 노드 온프레미스

배치 조건은 정의의 다섯 번째 마디라기보다 앞의 네 마디를 감싸는 전제입니다. 문서와 질의와 답변이 조직 경계를 넘지 않는 구성이므로, 운영 문서에 담긴 호스트 이름과 구성 값이 바깥으로 나갈 경로가 만들어지지

않습니다. 이 전제가 무너지면 나머지 네 마디의 의미도 함께 달라집니다. 자원 규모가 판단 기준이 되는 조직이라면 10장의 배치 조건을 먼저 읽고 이 장으로 돌아오셔도 됩니다.



계층	담당하는 일	이 계층만으로는 답이 담기지 않는 질문
검색 계층	파일 이름과 본문 낱말로 문서 찾기	여러 문서에 흩어진 같은 항목을 한 답으로 묶어 달라
운영 도구 계층	상태 수집, 경보, 사건 처리	지금 이 경보에 대응했던 지난 절차와 기록이 무엇이었나
운영 지식 계층	문서를 장 단위로 나눠 질의에 답하고 근거를 되짚음	(이 계층이 맡는 자리)

계층이 다르다는 말은 기존 도구를 걷어 내야 한다는 뜻이 아닙니다. 원본과 권한은 문서 저장 계층에 그대로 두고, 경보와 사건 처리도 운영 도구 계층에 남습니다. 도입 판단에서 가를 것은 무엇을 없앨지가 아니라, 지금 답이 담기지 않는 질문이 조직에 얼마나 자주 생기는지입니다. 그 빈도가 낮다면 이 계층을 새로 엮을 이유도 크지 않습니다.

4.2 제품 경계 — 이 백서가 하지 않는 주장

경계는 두 갈래로 적습니다. 하나는 제품 기능으로 세지 않는 일이고, 다른 하나는 구현 수단으로 전제하지 않는 도구입니다. 앞의 것은 운영 조직이 직접 맡아야 하므로 도입 계획에 인원과 시간으로 잡혀야 하고, 뒤의 것은 도입 전에 없다고 해서 걸림돌이 되지 않으므로 준비물 목록에서 빼도 됩니다. 두 갈래를 섞어 읽으면 계획이 부풀거나 헐거워집니다.

4.2.1 제품 기능으로 적지 않는 것

문서 머리말의 메타 규약은 제품 기능이 아닙니다. 마크다운 문서 맨 앞에 작성자, 대상 시스템, 적용 버전, 유효 기간 같은 항목을 어떤 이름으로 몇 개나 들지는 조직마다 다르고, 그 규약을 정하고 지키게 하는 일은 운영 조직의 몫입니다. 제품은 그렇게 작성된 문서를 컬렉션에 적재하고 검색 대상으로 삼습니다. 규약이 느슨하면 검색 결과의 선별도 느슨해지므로, 첫 적재를 시작하기 전에 항목 이름을 맞춰 두는 편이 낫습니다.

유효 기간이 지난 문서를 골라 담당자에게 알리는 자동화도 제품 기능으로 세지 않습니다. 어떤 문서를 몇 달 주기로 다시 보게 할지는 조직의 점검 주기와 인사 구조를 따라 정해지는 문제이고, 알림을 받을 사람과 받을 경로도 조직마다 다릅니다. 이 일은 기존 점검 일정에 엮는 편이 자연스럽습니다. 제품이 맡는 부분은 문서가 언제 어떤 리비전으로 갱신됐는지를 이력으로 남기는 데까지입니다.

구분	제품이 맡는 범위	운영 조직이 맡는 범위
문서 머리말	작성된 문서를 컬렉션에 적재하고 검색	머리말 항목 이름과 필수 여부를 정하고 지키게 함
문서 신선도	리비전 이력 누적, 갱신 시점 기록	재검토 주기 결정, 만료 문서 알림과 담당자 지정
문서 분류	열한 종 컬렉션이라는 적재 자리	자기 조직 문서를 어느 컬렉션에 넣을지 판단
문서 품질	장 단위 검색과 인용 추적	본문 내용의 정확성과 최신성 확인

빠진 것을 먼저 말하는 편이 도입 계획을 튼튼하게 만듭니다. 위 표의 오른쪽 칸은 도구를 사도 줄어들지 않는 일이며, 담당자와 주기를 정하지 않으면 저장소가 채워진 뒤에도 답변 품질이 오르지 않습니다. 이 뒤편에 드는 부담은 적재하는 문서 종류와 생산 주기에 따라 달라지므로, 첫 컬렉션 하나로 규모를 재보고 나서 나머지를 확장하시기 바랍니다.

4.2.2 구현 수단으로 적지 않는 것

이 백서는 버전 관리 도구 기반 워크플로를 구현 수단으로 서술하지 않습니다. 문서를 저장소에 올리고 변경 요청으로 검토하고 병합하는 절차를 이미 쓰고 있는 조직이 있고, 그런 절차가 없는 조직도 있습니다. 제품이 요구하는 것은 마크다운으로 작성된 문서와 그 문서를 넣을 컬렉션이며, 문서가 어떤 경로로 작성되어 들어 오는지는 조직의 선택으로 남습니다. 이미 익숙한 도구가 있다면 그것을 그대로 쓰시면 됩니다.

리비전 이력도 마찬가지입니다. 제품은 문서마다 리비전 이력을 누적하므로, 변경 이력을 얻기 위해 별도의 도구를 먼저 도입해야 하는 관계가 아닙니다. 경계를 이렇게 적는 이유는 규범을 세우려는 것이 아니라 준비물 목록을 정확히 하기 위해서입니다. 전환에 드는 작업량은 기존 문서의 형식과 수량에 따라 달라지므로, 도입 규모에 맞춰 표본 문서 몇 건으로 먼저 재보시기 바랍니다.

5장: 운영 문서 열한 종이 한 책장에 서는 모습

저장소는 빈 상자로 시작하지 않습니다. OpsKnow 는 운영 문서를 열한 종의 컬렉션으로 나누어 받으며, 각 컬렉션은 현장에서 실제로 생산되는 문서 종류에 맞춰 자리를 갖고 있습니다. 제품 메뉴얼, 정기점검보고서, 장애보고서, 장애예방보고서, 설치계획서, 설치보고서, 작업가이드, 작업계획서, 작업보고서, CogentAL_보고서, FeedBack 입니다. 이 목록은 도입 첫날 독자가 마주하는 책장의 칸 이름이기도 합니다.

이 장의 쓰임은 하나입니다. 자기 조직에서 지금 만들어지고 있는 문서를 이 열한 칸에 대입해 보고, 그 가운데 먼저 넣을 문서를 고르는 것입니다. 열한 칸을 모두 채우고 시작할 필요는 없으며, 채워지지 않은 칸이 남아 있어도 검색은 채워진 칸에서 동작합니다. 5.1 에서 칸의 구성을 보고, 5.2 에서 첫 적재 대상을 고르는 기준을 잡으시면 됩니다.

5.1 열한 종 컬렉션의 구성

열한 개를 순서대로 외울 필요는 없습니다. 성격이 같은 것끼리 묶으면 다섯 갈래로 줄고, 다섯 갈래는 운영 조직이 이미 머릿속에 갖고 있는 구분과 거의 겹칩니다. 여기서는 먼저 갈래별로 묶어 목록을 보이고, 그다음 열한 종 가운데 계층이 가장 깊은 제품 메뉴얼의 구조를 따로 떼어 설명합니다.

5.1.1 열한 종 컬렉션 목록과 성격

문서는 만들어지는 계기가 성격을 결정합니다. 기준과 절차를 적어 두고 필요할 때마다 꺼내 보는 문서가 있고, 어떤 일을 하기 전에 계획으로 쓰는 문서가 있으며, 그 일을 마친 뒤 결과로 남기는 문서가 있습니다. 주기가 정해져 반복 생산되는 점검 문서가 있고, 사건이 벌어졌을 때만 생기는 문서가 있으며, 이 모든 기록을 되돌아보며 분석과 의견을 담는 문서가 있습니다. 열한 종은 이 다섯 갈래에 그대로 나뉩니다.

성격	컬렉션	생산 계기	다시 찾게 되는 상황
기준·절차	제품 메뉴얼, 작업가이드	제품 도입과 절차 표준화 시점	설치·설정·조치 절차를 확인할 때
계획	설치계획서, 작업계획서	작업 착수 전	같은 유형의 작업을 다시 계획할 때
수행 기록	설치보고서, 작업보고서	작업 종료 직후	구성 변경 시점과 내용을 되짚을 때
주기 점검·예방	정기점검보고서, 장애예방 보고서	정해진 주기와 매일	항목의 추세와 이전 소견을 볼 때
사건·환류	장애보고서, CogentAL_보고서, FeedBack	장애 발생과 그 이후 분석	유사 증상의 과거 사례를 대조할 때

갈래로 묶어 보면 자기 조직 문서를 대입하기가 쉬워집니다. 이름이 정확히 같지 않아도 계기가 같으면 같은 칸에 들어갑니다. 점검 결과를 표 하나로만 남기고 있다면 정기점검보고서 칸에, 작업 요청서와 결과 회신을 함께 쓰고 있다면 작업계획서와 작업보고서 두 칸에 나누어 들어갑니다. 조직에서 부르는 이름을 그대로 두고 칸만 맞춰도 검색 결과는 갈래별로 정돈되어 나옵니다.

한 가지 짚어 둘 점은 칸이 비어 있어도 무방하다는 것입니다. 외주 설치를 받는 조직이라면 설치계획서와 설치보고서 칸이 오래 비어 있을 수 있고, 자체 개발 비중이 낮은 조직이라면 작업가이드가 몇 건에 그칠 수 있습니다. 비어 있는 칸은 나중에 채우면 되고, 채워진 칸끼리는 그 자체로 서로를 보강합니다. 장애보고서 한 건이 같은 시기의 정기점검보고서와 나란히 검색되면 그때부터 맥락이 생깁니다.

5.1.2 제품 메뉴얼의 계층

열한 종 가운데 제품 메뉴얼은 계층이 가장 깊습니다. 맨 위에 제품이 있고, 그 아래 컴포넌트가 놓이며, 컴포넌트마다 설치 가이드와 운영 매뉴얼이 갈라지고, 그 아래에 장 단위 문서가 늘어섭니다. 설치 개요, 사전 준비, 설치 절차, 설치 검증, 업그레이드, 보안 설정, 트러블슈팅, 설정 레퍼런스처럼 장마다 다루는 주제가 다르고 길이도 다릅니다. 계층을 이렇게 내린 결과 검색 단위가 문서 한 편이 아니라 장이 됩니다.

검색 단위가 장이라는 사실은 답변 품질에 곧바로 작용합니다. 문서 한 편이 통째로 검색 결과에 오르면 그 안 어디를 보라는 말인지가 남지 않고, 읽는 사람이 다시 목차를 뒤져야 합니다. 장 단위로 서 있으면 "업그레이드 전 확인 항목"을 물었을 때 업그레이드 장이 직접 걸리고, 인용 추적을 따라가면 그 장의 원문이 열립니다. 계층을 알게 두면 검색이 넓어지고 깊게 두면 좁아지므로, 문서를 옮길 때 장 경계를 무너뜨리지 않는 편이 좋습니다.

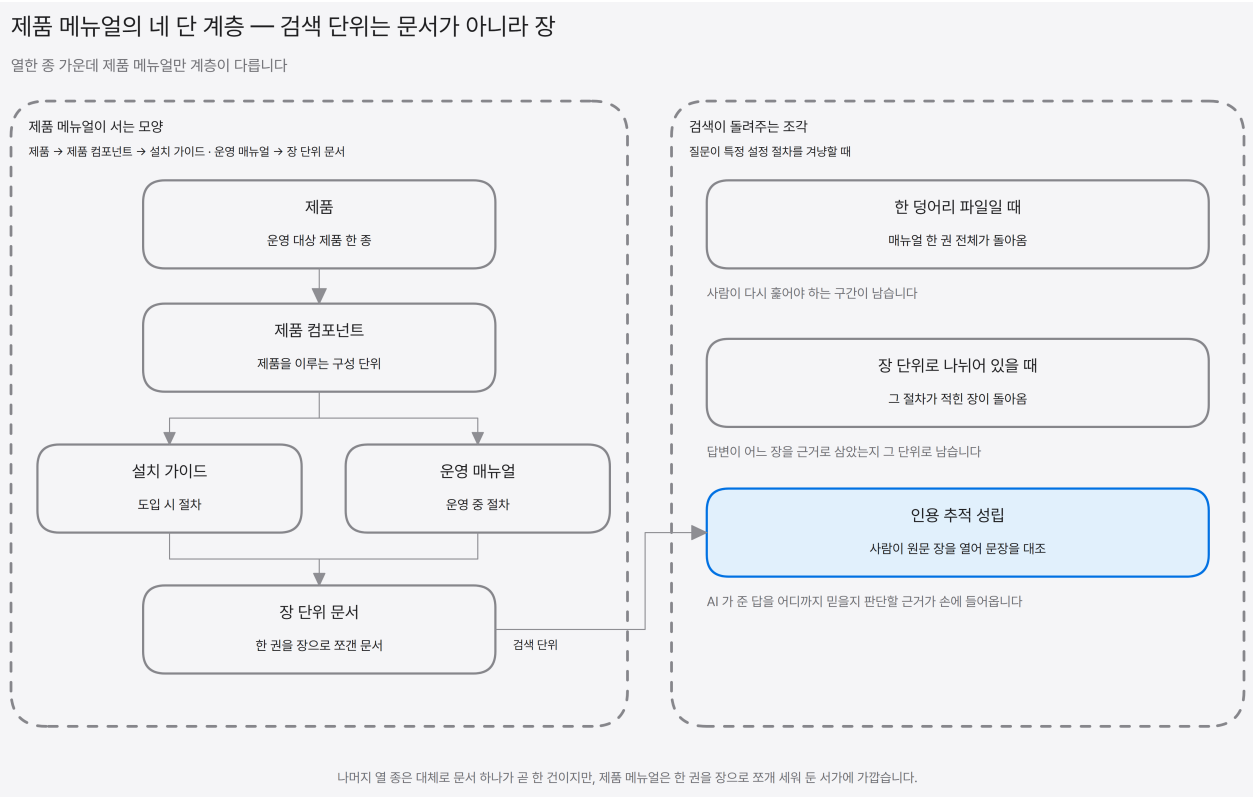


그림 8. 제품 메뉴얼 계층과 장 단위 검색

자기 조직 문서를 옮길 때 이 계층을 그대로 흉내 낼 필요는 없습니다. 제품이 하나뿐이라면 최상위 칸은 얇아지고, 컴포넌트가 많다면 그 층이 두꺼워집니다. 중요한 것은 깊이의 숫자가 아니라 헤딩이 내용의 경계와 일치하는지입니다. 편집 서식으로만 굵게 키워 둔 제목은 검색에서 경계 구실을 하지 못하므로, 옮기는 과정에서 헤딩 수준을 다시 매겨 두시기 바랍니다.

5.2 첫 적재 대상을 고르는 기준

열한 칸을 한꺼번에 채우려는 계획은 대체로 늦어집니다. 문서 형식을 바꾸는 작업이 부서마다 다른 속도로 진행되고, 어느 한 칸이 막히면 전체 일정이 함께 밀리기 때문입니다. 먼저 채울 칸을 하나에서 둘 사이로 고르고 그 칸에서 효용을 확인한 뒤 넓히는 편이 안전합니다. 이 절은 그 선택 기준과 함께, 채워진 문서가 시간이 지나며 어떤 모습으로 쌓이는지를 봅니다.

5.2.1 먼저 넣을 문서의 조건

조건은 셋이면 충분합니다. 첫째, 이미 정형화되어 있어야 합니다. 항목과 순서가 매번 달라지는 문서는 형식을 옮기는 일이 곧 문서를 다시 쓰는 일이 되고, 그 부담이 첫 단계를 멈추게 만듭니다. 둘째, 주기적으로 생산되어야 합니다. 한 번 형식을 정하면 그다음 회차부터 같은 모양이 자동으로 쌓이므로, 초기 비용이 회차 수로 나뉩니다. 셋째, 재참조 빈도가 높아야 합니다. 넣어 두고 아무도 찾지 않는 문서는 검색 품질을 재는 데도 쓰이지 못합니다.

