

Claude Code on Amazon Bedrock

CCB 도입 교육 프로그램

영상 10개로 완성하는 Claude Code — AI로 바꾸는 개발, 팀의 경쟁력이 됩니다.

AWS 전문가가 Claude Code on Amazon Bedrock의 도입부터 환경 세팅, 엔터프라이즈 운영까지를 아우르는 10개 모듈 주제에 대한 커리큘럼을 실전 강의 영상으로 제공하는 프로그램입니다.

10개

총 모듈 수

4단계

학습 단계

기초부터

실전까지

aws

POWERED BY AWS

프로그램 소개

Claude Code on Amazon Bedrock(CCB)는 Amazon Bedrock 위에서 동작하는 Anthropic의 AI 코딩 에이전트입니다. 본 교육 프로그램은 CCB를 처음 도입하는 고객이 환경 셋업부터 엔터프라이즈 운영까지 단계적으로 역량을 쌓을 수 있도록 설계되었습니다.

기본적인 사용법부터 **AI와 함께 개발하는 마인드셋**의 전환, 팀 단위 협업 워크플로우, 보안과 비용 거버넌스까지 — 실무에서 바로 적용 가능한 내용으로 구성되어 있습니다.

모든 모듈은 **AWS 현장 전문가가 직접 강의하는 영상**으로 제공됩니다. AWS Skill Builder와 YouTube를 통해 시청하실 수 있습니다.

▸ CCB 온라인 교육 프로그램 홈페이지

AWS 전문가가 직접 강의하는 10개 모듈 커리큘럼을 한눈에 확인하세요.

Skill Builder와 YouTube를 통해 지금 바로 수강하실 수 있습니다.

[홈페이지 바로가기 →](#)[Skill Builder 학습 플랜 →](#)

■ **기반 구축** — 환경 셋업 ■ **개인 역량** — AI 코딩 마인드셋 ■ **팀 협업** — 협업 개발 & 자동화 ■ **엔터프라이즈** — 조직 운영

모듈 구성 — M0 ~ M9

M0 기반 구축

안효빈 시니어 솔루션즈 아키텍트, AWS

Claude Code를 다룬다는 것: Amazon Bedrock 위에서 보는 핵심 구성요소

같은 프롬프트를 입력해도 결과가 달라지는 이유는 환경 선언(설정)이 다르기 때문입니다. 이 모듈은 Claude Code의 동작을 결정하는 핵심 구성요소를 Amazon Bedrock 위에서 체계적으로 다룹니다. Bedrock 기반 인증·모델 설정부터 .claude/ 파일 구조, Skill·Sub-agent·Hook을 활용한 자동화, MCP·Plugin 연결까지 — "환경이 곧 스펙"이라는 원칙 아래 재현 가능한 프로덕션 레디 환경을 직접 구성합니다.

학습 목표

- Amazon Bedrock 기반 Claude Code의 엔터프라이즈 장점(보안·데이터 주권·과금·컴플라이언스) 이해
- 통합 인증·모델 핀·리전·Cross-Region Inference 설정 방법 습득
- .claude/ 파일 구조와 settings.json·CLAUDE.md의 4계층 우선순위 체계 이해
- Skill·Sub-agent·Hook·MCP·Plugin으로 재현 가능한 프로덕션 레디 환경 완성

[▶ Skill Builder](#)[▶ YouTube](#)[📄 발표자료](#)

M1 개인 역량

문곤수 시니어 AI 스페셜리스트 솔루션즈 아키텍트, AWS

AI 코딩 마인드셋: 같은 도구, 다른 코드

같은 AI 도구를 써도 결과가 다른 이유는 사용법이 아니라 사고방식에 있습니다. 내 머릿속의 의도와 실제로 AI에게 전달된 내용 사이의 간극을 줄이는 법, AI와 협업할 때 개발자가 가져야 할 세 가지 역할, 그리고 자주 빠지는 6가지 실수 패턴을 금융 에이전트 데모와 함께 배웁니다.

