— STUDY PACK · COMPLETE EDITION —

# AWS Summit Seoul 2026 — 전체 지도

## Models · Factory · Fuel · Engine · Fortress · Applications

**대상** Samsung 마케팅 PM (David)
**구성** 6 chapters + 3 appendices
**분량** 100+ pages · 65 terms · 4 speaking scripts
**날짜** 2026년 5월

AWS SUMMIT SEOUL 2026

# 목차 — Contents

총 6 chapters + 3 appendices

CH 01 · OVERVIEW

# 전체 지도 — The Atlas

이틀간 18개 트랙·100여 개 용어가 모인 행사를 한 줄 요약하면 "에이전틱 AI 시대로의 풀스택 전환". 이 한 문장이 나머지 모든 챕터의 출발점입니다.

---

## 왜 한 행사인가

트랙 이름은 9개씩 두 날 — 총 18개로 다양해 보이지만, 본질은 하나로 수렴합니다. **"AI 에이전트를 어떻게 만들고, 돌리고, 지키고, 팔지"**라는 한 질문의 다른 면들이에요. 기존 클라우드 부품(Bedrock·SageMaker·S3·EC2 등)이 모두 "에이전트 친화적"으로 재포장된 게 핵심 신호입니다.

## 8개 범주로 보는 그림

AREA 01 · MODELS & AGENTS

### 🧠 AI 두뇌 — 생각하고 도구 쓰는 비서

LLM 본체(파운데이션 모델)와, 그 모델을 도구로 부려서 일을 시키는 에이전트 프레임워크. `Bedrock · Nova · AgentCore · Strands SDK · MCP · A2A`

AREA 02 · DEV & OPERATIONS

### 🏭 AI 공장 — 코드·운영을 매일 갱신하는 컨베이어

코드·인프라·모델을 자동으로 찍어내는 생산라인. AI가 코딩까지 같이 하는 새 개발 사이클.
`Kiro · AI-DLC · IaC · CI/CD · MLOps · AIOps · DevSecOps`

AREA 03 · DATA PLATFORM

### ⛽ AI 연료 — 에이전트가 먹고 일하는 회사 데이터

회사 데이터를 모델이 곧바로 쓸 수 있게 가공·검색·연결. `S3 · OpenSearch · RAG · GraphRAG · CDC · Lakehouse`

AREA 04 · COMPUTE INFRA

### ⚙️ AI 엔진 — 모델이 실제로 도는 실리콘

GPU·전용 칩·클러스터. "초당 토큰" 비용이 여기서 결정됨. `Trainium · Inferentia · EC2 P/G · HyperPod · Nitro · Slurm on EKS`

AREA 05 · SECURITY & GOVERNANCE

### 🛡️ AI 요새 — 에이전트가 회사 자산을 만지는 걸 지킴

에이전트가 회사 데이터를 만지기 시작하니, 보안이 "AI 워크로드 전용"으로 다시 짜여야 함.

`Zero Trust · KMS · CloudHSM · DevSecOps · Guardrails`

AREA 06 · PHYSICAL & INDUSTRY

### 🤖 AI 응용 — 피지컬

모니터 밖으로 나간 AI. 로봇·공장·자동차·헬스케어. 시뮬레이션으로 학습시켜 현실에 투입.

`Physical AI · Sim-to-Real · Robotics FM · AI Factory · AI for MI`

AREA 07 · BUSINESS APPS

### 💼 AI 응용 — 비즈니스

고객접점·내부생산성에 바로 꽂히는 완성형 에이전트. 마케팅 PM이 가장 자주 부딪힐 층.

`Amazon Connect · Quick Suite · AI 컨시어지 · Audience Engine`

AREA 08 · PARTNER ECOSYSTEM

### 🤝 파트너 생태계

AWS 혼자가 아니라 Anthropic·Cloudflare·Datadog·Snowflake 등이 같이 만드는 스택.

`Anthropic · Cloudflare · Datadog · Snowflake · Red Hat ROSA · Nutanix`

## 한 요청이 도는 길 — 6영역 합주

8개 범주 중 "이 모든 걸 가능하게 하는" 6개는 사용자 한 마디 요청을 위해 동시에 합주합니다:

① **비즈니스 앱** — 사용자 요청 수신 (채팅·SmartThings·콜센터)

↓

② **두뇌** — 에이전트가 모델·도구 선택

↓

③ **연료** — RAG로 회사 데이터 끌어옴

↓

④ **엔진** — GPU/Trainium에서 추론 실행

↓

⑤ **요새** — 권한·로그·키 검증 통과

↓

⑥ **공장** — 전체가 CI/CD로 매일 갱신

## David의 SmartThings Scenario Agent 위치

> v2 아키텍처(5-Agent 파이프라인 + BYOK + prompt.txt SSOT)는 정확히 **①비즈니스 앱(SmartThings) ←→ ②두뇌(Bedrock/Anthropic/Gemini + Multi-Agent) ←→ ③연료(27-시나리오 JSON DB = mini-RAG)** 층에 걸쳐 있습니다. 아직 안 닿은 곳: ④엔진(Cloudflare Workers에 얹혀있어 우회), ⑤요새(BYOK라 위임), ⑥공장(수동 배포). 이게 v3 로드맵의 후보입니다.

CH 02 · BRAIN

# AI 두뇌 — Models & Agents

판단하고 행동하는 부분. 모델 본체부터, 그 모델을 "비서"로 만들어 도구를 쓰게 하는 에이전트 프레임워크·프로토콜·아키텍처까지 한 묶음.

---

## 왜 "두뇌"인가

LLM은 갓 졸업한 천재 신입에 가깝습니다. 일반 상식은 압도적이지만, 우리 회사 메일을 보낼 줄도, DB를 뒤질 줄도 모릅니다. **"에이전트화"**란 이 신입에게 (1) 도구함, (2) 회사 매뉴얼, (3) 다른 동료와 협업하는 규약을 쥐어주는 작업입니다. 이번 Summit에서 두뇌 카테고리의 95%는 결국 이 세 가지를 어떻게 표준화·관리형 서비스로 제공하느냐의 이야기입니다.

## A. 파운데이션 모델 — 뇌 자체

FOUNDATION MODEL (FM)

### 파운데이션 모델 — 천재 신입사원

정의 대용량 데이터로 사전학습된 범용 거대모델. GPT-4·Claude·Gemini·Nova가 전부 FM.

발화 "이 기능을 FM 위에 얹을지, 직접 fine-tune할지부터 정합시다."

DAVID v2의 BYOK가 OpenAI·Anthropic·Gemini를 갈아끼울 수 있는 이유 — 셋 다 같은 "FM API 호출" 추상 위에 있어서.

AMAZON BEDROCK

### 베드락 — 모델 뷔페

정의 여러 회사의 FM(Claude·Llama·Nova 등)을 단일 API로 호출하는 AWS 서버리스 게이트웨이. 가드레일·RAG·에이전트 통합.

발화 "Bedrock으로 Claude 쓰면 회사 IAM 정책으로 거버넌스가 잡힙니다."

DAVID 너의 BYOK에서 `provider="anthropic"` 분기 자리 — Samsung 사내 정식화 시 Bedrock 호출이 들어감.

AMAZON NOVA · NOVA 2

### 노바 — AWS가 직접 키운 자체 모델

정의 AWS가 Trainium 칩으로 직접 만든 FM 패밀리. Nova 2가 이번 Summit 주력 발표.

발화 "Nova 2로 비용·지연 잡고 외부 모델 의존도 줄였다"가 사례 발표 단골 메시지.

DAVID v3에서 한국어 품질·비용 모두 잡고 싶다면 Bedrock-Nova 2를 BYOK 후보에 추가, A/B 테스트.

ANTHROPIC CLAUDE

### 앤트로픽 클로드 — Bedrock의 단골 메뉴

정의 Anthropic의 FM. 추론·코딩·긴 문서 처리에 강하고 안전성 설계 깐깐. Bedrock 단골. MCP 원작자.

DAVID v2 Copy Consult(Mode2) 품질 튜닝(P7-B) — Claude Sonnet 계열이 한국어 카피라이팅 결에서 유리할 수 있음.

NOVA FORGE · BEDROCK RFT

### 노바 포지 / RFT — 신입을 회사 일에 재교육

정의 범용 FM을 도메인 데이터에 맞게 재학습. RFT(Reinforcement Fine-Tuning)는 "좋은 답/나쁜 답" 보상신호로 행동 교정.

DAVID 27-시나리오 JSON DB가 더 쌓이면 RFT 보상 데이터로 활용 — Samsung 시나리오 스타일 특화 모델 가능. v4 고민거리.

## B. 관리형 ML 플랫폼

MANAGED SERVICE

### 관리형 서비스 — "서버 관리는 우리가, 너는 일만"

정의 서버 설치·패치·확장·백업을 클라우드가 다 해주고, 사용자는 "쓰기"에만 집중. Self-managed의 반대.

DAVID v2가 Cloudflare Workers 위에 얹힌 것도 같은 발상 — "서버 직접 운영 ✗, 함수 단위 호출"이 Workers의 관리형 모델.