우리 조직의 문서	이미 정형화됐는가	주기적으로 생산되는가	재참조 빈도가 높은가	판단
(예) 일일 점검 기록	예	매일	높음	1순위
(예) 정기 점검 결과	예	월·분기	중간	1순위
(예) 장애 보고	부분	사건 발생 시	높음	2순위
(예) 설치·작업 결과	부분	작업 시	중간	2순위
(예) 제품 설명 자료	예	드물	높음	2순위
(기입)				
(기입)				

표의 빈칸에 조직의 문서 이름을 적어 넣고 세 열을 채워 보시면 순위가 저절로 갈립니다. 세 열이 모두 "예"에 가까운 문서가 1순위이고, 둘까지만 맞는 문서가 2순위입니다. 셋 다 어긋나는 문서는 지금 옮기지 않아도 되며, 나중에 저장소가 자리를 잡은 뒤 다시 판단하면 됩니다. 세 조건을 넘겨 짚는 결정은 대개 문서의 중요도로 고르려다 생기는데, 중요한 문서가 반드시 먼저 넣기 쉬운 문서는 아닙니다.

장애예방보고서와 정기점검보고서가 세 조건을 함께 만족하는 대표 사례입니다. 항목이 고정되어 있고, 정해진 주기로 생산되며, 같은 항목의 지난 기록을 찾는 일이 잦습니다. 이 두 칸을 먼저 채우면 첫 달부터 "지난 기록에서 이 항목이 어땠는가"라는 질의를 실제로 던져 볼 수 있습니다. 검색 품질이 규모에 따라 어떻게 달라지는지도 이 단계에서 확인해 두시는 편이 좋습니다.

5.2.2 리비전 이력이 쌓이는 방식

문서는 한 번 넣고 끝나지 않습니다. 점검 항목은 체크 목록 형태로 본문에 들어가고, 문서가 갱신될 때마다 리비전 이력이 그 문서에 누적됩니다. 같은 문서의 이전 상태가 사라지지 않으므로, 어느 시점에 어떤 항목이 추가되고 어떤 기준이 바뀌었는지가 문서 안에 남습니다. 회차마다 새 파일을 만들어 다른 폴더에 두는 방식과 달라지는 지점이 여기입니다.

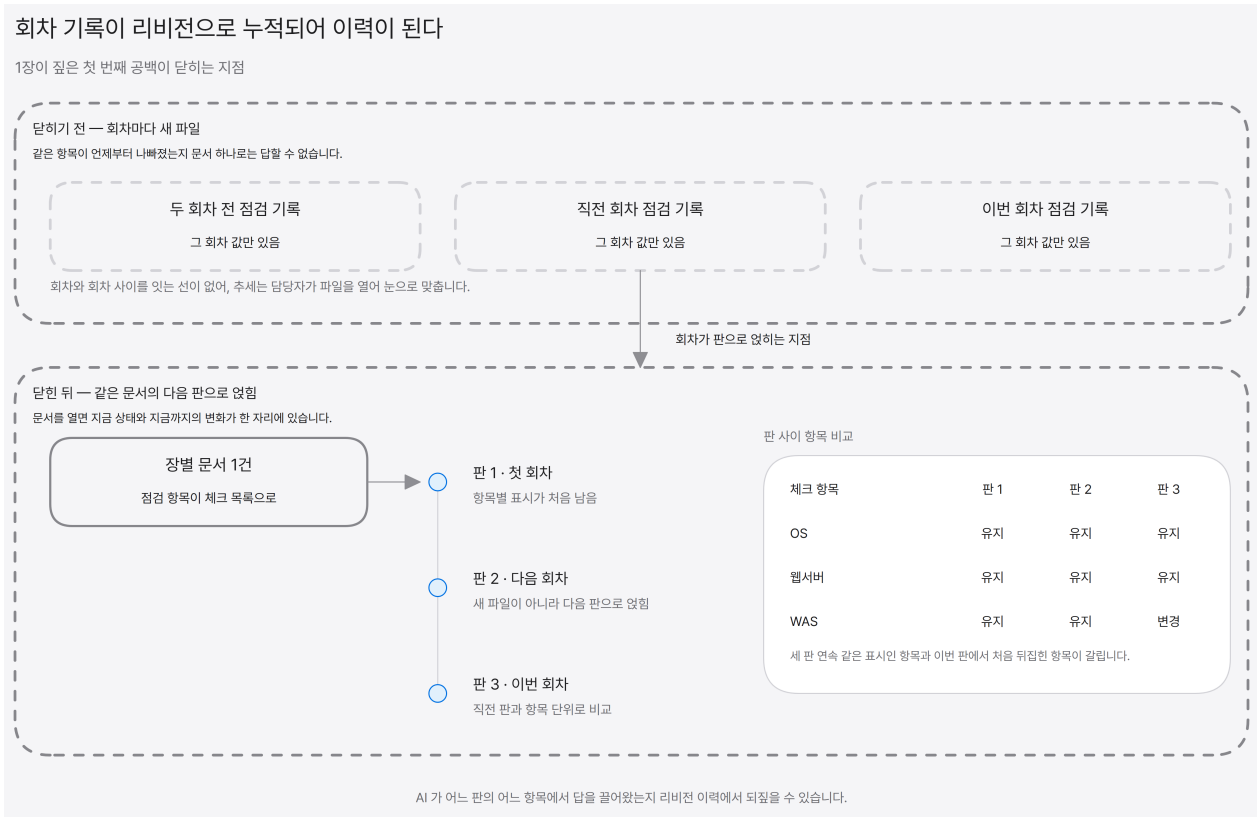


그림 9. 회차 기록이 리비전으로 누적되어 이력이 되는 과정

1장에서 짚은 이력의 공백은 이 지점에서 닫힙니다. 회차마다 기록이 새로 시작되면 항목의 추세가 어디에도 남지 않고, 여섯 달 동안 같은 항목이 어떻게 움직였는지를 묻는 질문에 답할 근거가 생기지 않습니다. 문서마다 리비전 이력이 누적되고 점검 항목이 체크 목록으로 서 있으면, 같은 항목을 축으로 회차를 가로질러 읽을 수 있게 됩니다. 검색 결과에 걸리는 것은 개별 회차 문서이지만, 인용 추적을 따라 이전 리비전으로 내려가면 그 항목의 지난 상태가 이어집니다.

이 누적이 값을 하려면 항목 이름을 바꾸지 않는 규율이 필요합니다. 같은 점검 항목을 회차마다 다른 낱말로 적으면 문서는 쌓여도 항목은 이어지지 않습니다. 항목 이름을 정하고 지키는 일은 4.2에서 적었듯 운영 조직의 몫이며, 첫 적재를 시작할 때 함께 결정해 두는 편이 가장 부담이 적습니다. 항목 수가 늘어나는 상황에서 검색 품질이 어떻게 유지되는지는 문서 수와 문서당 길이에 따라 달라지므로, 도입 규모에 맞춰 확인하시기 바랍니다.

6장: 매일 쌓이는 장애예방보고서

앞 장이 책장의 칸을 보였다면 이 장은 그 칸 하나가 매일 어떻게 두꺼워지는지를 봅니다. 열한 종 가운데 장애예방보고서는 생산 주기가 가장 짧습니다. 연도와 분기와 일자로 내려가는 트리를 따라 매일 한 건이 적재되고, OS·웹서버·WAS·DB·이벤트 항목이 그 안에 함께 들어갑니다. 저장소가 정적인 서고가 아니라 날마다 자라는 기록이라는 사실이 이 컬렉션에서 가장 뚜렷하게 드러납니다.

이 장은 두 걸음으로 나뉩니다. 6.1은 문서가 어디에 어떤 모양으로 놓이는지를, 6.2는 판정 결과가 어떤 형태의 문서로 고정되어 다음 질의의 근거가 되는지를 다룹니다. 두 걸음을 이어 보면 오늘의 출력이 내일의 입력이 되는 순환이 보입니다. 순환이 돌기 시작하면 저장소는 도입 시점보다 여섯 달 뒤에 더 쓸모 있어집니다.

6.1 날짜 트리 적재 규칙

날짜로 내려가는 트리는 단순해 보이지만 조회 방식을 결정합니다. 시점을 알고 문서를 찾는 경로와 내용을 알고 문서를 찾는 경로가 둘 다 열려 있어야 운영 기록이 이력 구실을 하기 때문입니다. 이 절은 트리 구조를 먼저 보이고, 그 안에 매일 들어가는 점검 항목의 구성을 이어서 적습니다.

6.1.1 연도·분기·일자 트리 구조

장애예방보고서는 연도, 분기, 일자 순서로 내려가는 트리에 적재됩니다. 최상위에 연도가 있고 그 아래 분기가 놓이며, 분기 아래에 일자별 문서가 늘어섭니다. 매일 한 건이 그날의 자리에 들어가므로 문서가 쌓여도 위치를 찾는 규칙은 바뀌지 않습니다. 어느 날의 기록을 보려면 그날의 자리로 내려가면 되고, 한 분기의 흐름을 보려면 그 분기 아래를 훑으면 됩니다.

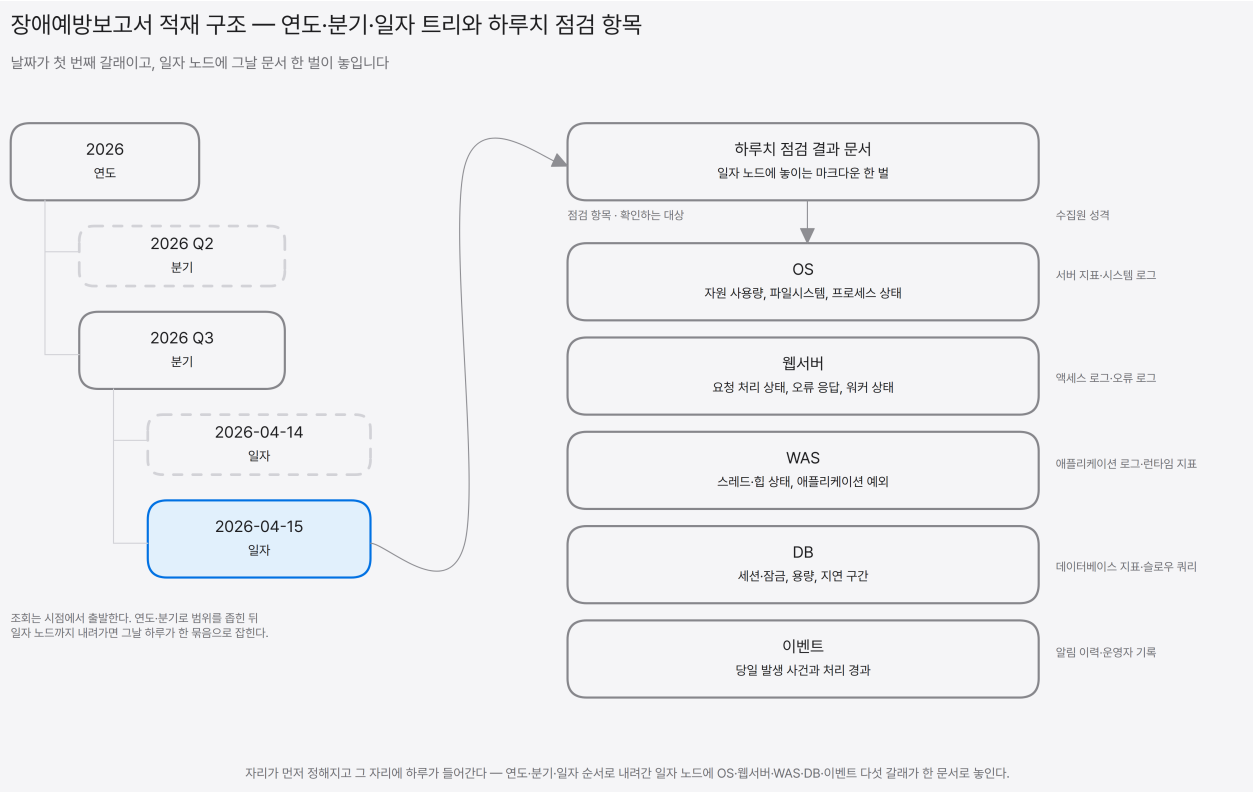


그림 10. 장애예방보고서의 연도·분기·일자 트리와 점검 항목

시점 기준 조회가 열려 있는지가 이력 추적을 가릅니다. 장애가 난 뒤 원인을 좁힐 때 가장 먼저 하는 일이 "그 며칠 전에 무슨 기록이 있었는가"를 확인하는 작업인데, 문서가 날짜로 서 있지 않으면 이 확인이 곧바로 사람 기억에 의존하게 됩니다. 날짜 트리는 그 확인을 위치 이동으로 바꿉니다. 내용으로 찾는 경로는 검색이 말고, 시점으로 찾는 경로는 트리가 맡아 두 경로가 서로를 보완합니다.

트리 깊이가 연도와 분기에서 멈추는 데에도 이유가 있습니다. 매일 한 건씩 쌓이는 문서를 월 단위까지 더 나 누면 한 자리에 들어가는 문서 수가 지나치게 적어지고, 반대로 연도 아래에 일자를 바로 두면 한 해치가 한 자리에 몰립니다. 분기는 운영 조직이 점검 주기와 보고 주기를 잡는 단위이기도 해서, 분기 아래를 훑는 동작 이 실제 업무 리듬과 어긋나지 않습니다.

6.1.2 점검 대상 항목의 구성

매일 기록되는 항목은 다섯 갈래입니다. OS, 웹서버, WAS, DB, 그리고 이벤트입니다. 앞의 네 갈래는 계층 이 서로 다른 구성 요소를 하나씩 맡고, 마지막 갈래는 그날 발생한 사건을 담습니다. 다섯 갈래를 한 문서 안 에 함께 두는 이유는 장애의 원인이 계층을 가로질러 나타나기 때문입니다. 응답이 느려진 원인이 WAS 설정 에 있는지 DB 대기에 있는지 OS 자원에 있는지는 세 기록을 나란히 놓아야 갑니다.

점검 갈래	이 갈래가 담는 것	대응하는 수집원	자기 환경에 대입할 자리
OS	자원 사용과 시스템 상태	서버 운영체제	사용하는 배포판과 커널 계열
웹서버	요청 처리와 접속 처리 상 태	웹 서버 제품	앞단에 둔 웹 서버와 로드 밸런서 구성
WAS	애플리케이션 실행 상태	애플리케이션 서버	운영 중인 WAS 제품과 인 스턴스 수
DB	데이터 계층 상태	데이터베이스	사용하는 DB 제품과 이중 화 구성
이벤트	그날 발생한 사건과 경보	운영 중 발생 기록	경보 등급 체계와 처리 절 차

표의 오른쪽 칸은 비워 둔 자리입니다. 조직마다 쓰는 제품과 구성이 다르므로, 다섯 갈래가 자기 환경의 무엇 에 대응하는지를 먼저 적어 보시면 적재 준비가 절반은 끝납니다. 갈래 이름이 정확히 맞지 않아도 계층이 겹 치면 같은 자리에 들어갑니다. 컨테이너 기반으로 운영한다면 WAS 갈래가 애플리케이션 컨테이너 상태를 담게 되고, 데이터 계층을 관리형으로 쓴다면 DB 갈래가 제품 지표 대신 서비스 지표를 담습니다.

다섯 갈래가 매일 같은 순서로 반복된다는 점이 이 컬렉션의 힘입니다. 항목이 고정되어 있으면 회차 사이의 비교가 가능해지고, 비교가 가능해야 추세가 보입니다. 어제와 오늘 사이의 한 줄 차이는 눈에 띄지 않지만, 같은 항목을 분기 단위로 훑으면 서서히 나빠지는 값이 드러납니다. 항목 수를 늘리는 결정은 이 비교 가능성 을 해치지 않는 선에서 내리시는 편이 좋습니다.

6.2 판정에서 문서 고정까지

기록을 모으는 일과 그 기록에서 판단을 만드는 일은 다릅니다. 그리고 판단을 만드는 일과 그 판단을 다음에 다시 쓸 수 있게 남기는 일도 다릅니다. 이 절은 뒤의 두 단계를 잇습니다. 판정 결과가 어떤 형태의 마크다운 문서로 고정되는지, 그리고 그 문서가 어떤 경로로 다시 검색되는지를 차례로 봅니다.