학습 목표

- AI 코딩에서 의도(Meant)와 명시(Said)의 간극 이해
- Orchestrator·Architect·Reviewer 세 가지 역할과 AI 위임·감독 영역 구분
- AI 코딩 시 자주 발생하는 6가지 안티패턴 진단
- Spec-driven Development(SDD)의 원리와 적용 방법 이해

[▶ Skill Builder](#)[▶ YouTube](#)[📄 발표자료](#)

Spec-Driven Development: 스펙으로 개발하기

"만들어줘" 한 줄로 빠르게 코드를 얻을 수 있지만, 맥락이 길어질수록 구조와 일관성이 무너집니다. 이 모듈은 그 한계를 넘는 대안인 Spec-Driven Development(SDD, 명세 기반 개발)를 다룹니다. 무엇을, 어떻게 만들지 먼저 스펙으로 정의하고 개발하는 방식을 배우고, 쇼핑몰 장바구니·쿠폰·배송비 기능 데모를 통해 예측 가능하고 일관성 있는 개발 흐름을 직접 경험합니다.

학습 목표

- AI 어시스트 코딩 트렌드를 이해하고 나에게 맞는 개발 방법 파악
- SDD(명세 기반 개발) 개념과 Vibe Coding과의 차이점 이해
- Claude Code 환경에서 SDD를 활용하는 세 가지 방법 학습
- Superpowers 워크플로우와 핵심 스킬을 실습 데모를 통해 바로 적용

[▶ Skill Builder](#)[▶ YouTube](#)[📄 발표자료](#)

Team SDD — 3명이 함께 스펙으로 개발하기 (feat. AI-DLC)

AI가 코드를 짜주는 시대인데, 왜 팀의 생산성은 기대만큼 오르지 않을까요? 이 모듈은 AWS의 AI 네이티브 방법론인 AI-DLC(AI-Driven Development Lifecycle)를 통해 팀이 함께 AI와 협업하는 방법을 다룹니다. 의도를 합의하고 → 역할을 나눠 병렬로 구축하고 → 배포·운영하는 3단계 흐름을 3명이 함께 실습하며, 팀 단위 AI 협업 워크플로우를 직접 설계할 수 있습니다.

학습 목표

- 코딩 속도와 개발 속도의 차이를 인식하고 팀 생산성에 SDD가 필요한 이유 이해
- AI-DLC 3단계(Inception→Construction→Operations)와 AI-사람 협업 루프 이해
- Claude Code 환경에서 AI-DLC 거버넌스 체계 구성 실습
- 팀 병렬 개발에 필요한 Claude Code tools·skills 노하우 습득

[▶ Skill Builder](#)[▶ YouTube](#)[📄 발표자료](#)

코드베이스 이해와 기여 — Claude Code로 기존 프로젝트에 뛰어들고 안전하게 코드 수정하기

코드베이스를 이해하려면 무작정 전체를 보는 것이 아니라, 진입점에서 수정 영역, 서비스 접점, 코드 밖 로직 순으로 범위를 좁혀가며 탐색해야 합니다. 탐색한 내용은 CLAUDE.md에 누적해 "자라는 지도"를 만들고, 서브에이전트와 심볼 단위 추적으로 컨텍스트를 깨끗하게 유지합니다. 레거시·인증 같은 민감 영역은 규칙으로 보호하고, 새 기능은 코드보다 스펙을 먼저 작성한 뒤 구현합니다.

학습 목표

- 대규모 코드베이스를 한 번에 이해하려는 시도가 실패하는 이유 4가지 차원으로 이해
- Entry point→수정 영역→접점→코드 밖 로직 순의 점진적 탐색 전략 습득
- CLAUDE.md를 세션마다 누적하는 "자라는 지도" 구성과 컨텍스트 위생 유지 방법
- "코드에 없는 로직"(DB 트리거·죽은 코드) 함정 진단 및 검증 방법 이해

[▶ Skill Builder](#)[▶ YouTube](#)[📄 발표자료](#)

AI 코드 검증 — AI가 만든 Agent 코드를 어떻게 믿어?

30초 만에 AI가 만든 코드, 그대로 프로덕션에 올려도 될까요? AI가 썼기 때문에 못 믿는 게 아니라, 검증을 거치지 않았기 때문에 못 믿는 겁니다. 이 모듈은 코드를 "읽어서" 결함을 잡는 정적 리뷰와, "돌려서" 행동을 검증하는 실행 검증, 두 축으로 AI 생성 코드를 체계적으로 신뢰할 수 있는 워크플로우를 구축합니다.