AMAZON SAGEMAKER

### 세이지메이커 — ML 작업장 전체

정의 데이터→학습→배포→모니터링까지 ML 전 과정 통합 플랫폼. Notebook·Training Job·Endpoint·Pipeline 한 묶음.

구분 Bedrock = "이미 만든 FM API 사용" / SageMaker = "내가 직접 만들고 굴림".

DAVID 지금 단계엔 불필요. "한국어 마케팅 카피 평가 모델 직접 만들고 싶을 때"부터 등장.

SAGEMAKER UNIFIED STUDIO · CATALOG

### 유니파이드 스튜디오 / 카탈로그 — IDE + 자산 도서관

정의 Unified Studio = 데이터·노트북·파이프라인 통합 IDE. Catalog = 사내 모델·데이터셋 검색·관리(메타데이터·계보).

DAVID 너의 prompt.txt SSOT 컨셉을 회사 단위로 확장한 게 Catalog.

## C. 에이전트 프레임워크

AGENT · AGENTIC AI

### 에이전트 — LLM + 도구사용 + 자율판단

정의 단순 챗봇이 아니라 "목표 → 계획 → 도구 → 결과 검증 → 다음 행동" 루프를 도는 LLM 시스템. `Chatbot ≠ Agent`.

DAVID 너의 v2는 이미 에이전트. 다만 "도구사용"이 약함 — Activator가 실제로 SmartThings API를 호출해서 시나리오를 등록하면 진짜 Agentic으로 도약.

AMAZON BEDROCK AGENTCORE

### 에이전트코어 — 에이전트의 관리형 사무실

정의 에이전트를 실행시키는 AWS 관리형 런타임. 메모리·도구연결·로그·세션 자동 관리. Runtime + Memory + Identity + Gateway 묶음.

DAVID 지금 너가 Workers로 5-Agent를 직접 코딩한 부분 — AWS 정식 도입 시 그 자리에 AgentCore가 들어옴. "세션 상태·메모리·로그"가 무료 부수효과.

STRANDS AGENT SDK

### 스트랜즈 SDK — 에이전트 양성 키트

정의 AWS 오픈소스 Python SDK. `pip install` 로 시작, AgentCore에 올리면 프로덕션. Bedrock·Anthropic·OpenAI 다 지원 = BYOK 친화적.

DAVID 파이썬 학습 중인 너에게 적합한 진입점. v3 학습 단계로 "5-Agent를 Strands로 재작성" 추천. LangGraph보다 학습곡선 완만.

## D. 에이전트 통신 프로토콜

MCP — MODEL CONTEXT PROTOCOL

### MCP — AI의 USB-C

정의 모델 ↔ 외부 도구/데이터 연결의 표준 규격. Anthropic 주도, OpenAI·Google·AWS 채택. JSON-RPC 기반.

핵심 효익 모델별로 도구 연동 코드를 따로 짤 필요 없음. "LLM의 USB-C"로 자리잡는 중.

DAVID SmartThings API·시나리오 DB·prompt.txt를 각각 MCP 서버로 노출하면 어떤 모델이든 동일 도구 호출 가능. **v3 최대 임팩트 리팩토링 후보.**

A2A — AGENT-TO-AGENT

### A2A — 에이전트끼리의 사내 메신저 규약

정의 서로 다른 회사·프레임워크 에이전트가 협업할 때 쓰는 통신 표준. 구글 주도, AWS·Anthropic 합류.

`MCP = 모델↔도구 / A2A = 에이전트↔에이전트`

DAVID 너의 5-Agent는 지금 함수 호출로 연결 — 모놀리식 파이프라인. Curator가 별도 서비스가 되고, Localizer가 다른 팀 에이전트가 되는 순간 A2A가 의미 있어짐.

# E. 에이전트 아키텍처 — 4 패턴

MULTI-AGENT PATTERNS

## 멀티 에이전트 — 1인 만능 ✗, 전문가 팀 ◯

SINGLE 모든 일을 한 에이전트가. 간단하지만 한계.

PIPELINE 컨베이어 — A→B→C 순차. **David v2의 패턴.**

ORCHESTRATOR 팀장 에이전트가 Sub-agent에게 분담. Claude Code의 sub-agent 구조 대표.

AUTONOMOUS 에이전트들이 자율 협상·합의. 가장 어렵고 가장 강력.

DAVID 진화 (1) Curator를 Orchestrator로 승격 → 시나리오별 동적 분기 (2) Story Chat을 Sub-agent로 정식 등록.

# 5층 스택 구조

두뇌 안에서 각 용어가 어디 층에 있는지:

**5층 · 아키텍처** — Multi-Agent / Pipeline / Orchestrator (몇 명을 어떻게 묶을지)

**4층 · 프로토콜** — MCP / A2A (에이전트가 외부·서로 대화하는 표준)

**3층 · 프레임워크** — AgentCore / Strands SDK (모델을 비서로 만드는 키트와 런타임)

**2층 · 모델 API** — Bedrock / OpenAI / Anthropic (FM 호출 추상화)

**1층 · 모델 본체** — Nova / Claude / Llama / Gemini (Foundation Model)

CH 03 · FACTORY

# AI 공장 — Dev & Operations

모델·에이전트를 매일 찍어내는 생산라인. AI가 코드를 쓰고 AI가 운영을 감시하기 시작한 순간, "개발"의 정의가 바뀌었습니다.

---

## 패러다임 전환 — before / after

**이전:** 사람이 코드 작성 → 사람이 리뷰 → 사람이 배포 → 사람이 모니터링

**지금:** Spec 작성 → **에이전트**가 코드 생성 → 사람이 검수 → **파이프라인** 자동배포 → **AIOps**가 자가복구

## A. 새로운 개발 사고방식

AI NATIVE DEVELOPMENT

### AI Native 개발 — AI를 손님이 아니라 동료로

정의 AI가 코드를 같이 짠다를 "기본 전제"로 깔고 워크플로우를 처음부터 다시 짜는 방식.

구분 AI-Assisted(기존 워크플로우에 AI 추가) ≠ AI Native(처음부터 AI 중심 재설계).

발화 "우리는 AI Native로 갑니다. AI가 1차 코드 쓰고 사람이 검수·결정하는 구조를 디폴트로."

DAVID 너는 이미 AI Native 사용자 — Antigravity · Claude CLI · Codex CLI + prompt.txt SSOT.

SPEC-DRIVEN DEVELOPMENT

### Spec-driven — 요구사항을 코드처럼 관리

정의 "코드부터 짜자" 대신 "Spec을 정확히 적자. 코드는 거기서 자동 생성"하는 접근. AWS Kiro가 대표 구현체.

DAVID 너의 prompt.txt SSOT + TODO.md + 마커 추출이 mini-Spec-driven. "Spec-driven 변형 적용 사례"로 명명 가능.

AI-DLC — AI-DRIVEN DEVELOPMENT LIFECYCLE

### AI-DLC — 새 SDLC

정의 기존 SDLC를 AI 시대에 맞춰 다시 그린 것. Spec → AI 코드생성 → AI 테스트 → CI/CD → AIOps 운영의 전체 루프.

발화 "기존 SDLC가 인간 개발자 중심이었다면, AI-DLC는 에이전트가 1차 작성하고 사람이 검토하는 새 사이클."

DAVID v1 실패 → v2 재설계 → TODO.md 회고 전체가 작은 AI-DLC 1회차.

## B. AI 코딩 어시스턴트 — 4개 도구의 분담

AWS KIRO

### Kiro — Spec 받으면 코드 짜는 AWS 에이전트

정의 Spec 파일 주면 코드·테스트·PR까지 만드는 AWS IDE/에이전트. Spec mode + Vibe mode.

발화 "Kiro는 spec을 입력으로 받아 코드·테스트·PR까지 한 번에 만들어주는 AWS 코딩 에이전트입니다."

CLAUDE CODE

### Claude Code — 터미널에 사는 코딩 에이전트

정의 Anthropic의 CLI 코딩 에이전트. IDE 없이 터미널에서 파일 직접 읽고 수정. Sub-agent 시스템으로 작업 위임·병렬화.

DAVID v2 리팩토링·P7-B 같은 작업에 sub-agent로 분리하면 효율 ↑. 단 회사 데이터 외부 노출 정책 확인 필수.

AWS DEVOPS AGENT

### DevOps Agent — 빌드·배포·장애대응 담당

정의 코드는 안 짜고, 운영 작업(배포·롤백·장애조사)을 자동화하는 AWS 도메인 특화 에이전트.

DAVID 너가 PAT로 직접 git push하고 Cloudflare 콘솔에서 수동 배포하는 부분이 이 자리.

AWS TRANSFORM

### AWS Transform — 레거시 코드 자동 현대화

정의 오래된 .NET·메인프레임·Java 6 같은 코드를 AI가 분석해 최신 클라우드 네이티브로 자동 변환·재작성.

구분 Kiro=신규(0→1) / Claude Code=수정(기존 코드) / Transform=현대화(1→1.5) / DevOps Agent=운영.