6.2.1 요약·소견·조치 우선순위가 문서로 고정되는 단계

판정 결과는 세 부분을 갖춘 마크다운 문서로 남습니다. 맨 앞에 그날 상태를 압축한 요약이 오고, 그 뒤에 다섯 갈래에 대응하는 항목별 소견이 이어지며, 마지막에 조치 우선순위가 붙습니다. 세 부분은 읽는 순서이기도 합니다. 요약만 보고 지나갈 날과 소견까지 읽어야 할 날이 갈리고, 조치 우선순위는 그날 손을 대야 할 대상을 짚은 목록으로 남깁니다.



그림 11. 판정이 마크다운 문서로 고정되는 단계

문서로 고정되는 단계가 왜 별도의 단계인지는 반대 경우를 생각하면 분명해집니다. 판정이 화면에만 떠 있다가 사라지면 그날의 판단은 그날로 끝나고, 이력이 되지 않습니다. 같은 증상이 두 달 뒤에 다시 나타났을 때 지난번 소견이 무엇이었는지를 확인할 길이 없어지고, 확인할 길이 없으면 판단은 다시 처음부터 만들어져야 합니다. 마크다운 문서로 고정되면 그 판단이 다른 운영 문서와 같은 자리에 서고, 같은 검색 대상이 됩니다.

형식이 마크다운이라는 점도 이 단계에서 값을 합니다. 요약과 소견과 조치가 헤딩으로 나뉘어 있으면 검색이 세 부분을 각각 다른 조각으로 다루므로, "조치가 필요하다고 적힌 날"과 "소견에 특정 항목이 언급된 날"을 서로 다른 질의로 물을 수 있습니다. 편집 서식으로만 구분해 두면 이 경계가 사라집니다. 출력 형식을 정할 때 세 부분의 헤딩 수준을 고정해 두시기 바랍니다.

6.2.2 재검색으로 이어지는 경로

오늘 고정된 문서는 내일부터 검색 대상입니다. 경로는 단순합니다. 문서가 컬렉션에 적재되면 다른 운영 문서와 같은 검색 순위에 들어가고, 질의가 들어오면 관계형 조회와 벡터 검색이 함께 돌아 결과를 하나의 순위

로 병합합니다. 답변에는 참조한 문서가 달려 나오므로, 어느 날짜의 어느 소견에서 나온 말인지를 되짚어 원문을 열 수 있습니다. 트리를 따라 그날의 자리로 내려가는 경로도 그대로 남아 있습니다.

이 순환은 특별한 장치가 아니라 절차입니다. 매일 한 건이 들어가고, 들어간 문서가 다음 질의의 후보가 되며, 답변에 쓰인 문서가 인용으로 표시되어 다시 열립니다. 한 달치가 쌓이면 한 달 안에서 대조가 가능해지고, 한 분기치가 쌓이면 분기 흐름을 물을 수 있게 됩니다. 누적 효과는 시간이 만드는 것이지 도입 첫날에 생기는 것이 아니며, 그 속도는 적재하는 문서 종류와 생산 주기에 따라 달라집니다.

한 가지 조건을 덧붙입니다. 검색이 답을 잘 골라 오려면 문서 자체가 정확해야 합니다. 인용 추적은 답이 어느 문서에서 왔는지를 보여 줄 뿐 그 문서의 내용이 옳았는지까지 보증하지 않으므로, 소견을 적는 단계의 규율이 결국 답변 품질의 상한이 됩니다. 이 경계를 어떻게 다룰지는 11장에서 이어 다룹니다.

7장: AI 에게 묻은 질문이 그대로 운영 일지가 되는 흐름

운영 현장에서 오가는 질문은 대개 흔적을 남기지 않습니다. 담당자가 옆자리에 묻고 답을 듣고 조치를 마치면 그 대화는 그 자리에서 끝납니다. 같은 질문이 몇 달 뒤 다른 사람의 입에서 다시 나와도 조직은 처음과 같은 시간을 다시 씁니다. 답이 없어서 느린 것이 아니라 이미 나온 답이 남지 않아서 느린 것입니다.

OpsKnow 는 질문을 받는 자리와 기록이 남는 자리를 하나로 둡니다. 운영자가 저장소에 질문을 던지면 답변이 돌아오고, 답변에는 근거가 된 문서가 함께 붙습니다. 질문과 답변과 참조 문서가 한 묶음으로 남으므로, 답을 받는 동작이 곧 기록을 만드는 동작이 됩니다. 기록을 따로 남기려고 별도의 화면을 열 필요가 없다는 뜻입니다.

이 장은 그 묶음이 만들어지는 절차를 따라갑니다. 앞 절에서는 질의가 기록으로 남는 형태를 정리하고, 그 가운데 문자열이 그대로 맞아야 하는 질의 유형을 따로 가릅니다. 뒤 절에서는 기존 운영 일지 작성 절차와 겹치는 지점, 그리고 사람이 여전히 채워야 하는 항목을 봅니다. 읽는 분이 가져가실 판단은 우리 조직의 질의가 일지의 어느 칸까지 채워 주는가입니다.

7.1 질의가 기록으로 남는 절차

질의 기록은 따로 마련한 입력 양식에서 만들어지지 않습니다. 운영자가 답을 얻으려고 한 동작이 그대로 기록의 재료가 됩니다. 이 절은 그 재료가 어떤 조각으로 이루어지는지, 그리고 그 조각 가운데 근사값이 허용되지 않는 자리가 어디인지를 나눠 봅니다.

7.1.1 질문과 답변이 함께 남는 형태

저장소에 던진 질문은 답변과 짝을 이뤄 남습니다. 답변 아래에는 그 답을 만들 때 근거로 쓰인 문서가 목록으로 붙고, 목록의 항목을 열면 원문 자리까지 이어집니다. 인용 추적이 함께 붙으므로 답변의 한 대목이 어느 문서의 어느 대목에서 나왔는지 되짚을 수 있습니다.

이 묶음이 기록으로서 값을 갖는 까닭은 답만 남지 않기 때문입니다. 답만 남으면 나중에 그 답이 맞았는지 가릴 방법이 없지만, 참조 문서가 함께 있으면 근거를 다시 읽어 판단을 되짚을 수 있습니다. 질문 원문이 남는다는 점도 같은 묶음입니다. 같은 상황을 만난 다른 담당자가 자기 질문을 만들기 전에 앞서 사람이 어떤 표현으로 물었는지 먼저 볼 수 있습니다.

구성 요소	남는 내용	나중에 쓰이는 방식
질문 원문	운영자가 입력한 문장 그대로	같은 상황의 질의를 다시 만들 때 출발점
답변 본문	저장소가 돌려준 답	조치 근거의 요약본
참조 문서 목록	답을 만들 때 근거로 쓰인 문서	원문을 열어 맥락을 확인하는 경로
인용 자리	답변의 각 대목이 나온 문서 안쪽 위치	답을 의심할 때 되짚는 지점

구성 요소	남는 내용	나중에 쓰이는 방식
질의 시각	질문과 답이 오간 시점	장애·작업 기록과 시간으로 이어 붙이는 고리

표의 마지막 줄이 운영 일지와 잇는 고리입니다. 장애 대응 도중에 던진 질문은 그 장애의 시간대 안에 놓고, 정기 작업 중에 던진 질문은 그 작업의 시간대 안에 놓입니다. 그래서 나중에 하나의 사건을 되짚을 때 그 날 무엇을 물었고 무엇을 근거로 삼았는지가 사건의 시간축 위에서 함께 읽힙니다. 이 순환이 하루 단위로 반복되면 장애예방보고서처럼 날짜로 쌓이는 컬렉션과도 자연스럽게 맞물립니다.

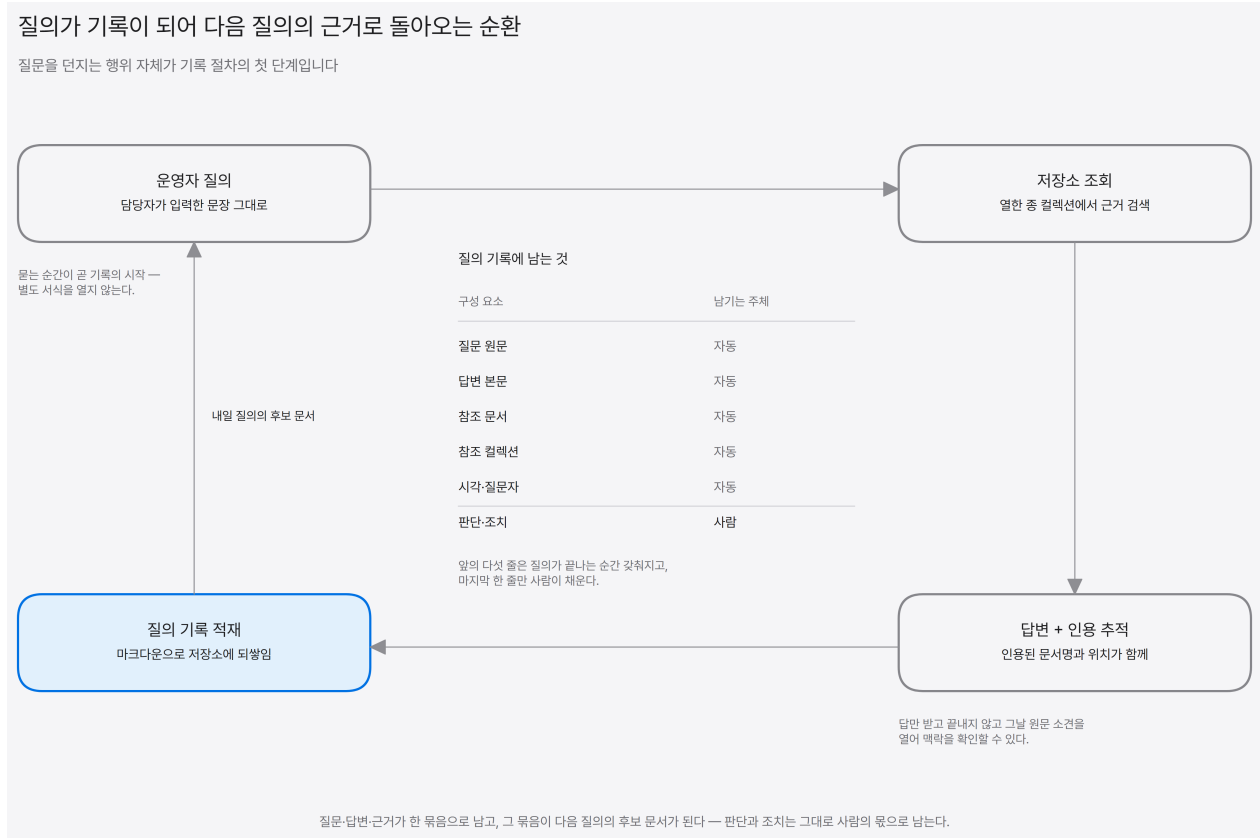


그림 12. 질의가 기록이 되어 다음 질의의 근거로 돌아오는 순환

7.1.2 문자열 그대로 맞아야 하는 질의 유형

운영 질의에는 뜻만 맞으면 되는 것과 글자가 맞아야 하는 것이 섞여 있습니다. "지난 분기에 비슷한 증상이 있었는가" 같은 질문은 표현이 달라도 같은 상황을 다룬 문서를 찾아 주면 됩니다. 반면 "이 컴포넌트가 지원 하는 버전이 무엇인가", "설정 파일이 어느 경로에 있는가" 같은 질문은 비슷한 값을 돌려받는 순간 답이 아니라 사고의 원인이 됩니다.

이 구분이 중요한 까닭은 두 유형이 요구하는 검색 방식이 다르기 때문입니다. 의미가 비슷한 문서를 찾는 방식은 표현의 차이를 견디도록 만들어져 있고, 같은 성질 때문에 한 글자 차이도 견뎌 버립니다. 버전 번호 하나, 경로 한 마디가 다른 문서를 "가까운 문서"로 올려 줄 수 있다는 뜻입니다. 온프레미스 환경에서 검색을 구성한 공개 사례들이 값 조회와 유사도 검색을 함께 두는 이유가 여기에 있습니다.

질의 유형	질문 예	요구되는 일치
버전·지원 범위	이 컴포넌트가 지원하는 버전 범위	문자열 그대로
경로·파일 위치	설정 파일이 놓이는 자리	문자열 그대로
설정값·파라미터	권장 설정값과 단위	문자열 그대로
오류 코드·메시지	특정 코드가 가리키는 상태	문자열 그대로
유사 상황 조회	비슷한 증상을 다룬 과거 사례	뜻이 가까우면 됨
절차·배경 설명	이 작업을 수행하는 순서	뜻이 가까우면 됨

앞의 네 줄이 근사값을 허용하지 않는 자리입니다. 이 네 줄에서 답이 한 글자라도 어긋나면 운영자는 틀린 값을 근거로 손을 대게 되고, 질의 기록에는 틀린 값이 그대로 남아 다음 사람에게 전달됩니다. 그래서 저장소가 어느 질의를 어떤 방식으로 처리하는지는 편의의 문제가 아니라 기록의 신뢰 문제입니다. 이 문제의 답은 9장에서 관계형 조회와 벡터 검색의 역할 분담으로 다시 나옵니다.

7.2 운영 일지와의 접합

질의 기록이 쌓인다고 해서 운영 일지가 사라지지는 않습니다. 두 기록은 서로 다른 목적으로 만들어졌고 겹치는 구간과 겹치지 않는 구간이 나뉩니다. 이 절은 그 경계를 먼저 그리고, 경계 바깥에 남는 사람의 뒀을 이어서 봅니다.

7.2.1 기존 일지 작성 절차와의 겹침

운영 일지는 대개 그날 들어온 문의, 확인한 내용, 취한 조치, 남은 사항 순서로 적힙니다. 질의 기록은 이 가운데 앞의 두 항목과 겹칩니다. 무엇을 물었는지와 무엇을 확인했는지가 이미 묶음으로 남아 있기 때문입니다. 일지를 적는 사람이 기억을 되살려 다시 문장으로 옮기는 단계가 그만큼 줄어듭니다.

겹친다는 말과 대체한다는 말은 같지 않습니다. 질의 기록에는 조치의 결과가 들어 있지 않습니다. 저장소는 무엇을 물었고 어떤 근거로 어떤 답이 나왔는지를 남길 뿐, 운영자가 실제로 어떤 명령을 넣었고 그 뒤 상태가 어떻게 바뀌었는지는 알지 못합니다. 아래 표는 어디까지가 저절로 채워지고 어디부터 사람의 손이 필요한지를 나눈 것입니다.

기존 일지 항목	질의 기록에 남는 부분	사람이 채우는 부분
문의·요청 내용	질문 원문 그대로	요청자와 요청 경위
확인 내용	답변 본문과 참조 문서 목록	답변과 실제 상태를 대조한 결과
조치 내용	남지 않음	실제로 수행한 작업과 순서
결과·후속	남지 않음	조치 뒤의 상태와 남은 과제

오른쪽 열이 비어 있지 않다는 점을 먼저 보아 두시기 바랍니다. 도입 판단에서 흔히 어긋나는 대목이 여기입니다. 질의 기록이 일지를 통째로 대신한다고 기대하면 도입 뒤에 남는 작업이 예상보다 크게 느껴지고, 반대로 겹치는 구간을 세어 보지 않으면 줄어드는 뒀을 놓칩니다. 우리 조직의 일지 양식을 이 표의 왼쪽 열에 대응시켜 보면 겹치는 칸의 수가 곧 줄어드는 작업량의 크기입니다.

7.2.2 남는 수작업

사람이 채워야 하는 항목은 크게 셋입니다. 첫째는 조치의 결과입니다. 무엇을 실행했고 그 뒤 상태가 어떻게 바뀌었는지는 운영자만 압니다. 둘째는 저장소 밖에서 온 근거입니다. 벤더와 주고받은 회신, 현장에서 눈으로 확인한 표시등, 전화로 들은 상황 같은 것들은 문서로 옮겨 놓지 않으면 남지 않습니다. 셋째는 기록을 어느 컬렉션에 붙일지 정하는 판단입니다.

이 부담의 크기는 조직마다 다릅니다. 적재하는 문서 종류와 생산 주기에 따라 달라지므로, 도입을 검토하실 때는 하루에 오가는 질의 건수와 그 가운데 일지로 남겨야 하는 비율을 먼저 세어 보시기 바랍니다. 질의는 많고 일지 요건이 가벼운 조직이라면 겹치는 구간이 넓고, 질의는 적는데 결재용 일지 요건이 무거운 조직이라면 좁습니다. 같은 제품을 놓아도 체감이 갈리는 이유가 여기에 있습니다.

한 가지는 분명히 해 두겠습니다. 질의가 늘어난다고 해서 일지가 저절로 완성되지는 않습니다. 저장소가 만드는 것은 완성된 일지가 아니라 일지의 앞부분을 채울 재료이며, 그 재료를 어느 문서에 어떤 형식으로 붙일지는 조직의 절차가 정합니다. 마크다운 문서로 남는다는 점이 이 대목에서 도움이 됩니다. 사람이 그대로 읽을 수 있고 기계가 구조를 잃지 않고 다시 읽을 수 있으므로, 질의 묶음을 일지 양식에 옮겨 붙이는 작업이 형식 변환이 아니라 문단 이동으로 끝납니다.

8장: 사람과 에이전트가 같은 책장을 보는 접점

저장소를 여는 쪽은 둘입니다. 하나는 브라우저를 열어 문서를 찾아 읽는 사람이고, 다른 하나는 사람 손을 거치지 않고 저장소에 질의를 던지는 코드 에이전트입니다. 두 쪽이 서로 다른 사본을 보는 순간 같은 질문에 다른 답이 나오므로, OpsKnow 는 하나의 저장소를 두 접점에서 여는 구성을 씁니다.

같은 책장이라는 말에는 조건이 붙습니다. 사람이 읽는 문서와 에이전트가 검색하는 문서가 같은 파일이어야 하고, 사람이 문서를 고치면 에이전트 쪽 결과도 고쳐진 문서를 가리켜야 합니다. 열람용 사본과 색인용 사본을 따로 두면 두 결과가 어긋나기 시작하고, 어긋난 다음에는 어느 쪽이 맞는지 가리는 일이 새로운 업무로 늘어납니다. 문서 형식을 마크다운으로 고정하는 선택이 이 조건을 떠받칩니다.

이 장은 두 접점을 나란히 놓고 그 사이의 경계를 그립니다. 앞 절은 사람이 문서에 닿는 경로와 문서를 고칠 때 남는 리비전을 봅니다. 뒤 절은 코드 에이전트가 제품 메뉴얼을 검색하는 경로와, 그 접점이 열어 주는 범위의 끝이 어디인지를 봅니다. 두 절 모두 화면 조작 방법이 아니라 무엇이 보이고 무엇이 보이지 않는지를 다룹니다.

8.1 사람이 쓰는 화면

사람이 저장소를 쓰는 방식은 두 가지로 갈립니다. 하나는 구조를 따라 내려가 문서에 닿는 방식이고 다른 하나는 문서를 고쳐 저장소를 자라게 하는 방식입니다. 이 절은 두 동작을 차례로 보면서, 각각이 무엇을 남기는지 확인합니다.

8.1.1 문서 탐색과 열람 경로

검색어를 떠올리지 못해도 문서에 닿을 수 있어야 합니다. 첫 갈래는 컬렉션입니다. 제품 메뉴얼, 정기점검보고서, 장애보고서, 장애예방보고서, 설치계획서, 설치보고서, 작업가이드, 작업계획서, 작업보고서, CogentAI_보고서, FeedBack 열한 종이 서가처럼 나뉘어 있고, 운영자는 찾는 문서가 어떤 성격인지만 알면 첫 칸을 고를 수 있습니다.

컬렉션 안쪽 구조는 문서의 성격을 따릅니다. 제품 메뉴얼은 제품에서 컴포넌트로 내려가고 다시 설치 가이드와 운영 매뉴얼로 갈라지며 마지막이 장 단위 문서입니다. 장애예방보고서는 연도와 분기와 일자 순서의 날짜 트리를 따라 놓입니다. 앞쪽이 주제로 갈라지고 뒤쪽이 시점으로 갈라지는 셈이라, 운영자는 "무엇에 관한 문서인가"와 "언제 문서인가" 가운데 자기가 아는 쪽으로 먼저 좁힐 수 있습니다.

문서에 닿는 세 갈래 길

컬렉션 이름 · 제품 계층 · 날짜 트리 — 기억하는 단서가 달라도 도착하는 문서는 하나다

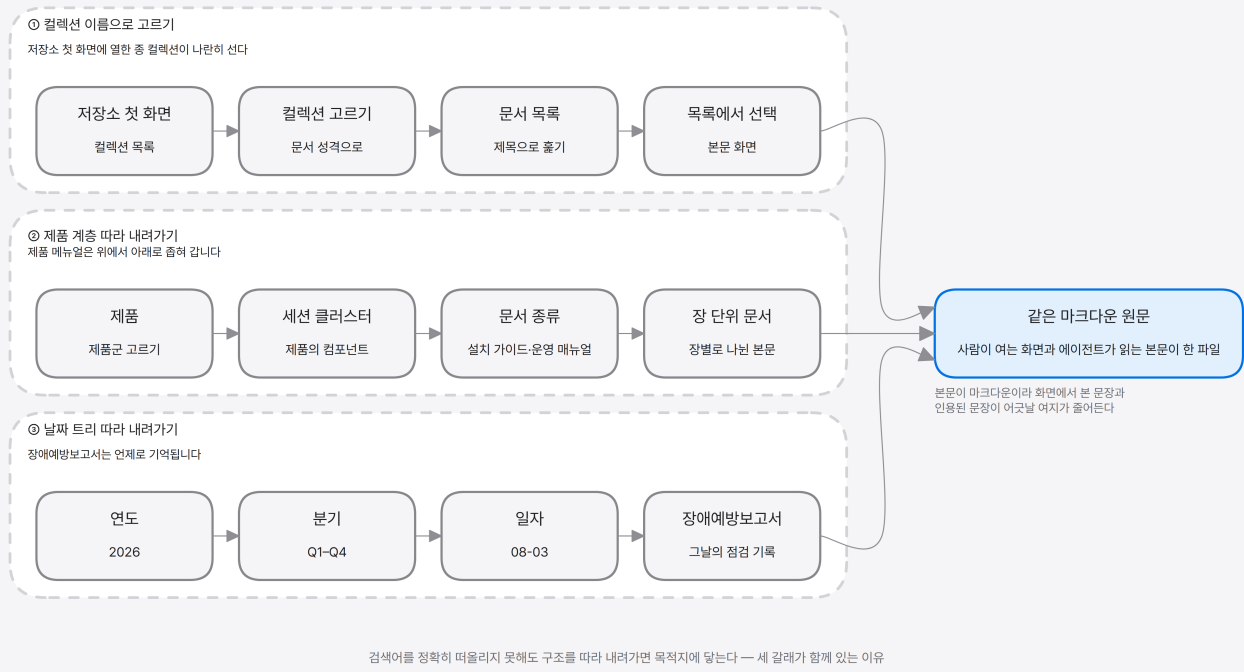


그림 13. 한 문서에 닿는 세 갈래 탐색 경로

기존 위키를 써 온 조직이라면 이 경로가 낯설지 않습니다. 달라지는 지점은 마지막 단계입니다. 제품 메뉴얼의 검색 단위가 문서 전체가 아니라 장이므로, 매뉴얼 한 권을 열어 원하는 대목을 눈으로 훑는 대신 그 대목이 담긴 장에 곧장 닿습니다. 탐색으로 좁히든 검색으로 좁히든 도착하는 자리가 같은 장이라는 점이 이 구조의 실익입니다.

8.1.2 편집과 리비전

문서를 고쳐도 이전 상태가 사라지지 않습니다. 문서마다 리비전 이력이 누적되므로 언제 누가 어느 대목을 바꿨는지가 남습니다. 점검 항목은 체크 목록 형태로 적히므로, 판과 판 사이에서 어떤 항목이 채워졌고 어떤 항목이 빠졌는지가 문장이 아니라 목록 단위로 드러납니다.

증적 관점에서 이 이력이 갖는 값은 두 가지입니다. 하나는 문서 변경 자체가 운영 변경의 흔적이 된다는 점입니다. 설치 구성이 바뀌어 운영 매뉴얼의 한 장을 고쳤다면, 그 리비전이 곧 구성 변경이 반영된 시점의 표시가 됩니다. 다른 하나는 답변이 인용한 문서가 그 뒤에 바뀌었는지 확인할 수 있다는 점입니다. 답변을 받은 시점과 문서의 최신 리비전 시점이 어긋나 있으면 답을 다시 받아 볼 근거가 생깁니다.

확인해 두실 지점은 조직의 증적 요건이 요구하는 항목과 리비전이 남기는 항목이 맞물리는 것입니다. 변경자와 변경 시점과 변경 내용이 필요한 요건이라면 이 이력이 그대로 대응합니다. 변경의 사유나 승인자를 함께 남겨야 하는 요건이라면 그 항목은 문서 본문 안에 자리를 만들어 적어야 하며, 이 판단은 문서 양식을 정할 때 미리 내려 두시는 편이 낫습니다.

8.2 에이전트가 쓰는 접점

저장소를 여는 두 번째 쪽은 사람이 아닙니다. 코드 에이전트가 직접 저장소에 붙어 필요한 대목을 찾아 답을 만듭니다. 이 절은 그 경로가 어떻게 놓이는지, 그리고 그 경로가 어디에서 끊기는지를 봅니다.

8.2.1 코드 에이전트의 제품 메뉴얼 검색

OpsKnow 는 사람 화면 말고 접점을 하나 더 엽니다. 코드 에이전트가 MCP 로 저장소에 붙어 제품 메뉴얼을 검색하고, 검색 결과를 근거로 답을 만듭니다. 검색 단위가 문서 전체가 아니라 장이므로 에이전트가 받는 것은 매뉴얼 한 권이 아니라 질문에 걸리는 장 몇 개이며, 답변의 근거도 그 장에서 나옵니다.

이 구성이 실무에서 갖는 뜻은 조회 경로가 사람의 일정에 묶이지 않는다는 것입니다. 담당자가 자리에 없거나 야간이라 문의를 넣을 곳이 없을 때도 에이전트는 매뉴얼의 해당 장을 열어 근거를 가져올 수 있습니다. 답변에는 사람 화면과 같은 인용 추적이 붙으므로, 에이전트가 내놓은 답도 어느 문서의 어느 대목에서 나왔는지 되짚어 원문을 열 수 있습니다.

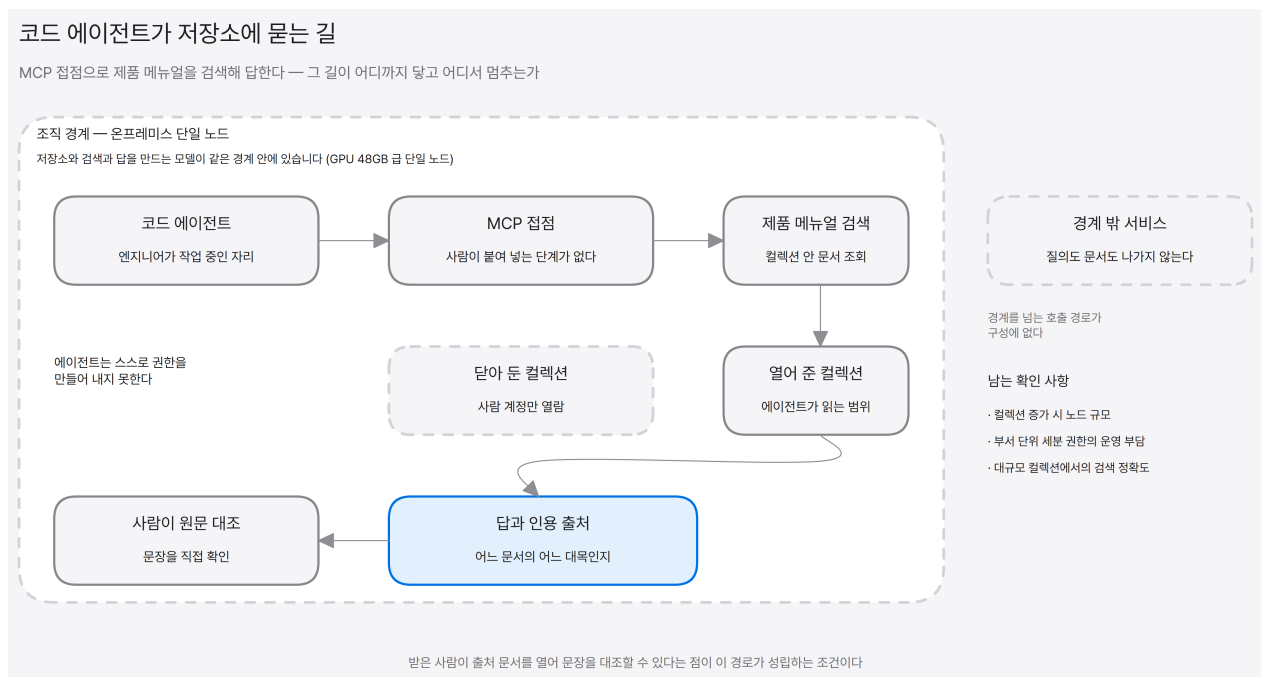


그림 14. 코드 에이전트의 검색 경로와 조직 경계

범위는 분명히 해 두겠습니다. 여기서 말하는 접점은 제품 메뉴얼을 검색해 답하는 조회 경로입니다. 에이전트가 운영 작업을 대신 수행하거나 시스템 상태를 바꾸는 일은 이 접점의 몫이 아닙니다. 조회와 실행은 성격이 다른 권한이며, 이 접점이 여는 것은 앞의 하나뿐입니다.

8.2.2 접근 범위와 권한 경계

이런 구성을 처음 보는 자리에서 나오는 첫 질문은 대개 같습니다. 문서가 조직 밖으로 나가는가입니다. OpsKnow 는 GPU 48GB 급 단일 노드 온프레미스 배치를 전제로 하며, 문서와 질의와 답변이 조직의 경계를 넘지 않는 구성으로 놓입니다. 답을 만드는 모델도 같은 경계 안에 자리하므로 질문 문장이 외부로 실려 나가는 구간이 생기지 않습니다.

경계는 저장소의 바깥만 가르지 않습니다. 에이전트 접점이 여는 범위는 저장소가 허용한 컬렉션까지이고, 그 밖의 시스템은 이 접점이 닿는 자리가 아닙니다. 아래 표는 무엇이 경계 안에 있고 무엇이 밖에 있는지를 구성 사실대로 나눈 것입니다.

구분	경계 안	경계 밖
문서 원본	조직이 운영하는 저장소 노드	외부 저장소로의 복제·전송
질의와 답변	같은 노드 안에서 처리	외부 서비스로의 전달
답변 생성 모델	같은 노드에 자리	외부 모델 호출
에이전트 조회 범위	저장소가 허용한 컬렉션	운영 시스템 조작·상태 변경

표를 우리 환경에 맞춰 읽으실 때 확인하실 지점은 둘입니다. 하나는 GPU 48GB 급 단일 노드를 놓을 자리와 그 노드를 관리할 사람이 있는가입니다. 다른 하나는 열한 종 컬렉션 가운데 어디까지를 에이전트 접점에 열 것인가입니다. 뒤쪽은 제품 설정이 아니라 조직의 문서 등급 정책이 정하는 문제이며, 문서를 적재하기 전에 등급을 나눠 두면 나중에 컬렉션을 다시 가르는 수고를 덜 수 있습니다.

9장: 관계형 데이터베이스와 벡터 데이터베이스를 함께 쓰는 이유

7장에서 운영 질의가 두 갈래로 갈린다는 점을 보았습니다. 글자가 그대로 맞아야 하는 질의와 뜻이 가까우면 되는 질의입니다. 검색 방식을 한 가지만 두면 둘 가운데 한쪽이 반드시 흔들립니다. 값 조회만 두면 표현이 다른 과거 사례를 놓치고, 유사도 검색만 두면 버전과 경로가 어긋난 답이 그럴듯한 얼굴로 올라옵니다.