## C. 자동화 파이프라인

INFRASTRUCTURE AS CODE (IaC)

### IaC — 인프라를 코드로 적기

정의 서버·DB·네트워크를 콘솔 클릭이 아니라 코드 파일로 정의.
Terraform · CloudFormation · CDK · Pulumi 대표.

DAVID 너의 Cloudflare wrangler.toml이 mini-IaC. "내 v2도 IaC 사용 중"으로 설명 가능.

CI/CD — CONTINUOUS INTEGRATION/DELIVERY

### CI/CD — 코드 푸시 → 자동 빌드·테스트·배포

정의 CI = 코드 올리면 자동 빌드·테스트. CD = 통과 시 자동 배포. GitHub Actions · CodePipeline 대표.

구분 IaC = "인프라"를 코드로 / CI/CD = "코드 변경"을 자동 배포. 보통 함께 씀.

DAVID 너의 수동 git push → wrangler-publish GitHub Action으로 자동화 가능. **v3 후보.**

API MANAGEMENT

### API 관리 — 외부·내부 호출 통제 게이트

정의 API 호출의 인증·요금·트래픽을 통제하는 게이트웨이. AI 시대엔 "에이전트 외부 도구 호출의 단일 통제점"으로 의미 확장.

DAVID 5-Agent가 BYOK로 외부 LLM 호출하는 경로 — Samsung 정식화 시 사내 API Gateway 한 번 거치는 구조로. 감사 로그 확보.

# D. AI 시대의 운영 — Ops 3형제

MLOPS · AIOPS · DEVSECOPS

## 3형제 — 대상이 결정적으로 다름

MLOPS 대상 = **모델**. 모델의 데이터→학습→배포→재학습 라이프사이클 자동화.

AIOPS 대상 = **시스템 운영**. 로그·알람·메트릭을 AI가 분석, 장애 진단·복구.

DEVSECOPS 대상 = **보안**. 보안 검사를 코딩·빌드·배포 전 단계에 자동 삽입. "Shift Left Security".

암기키 `MLOps=모델 / AIOps=시스템 / DevSecOps=보안`

DAVID 너의 PAT 즉시 폐기 = 개인 단위 DevSecOps 시크릿 관리. 사내 정식화 시 git push 시 토큰 노출 자동 차단으로 진화.

# 임원 설명 30초 템플릿 — 여러 용어를 한 호흡에

**▸ 상황 1 — 임원: "AWS Summit 뭐가 핵심이었나"**

"한 줄로 정리하면, **AI Native 사고방식**이 사이클 단위로 굳어진 게 핵심입니다. **Spec-driven**으로 요구사항을 적고, **Kiro·Claude Code** 같은 에이전트가 코드를 1차로 짜고, 그 결과가 **CI/CD** 위에서 자동 배포되고, 운영은 **AIOps·DevSecOps**가 받습니다. 이걸 묶은 사이클을 **AI-DLC**라고 부르더라고요."

**▸ 상황 2 — "우리도 AI 코딩 도구 도입하자"**

"도구가 한 종류로 묶이는 게 아니라 역할이 셋입니다. **Kiro**는 spec 받아 새 코드를 짜는 쪽, **Claude Code**는 터미널에서 기존 코드 수정·디버깅 쪽, **AWS Transform**은 레거시 .NET·메인프레임 현대화 쪽이에요. 신규 개발 많은 팀엔 Kiro, 운영 부담 큰 팀엔 Claude Code, 사내 레거시 정리엔 AWS Transform — 3축으로 검토."

**▸ 상황 3 — "MLOps랑 AIOps 뭐가 다른데"**

"셋 다 'Ops'지만 대상이 달라요. **MLOps**는 **모델 자체**의 운영—학습·배포·재학습. **AIOps**는 **시스템 운영**에 AI를 쓰는 거예요. 로그·알람을 AI가 분석해서 장애 원인 자동 진단. **DevSecOps**는 **보안**을 개발 사이클 전 단계에 자동으로 끼워넣는 접근이고요."

CH 04 · FUEL & ENGINE

# AI 연료 & 엔진 — Data & Infrastructure

두뇌가 먹는 연료(데이터)와 두뇌가 굴러갈 엔진(컴퓨트). 둘 중 하나라도 빠지면 에이전트는 한 발짝도 못 움직입니다.

▶ SPEAKING SCRIPT — DAVID'S READ 약 2분 30초

OPENING · 15초

이번 AWS Summit에서 데이터 트랙과 인프라 트랙을 들으면서 깨달은 게 하나 있습니다… AI 에이전트가 제대로 일하려면 두 가지가 필요하다는 거예요… 먹을 연료, 그리고 굴러갈 엔진.

PART 1 · 연료 = 데이터 · 70초

연료 쪽은 AI-Ready Data라는 개념으로 시작합니다. 한 줄로 말하면, "우리 회사 데이터가 LLM이 곧바로 쓸 수 있는 상태인가" 라는 질문입니다.

보통 데이터는 Amazon S3 같은 객체 스토리지에 깔리고요, 그걸 검색 가능하게 만드는 게 Amazon OpenSearch입니다. 단순 키워드 검색으론 부족하니까, 텍스트를 벡터로 임베딩해서 의미 기반 검색을 합니다… 이게 바로 RAG—Retrieval-Augmented Generation—의 핵심이에요. 모델이 답하기 전에 회사 데이터에서 관련 문서를 먼저 끌어오는 거죠.

올해의 진화는 GraphRAG입니다. 그냥 문서를 찾는 게 아니라, 문서 사이의 관계까지 그래프로 두고 그 위에서 검색합니다. 미래에셋증권 상품 지식 DB가 대표 사례였습니다.

데이터 파이프라인 쪽에선 CDC와 Lakehouse가 두 키워드예요. CDC는 운영 DB 변경분만 실시간 스트리밍, Lakehouse는 그걸 받아 분석·AI 학습에 쓰는 통합 저장소. 둘이 짝을 이뤄 "실시간 AI-Ready 데이터"를 만듭니다.

PART 2 · 엔진 = 인프라 · 70초

이제 엔진 쪽입니다. AI 모델을 돌리는 실리콘 얘기예요.

크게 두 종류—훈련용 칩과 추론용 칩. AWS는 자체 칩으로 Trainium과 Inferentia를 갖고 있어요. Trainium은 학습 전용, Inferentia는 추론 전용. NVIDIA GPU 의존도를 낮추고 비용 잡겠다는 전략이죠.

대안으로 NVIDIA GPU를 쓰고 싶으면 EC2 P/G 인스턴스가 있고, 그 모든 컴퓨트를 안전하게 격리하는 토대가 AWS Nitro System입니다.

대규모 학습은 한 대로 안 되니 클러스터가 필요합니다. HyperPod이 그 답이에요—수백 대 GPU를 한 묶음으로 묶고, 장애 자동 복구하는 관리형 클러스터. 그 위에 잡 스케줄러로 Slurm이 EKS 위에서 도는 구성이 올해 부각됐습니다. 마지막으로 EC2 Capacity Blocks는 GPU 미리 예약권 — GPU 품귀 시대의 안전장치입니다.

CLOSING · 25초

정리하면… 데이터는 S3에 깔고, OpenSearch와 RAG·GraphRAG로 검색 가능하게 만들고, CDC와 Lakehouse로 실시간 갱신. 그게 AI 연료. 그걸 Trainium·Inferentia·GPU 위에서 돌리고, HyperPod 클러스터에 묶어 굴리는 게 AI 엔진. 이 둘이 갖춰져야 비로소 두뇌가 일을 할 수 있다… 이게 핵심이었습니다.

# A. AI 연료 — 데이터

AI-READY DATA · S3 · OPENSEARCH

## 데이터 기반 — 저장 + 검색 가능 상태

정의 AI-Ready Data = "LLM이 곧바로 쓸 수 있는 상태"의 회사 데이터. S3(저장) + OpenSearch(검색)가 그 상태를 만드는 기본 부품.

발화 "데이터는 S3에 깔리고, OpenSearch가 벡터 검색까지 해주는 검색층 역할입니다."

RAG — RETRIEVAL-AUGMENTED GENERATION

## RAG — 답하기 전에 회사 자료부터 뒤짐

정의 LLM이 회사 DB에서 관련 문서를 먼저 검색한 다음 그 컨텍스트로 답하는 방식. 환각(hallucination) 감소의 표준 처방.

RAG VS 파인튜닝 RAG는 모델 그대로, 데이터만 갱신. Fine-tuning은 모델 자체를 재학습. **먼저 RAG, 한계 도달 시 Fine-tune**이 표준 순서.

DAVID 너의 27-시나리오 JSON DB + facet retrieval = mini-RAG. 정통 RAG로 발전 = S3 + OpenSearch 벡터 인덱스로 이동.

GRAPHRAG

## GraphRAG — 문서 관계까지 따라가는 검색

정의 RAG가 "비슷한 문서 찾기"라면, GraphRAG는 문서·개념 사이 관계를 그래프로 두고 검색. 다단계 질문에 강함.

현장 미래에셋증권 GraphRAG 기반 상품지식DB — 금융처럼 규정·관계 복잡한 도메인에 적합.

CDC · LAKEHOUSE

### CDC + Lakehouse — 실시간 AI-Ready 만들기

정의 CDC = Change Data Capture, 운영 DB의 binlog/WAL 스트리밍. Lakehouse = Data Lake + Warehouse 통합 아키텍처 (Iceberg·Delta Lake). 배치 ETL 대체.

발화 "CDC가 운영 DB 변경을 실시간 흘려보내고, Lakehouse가 분석·AI 학습에 쓰는 구조입니다."

## B. AI 엔진 — 인프라

AWS TRAINIUM · INFERENTIA

### Trainium / Inferentia — AWS 자체 AI 칩 듀오

정의 Trainium = 학습 전용 칩, Inferentia = 추론 전용 칩. 둘 다 AWS 직접 제작. NVIDIA 의존도 ↓ 비용 ↓.

발화 "Trainium은 학습 전용, Inferentia는 추론 전용. AWS가 NVIDIA 의존도를 낮추려고 직접 만든 자체 칩 듀오입니다."

EC2 P/G · NITRO SYSTEM

### EC2 P/G + Nitro — NVIDIA GPU 옵션 + 신뢰 기반

정의 EC2 P/G = NVIDIA GPU 얹은 EC2 시리즈 (P=학습 고성능, G=추론·그래픽). Nitro = 모든 EC2를 안전하게 격리·가속하는 하드웨어 시스템.

HYPERPOD · SLURM ON EKS · CAPACITY BLOCKS

### 대규모 학습 클러스터 — 수백 대 GPU를 한 묶음으로

정의 HyperPod = 노드 장애 자동복구 + 체크포인트 자동저장 관리형 학습 클러스터. Slurm on EKS = HPC 표준 스케줄러를 Kubernetes 위에서. Capacity Blocks = GPU 미리 예약권.

현장 하이퍼커넥트의 HyperPod 기반 Slurm on EKS 도입기 — 영상 AI 학습 안정성 향상 사례.

## 한 번의 RAG 호출이 지나는 6층

사용자 질문 → 답 한 줄 사이에서 데이터·인프라가 합주하는 모습:

① **데이터 · STORAGE** — S3에 회사 문서·로그 적재. CDC가 운영 DB 변경분 흘려보내고 Lakehouse가 받음.

② **검색 · RETRIEVAL** — OpenSearch가 벡터로 인덱싱. 질문이 임베딩되어 RAG/GraphRAG로 관련 문서 추출.

③ **컨텍스트 · CONTEXT** — 검색된 문서를 LLM 프롬프트에 함께 끼움. 환각 ↓, 정확성 ↑.

④ **추론 · INFERENCE** — Inferentia 또는 NVIDIA GPU(EC2 P/G)에서 답 생성. Nitro로 격리.

⑤ **학습 · TRAINING** — 주기적으로 새 데이터로 재학습 시 Trainium + HyperPod + Slurm on EKS로 분산 학습.

⑥ **답 · OUTPUT** — 사용자에게 한 줄 답변. 뒤에서는 6층이 합주한 결과.

CH 05 · FORTRESS & APPLICATIONS

# AI 요새 & 응용 — Security · Physical · Business

두뇌·공장·연료·엔진을 다 갖췄으면, 남은 건 두 가지. 이 모든 걸 지키는 보안, 그리고 이 모든 게 가치를 만드는 응용.

▶ SPEAKING SCRIPT — DAVID'S READ 약 3분

OPENING · 15초

이번 Summit의 후반전이 이 두 가지였습니다… AI 인프라를 다 갖췄을 때 추가로 필요한 것… 안전하게 지키는 요새, 그리고 실제로 가치를 만드는 응용.

PART 1 · 요새 = 보안 · 55초

AI 에이전트가 회사 데이터를 진짜로 만지기 시작하면서, 보안이 다시 짜이고 있습니다. 핵심 키워드 세 가지예요… Zero Trust, KMS / CloudHSM, 그리고 AI 워크로드 심층 방어.

Zero Trust는 한 줄로 "아무도 기본 신뢰하지 않는다"는 원칙. 예전엔 사내 네트워크 안이면 일단 믿었는데, 이젠 에이전트가 호출하는 모든 요청을 매번 검증합니다. Zscaler, Palo Alto Networks 같은 파트너가 다 이 메시지로 들어왔습니다.

KMS와 CloudHSM은 암호화 키 관리 도구. KMS는 관리형, CloudHSM은 전용 하드웨어. 금융권·정부처럼 키 격리 등급이 빡빡한 곳이 CloudHSM. 현대카드의 대규모 서명키 관리 사례가 대표적.

마지막으로 AI 워크로드 심층 방어… 이게 새로운 영역. 프롬프트 인젝션·에이전트 행동 가드레일·민감 데이터 유출 차단 같은 AI 전용 위협 대응 층이에요. 기존 보안 위에 한 겹이 더 깔린 셈입니다.

PART 2 · 응용 = 피지컬 AI · 50초

이번 Summit에서 가장 시각적으로 강했던 게 피지컬 AI입니다. AI가 모니터 안에서 나와 로봇·자동차·공장으로 들어가는 흐름이에요.

핵심은 Robotics Foundation Model. 텍스트·이미지 FM처럼, 이제 로봇 동작 전용 거대 모델이 만들어지는 시기예요. POSCO와 Config가 AWS 위에서 자기 Robotics FM 개발 발표.

그리고 Sim-to-Real—시뮬레이션에서 학습한 걸 실제 로봇으로 옮기는 기술. 위로보틱스 RLWRLD 사례가 인상적. 현대자동차 AI for MI, POSCO 자율 예지정비 — 피지컬 AI가 이미 공장 라인에서 돌고 있다는 신호.

PART 3 · 응용 = 비즈니스 · 45초

마케팅 PM 입장에서 가장 가까운 층. AWS 완성형 에이전트 두 개… Amazon Connect는 콜센터·고객응대용 AI, Quick Suite는 사내 데이터·생산성 통합 도구.

그 위에 한국 기업들이 자기 분야 응용을 가져왔습니다… GS SHOP의 영상 추천 플랫폼, KB국민은행의 임베디드 금융, 하나투어·AK아이에스의 AI 컨시어지. 전부 같은 패턴이에요. "고객접점에 에이전트를 꽂아 전환율을 올린다."

우리의 SmartThings 시나리오 에이전트도 정확히 이 카테고리. "사용자가 시나리오를 발견하는 접점에 에이전트를 꽂아 활성화율을 올린다"로 한 줄 정의 가능.

CLOSING · 20초

정리하면… 요새는 Zero Trust · KMS · AI 심층 방어, 응용은 피지컬 AI와 비즈니스 에이전트. 이걸로 두뇌·공장·연료·엔진까지 합쳐 6개 영역이 모두 닿았습니다.

## A. AI 요새 — 3개 기둥

ZERO TRUST · SECURE AI BY DESIGN

### Zero Trust — 아무도 기본 신뢰하지 않는다

정의 "Never trust, always verify". 매 요청마다 ID·기기·문맥 동적 검증. 사내/사외 구분 없음. AI 시대엔 에이전트의 API 호출 전부 재확인.

현장 Bedrock AgentCore로 AI 시대의 Zero Trust 구현 · Palo Alto Networks "Secure AI by Design" — AI 전용 Zero Trust가 별도 카테고리화.

AWS KMS · CLOUDHSM

### KMS / CloudHSM — 암호화 키의 금고

정의 KMS = 관리형 키 서비스 (FIPS 140-2 Level 3). CloudHSM = 전용 하드웨어 보안 모듈, 키가 물리적으로 격리. 금융·정부처럼 키 격리 빡빡한 곳용.

현장 현대카드 CloudHSM 기반 대규모 서명키 관리 시스템 사례.

AI WORKLOAD DEFENSE · PROMPT INJECTION · GUARDRAILS

### AI 워크로드 심층 방어 — AI 전용 위협이 따로 있음

정의 프롬프트 인젝션·모델 탈옥·민감정보 유출·에이전트 오·남용 대응 별도 보안층. Bedrock Guardrails·프롬프트 필터·출력 검증.

발화 "AI 워크로드 심층 방어는 프롬프트 인젝션이나 에이전트 오·남용 같은 AI 전용 위협에 대응하는 새 보안층입니다. 기존 보안 위에 한 겹 더 쌓이는 셈이에요."

DAVID 너의 BYOK 구조에서 사용자 키 노출, prompt.txt에 악의적 지시 끼어드는 경우 — 정확히 이 위협. v3에서 "입력 검증 + 시나리오 출력 검증" 명시적 추가가 베이비 가드레일.

## B. 피지컬 AI

PHYSICAL AI · ROBOTICS FOUNDATION MODEL · SIM-TO-REAL

### 피지컬 AI — 로봇용 FM + 시뮬레이션 학습

정의 Physical AI = 우산 개념. Robotics FM = 로봇 동작 전용 거대 모델 (vision + 동작 시퀀스). Sim-to-Real = 시뮬→현실 전이 기술.