OpsKnow 는 관계형 조회와 벡터 검색을 함께 두고 두 결과를 하나의 순위로 병합합니다. 두 저장 방식은 경쟁 관계가 아니라 서로 다른 질문을 나눠 맡는 관계입니다. 온프레미스 환경에서 검색을 구성한 공개 사례에서도 두 방식을 함께 두는 구성이 나타나며, 자체 호스팅한 모델을 같은 경계 안에 배치하는 형태가 함께 쓰입니다.

이 장은 그 나눔과 합침을 차례로 봅니다. 앞 절은 어떤 질의를 어느 쪽이 맡는지 가르고, 뒤 절은 두 결과가 하나의 순위로 합쳐지는 절차와 그 합침이 어긋날 때의 거동을 봅니다. 읽는 분이 가져가실 판단은 우리 조직의 질의가 어느 쪽으로 기울어 있는가이며, 그 기울기가 곧 병합 설정을 어디로 조정할지의 근거가 됩니다.

9.1 역할 분담

두 저장 방식은 문서를 다르게 붙잡습니다. 한쪽은 값이 같은지를 보고 다른 한쪽은 뜻이 가까운지를 봅니다. 이 절은 각 방식이 감당하는 질의를 나누고, 그 나눔이 7장의 질의 유형표와 어떻게 맞물리는지 확인합니다.

9.1.1 정확 일치가 필요한 질의

관계형 조회는 값이 맞는지 아닌지를 가립니다. 버전 문자열, 파일 경로, 설정 키와 값, 오류 코드처럼 답이 하나로 정해진 항목에서는 "가깝다"는 판정이 쓸모가 없습니다. 운영자가 필요한 것은 근접한 후보 세 개가 아니라 맞는 값 하나이며, 그 하나가 아닌 답은 답이 아니라 오답입니다.

마크다운 문서의 구조가 이 자리에서 뒹굴 합니다. heading과 목록 경계가 살아 있으므로 문서에서 값에 해당하는 대목을 항목 단위로 끊어 낼 수 있고, 끊어 낸 항목이 조회의 대상이 됩니다. 제품 메뉴얼의 검색 단위가 장이라는 점도 같은 방향입니다. 장이라는 경계가 있어야 어느 제품의 어느 컴포넌트에 딸린 값인지가 흐려지지 않습니다.

질의 유형	담당하는 방식	판정 기준
버전·지원 범위	관계형 조회	값이 같은가
파일 경로·자리	관계형 조회	문자열이 같은가
설정 키와 값	관계형 조회	키가 있고 값이 같은가
오류 코드	관계형 조회	코드가 같은가
유사 증상 사례	벡터 검색	뜻이 얼마나 가까운가
절차·배경 설명	벡터 검색	뜻이 얼마나 가까운가
항목 이름과 기간을 함께 건 조회	두 갈래 함께	값이 맞는 범위 안에서 뜻이 가까운가



그림 15. 관계형 조회와 벡터 검색의 역할 분담

7장의 질의 유형표와 이 표는 같은 줄을 다른 각도에서 본 것입니다. 앞의 표가 "이 질문에 근사값이 허용되는가"를 물었다면 이 표는 "그래서 어느 쪽이 그 질문을 받는가"에 답합니다. 마지막 줄이 두 방식을 함께 쓰는 자리이며, 실제 운영 질의에서 가장 자주 나오는 형태이기도 합니다. 특정 컴포넌트 이름으로 범위를 좁힌 뒤 그 안에서 비슷한 상황을 찾는 질문이 여기에 해당합니다.

9.1.2 의미가 비슷한 문서를 찾는 질의

1장에서 본 원인 선별 지연은 대조할 과거 사례를 찾지 못하는 데서 옵니다. 지금 눈앞의 증상을 적은 말과 몇 해 전 보고서에 적힌 말이 같을 이유는 없습니다. "응답이 느려짐"과 "요청 처리 지연"과 "커넥션 대기 증가"는 서로 다른 문자열이지만 같은 상황을 가리킬 수 있고, 검색어가 한 글자도 겹치지 않는 탓에 값 조회만으로는 서로 만나지 못합니다.

벡터 검색이 이 자리를 감당합니다. 문장을 뜻에 가까운 형태로 바꿔 두고, 질의와 뜻이 가까운 문서를 후보로 올립니다. 표현이 달라도 같은 상황을 다룬 장애보고서나 장애예방보고서의 항목별 소견이 함께 올라오므로, 대응 인력은 자기 기억이 아니라 조직의 기록을 대조 대상으로 삼을 수 있습니다. 담당자마다 다르고 이직과 함께 사라지던 경험의 자리를 문서가 대신하는 지점이 여기입니다.

예를 들어 야간에 특정 서비스의 응답이 늘어졌다면, 운영자는 증상을 자기 말로 적어 던지고 비슷한 상황을 다룬 과거 문서를 후보로 받습니다. 후보에는 그때의 소견과 조치 순서가 함께 들어 있으므로 원인 후보를 좁히는 출발점이 생깁니다. 다만 후보의 품질은 저장소에 무엇이 얼마나 들어 있는지에 달려 있습니다. 문서 수와 문서당 길이에 따라 달라지므로 도입 규모에 맞춰 확인하시기 바랍니다.

9.2 순위 병합

두 검색이 각자 후보를 내놓으면 운영자에게는 목록이 둘 생깁니다. 목록이 둘인 채로 화면에 오르면 어느 쪽을 먼저 볼지 판단하는 일이 사람 몫으로 넘어옵니다. 이 절은 두 목록을 하나로 합치는 절차와, 그 합침이 매끄럽지 않을 때 무엇이 보이는지를 봅니다.

9.2.1 병합 방식과 기본값

두 검색은 서로 다른 점수 체계를 씁니다. 관계형 조회는 맞았는지 아닌지에 가깝고 벡터 검색은 얼마나 가까운지를 숫자로 냅니다. 성격이 다른 두 숫자를 그대로 견줄 수 없으므로, 병합은 점수 자체가 아니라 각 결과 안에서의 순위를 재료로 삼습니다. 어느 쪽이 몇 점을 주었는가가 아니라 어느 쪽에서 몇 번째로 올라왔는가를 보는 방식입니다.

절차는 세 단계입니다. 먼저 두 검색이 각자 후보 목록을 만들고, 다음으로 같은 문서가 양쪽에 나타나면 하나로 합치며, 마지막으로 합쳐진 목록을 하나의 순위로 다시 세웁니다. 양쪽에 모두 걸린 문서는 한쪽에만 걸린 문서보다 위로 올라갑니다. 값도 맞고 뜻도 가까운 문서가 답의 근거로 가장 먼저 쓰인다는 뜻입니다.

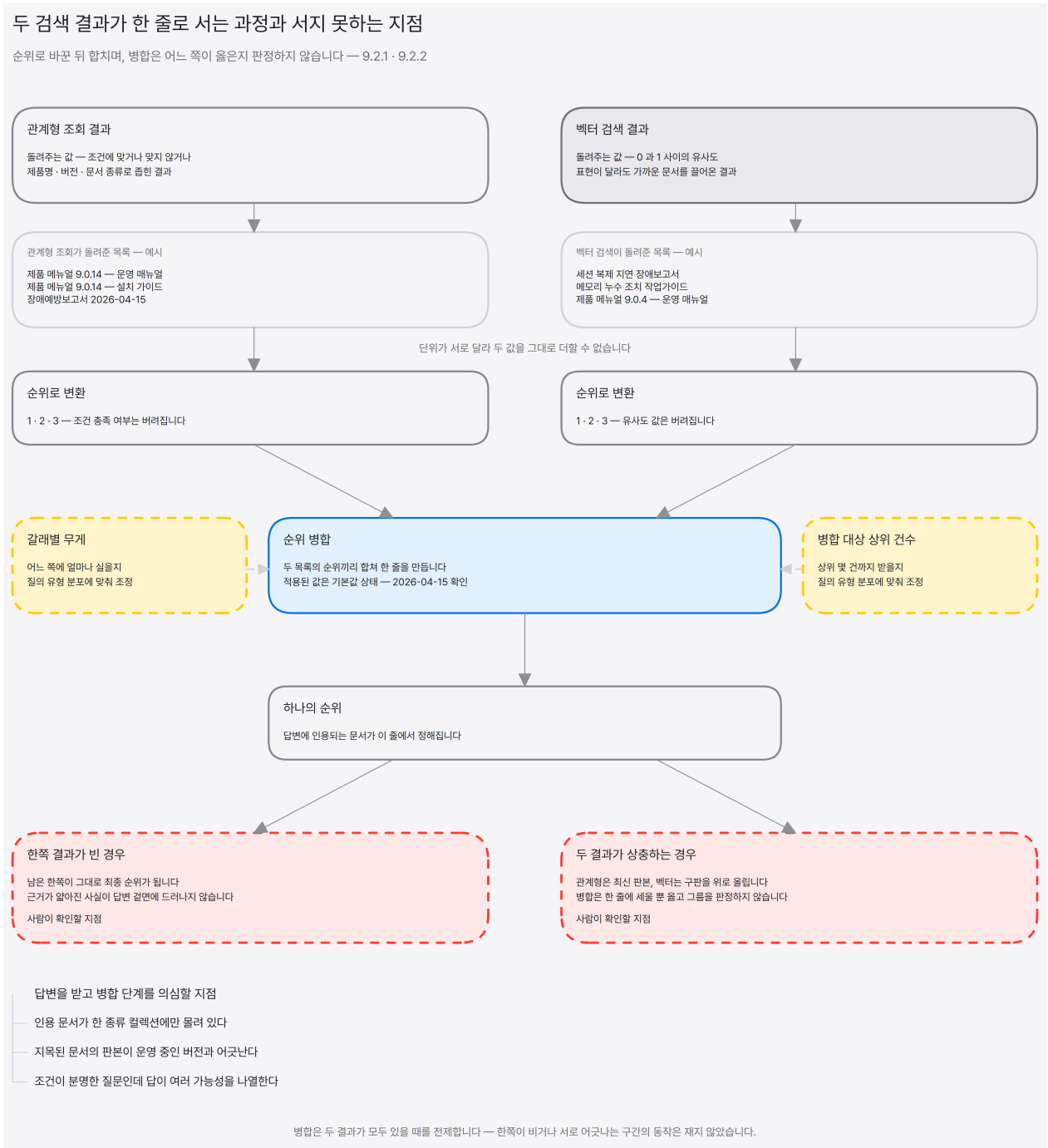


그림 16. 두 검색 결과가 하나의 순위로 합쳐지는 흐름

어느 쪽에 무게를 더 둘지는 고정된 값이 아닙니다. 질의 유형 분포에 맞춰 조정할 수 있습니다. 버전과 설정 값을 확인하는 질의가 많은 조직과 과거 사례를 뒤지는 질의가 많은 조직은 알맞은 무게가 서로 다릅니다. 도입 초기에 오간 질의를 유형별로 세어 보면 어느 쪽으로 기울여야 할지가 숫자로 드러나므로, 그 셈을 운영 초기 과제로 잡아 두시기 바랍니다.

9.2.2 병합이 실패하는 경우

병합이 늘 매끄럽지는 않습니다. 어긋나는 형태는 셋입니다. 한쪽 결과가 비는 경우, 두 쪽이 서로 다른 문서를 위로 올리는 경우, 그리고 양쪽이 같은 문서를 올렸는데 그 문서가 지금 구성과 맞지 않는 시점의 것인 경

우입니다. 실패의 형태를 알아야 답을 의심할 자리를 알 수 있으므로, 각각이 화면에서 어떻게 보이는지를 정리해 두는 편이 낫습니다.

한쪽이 비는 경우가 가장 알아보기 쉽습니다. 값 질의를 던졌는데 관계형 결과가 비었다면 저장소에 그 값이 적힌 문서가 없거나, 값이 문서 안에서 항목으로 끊기지 않은 상태입니다. 이때 벡터 결과만으로 답이 만들어지면 "비슷한 값"이 답의 자리에 올라올 수 있습니다. 참조 문서 목록에 값이 적힌 문서가 아니라 그 값을 언급만 한 문서가 들어 있다면 이 경우를 의심하시기 바랍니다.

상충하는 경우는 눈에 덜 띵니다. 두 쪽이 각자 다른 문서를 1순위로 올리면 병합된 목록의 위쪽에 성격이 다른 문서가 나란히 서고, 답변은 그 둘을 함께 읽어 만들어집니다. 문장 하나 안에서 두 문서의 내용이 섞이면 각 대목은 맞는데 합쳐 놓은 문장은 어긋나는 상태가 생깁니다. 아래는 답을 받은 뒤 사람이 확인할 지점입니다.

- 값 질의의 답이 참조 문서 원문의 값과 글자까지 같은가
- 참조 문서의 리비전 시점이 지금 구성과 맞는 시점인가
- 답변의 한 문장이 서로 다른 문서 두 곳을 섞어 만든 것은 아닌가
- 참조 목록에 같은 주제를 다른 문서가 하나뿐인가

네 줄 모두 인용 추적이 있어야 확인할 수 있는 항목입니다. 답변만 놓고는 어느 것도 가릴 수 없고, 답변에 붙은 참조 문서를 열어야 비로소 대조가 시작됩니다. 이 확인이 왜 사람의 몫으로 남는지, 인용 추적이 무엇을 보장하고 무엇까지는 보장하지 않는지는 11장에서 이어 봅니다.

10장: GPU 48GB 에서 도는 온프레미스 구성과 그 규모

앞 장까지는 무엇을 모으고 어떻게 찾는지를 다뤘습니다. 이 장이 답하는 질문은 하나로 좁혀집니다. 그 구성이 우리 전산실 안에서 실제로 돌아가느냐입니다. 검색 품질이 아무리 좋아도 전용 GPU 집합과 여러 대의 노드가 전제 조건이라면, 대부분의 운영 조직에서 검토는 그 자리에서 멈춥니다.

OpsKnow 의 배치 조건은 GPU 메모리 48GB 급 단일 노드 온프레미스입니다. 노드 하나 안에서 문서 보관과 색인, 관계형 조회와 벡터 검색, 답변 생성과 인용 반환이 모두 끝납니다. 문서와 질의와 답변은 조직 경계를 넘지 않으며, 운영 중 외부 서비스로 나가는 호출 경로를 두지 않습니다. 숫자를 먼저 적는 까닭은 이 조건이 도입 가능 여부를 가르는 첫 관문이기 때문입니다.

읽는 순서는 구성 조건(10.1)에서 감당 범위(10.2)입니다. 앞 절은 어떤 하드웨어 위에서 무엇이 함께 도는지를 적고, 뒤 절은 그 한 대가 어느 정도 규모까지 받아 내는지를 독자가 자기 부서 숫자로 대입하게 만듭니다. 뒤 절의 마지막 항은 규모가 커질 때 무엇을 다시 재봐야 하는지를 조건으로 남기므로, 11장의 신뢰 경계와 이어서 읽으시기 바랍니다.