현장 POSCO 광양제철소 · Config의 Robotics FM 개발 · 위로보틱스 RLWRLD 휴머노이드 조작 · SK인텔릭스 실+합성 데이터로 Sim-to-Real 가속화. **한국 제조업이 가장 많이 등장한 영역.**

DAVID SmartThings = 가전 제어. Activator가 실제 SmartThings API로 가전 제어하는 순간, 너의 에이전트도 mini Physical AI의 한 형태.

AI FOR MI · 자율 예지정비

### 산업 응용 — 마켓 인텔리전스 · 예지정비

정의 AI for MI(Market Intelligence) = 시장 데이터 AI 자동 분석. 자율 예지정비 = 공장 설비 센서로 고장 시점 미리 예측 + 자율 조치.

현장 현대자동차 AI for MI · 두산 디지털이노베이션 Agentic AI 기반 자율 예지정비.

# C. 비즈니스 응용

AMAZON CONNECT

## Amazon Connect — 콜센터·고객응대용 AI 에이전트

정의 콜센터·챗봇·상담을 AI 에이전트가 1차 처리하는 AWS 완성형 서비스. 음성·텍스트 모두. CCaaS + Bedrock 통합.

현장 "Amazon Connect AI Agent가 다시 쓰는 고객 경험" — 이번 Summit 슬로건.

QUICK SUITE · AMAZON QUICK

## Quick Suite — 사내 데이터·생산성 통합 에이전트

정의 사내 데이터·문서·업무를 AI 에이전트가 통합. Microsoft Copilot의 AWS 버전. "20만 Amazonian 내재화 도구"로 마케팅.

한국 기업 응용 패턴

## 산업별 응용 — 같은 패턴, 다른 도메인

패턴 "고객접점에 에이전트를 꽂아 전환율을 올린다." 한 패턴의 변주.

사례 AI 컨시어지(하나투어·AK아이에스) · Audience Engine(무신사·GS SHOP) · 임베디드 금융(KB국민은행) · K-POP 글로벌 라이브(CJ ENM Mnet) · 제주항공 업무 혁신.

DAVID "SmartThings 시나리오 발견 접점에 에이전트 꽂아 활성화율 올림" — Samsung 사내 보고 한 줄로 카테고리 정렬 끝.

CH 06 · SYNTHESIS · 마스터

# 통합편 — 한 장으로

이틀, 18개 트랙, 100여 개 용어, 6개 영역. 이 마지막 챕터가 그 모든 것의 한 줄 요약입니다.

▶ MASTER SPEAKING SCRIPT · FULL　　약 4분 · 임원 보고 풀버전

OPENING · 25초

AWS Summit Seoul 2026, 이틀간 18개 트랙·100여 개 용어 들었습니다. 다 듣고 나서 정리한 한 문장은 이거예요… **"에이전틱 AI 시대로의 풀스택 전환"**. 다른 말로 하면, 클라우드의 모든 부품이 "AI 에이전트를 어떻게 만들고·돌리고·지키고·팔지"라는 한 질문 중심으로 재배치됐다는 뜻입니다.

PART 1 · 6개 영역 · 110초

전체 그림은 6개 영역으로 나뉘었습니다.

**첫째, 두뇌**. 모델 본체와 에이전트 프레임워크예요. Bedrock·Nova·Claude 같은 파운데이션 모델 위에 AgentCore·Strands SDK·MCP가 얹혀서, 모델이 "도구 쓰는 비서"로 변합니다.

**둘째, 공장**. AI Native 사고방식 + Kiro·Claude Code 같은 AI 코딩 도구 + CI/CD·IaC 자동화 + MLOps·AIOps·DevSecOps 운영. 코드를 매일 갱신하는 컨베이어 벨트예요.

**셋째, 연료**. S3에 깔리고 OpenSearch가 검색층이 되고, RAG·GraphRAG로 회사 데이터가 모델에 닿습니다. CDC와 Lakehouse가 실시간 갱신을 담당하고요.

**넷째, 엔진**. Trainium·Inferentia 자체 칩과 NVIDIA GPU를 얹은 EC2 P/G 위에서, HyperPod 클러스터로 수백 대 GPU를 묶어 굴립니다.

**다섯째, 요새**. Zero Trust 원칙 + KMS·CloudHSM 키 관리 + AI 전용 위협 방어. 에이전트가 회사 데이터를 진짜로 만지기 시작하면서 보안이 한 겹 더 두꺼워졌어요.

**여섯째, 응용**. 피지컬 AI—로봇·자동차·공장—와 비즈니스 에이전트—Amazon Connect·Quick Suite·산업별 응용. 가치가 실제로 만들어지는 층입니다.

PART 2 · 6개가 한 흐름으로 · 70초

이 여섯 개가 어떻게 한 흐름으로 묶이느냐… 사용자 질문 한 줄이 도는 길을 따라가면 명확합니다.

사용자가 **비즈니스 앱**에 질문을 던집니다. **요새**가 인증·권한을 통과시키고, **두뇌**의 에이전트가 받아서, **연료**에서 RAG로 회사 데이터를 끌어오고, **엔진** 위에서 추론을 돌립니다. 답이 다시 요새를 거쳐 사용자에게 닿고요. 그리고 이 모든 코드와 운영을 **공장**이 24시간 갱신하고 있어요.

한 줄로 정리하면… 두뇌가 일하고, 연료를 먹고, 엔진에서 돌고, 요새가 지키고, 공장이 갱신하고, 응용이 가치를 만든다.

PART 3 · 우리에게 의미 · 65초

우리에게 의미가 뭐냐… 이미 사내에서 만들고 있는 SmartThings Scenario Agent를 이 6개 그림 위에 올려보면, **응용·두뇌·연료** 세 층까지는 v2로 닿아 있습니다. 5-Agent Pipeline에 BYOK 모델, 27-시나리오 JSON DB가 그 증거예요.

남은 세 층은… **공장(CI/CD 자동화)**, **요새(사내 KMS 연동)**, **엔진(인프라 선택)**… 이게 v3에서 우리가 채워야 할 자리들입니다.

특히 한 가지만 짚자면 — **MCP 서버화**입니다. 시나리오 DB를 MCP라는 표준 도구로 노출하면, 모델이 바뀌어도, 다른 팀 에이전트가 호출해도 그대로 쓸 수 있는 자산이 됩니다. 가장 임팩트 큰 다음 한 수예요.

CLOSING · 30초

정리하면… 100개 용어는 6개 영역의 분담이었고, 6개 영역은 사용자 한 마디 답변을 위한 합주였고, 그 합주는 우리도 이미 일부 하고 있었다, 입니다.

다음 분기에 보일 가장 큰 변화는… "AI 에이전트가 우리 인프라의 디폴트 가정이 된다"는 점입니다. 감사합니다.

## 6개 영역 한 페이지 정리

| 영역 | 역할 | 핵심 키워드 |
|---|---|---|
| 🧠 **두뇌** Models & Agents | 생각하고 도구 쓰는 비서. | Bedrock · Nova · Anthropic · AgentCore · Strands · MCP · A2A · Multi-Agent |
| 🏭 **공장** Dev & Operations | 코드·운영을 매일 갱신하는 컨베이어. | AI Native · Spec-driven · AI-DLC · Kiro · Claude Code · CI/CD · IaC · MLOps · AIOps · DevSecOps |
| ⛽ **연료** Data Platform | 에이전트가 먹고 일하는 회사 데이터. | S3 · OpenSearch · RAG · GraphRAG · CDC · Lakehouse · AI-Ready Data |
| ⚙️ **엔진** Compute Infra | 모델이 실제로 도는 실리콘. | Trainium · Inferentia · EC2 P/G · Nitro · HyperPod · Slurm on EKS · Capacity Blocks |
| 🛡️ **요새** Security | 에이전트가 회사 자산을 만지는 걸 지킴. | Zero Trust · KMS · CloudHSM · Guardrails · DevSecOps · Secure AI |

| 영역 | 역할 | 핵심 키워드 |
|---|---|---|
| 💼 **응용** Physical & Business | 가치가 실제로 만들어지는 사용자 접점. | Amazon Connect · Quick Suite · Physical AI · Robotics FM · Sim-to-Real · AI 컨시어지 · Audience Engine |

## 통합 흐름도 — "오늘 출퇴근 시간 시나리오 추천해줘"의 뒷면

> **USER INPUT:** "오늘 출퇴근 시간 시나리오 추천해줘"

① **응용** — SmartThings 앱이 입력 수신. Amazon Connect·Quick Suite·또는 자체 앱이 1차 접점.

② **요새** — Zero Trust로 요청자 검증. KMS가 호출 키 안전하게 발급. AI 전용 Guardrails가 프롬프트 인젝션 차단.

③ **두뇌** — AgentCore에서 도는 5-Agent가 입력 수신. Curator가 시나리오 후보 분류, Localizer가 한국 맥락 반영. MCP가 외부 도구 호출 표준.

④ **연료** — RAG가 27-시나리오 DB(S3+OpenSearch)에서 관련 시나리오 검색. 운영 DB 변경분은 CDC+Lakehouse 실시간 동기화.