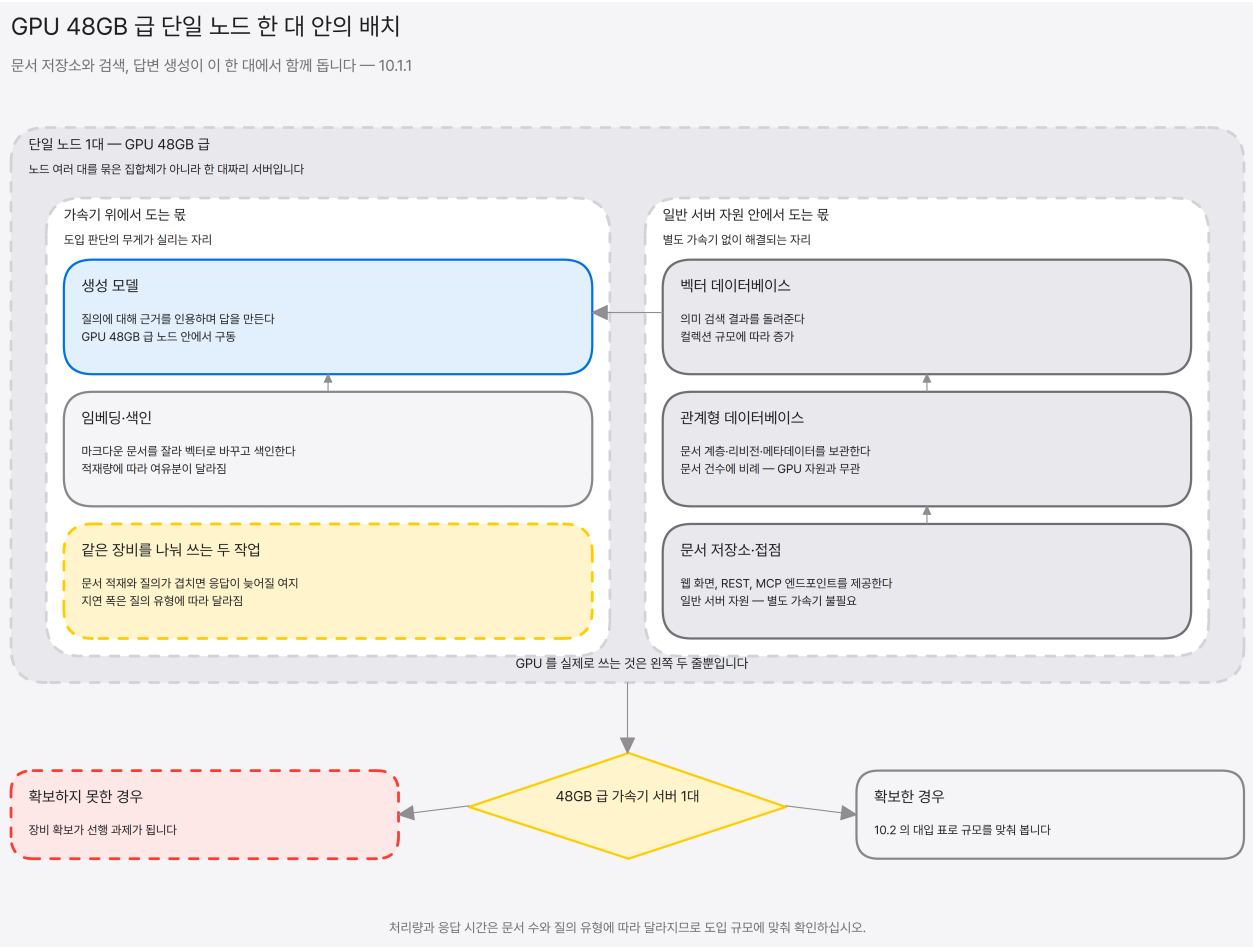


그림 17. GPU 48GB 급 단일 노드 안의 구성

10.1 GPU 48GB 구성의 조건

구성 조건은 두 갈래로 갈립니다. 하나는 어떤 자원 위에서 도느냐이고, 다른 하나는 그 자원 밖으로 무엇이 나가느냐입니다. 앞쪽은 예산과 조달의 문제이고 뒤쪽은 반출 통제의 문제이므로, 두 질문은 결재 라인에서 서로 다른 사람에게 갑니다. 이 절은 자원 조건(10.1.1)과 데이터 이동 경계(10.1.2)를 차례로 봅니다.

10.1.1 실측 구성과 구동 조건

조건부터 숫자로 적습니다. OpsKnow 는 GPU 메모리 48GB 급 단일 노드에서 도는 온프레미스 구성입니다. 노드가 한 대라는 말은 답변 생성 모델과 임베딩 모델, 관계형 데이터베이스와 벡터 검색, 문서 저장소가 같은 기계 안에 함께 상주한다는 뜻입니다. 검색 계층과 추론 계층을 별도 클러스터로 벌려 놓는 전제가 없으므로, 도입 검토에서 따져야 할 항목은 서버 대수가 아니라 그 한 대의 자원 배분입니다.

자원은 성격에 따라 나뉩니다. 언어 모델과 임베딩 모델은 GPU 메모리를 차지하며 이 둘이 48GB 안에서 함께 살아야 합니다. 반면 관계형 조회와 벡터 색인, 마크다운 원문 보관은 GPU 를 쓰지 않고 CPU 와 메모리, 디스크에 부담을 겁니다. 두 부담이 서로 다른 자원에 걸리기 때문에 한 노드 안에서 역할을 나누는 구성이 성립합니다.

구성 요소	하는 일	자원 조건
답변 생성 모델	검색된 문서를 근거로 답변 문장을 만듭니다	GPU 메모리 상주
임베딩 모델	문서와 질의를 검색용 벡터로 바꿉니다	GPU 메모리 상주
벡터 검색	표현이 달라도 의미가 가까운 문서를 찾습니다	CPU·메모리, 색인은 디스크
관계형 데이터베이스	제품·컴포넌트·문서 종류·일자 조건으로 범위를 좁힙니다	CPU·메모리, 데이터는 디스크
문서 저장소	마크다운 원문과 리비전 이력을 보관합니다	디스크
질의 처리부	두 검색 결과를 하나의 순위로 병합하고 인용을 붙입니다	CPU

표에서 GPU 메모리를 요구하는 항목은 위쪽 두 줄뿐입니다. 나머지 네 줄은 일반적인 서버 자원 위에서 돌아가므로, 증설 판단이 필요한 지점은 문서가 늘어날 때의 디스크와 색인 메모리 쪽입니다. 문서를 마크다운으로 보관하는 선택이 여기서도 값을 합니다. 원문이 이진 형식이 아니라 텍스트이므로 보관 용량이 상대적으로 가볍고, 리비전 이력이 쌓여도 증가 폭을 가늠하기 쉽습니다.

구동 조건에서 함께 확인할 것은 문서 종류의 폭입니다. 열한 종 컬렉션은 제품 메뉴얼처럼 값이 고정된 문서와 장애예방보고서처럼 매일 새로 쌓이는 문서를 함께 담습니다. 앞쪽은 한 번 적재하면 개정 시점에만 손이 가고, 뒤쪽은 매일 색인 대상이 늘어납니다. 노드 한 대의 자원 계획은 이 두 성격이 각각 얼마나 되는지에 따라 달라지므로, 다음 절의 대입 표에서 자기 부서 숫자로 채워 보시기 바랍니다.

10.1.2 문서를 외부로 내보내지 않는 구성

운영 문서에는 호스트명과 계정 체계, 네트워크 구간과 장애 당시의 판단 근거가 함께 적혀 있습니다. 이런 문서를 검색 대상으로 올린다는 말은 조직에서 가장 민감한 텍스트를 한곳에 모은다는 뜻입니다. 그러므로 공공과 금융 부문 검토자가 첫 회의에서 확인하는 항목은 검색 정확도가 아니라 그 텍스트가 어디까지 이동하느냐입니다.

OpsKnow의 배치 조건에서 문서와 질의와 답변은 조직 경계를 넘지 않습니다. 문서 적재와 임베딩 생성, 벡터 색인과 관계형 조회, 답변 생성과 인용 반환이 모두 같은 노드 안에서 일어납니다. 모델 가중치는 배치 시점에 반입해 노드 안에 두고 쓰며, 운영 중 외부 추론 서비스를 호출하는 경로를 두지 않습니다. 질의를 받아 답변을 돌려주는 왕복 전체가 내부에서 닫힙니다.

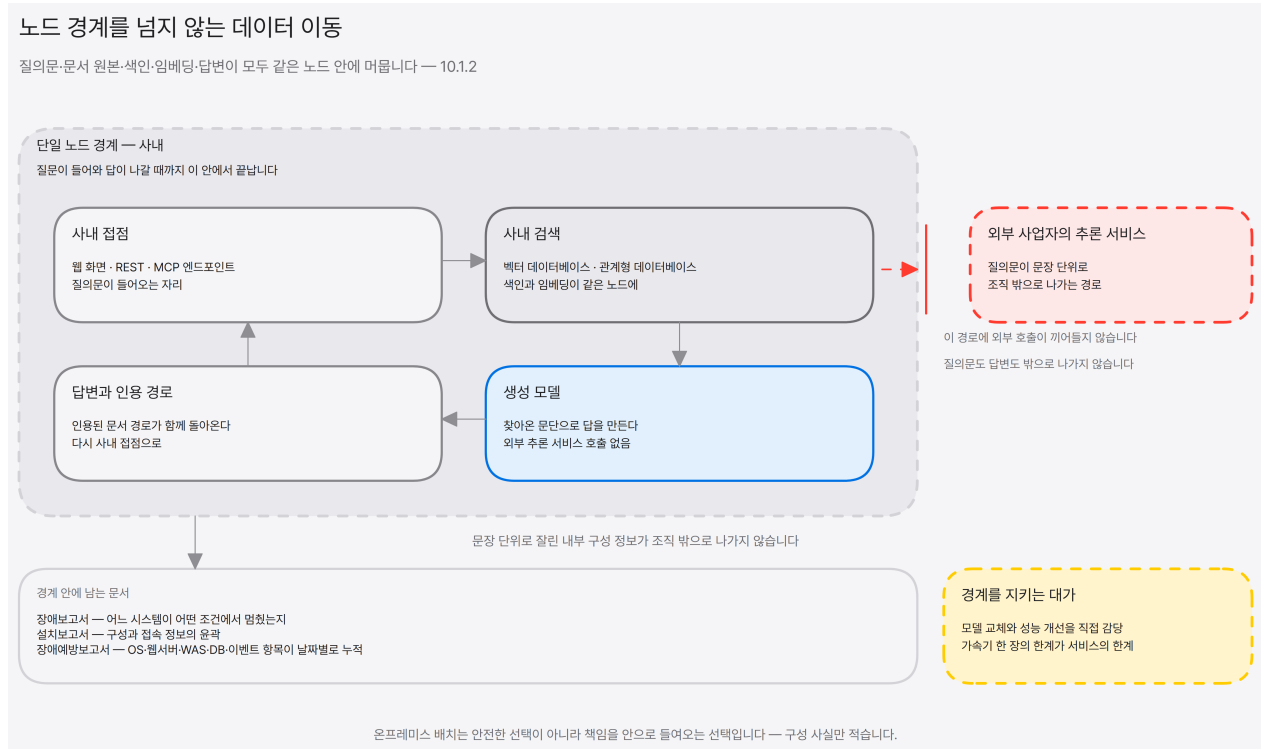


그림 18. 노드 경계를 넘지 않는 데이터 이동

단계	무엇이 움직이는가	어디까지 가는가
문서 적재	마크다운 원문	노드 내부 문서 저장소
임베딩 생성	문서 본문과 질의 문장	노드 내부 GPU
색인	벡터와 문서 속성	노드 내부 벡터 색인·관계형 데이터베이스
질의	운영자가 입력한 문장	노드 내부 질의 처리부
답변 생성	검색된 문서와 생성 문장	노드 내부 GPU
인용 반환	참조 문서 식별자와 원문 링크	노드 내부 문서 저장소

표에 외부로 향하는 행이 없다는 점이 이 구성의 성질입니다. 반출 심사에서 확인할 대상은 사용 약관이나 처리 위탁 범위가 아니라, 노드에 걸린 네트워크 정책과 계정 권한이라는 익숙한 항목으로 옮겨 갑니다. 코드에

이전트가 MCP 로 제품 메뉴얼을 검색해 답하는 점점 역시 같은 경계 안에서 동작하므로, 개발자 도구가 붙었다는 이유로 경계가 넓어지지 않습니다.

경계가 닫혀 있다는 사실이 접근 통제를 대신하지는 않습니다. 열한 종 컬렉션이 한 저장소에 모이면 이전에는 부서별 공유 폴더에 흩어져 있던 문서가 한 번의 질의로 함께 검색됩니다. 그러므로 어떤 컬렉션을 누구에게 열지, 장애보고서와 작업계획서를 같은 권한으로 묶을지는 도입 시점에 조직이 정해야 할 항목으로 남습니다.

10.2 그 규모가 감당하는 범위

노드 한 대라는 조건을 받아들였다면 다음 질문은 그 한 대가 우리 부서 규모를 받아 내느냐입니다. 이 질문은 일반론으로 답할 수 없습니다. 같은 대상 시스템 수라도 제품 메뉴얼이 수천 장인 조직과 수백 장인 조직은 색인 규모가 한 자릿수 배 이상 벌어지기 때문입니다. 이 절은 자기 숫자를 대입할 표(10.2.1)와 규모가 커질 때 달라지는 항목(10.2.2)을 차례로 봅니다.

10.2.1 대상 시스템 수 대입 기준

대입의 출발점은 대상 시스템 수입니다. 다만 검색 대상은 시스템이 아니라 문서이므로, 시스템 수에 문서 생산 규칙을 곱해야 실제 색인 대상이 나옵니다. 제품 메뉴얼은 제품에서 컴포넌트로, 다시 설치 가이드와 운영 매뉴얼로, 마지막에 장 단위 문서까지 내려가며 검색 단위가 문서가 아니라 장이라는 점도 함께 곱해집니다.

아래 표는 빈칸으로 두었습니다. 오른쪽 칸을 자기 부서 숫자로 채우면 색인 대상 규모가 나오고, 그 규모가 노드 한 대에 올릴 만한 크기인지 판단할 근거가 생깁니다. 산정 방법 열은 어디서 그 숫자를 구할지를 적어 두었으므로, 자료를 새로 만들지 않고 기존 운영 대장에서 옮겨 적으실 수 있습니다.

항목	우리 부서 값	산정 방법
대상 시스템 수		운영 중인 시스템 대장의 행 수
시스템당 제품 메뉴얼 장 수		대표 시스템 한 곳의 설치 가이드·운영 매뉴얼 장 수
하루에 쌓이는 장애예방보고서 건수		점검 대상 수 × 하루 점검 회차
분기 정기점검보고서 건수		대상 시스템 수 × 분기 점검 회차
월 작업 관련 문서 건수		월 작업 승인 건수 × 작업계획서·작업보고서 2건
누적 장애보고서 건수		최근 3년 장애 대장의 행 수
설치 관련 문서 건수		연간 신규·이관 구축 건수 × 설치계획서·설치보고서 2건

표를 채우고 나면 두 가지가 드러납니다. 하나는 첫 적재로 한 번에 올라갈 기존 문서량이고, 다른 하나는 매일 또는 매월 새로 붙는 증가분입니다. 장애예방보고서는 연도와 분기와 일자 트리를 따라 매일 적재되며 OS·웹서버·WAS·DB·이벤트 항목을 각각 남기므로, 증가분에서 큰 몫을 차지하는 쪽은 대체로 이 컬렉션입니다.

두 숫자를 구분하는 일이 실제 판단을 바꿉니다. 기존 문서량이 크고 증가분이 작은 조직은 첫 전환 작업이 고비이고 그 뒤로는 손이 덜 갑니다. 반대로 기존 문서량이 작아도 매일 점검 기록이 쏟아지는 조직은 적재 자동화를 먼저 준비해야 합니다. 어느 쪽인지에 따라 도입 초기에 배치할 인력의 성격이 달라지므로, 표는 결재 문서에 그대로 붙일 수 있게 채워 두시는 편이 좋습니다.

10.2.2 규모에 따라 달라지는 것

색인 규모가 커진다고 검색 품질이 같은 비율로 나빠지지는 않습니다. 다만 문서 수와 문서당 길이에 따라 달라지므로, 도입 규모에 맞춰 확인하십시오. 문서가 늘면 같은 질의에 걸리는 후보 문서가 함께 늘고, 그중에서 상위 몇 건을 답변 근거로 올릴지에 따라 사람이 읽어야 할 원문의 양이 달라집니다. 문서당 길이도 함께 작용합니다. 한 장 안에 여러 주제가 섞여 있으면 검색 단위가 커져 근거가 뭉뚱해지고, 장이 잘게 나뉘어 있으면 근거가 뽕족한 대신 여러 건을 함께 읽어야 합니다.

순위 병합도 같은 성질을 가집니다. 관계형 조회 결과와 벡터 검색 결과에 각각 얼마나 무게를 실을지는 질의 유형 분포에 맞춰 조정할 수 있습니다. 버전과 경로를 묻는 질의가 많은 조직과 과거 사례를 되짚는 질의가 많은 조직은 적절한 배분이 서로 다릅니다. 어느 쪽이 많은지는 도입 후 몇 주간의 질의 기록으로 드러나므로, 초기 기본값을 그대로 쓰다가 분포가 보이면 조정하는 순서가 무난합니다.