⑤ **엔진** — Bedrock이 Inferentia 또는 NVIDIA GPU(EC2 P/G)에서 답 생성. Nitro 격리. 응답시간 수백 ms.

⑥ **요새** — 출력에 민감 데이터·환각·정책 위반 없는지 가드레일이 한 번 더 검사.

⑦ **응용** — SmartThings 앱에 시나리오 카드 표시. 사용자는 한 번의 응답만 보지만, 뒤에서는 6개 영역이 합주.

> **USER OUTPUT:** "☕ 모닝 루틴: 7:00 자동 기상등 + 커피 머신 ON + 출퇴근 교통 안내"

↑ 그리고 한 가지 더 — 이 모든 흐름의 코드·인프라·모델은 **공장(AI-DLC + CI/CD + AIOps)**이 백그라운드에서 매일 갱신·감시 중입니다.

# SmartThings Scenario Agent — v3 후보 5개

> **현 상태 진단 (v2):**
> ✓ 응용 — SmartThings 사용자 접점 / ✓ 두뇌 — 5-Agent Pipeline + BYOK / △ 연료 — mini-RAG (정통 RAG 아님) / △ 엔진 — Cloudflare 위임 / ✗ 요새 — 개인 단위만 / ✗ 공장 — 수동 배포·수동 운영

## 01 MCP 서버화 연료 + 두뇌

27-시나리오 JSON DB와 prompt.txt를 MCP 표준 도구로 노출. 모델/팀이 바뀌어도 동일 도구로 호출. **자산화의 가장 큰 한 수.**

IMPACT — 매우 큼

## 02 Orchestrator 패턴 진화 두뇌

고정 Pipeline → Curator를 Orchestrator로 승격. 시나리오 복잡도에 따라 Localizer 건너뛰기·Expander 두 번 호출 같은 동적 분기. Story Chat을 Sub-agent로 정식 등록.

IMPACT — 큼

## 03 CI/CD 자동화 공장

git push → wrangler-publish GitHub Action으로 Cloudflare 자동 배포. 수동 콘솔 사라짐. P7-B 같은 잦은 튜닝 사이클에 가장 큰 시간 절약.

IMPACT — 큼

## 04 정통 RAG 전환 + GraphRAG 검토 연료

현재 facet 기반 retrieval → S3 + OpenSearch 벡터 인덱스로 정식 RAG. 시나리오 간 관계(같은 디바이스·시간대)를 살리려면 GraphRAG. 시나리오 100개 넘으면 임계점.

IMPACT — 중간

## 05 AIOps 도입 공장

Datadog 또는 Cloudflare Analytics로 P95 지연·LLM 호출 에러율 자동 감지. 너가 직접 로그 보지 않고도 이상 알림 받는 단계. 사용자 수 늘면 필수.

IMPACT — 중간

APPENDIX A · GLOSSARY

# 글로사리 — A→Z 65 terms

모든 용어를 영문 알파벳순으로. 막힐 때 찾아오는 자리.

---

### A2A 에이전트 간 통신 프로토콜 BRAIN

서로 다른 회사·프레임워크 에이전트들이 협업할 때 쓰는 표준 통신 프로토콜. 구글 주도. MCP가 모델↔도구라면 A2A는 에이전트↔에이전트.

### Agent · Agentic AI 에이전트 / 에이전틱 AI BRAIN

LLM에 도구사용·계획·자율판단을 더한 시스템. 단발 응답이 아니라 'Plan→Tool→Observe→Act' 루프를 도는 다단계 의사결정 AI. Chatbot ≠ Agent.

### AI Concierge AI 컨시어지 APPS

호스피탈리티·여행 도메인의 AI 응대 에이전트. 하나투어·AK아이에스 사례. '고객접점에 에이전트 + 도메인 데이터' 패턴의 한 변종.

### AI for MI 마켓 인텔리전스 자동화 APPS

시장 데이터를 AI가 자동 분석·요약하는 응용. 현대자동차 사례. '에이전트 + 도메인 데이터 + 정형 출력' 패턴.

### AI Native Development AI Native 개발 FACTORY

AI가 코드 짜는 걸 기본 전제로 두고 워크플로우를 처음부터 재설계. AI-Assisted(추가)와 다름. Cloud Native의 후속 개념.

### AI Workload Defense AI 워크로드 심층 방어 FORTRESS

AI 전용 위협(프롬프트 인젝션·모델 탈옥·민감정보 유출) 대응 보안층. Bedrock Guardrails·출력 검증·행동 제한. DevSecOps의 AI 시대 확장.

### AI-DLC AI 주도 개발 라이프사이클 FACTORY

기존 SDLC의 AI 시대 버전. Spec→AI 코드생성→AI 테스트→CI/CD→AIOps 전체 루프. AI Native의 사이클화·표준화.

### AI-Ready Data AI 준비 데이터 FUEL

'LLM이 곧바로 쓸 수 있는 상태'의 회사 데이터를 가리키는 개념. S3·OpenSearch·임베딩·RAG로 이 상태를 만듦.

### AIOps 에이아이옵스 · 시스템 운영 AI

FACTORY

시스템 운영(로그·알람·메트릭)을 AI가 분석해 장애를 자동 진단·복구. MLOps=모델 / AIOps=시스템 / DevSecOps=보안.

### Amazon Bedrock 베드락

BRAIN

여러 회사 FM(Claude·Llama·Nova 등)을 단일 API로 호출하는 AWS 서버리스 게이트웨이. 가드레일·RAG·에이전트 기능 통합.

### Amazon Bedrock AgentCore 에이전트코어

BRAIN

에이전트를 실행하는 AWS 관리형 런타임. 메모리·도구연결·로그·세션을 자동 관리. 코드는 너, 운영은 AWS.

### Amazon Connect 아마존 커넥트

APPS

콜센터·고객응대를 AI 에이전트가 1차 처리하는 AWS 완성형 서비스. 음성·텍스트 모두 지원. CCaaS + Bedrock 통합.

### Amazon Nova · Nova 2 노바 / 노바 2

BRAIN

AWS가 직접 만든 FM 패밀리. 텍스트·이미지·멀티모달·추론. Trainium 칩에 최적화돼 비용·지연 유리. Nova 2가 이번 Summit 주력.

### Amazon OpenSearch 오픈서치

FUEL

키워드 검색 + 벡터 검색을 모두 지원하는 분산 검색엔진. RAG의 retrieval 층 기본 부품. Elasticsearch 포크에서 출발.

### Amazon Quick · Quick Suite 아마존 퀵 / 퀵 스위트

APPS

사내 데이터·문서·업무를 AI 에이전트가 통합해주는 AWS 생산성 도구. Microsoft Copilot의 AWS 버전.

### Amazon S3 S3 / 객체 스토리지

FUEL

AWS 객체 스토리지. 모든 종류의 파일(이미지·문서·로그) 적재. AI-Ready 데이터의 기본 저장소. 거의 무한 확장.

### Amazon SageMaker 세이지메이커

BRAIN

데이터→학습→배포→모니터링까지 ML 전 과정 통합 플랫폼. Bedrock이 '이미 만든 FM 사용', SageMaker는 '내가 직접 만들고 굴림'.

### Anthropic Claude 앤트로픽 클로드

BRAIN

Anthropic의 FM. 추론·코딩·긴 문서 처리에 강하고 안전성 설계가 깐깐. Bedrock 단골. MCP 프로토콜 원작자.

### API Management API 관리 · 게이트웨이

FACTORY

서비스 API 호출의 인증·요금·트래픽 통제 게이트. AI 시대엔 '에이전트의 외부 도구 호출 단일 진입점'으로 의미 확장.

### Audience Engine 오디언스 엔진

APPS

미디어·커머스에서 사용자 세그먼트 자동 분류·타깃팅 엔진. 무신사·GS SHOP 사례. 추천 플랫폼의 핵심 부품.

### AWS CloudHSM 클라우드 HSM · 전용 HSM

FORTRESS

전용 하드웨어 보안 모듈. 키가 물리적으로 격리되는 한 단계 더 강한 옵션. 금융·정부처럼 규제 빡빡한 곳용.

### AWS Inferentia 인퍼런시아 · 추론 칩

ENGINE

AWS 자체 AI 칩 — 추론(inference) 전용. Trainium의 페어. NVIDIA GPU 의존도 ↓ 비용 ↓.

### AWS Kiro 키로 · AI 코딩 에이전트

FACTORY

Spec 파일 주면 코드·테스트·PR까지 자동 생성하는 AWS 코딩 에이전트. Spec mode + Vibe mode. Kiro=신규/Claude Code=수정/AWS Transform=현대화.

### AWS KMS 관리형 키 서비스

FORTRESS

AWS 관리형 암호화 키 서비스. 키 생성·로테이션·접근제어 자동. FIPS 140-2 Level 3.

### AWS Nitro System 나이트로 시스템

ENGINE

EC2의 하드웨어 보안·격리 기반. 하이퍼바이저+카드 분리로 호스트 OS 의존 ↓. AI 워크로드 보안 베이스라인.

### AWS Trainium 트레이니움 · 학습 칩

ENGINE

AWS 자체 AI 칩 — 학습(training) 전용. Inferentia와 페어. Nova가 이 칩 위에 최적화.

### AWS Transform 트랜스폼 · 레거시 현대화

FACTORY

오래된 .NET·메인프레임·Java 6 같은 코드를 AI가 분석해 최신 클라우드 네이티브로 자동 변환. 마이그레이션 비용 ↓.

### Bedrock RFT 강화학습 미세조정

BRAIN

Reinforcement Fine-Tuning. '좋은 답/나쁜 답' 보상신호로 FM을 추가 학습. 일반 SFT보다 행동 교정에 강함.

### CDC 변경 데이터 캡처

FUEL

Change Data Capture. 운영 DB의 변경분(insert/update/delete)만 실시간 스트리밍. binlog·WAL 기반. 배치 ETL 대체.

### CI/CD 지속적 통합·배포

FACTORY

코드 push → 자동 빌드·테스트·배포 컨베이어. AI 시대엔 PR 리뷰까지 AI가 자동화. GitHub Actions·CodePipeline 등.

### Claude Code 클로드 코드

FACTORY

터미널 기반 코딩 에이전트(Anthropic). IDE 없이 명령어로 파일을 직접 읽고 수정. Sub-agent 시스템으로 병렬 작업.

### DevSecOps 데브섹옵스 · 보안 통합 운영

FACTORY

보안 검사를 출시 직전이 아니라 코딩·빌드·배포 전 단계에 자동 삽입. 'Shift Left Security'. AI 시대엔 프롬프트 인젝션 검사 포함.

### EC2 Capacity Blocks 캐파시티 블록

ENGINE

GPU 인스턴스를 미리 예약하는 서비스. GPU 품귀 시대의 안전장치. 학습 일정 잡힌 곳이 미리 확보.

### EC2 P/G Instances EC2 P/G 인스턴스

ENGINE

NVIDIA GPU를 얹은 EC2 시리즈. P시리즈=학습용 고성능 GPU, G시리즈=추론·그래픽. Trainium의 NVIDIA 대안.

### Embedded Finance 임베디드 금융

APPS

비금융 서비스 안에 금융 기능을 끼워넣는 패턴(결제·송금·대출). KB국민은행 사례. AI로 추천·심사 자동화.

### Embedding 임베딩 · 벡터화

FUEL

텍스트·이미지를 의미를 담은 숫자 벡터로 변환. 의미 기반 검색(벡터 검색)의 전제. RAG의 핵심 부품.

### Foundation Model (FM) 파운데이션 모델

BRAIN

대용량 데이터로 self-supervised 사전학습된 거대 신경망. GPT·Claude·Nova·Gemini가 전부 FM. 다양한 task에 재활용 가능해 'Foundation'.

### GraphRAG 그래프 RAG

FUEL

문서 사이 관계를 그래프로 두고 그 위에서 검색. 단순 유사도 검색을 넘어 인과·계층 관계 살림. 미래에셋증권 상품DB 사례.

### Guardrails 가드레일 · AI 행동 제한

FORTRESS

AI 모델 입력·출력에 정책 필터를 거는 안전장치. 민감정보 출력 차단, 금지 주제 차단, 톤·형식 제약. Bedrock Guardrails가 대표.

### HyperPod 하이퍼팟 · 관리형 학습 클러스터

ENGINE

수백 대 GPU를 한 묶음으로 굴리는 AWS의 관리형 학습 클러스터. 노드 장애 자동복구·체크포인트 자동저장. 하이퍼커넥트 사례.

### IaC 코드형 인프라

FACTORY

Infrastructure as Code. 서버·DB·네트워크를 콘솔 클릭이 아니라 코드 파일로 정의·실행. Terraform·CloudFormation·CDK. 재현 가능·버전관리 가능.

### Lakehouse 레이크하우스

FUEL

Data Lake + Data Warehouse 통합 아키텍처. Iceberg·Delta Lake 등이 대표. 배치 분석과 실시간 분석을 한 자리에서.

### Managed Service 관리형 서비스

BRAIN

서버 설치·패치·확장·백업을 클라우드가 다 해주고 사용자는 '쓰기'에만 집중하는 형태. Self-managed의 반대. 엔터프라이즈 선호.

### MCP 모델 컨텍스트 프로토콜

BRAIN

모델 ↔ 외부 도구·데이터 표준 연결 규격. Anthropic 주도, OpenAI·Google·AWS 채택. 'AI의 USB-C'. JSON-RPC 기반.

### MLOps 엠엘옵스 · 모델 운영

FACTORY

모델의 데이터→학습→배포→모니터링→재학습 라이프사이클 자동화. SageMaker AI MLOps가 대표. 모델 버전관리·실험추적·드리프트 감지.

### Multi-Agent 멀티 에이전트

BRAIN

에이전트 여러 명을 역할별로 분담시켜 협업. 4가지 패턴 — Single/Pipeline/Orchestrator+Sub-agents/Autonomous. David의 v2 = Pipeline 패턴.

### Nova Forge 노바 포지

BRAIN

Nova 모델을 도메인 데이터로 커스터마이징하는 AWS 도구. Fine-tuning 워크플로우 관리형. RFT도 여기 묶임.

### Orchestrator pattern 오케스트레이터 패턴

BRAIN

팀장 에이전트가 Sub-agent들에게 작업 분담·통합. Pipeline보다 동적 분기 가능. Claude Code의 sub-agent 구조가 대표.

### Physical AI 피지컬 AI

APPS

AI가 모니터를 나와 로봇·자동차·공장·가전으로 들어가는 흐름의 우산 개념. Robotics FM + Sim-to-Real이 핵심 부품.

### Pipeline pattern 파이프라인 패턴

BRAIN

에이전트 A→B→C 순차 호출. 컨베이어 벨트. 단순하고 안정적이지만 동적 분기 어려움. David v2 = 5-Agent Pipeline.

### Prompt Injection 프롬프트 인젝션 FORTRESS

악의적 명령을 프롬프트에 끼워넣어 모델 정책을 우회하는 공격. AI 시대 전용 위협의 대표격. Guardrails로 대응.

### RAG 검색증강생성 FUEL

Retrieval-Augmented Generation. LLM이 답하기 전에 회사 DB에서 관련 문서를 먼저 검색해 함께 넘기는 방식. 환각 ↓, 데이터 갱신 즉시 반영. Fine-tuning의 대안.

### Robotics Foundation Model 로봇용 파운데이션 모델 APPS

텍스트 FM처럼 로봇 동작 전용 거대 모델. Vision + 동작 시퀀스 학습. RT-2·π0 계열. POSCO·Config 사례.

### SageMaker Catalog 세이지메이커 카탈로그 BRAIN

'회사 안에 어떤 모델·데이터셋이 있고 누가 만들었고 어떻게 쓰는지' 검색·관리. 메타데이터·계보·거버넌스. 중복 개발 차단.

### SageMaker Unified Studio 유니파이드 스튜디오 BRAIN

데이터 탐색·노트북·파이프라인을 한 화면에 통합한 IDE. SageMaker Studio + Data Wrangler + Glue Studio 등 통합.

### Secure AI by Design 시큐어 AI 바이 디자인 FORTRESS

설계 단계부터 AI 보안을 내장하는 접근. Palo Alto Networks 사례. Zero Trust + Guardrails + 거버넌스 결합.

### Sim-to-Real 심투리얼 · 시뮬→실세계 전이 APPS

시뮬레이션에서 학습한 모델을 실제 로봇·환경으로 옮기는 기술. 도메인 랜덤화·domain adaptation. 현실 실수 비용 큰 산업에 필수.

### Slurm on EKS EKS 위의 슬럼 ENGINE

HPC 표준 잡 스케줄러 Slurm을 Kubernetes(EKS) 위에서 돌리는 구성. 학습 작업 큐 관리. 하이퍼커넥트 도입 사례.

### Spec-driven Development 스펙 주도 개발 FACTORY

요구사항(spec)을 1급 시민으로. 코드는 spec에서 자동 생성, spec ↔ code 양방향 동기화. AWS Kiro가 대표 구현.

### Strands Agent SDK 스트랜즈 SDK BRAIN

AWS의 오픈소스 에이전트 개발 키트(Python). 가볍고 BYOK 친화적(Bedrock·OpenAI·Anthropic 다 지원). LangChain·LangGraph의 AWS측 대안.

### Sub-agent 서브 에이전트 BRAIN

메인 에이전트(Orchestrator)가 호출하는 보조 에이전트. 작업을 위임·병렬화. Claude Code가 sub-agent 시스템 채택.

### Zero Trust 제로 트러스트

FORTRESS

'아무도 기본 신뢰하지 않는다.' 매 요청마다 ID·기기·문맥 동적 검증. 사내·사외 구분 없음. AI 시대엔 에이전트의 API 호출 전부 재확인.

### 자율 예지정비 Autonomous Predictive Maintenance

APPS

공장 설비 센서 데이터로 '곧 고장날 시점'을 미리 예측. 에이전트가 진단+조치까지 자율 수행. POSCO·두산 사례.