운영 부담과 전환 작업량도 규모의 함수입니다. 운영 부담은 적재하는 문서 종류와 생산 주기에 따라 달라집니다. 열한 종을 한꺼번에 열어 두는 조직과 제품 매뉴얼과 장애예방보고서 두 종으로 시작하는 조직은 매일 드는 손이 다릅니다. 전환 작업량은 기존 문서의 형식과 수량에 따라 달라집니다. 이미 텍스트로 관리하던 조직은 헤딩 구조만 다듬으면 되고, 문서 편집기 파일과 표 이미지가 섞여 있는 조직은 변환 자체가 첫 과제가 됩니다.

그러므로 규모 판단은 한 번에 끝내는 계산이 아니라 도입 범위를 정하는 선택으로 다루시는 편이 낫습니다. 먼저 올릴 컬렉션을 좁히면 색인 규모와 전환 작업량과 운영 부담이 함께 줄고, 그 상태에서 우리 질의 분포와 검색 결과를 확인한 뒤 범위를 넓힐 수 있습니다. 어디까지 넓힐지는 앞의 대입 표에 채운 숫자가 알려 줍니다.

11장: 답이 흔들리지 않게 만드는 장치와 남는 한계

10장까지 오면 두 가지 질문에는 답이 섰습니다. 무엇을 먼저 넣을지, 그리고 우리 환경에서 돌릴 수 있는지입니다. 남은 질문은 성격이 다릅니다. 화면에 뜬 답변을 어디까지 믿고 운영 판단에 쓸 것이냐입니다. 이 질문은 기능 목록으로 답할 수 없고, 보장되는 범위와 사람이 확인할 범위를 함께 그어야 답이 됩니다.

OpsKnow 는 답변마다 참조 문서를 되짚어 원문을 열 수 있게 합니다. 이 장치가 서 있으면 답변은 검토 가능한 주장이 되고, 없으면 출처를 알 수 없는 문장이 됩니다. 다만 이 장치가 보장하는 것은 어느 문서에서 나왔느냐까지입니다. 그 문서를 그렇게 읽은 것이 옳으냐는 다른 층위의 문제이며, 이 장은 두 층위를 갈라 놓는데 지면을 씁니다.

읽는 순서는 인용 추적(11.1)에서 지속 부담(11.2)입니다. 앞 절은 답변과 원문 사이의 경로를 다루고, 뒤 절은 그 경로가 계속 살아 있으려면 조직이 무엇을 계속해야 하는지를 다룹니다. 두 절은 결이 다르지만 같은 결론으로 모입니다. 저장소는 세워 두는 물건이 아니라 유지하는 물건입니다.

신뢰 경계 — 인용 추적이 보장하는 범위와 사람이 확인하는 범위

11.1 인용 추적

기계의 몫 — 출처

인용 추적이 보장하는 범위

○

답변의 출처가 사내 저장소 문서

저장소 밖 지식으로 지어낸 문장과 갈라짐

○

문서의 위치와 판본

답변을 읽다가 곧바로 원문을 열 수 있음

○

문서마다 누적된 리비전 이력

특정 시점의 특정 판본에 도착

질의 유형 → 인용되는 문서

설치 절차 → 제품 매뉴얼 설치 가이드 장

같은 증상 이력 → 장애보고서

어제 지표 정상 여부 → 장애예방보고서 해당 일차

점검 항목 누락 → 정기점검보고서

지난주 변경 결과 → 작업계획서-작업보고서

기계는 어디서 가져왔는지까지 책임진다

책임의 경계

사람의 몫 — 해석

인용이 맞아도 원문을 열어야 판별되는 것

질의 유형	인용이 맞아도 생기는 문제	사람이 확인할 지점
조치 절차	적용 조건이 빠지고 절차만 남음	원문의 전체 조건과 예외 항목
설정값	특정 버전 값이 권장값으로 넓혀짐	대상 제품-버전의 일치 여부
원인	상관 기술이 인과 결론처럼 서술됨	원문이 확정인지 추정인지
이력	다른 시점 기록이 한 흐름으로 불음	각 기록의 일자와 당시 환경
문서에 없는 값	인접 문서로 매워 답함	인용 문서가 그 항목을 담는지
판단	문서에 없는 결론이 인용과 함께 나옴	결론의 근거가 원문에 있는지

여섯 유형 모두 인용 링크를 눌러 원문을 열어야 판별된다

지금 우리 환경에 맞는지는 담당자가 책임진다

인용 추적은 출처를 보장하지 해석을 보장하지 않는다.

그림 19. 인용 추적이 보장하는 범위와 사람 확인이 필요한 범위

11.1 인용 추적

인용 추적은 답변의 뒤쪽을 여는 장치입니다. 답변 문장이 어떤 문서의 어느 대목에 기대고 있는지를 되짚을 수 있으면, 읽는 사람은 문장을 그대로 받는 대신 근거를 직접 확인할 수 있습니다. 이 절은 그 장치가 보장하는 범위(11.1.1)와 그 범위를 넘어 남는 오독 가능성(11.1.2)을 차례로 봅니다.

11.1.1 인용 추적이 보장하는 것

운영 판단에서 답변의 값어치는 문장 자체가 아니라 그 문장을 되짚을 수 있느냐로 결정됩니다. 새벽에 응급 지연을 보고받은 담당자가 "예전에 같은 증상이 있었고 커넥션 풀을 늘려 해결했다"는 답을 받았다면, 그 답

2026-08-17

45 / 53

msap.ai

만으로는 손을 대기 어렵습니다. 어느 시스템의 어느 시점 기록인지, 그때 설정값이 지금과 같은지를 확인해야 조치가 성립하기 때문입니다.

OpsKnow 는 답변마다 참조 문서를 되짚어 원문을 열 수 있습니다. 답변 아래에 붙은 문서를 열면 마크다운 원문이 그대로 나오고, 문서에 누적된 리비전 이력을 함께 볼 수 있으므로 그 대목이 언제 쓰였고 이후에 고쳐졌는지도 확인됩니다. 제품 메뉴얼은 제품과 컴포넌트를 거쳐 장 단위까지 내려가는 계층 위에 있으므로, 열린 문서가 어느 제품의 어느 버전에 속한 장인지도 함께 드러납니다.

답변 문장	인용된 문서	원문에서 확인할 것
이 컴포넌트의 권장 커넥션 풀 최대 값은 이 범위입니다	제품 메뉴얼 운영 매뉴얼 해당 장	문서가 가리키는 제품 버전이 우리 버전과 같은지
같은 증상이 과거에 있었고 원인은 세션 복제 지연이었습니다	해당 시스템의 장애보고서	당시 구성과 지금 구성이 같은지, 조치가 이후에 바뀌었는지
어제 점검에서 이 항목에 이상 징후가 기록되었습니다	해당 일자의 장애예방보고서	기록된 항목이 지금 문제와 같은 구간인지
이 작업은 이런 순서로 수행합니다	작업가이드와 직전 작업보고서	가이드 판본이 최신인지, 직전 작업에서 예외가 있었는지
설치 시 이 값을 이렇게 잡았습니다	해당 시스템의 설치보고서	이후 작업보고서에서 값이 변경되었는지

표의 오른쪽 열이 인용 추적의 실제 쓸모입니다. 답변을 읽는 사람은 문장을 믿을지 말지 결정하는 대신, 무엇을 확인하면 되는지를 안내받습니다. 확인 대상이 한 건의 문서로 좁혀지므로 확인에 드는 시간이 짧아지고, 확인 결과가 답변과 어긋나면 그 자리에서 문서를 고칠 근거도 생깁니다.

인용이 붙는다는 사실은 저장소를 쓰는 방식도 바꿉니다. 답변을 근거로 조치한 뒤 그 조치가 새 작업보고서로 다시 적재되면, 다음 질의는 그 보고서까지 함께 인용합니다. 7장에서 질의 자체가 기록이 된다고 적었던 흐름이 여기서 닫힙니다.

11.1.2 인용이 있어도 남는 오독 가능성

여기서 경계를 분명히 그어야 합니다. 인용 추적은 출처를 보장하지 해석을 보장하지 않습니다. 인용된 문서가 실제로 그 답변의 근거가 맞다는 사실과, 그 문서를 그렇게 요약한 것이 옳다는 사실은 서로 다른 주장입니다. 앞의 것은 장치가 담보하고, 뒤의 것은 사람이 담보합니다.

요약 과정에서 뜻이 어긋나는 방식은 대체로 두 갈래입니다. 하나는 뜻이 좁아지는 경우입니다. 원문에 "야간 배치가 도는 시간대에는 이 값을 낮춘다"고 적혀 있는데 답변이 조건절을 떨어뜨리고 "이 값을 낮춘다"로만 남기면, 읽는 사람은 상시 적용으로 받아들입니다. 다른 하나는 뜻이 넓어지는 경우입니다. 특정 버전에만 해당하는 조치가 제품 전체의 권장 사항처럼 서술되면, 다른 버전을 쓰는 시스템에 그대로 적용됩니다.

문서 쪽에서 비롯되는 어긋남도 있습니다. 인용된 문서가 사실이어도 지금 시점의 사실이 아닐 수 있습니다. 설치보고서에 적힌 설정값이 이후 작업으로 바뀌었는데 그 변경이 작업보고서로 적재되지 않았다면, 답변은 오래된 값을 정확하게 인용합니다. 이런 어긋남은 검색이 만든 문제가 아니라 적재가 만든 문제이므로, 다음 질의 지속 부담과 이어집니다.

질의 유형	사람이 확인할 것
설정값·임계값을 묻는 질의	원문의 조건절과 적용 범위, 문서가 가리키는 제품 버전
과거 사례를 되짚는 질의	당시 구성과 현재 구성의 차이, 이후 변경 이력
조치 절차를 묻는 질의	가이드 판본의 최신 여부, 직전 작업에서의 예외 처리
여러 문서를 묶어 요약한 답변	문서 간에 서로 어긋나는 서술이 있는지
근거 문서가 한 건뿐인 답변	같은 주제를 다룬 다른 컬렉션의 문서가 있는지
변경 작업 직후의 질의	변경 내용이 문서로 적재되었는지

표의 항목은 답변을 의심하라는 뜻이 아니라 확인 지점을 미리 알려 준다는 뜻입니다. 인용이 붙어 있으므로 확인은 원문 한 건을 여는 일로 끝나며, 확인 없이 통과시킬 질의와 반드시 원문을 열 질의를 조직이 미리 갈라 둘 수 있습니다. 운영 판단의 무게가 큰 질의일수록 뒤쪽에 두는 편이 안전합니다.

11.2 도입 후 지속 부담

저장소는 세우는 순간이 아니라 유지되는 동안 값을 냅니다. 검색이 답을 주는 이유는 알고리즘이 좋아서가 아니라 그 시각까지의 기록이 들어와 있어서입니다. 이 절은 계속 드는 작업(11.2.1)과 환경에 따라 달라져 조직이 직접 재보아야 하는 항목(11.2.2)을 차례로 봅니다.

11.2.1 계속 들어가는 손

계속 드는 작업은 크게 셋으로 나뉩니다. 문서를 적재하는 일, 마크다운 형식과 구조를 유지하는 일, 컬렉션과 계층을 관리하는 일입니다. 어느 하나가 끊기면 검색은 곧바로 무너지지 않고 조용히 낡습니다. 답변은 계속 나오지만 근거가 과거에 머무는 상태가 가장 다루기 어렵습니다.

작업의 성격도 컬렉션마다 다릅니다. 장애예방보고서는 연도와 분기와 일자 트리를 따라 매일 붙으므로 적재가 일과에 붙어야 하고, 제품 메뉴얼은 제품 개정 시점에만 손이 가는 대신 한 번에 여러 장이 바뀝니다. 작업 계획서와 작업보고서는 작업 승인 절차에 맞물려 있으므로, 결재가 끝난 문서가 저장소까지 오는 경로를 만들어 두면 별도의 손이 거의 들지 않습니다.

작업 항목	언제 발생하는가	무엇을 유지하는가
일일 점검 기록 적재	매일	장애예방보고서 트리의 연속성
장애·작업 문서 적재	사건·작업이 끝날 때마다	최신 사실과 저장소 사이의 시차
마크다운 헤딩·목록 구조 유지	문서를 쓸 때마다	장 단위 검색이 성립하는 조건
점검 항목 체크 목록 유지	점검 서식이 바뀔 때	문서 간 비교 가능성
제품·컴포넌트 계층 갱신	신규 도입·버전 개정 시	조건으로 범위를 좁히는 조회의 정확도
컬렉션 권한 조정	조직 개편·담당 변경 시	열람 범위와 실제 담당 범위의 일치
리비전 이력 확인	개정이 잦은 문서에서 주기적으로	원문을 열었을 때의 판단 근거

이 부담의 크기는 적재하는 문서 종류와 생산 주기에 따라 달라집니다. 열한 종을 한꺼번에 여는 조직과 두세 종으로 시작하는 조직은 매일 드는 손이 다르고, 점검 회차가 하루 한 번인 조직과 교대마다 한 번인 조직도 다릅니다. 그러므로 도입 범위를 정할 때 검색 품질만 보지 말고 유지 가능한 적재량인지를 함께 보시기 바랍니다.

부담을 줄이는 방향은 문서를 더 쓰게 만드는 쪽이 아니라 이미 쓰는 문서가 저장소로 흘러오게 만드는 쪽입니다. 점검과 작업과 장애 대응은 어차피 기록을 남기는 절차이므로, 그 기록의 형식을 마크다운으로 맞추고 저장 위치를 저장소로 옮기면 새로 생기는 일은 크지 않습니다. 새 서식을 만들어 기존 절차 위에 얹으면 그때부터 부담이 두 배가 됩니다.

11.2.2 확인이 남은 항목

아래 항목은 제품이 정하지 않고 환경이 정합니다. 같은 구성이라도 조직마다 값이 다르게 나오므로, 도입 판단에서는 이 항목을 재는 방법과 재는 시점을 함께 잡아 두시는 편이 좋습니다. 각 항목에는 무엇이 값을 흔드는지와 어떻게 재는지를 함께 적었습니다.

항목	무엇에 따라 달라지는가	어떻게 재는가
컬렉션 규모 확대 시 검색 품질	문서 수와 문서당 길이	도입 규모에 맞춰 대표 질의 묶음을 정해 두고 규모를 늘려 가며 상위 결과를 비교합니다
순위 병합 배분	질의 유형 분포	초기 몇 주의 질의 기록에서 정확 일치형과 사례 탐색형의 비율을 세어 조정합니다
전환 작업량	기존 문서의 형식과 수량	대표 시스템 한 곳을 먼저 옮겨 보고 그 소요를 전체 수량에 곱합니다
운영 부담	적재하는 문서 종류와 생산 주기	한 달 치 문서 생산 건수를 컬렉션별로 세어 적재 경로별 소요를 나눕니다
답변을 그대로 쓸 질의의 범위	운영 판단의 무게와 조직의 승인 절차	질의 유형별로 원문 확인 의무를 둘지 조직이 정합니다

표의 마지막 행은 기술이 아니라 규칙의 문제입니다. 같은 답변이라도 개발 환경 문의에 쓰는 경우와 운영 변경 작업에 쓰는 경우는 요구되는 확인 강도가 다릅니다. 그 선을 어디에 그을지는 조직의 승인 절차와 맞물려 있으므로, 도입 초기에 정해 두면 이후 논의가 짧아집니다.

이 항목들을 미리 재 두면 도입 판단이 추측에서 계산으로 바뀝니다. 대표 시스템 한 곳을 먼저 옮겨 소요를 재고, 그 값을 앞 장의 대입 표에 채운 숫자에 곱하면 전체 규모가 나옵니다. 12장은 그렇게 얻은 숫자를 놓고 세 가지 기준으로 판정하는 방법을 정리합니다.

12장: 도입 판단에서 보아야 할 것

이 문서는 처음부터 세 가지 질문을 향해 왔습니다. 우리 조직에서 먼저 적재할 문서는 무엇인가, 그 구성이 우리 환경에서 도는가, 나온 답을 어디까지 믿을 것인가입니다. 앞의 열한 장은 각 질문에 필요한 자료를 하나씩 놓았고, 이 장은 그 자료를 판정 가능한 형태로 모읍니다. 판정의 주체는 제품이 아니라 읽는 조직입니다.