APPENDIX B · EN ↔ KR MAPPING

# 영-한 매핑 — 빠른 대응표

정의 없이 깔끔한 매핑만. "이 영어 용어 한국말로 뭐였지" 또는 그 반대.

| ENGLISH | 한국어 | AREA |
|---|---|---|
| **A2A** | 에이전트 간 통신 프로토콜 | BRAIN |
| **Agent · Agentic AI** | 에이전트 / 에이전틱 AI | BRAIN |
| **AI Concierge** | AI 컨시어지 | APPS |
| **AI for MI** | 마켓 인텔리전스 자동화 | APPS |
| **AI Native Development** | AI Native 개발 | FACTORY |
| **AI Workload Defense** | AI 워크로드 심층 방어 | FORTRESS |
| **AI-DLC** | AI 주도 개발 라이프사이클 | FACTORY |
| **AI-Ready Data** | AI 준비 데이터 | FUEL |
| **AIOps** | 에이아이옵스 · 시스템 운영 AI | FACTORY |
| **Amazon Bedrock** | 베드락 | BRAIN |
| **Amazon Bedrock AgentCore** | 에이전트코어 | BRAIN |
| **Amazon Connect** | 아마존 커넥트 | APPS |
| **Amazon Nova · Nova 2** | 노바 / 노바 2 | BRAIN |
| **Amazon OpenSearch** | 오픈서치 | FUEL |
| **Amazon Quick · Quick Suite** | 아마존 퀵 / 퀵 스위트 | APPS |
| **Amazon S3** | S3 / 객체 스토리지 | FUEL |
| **Amazon SageMaker** | 세이지메이커 | BRAIN |
| **Anthropic Claude** | 앤트로픽 클로드 | BRAIN |

| ENGLISH | 한국어 | AREA |
|---|---|---|
| **API Management** | API 관리 · 게이트웨이 | FACTORY |
| **Audience Engine** | 오디언스 엔진 | APPS |
| **AWS CloudHSM** | 클라우드 HSM · 전용 HSM | FORTRESS |
| **AWS Inferentia** | 인퍼런시아 · 추론 칩 | ENGINE |
| **AWS Kiro** | 키로 · AI 코딩 에이전트 | FACTORY |
| **AWS KMS** | 관리형 키 서비스 | FORTRESS |
| **AWS Nitro System** | 나이트로 시스템 | ENGINE |
| **AWS Trainium** | 트레이니움 · 학습 칩 | ENGINE |
| **AWS Transform** | 트랜스폼 · 레거시 현대화 | FACTORY |
| **Bedrock RFT** | 강화학습 미세조정 | BRAIN |
| **CDC** | 변경 데이터 캡처 | FUEL |
| **CI/CD** | 지속적 통합·배포 | FACTORY |
| **Claude Code** | 클로드 코드 | FACTORY |
| **DevSecOps** | 데브섹옵스 · 보안 통합 운영 | FACTORY |
| **EC2 Capacity Blocks** | 캐파시티 블록 | ENGINE |
| **EC2 P/G Instances** | EC2 P/G 인스턴스 | ENGINE |
| **Embedded Finance** | 임베디드 금융 | APPS |
| **Embedding** | 임베딩 · 벡터화 | FUEL |
| **Foundation Model (FM)** | 파운데이션 모델 | BRAIN |
| **GraphRAG** | 그래프 RAG | FUEL |
| **Guardrails** | 가드레일 · AI 행동 제한 | FORTRESS |
| **HyperPod** | 하이퍼팟 · 관리형 학습 클러스터 | ENGINE |

| ENGLISH | 한국어 | AREA |
|---|---|---|
| **IaC** | 코드형 인프라 | FACTORY |
| **Lakehouse** | 레이크하우스 | FUEL |
| **Managed Service** | 관리형 서비스 | BRAIN |
| **MCP** | 모델 컨텍스트 프로토콜 | BRAIN |
| **MLOps** | 엠엘옵스 · 모델 운영 | FACTORY |
| **Multi-Agent** | 멀티 에이전트 | BRAIN |
| **Nova Forge** | 노바 포지 | BRAIN |
| **Orchestrator pattern** | 오케스트레이터 패턴 | BRAIN |
| **Physical AI** | 피지컬 AI | APPS |
| **Pipeline pattern** | 파이프라인 패턴 | BRAIN |
| **Prompt Injection** | 프롬프트 인젝션 | FORTRESS |
| **RAG** | 검색증강생성 | FUEL |
| **Robotics Foundation Model** | 로봇용 파운데이션 모델 | APPS |
| **SageMaker Catalog** | 세이지메이커 카탈로그 | BRAIN |
| **SageMaker Unified Studio** | 유니파이드 스튜디오 | BRAIN |
| **Secure AI by Design** | 시큐어 AI 바이 디자인 | FORTRESS |
| **Sim-to-Real** | 심투리얼 · 시뮬→실세계 전이 | APPS |
| **Slurm on EKS** | EKS 위의 슬럼 | ENGINE |
| **Spec-driven Development** | 스펙 주도 개발 | FACTORY |
| **Strands Agent SDK** | 스트랜즈 SDK | BRAIN |
| **Sub-agent** | 서브 에이전트 | BRAIN |
| **Zero Trust** | 제로 트러스트 | FORTRESS |

| ENGLISH | 한국어 | AREA |
| --- | --- | --- |
| **자율 예지정비** | Autonomous Predictive Maintenance | APPS |

APPENDIX C · AWS VS AZURE VS GCP

# 클라우드 비교 — 같은 역할 다른 이름

다른 클라우드 얘기 들을 때 즉시 대응되는 매핑. 영역별로 정리.

## 🧠 두뇌 · Models & Agents

| 역할 | AWS | AZURE | GCP |
|---|---|---|---|
| FM 게이트웨이 | `Amazon Bedrock` | `Azure AI Foundry` | `Vertex AI Model Garden` |
| 1st-party FM | `Amazon Nova` | `Microsoft Phi` | `Google Gemini` |
| ML 플랫폼 | `SageMaker` | `Azure Machine Learning` | `Vertex AI` |
| 에이전트 프레임워크 | `AgentCore · Strands SDK` | `AI Foundry Agent · Semantic Kernel` | `Vertex AI Agent Builder · ADK` |

## 🏭 공장 · Dev & Operations

| 역할 | AWS | AZURE | GCP |
|---|---|---|---|
| AI 코딩 에이전트 | `AWS Kiro` | `GitHub Copilot` | `Gemini Code Assist` |
| CI/CD | `CodePipeline · CodeBuild` | `Azure DevOps · GitHub Actions` | `Cloud Build` |
| IaC (서비스) | `CloudFormation · CDK` | `ARM · Bicep` | `Deployment Manager` |

## ⛽ 연료 · Data

| 역할 | AWS | AZURE | GCP |
|---|---|---|---|
| 객체 스토리지 | `Amazon S3` | `Blob Storage` | `Cloud Storage` |
| 검색·벡터 DB | `OpenSearch` | `Azure AI Search` | `Vertex AI Search` |
| Lakehouse | `Lake Formation · Iceberg` | `Microsoft Fabric · Synapse` | `BigLake · BigQuery` |
| 데이터 카탈로그 | `Glue Catalog · SageMaker Catalog` | `Microsoft Purview` | `Dataplex` |

## ⚙️ 엔진 · Compute

| 역할 | AWS | AZURE | GCP |
|---|---|---|---|
| 학습용 AI 칩 | `AWS Trainium` | `Azure Maia` | `TPU` |
| 추론용 AI 칩 | `AWS Inferentia` | `Azure Cobalt` | `TPU (inference)` |
| GPU 인스턴스 | `EC2 P/G Series` | `NV · NC · ND Series` | `A2 · A3 · G2` |
| 학습 클러스터 | `SageMaker HyperPod` | `Azure ML Compute Clusters` | `Vertex AI Training · GKE` |

## 🛡️ 요새 · Security

| 역할 | AWS | AZURE | GCP |
|---|---|---|---|
| 관리형 키 서비스 | `AWS KMS` | `Azure Key Vault` | `Cloud KMS` |
| 전용 HSM | `CloudHSM` | `Managed HSM` | `Cloud HSM` |
| AI Guardrails | `Bedrock Guardrails` | `Azure AI Content Safety` | `Vertex AI Safety Filters` |

# 💼 응용 · Business

| 역할 | AWS | AZURE | GCP |
|---|---|---|---|
| 콜센터 AI | `Amazon Connect` | `Dynamics 365 Contact Center` | `Contact Center AI` |
| 사내 생산성 AI | `Quick Suite · Amazon Quick` | `Microsoft 365 Copilot` | `Gemini for Workspace` |

> **주의 사항** — 클라우드 제품명은 빠르게 바뀝니다 (예: Azure OpenAI Service → Azure AI Foundry 통합). 이 표는 "역할 기준 매핑"으로 봐주세요. 도입 검토 시엔 각 회사 공식 문서 최신본 확인 필수. 같은 칸이라도 가격·성능·생태계는 회사마다 크게 다릅니다.