그러므로 이 장은 도입을 원하는 자리가 아니라 도입이 맞는지 가르는 자리입니다. 맞지 않는 조건을 먼저 적고, 맞는 조건에서 제품이 무엇을 제공하는지를 판정 기준과 짝지어 정리합니다. 마지막으로 환경에 따라 달라져 조직이 직접 확인해야 하는 항목을 남깁니다.

읽는 순서는 판정 기준(12.1)에서 기술적 우위 요약(12.2)입니다. 앞 절은 세 가지 질문을 표로 옮기고 어느 장으로 되돌아가면 되는지를 연결하며, 뒤 절은 그 기준에 제품 기능과 확인 항목을 각각 붙입니다.

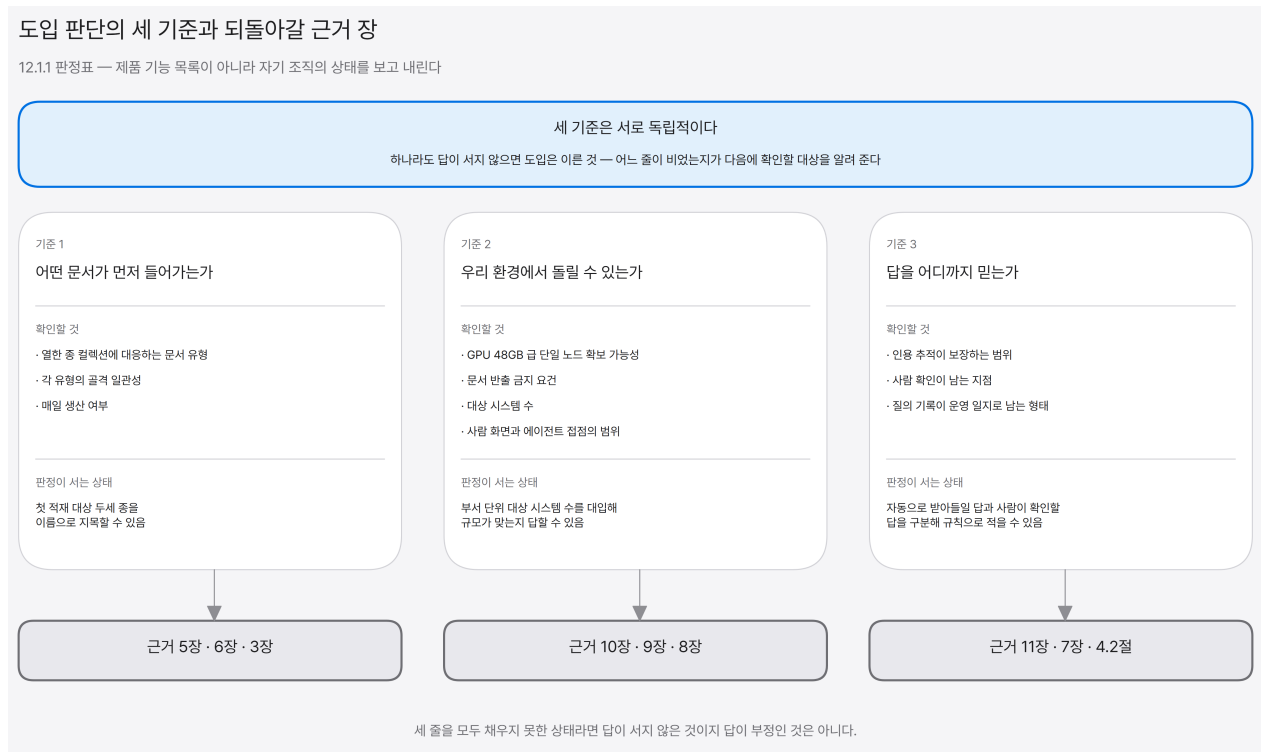


그림 20. 판정 기준 셋과 각각의 근거 장

12.1 세 가지 판정 기준

세 가지 질문은 서로 독립적이지 않습니다. 먼저 넣을 문서를 좁히면 구동 규모가 줄고, 구동 규모가 줄면 답을 검증할 범위도 좁아집니다. 그러므로 세 기준은 순서대로 통과하는 관문이 아니라 함께 조정하는 값입니다. 이 절은 판정표(12.1.1)와 판정이 갈리는 지점(12.1.2)을 차례로 봅니다.

12.1.1 판정표

판정은 각 기준마다 하나의 질문에 답하는 방식으로 진행합니다. 질문에 숫자나 명확한 문장으로 답할 수 있으면 통과이고, 답이 나오지 않으면 그 자리가 지금 확인해야 할 지점입니다. 오른쪽 열의 장 번호는 답을 만들 재료가 어디에 있는지를 가리키므로, 막히는 항목이 있으면 해당 장으로 되돌아가 보시기 바랍니다.

판정 기준	판정 질문	통과 조건	근거 장
먼저 넣을 문서	열한 종 컬렉션 가운데 우리가 이미 생산하고 있는 문서는 무엇인가	형식만 맞추면 옮길 수 있는 컬렉션을 두 종 이상 지목할 수 있다	3장·4장·5장·6장
먼저 넣을 문서	그 문서를 마크다운 구조로 옮기는 데 무엇이 걸리는가	대표 시스템 한 곳의 전환 소요를 말할 수 있다	3장
구동 가능 여부	GPU 메모리 48GB 급 단일 노드를 확보할 수 있는가	조달 또는 기존 자원 전용이 가능하다	10장
구동 가능 여부	우리 색인 대상 규모는 얼마인가	대입 표의 빈칸을 모두 채울 수 있다	10장
구동 가능 여부	문서가 조직 경계를 넘지 않아야 하는가	내부에서 달히는 구성이 요건과 맞는다	10장
답을 믿을 범위	답변에서 무엇을 되짚을 수 있어야 하는가	인용된 원문을 열어 확인하는 절차를 정할 수 있다	11장·9장
답을 믿을 범위	어떤 질의는 반드시 사람이 확인해야 하는가	확인 의무를 둘 질의 유형을 지목할 수 있다	11장·8장
답을 믿을 범위	적재를 계속 이어 갈 주체가 있는가	컬렉션별 적재 담당과 경로가 정해진다	11장·6장·7장

표를 채우다 막히는 칸이 나오면 그 칸이 곧 다음에 할 일입니다. 세 기준 가운데 두 번째는 대체로 숫자로 빨리 답이 나오고, 첫 번째와 세 번째는 조직 안에서 합의가 필요합니다. 합의가 필요한 항목을 먼저 꺼내 두면 자원 확보 논의와 병행할 수 있습니다.

한 가지만 덧붙입니다. 세 기준을 모두 최대치로 채우고 시작할 필요는 없습니다. 컬렉션 두 종과 대표 시스템 몇 곳으로 좁혀도 판정은 성립하며, 오히려 좁게 시작한 쪽이 세 번째 기준에 답하기 쉽습니다. 검증할 답변의 범위가 작을수록 확인 절차를 실제로 돌려 볼 수 있기 때문입니다.

12.1.2 판정이 갈리는 지점

맞지 않는 조건부터 적습니다. 운영 기록을 문서로 남기지 않는 조직에는 이 저장소가 값을 내지 못합니다. 점검 결과가 담당자 머릿속과 메신저 대화에만 있고 장애 대응이 끝나면 보고서가 남지 않는다면, 적재할 대상 자체가 없습니다. 이 경우 먼저 필요한 것은 검색 도구가 아니라 기록을 남기는 절차입니다.

문서를 남기더라도 형식이 저장소와 맞지 않으면 첫 고비가 커집니다. 표가 이미지로 들어간 문서, 서식이 담당자마다 다른 점검 기록, 본문 구조 없이 한 덩어리로 쓰인 보고서는 마크다운으로 옮기는 과정에서 손이 많이 갑니다. 전환 작업량은 기존 문서의 형식과 수량에 따라 달라지므로, 대표 시스템 한 곳을 먼저 옮겨 보고 판단하시기 바랍니다.

적재를 이어 갈 주체가 정해지지 않은 경우도 도입이 이릅니다. 저장소는 세운 뒤에 저절로 유지되지 않으며, 매일 붙는 점검 기록과 사건마다 붙는 장애·작업 문서가 들어와야 답변이 현재를 가리킵니다. 담당이 공석이거나 조직 개편이 예정되어 있다면, 적재 경로가 안정된 뒤에 시작하는 편이 낫습니다.

조건	판정
운영 기록을 문서로 남기지 않는다	도입이 이릅니다. 기록 절차를 먼저 세웁니다
문서는 있으나 형식이 제각각이고 표가 이미지로 들어 있다	전환 소요를 먼저 재고 범위를 좁혀 시작합니다
적재를 이어 갈 담당이 정해지지 않았다	담당과 경로를 정한 뒤 시작합니다
문서를 외부로 내보낼 수 없다	내부에서 닫히는 구성이므로 요건과 맞습니다
대상 시스템이 많고 담당자가 자주 바뀐다	기록이 사람에게 묶이지 않게 되므로 값이 큼니다
같은 유형의 장애가 반복된다	과거 기록이 검색으로 되돌아오므로 값이 큼니다
점검 기록이 매일 쌓이고 있다	적재 경로만 만들면 되므로 시작이 가볍습니다
제품 메뉴얼이 여러 제품·버전으로 나뉘어 있다	계층 조화가 범위를 좁히므로 값이 큼니다

표의 위쪽 세 줄과 아래쪽 다섯 줄은 방향이 반대입니다. 위쪽에 해당하면 지금 필요한 일은 도입이 아니라 준비이고, 아래쪽에 해당하면 좁게 시작해도 효과가 드러납니다. 두 무리에 함께 걸리는 조직도 혼잡합니다. 그런 경우에는 아래쪽에 해당하는 영역부터 컬렉션을 열고 위쪽 문제를 그 옆에서 정리하는 순서가 무난합니다.

12.2 기술적 우위 요약

이 절은 제품이 무엇을 제공하는지와 조직이 무엇을 확인해야 하는지를 나란히 놓습니다. 두 목록은 대립하지 않습니다. 앞쪽이 서 있어야 뒤쪽을 확인할 여지가 생기고, 뒤쪽을 확인해야 앞쪽이 실제 운영에서 값을 냅니다.



그림 21. 제품이 제공하는 것과 도입 전에 확인할 조건

12.2.1 OpsKnow 가 제공하는 것

기능을 나열하는 대신 판정 기준별로 묶습니다. 첫 번째 기준에는 무엇을 어떤 형태로 받아 두느냐가 걸려 있고, 두 번째 기준에는 그것이 어디에서 도느냐가, 세 번째 기준에는 나온 답을 어떻게 되짚느냐가 걸려 있습니다. 각 줄을 자기 조직 요건과 하나씩 맞춰 보시기 바랍니다.

판정 기준	제품이 제공하는 것
먼저 넣을 문서	제품 매뉴얼·정기점검보고서·장애보고서·장애예방보고서·설치계획서·설치보고서·작업가이드·작업계획서·작업보고서·CogentAI_보고서·FeedBack 열한 종 컬렉션 구성
먼저 넣을 문서	마크다운 원문 보관. 헤딩과 목록과 표 구조가 텍스트에 그대로 남습니다
먼저 넣을 문서	제품에서 컴포넌트로, 설치 가이드·운영 매뉴얼을 거쳐 장 단위까지 내려가는 계층. 검색 단위는 문서가 아니라 장입니다
먼저 넣을 문서	장애예방보고서를 연도·분기·일자 트리로 매일 적재. OS·웹서버·WAS·DB·이벤트 항목을 각각 남깁니다
먼저 넣을 문서	점검 항목을 체크 목록으로 남기고 문서마다 리비전 이력이 누적됩니다

판정 기준	제품이 제공하는 것
구동 가능 여부	GPU 메모리 48GB 급 단일 노드 온프레미스 배치
구동 가능 여부	문서와 질의와 답변이 조직 경계를 넘지 않는 구성
구동 가능 여부	관계형 조회와 벡터 검색을 함께 쓰고 결과를 하나의 순위로 병합
답을 믿을 범위	답변마다 참조 문서를 되짚어 원문을 열 수 있는 인용 추적
답을 믿을 범위	리비전 이력이 함께 열리므로 그 대목이 언제 쓰였고 이후 고쳐졌는지 확인됩니다
답을 믿을 범위	코드 에이전트가 MCP 로 제품 메뉴얼을 검색해 답하는 접점. 같은 경계 안에서 동작합니다

표를 세로로 읽으면 하나의 성질이 드러납니다. 세 기준 모두 마크다운 원문과 문서 계층 위에 서 있습니다. 무엇을 넣을지가 정해지면 검색 단위가 정해지고, 검색 단위가 정해지면 인용으로 되짚을 대상이 정해집니다. 형식 선택이 앞단의 편의가 아니라 뒷단의 검증 가능성을 결정한다는 뜻입니다.

가로로 읽으면 다른 성질이 보입니다. 열한 종 컬렉션은 종류만 늘어놓은 목록이 아니라 운영의 시간 축을 덮는 구성입니다. 설치계획서와 설치보고서가 시작을 남기고, 작업계획서와 작업보고서가 변경을 남기며, 정기 점검보고서와 장애예방보고서가 평상시를 남기고, 장애보고서가 사고를 남깁니다. 한 시스템에 무슨 일이 있었는지를 되짚을 때 끊기는 구간이 적다는 점이 검색 품질의 바탕이 됩니다.

12.2.2 도입 전에 확인할 조건

아래 항목은 제품이 정하지 않고 환경이 정합니다. 컬렉션 규모가 커질 때의 검색 품질은 문서 수와 문서당 길이에 따라 달라지므로, 도입 규모에 맞춰 확인하십시오. 대표 질의 묶음을 미리 정해 두고 규모를 늘려 가며 상위 결과를 비교하면, 어느 지점에서 근거가 뭉뚱해지는지 조직의 문서 분포 위에서 드러납니다.

순위 병합의 배분은 질의 유형 분포에 맞춰 조정할 수 있습니다. 버전과 경로를 확인하는 질의가 많은 조직과 과거 사례를 되짚는 질의가 많은 조직은 적절한 배분이 다르므로, 초기 기본값으로 운영하다가 분포가 드러나면 조정하는 순서가 무난합니다. 전환 작업량은 기존 문서의 형식과 수량에 따라 달라지고, 운영 부담은 적재하는 문서 종류와 생산 주기에 따라 달라집니다. 두 값 모두 대표 시스템 한 곳을 먼저 옮겨 보면 나머지에 곱할 계수가 나옵니다.

마지막으로 남는 조건은 기술이 아니라 규칙입니다. 인용 추적은 출처를 보장하지 해석을 보장하지 않으므로, 어떤 질의를 답변만으로 통과시키고 어떤 질의에서 원문 확인을 의무로 돌지는 조직이 정해야 합니다. 이 선은 운영 판단의 무게와 승인 절차에 맞물려 있어 제품이 대신 그어 줄 수 없습니다.

정리하면 OpsKnow 는 이미 만들어지고 있는 운영 문서를 마크다운 원문 그대로 한 저장소에 받아, 제품·컴포넌트·장 단위 계층과 일자 트리 위에 올리고, 관계형 조회와 벡터 검색을 함께 돌려 하나의 순위로 답을 세우며, 그 답을 원문까지 되짚게 하는 구성입니다. 이 모두가 GPU 메모리 48GB 급 단일 노드 안에서, 문서와 질의와 답변이 조직 경계를 넘지 않는 상태로 이뤄집니다. 남는 판단은 하나입니다. 우리 조직의 문서와 규모와 확인 규칙이 이 구성과 맞느냐입니다. 앞의 판정표가 그 답을 세우는 자리입니다.



OpsKnow AI 기반 IT 인프라 운영 지식 관리 시스템

CONTACT

WEB

msap.ai

www.msap.ai/

EMAIL

hello@msap.ai

TEL

02-6953-5427

0269535427

YOUTUBE

@msaptv

www.youtube.com/@msaptv

LINKEDIN

linkedin.com/showcas...

www.linkedin.com/showcase/msap-ai/

FACEBOOK

facebook.com/opennaru

www.facebook.com/opennaru



SCAN