학습 목표

- 신뢰의 출처가 작성자(Author)가 아닌 통과한 게이트(Gate)임을 이해
- Read 축(정적 리뷰)과 Run 축(실행 검증)의 상보적 구조와 활용 방법 이해
- Claude의 코드 리뷰 기능(security-review·code-review·pr-review-toolkit) 활용
- AI 생성 코드의 대표 위험 패턴과 검증 안티패턴 진단

[▶ Skill Builder](#)[▶ YouTube](#)[📄 발표자료](#)

하네스 엔지니어링 — 디자인하고, 측정하고, 관리하기

같은 모델을 써도 어떤 에이전트는 일을 끝내고, 어떤 에이전트는 끝내 실패합니다. 그 차이를 만드는 건 모델이 아니라 하네스 — 에이전트를 둘러싼 환경 전체입니다. 이 모듈은 Claude Code 하네스의 5가지 구성요소를 설계하고, 측정하고, 안전하게 운영하는 방법을 다룹니다.

학습 목표

- 하네스(Harness)의 정의와 Agent = Model + Harness 등식 이해
- Claude Code 하네스의 5대 구성요소(제약·도구·실행·상태·게이트) 설계 방법 습득
- 하네스 3대 원칙(평가자 분리·환경 강제·컨텍스트 관리) 체득
- 컨텍스트 엔지니어링과 하네스 개선 라이프사이클 운영

[▶ Skill Builder](#)[▶ YouTube](#)[📄 발표자료](#)

MCP와 Sub-agents: AI Agent의 능력 확장

단일 AI 에이전트는 코드 생성·분석엔 훌륭하지만, "실제 데이터로 구체적 절감액 산출"같은 요구엔 한계가 있습니다. 이 모듈은 그 한계를 넘는 두 축 — 외부 시스템을 직접 읽고 조작하는 MCP와, 독립 컨텍스트에서 전문 역할을 위임하는 Sub-agent를 다룹니다. FinOps Agent를 Level 0→1→2로 확장하며 멀티 에이전트 환경을 직접 구성합니다.

학습 목표

- 단일 에이전트의 가치와 접근 범위의 한계(능력 한계가 아닌) 구분
- MCP의 핵심 아키텍처(Client↔Server, Tool/Resource)와 전송 방식 이해
- Sub-agent의 본질(독립 컨텍스트)과 5가지 이점(컨텍스트 보존·비용 제어·재사용 등) 이해
- FinOps Agent를 Level 0→1→2로 확장하며 멀티 에이전트 환경 구성

[▶ Skill Builder](#)[▶ YouTube](#)[📄 발표자료](#)

Claude Code 자동화와 확장: 내 터미널 너머

Claude Code로 코드 리뷰를 해본 경험은 있으실 겁니다. 그런데 그런 내가 터미널을 열고 직접 물어봤을 때만 일어납니다. 새벽 3시에 올라온 PR은? 내가 휴가인 날은? 이 모듈은 Claude Code를 CI/CD 파이프라인에 연결해 사람의 개입 없이 자동으로 동작하게 만드는 방법과, 팀의 지식을 Plugin으로 축적해 AI 분석 품질을 팀과 함께 성장시키는 방법을 다룹니다.

학습 목표

- claude -p headless mode의 핵심 개념과 CI/CD 파이프라인 통합 방법 이해
- GitHub Actions 워크플로우 구축으로 PR 자동 분석 환경 실습
- Plugin 구조(skills/·references/·.claude-plugin/)와 마켓플레이스 배포 방법 습득
- 팀 아키텍처 지식을 축적하는 기여 사이클 구성과 조직 수준 확장 전략 수립

[▶ Skill Builder](#)[▶ YouTube](#)[📄 발표자료](#)

엔터프라이즈 CCB 운영 — 보안, 비용, 팀 도입

AI 코딩 도구를 팀 전체에 도입하면 반드시 맞닥뜨리는 현실적인 질문들을 다룹니다. 누가 어디까지 쓸 수 있게 할지, 비용은 어떻게 관리할지, 도구가 바뀌어도 운영 체계는 어떻게 유지할지 — LLM Gateway(거버넌스)와 활용 교육(Enablement) 두 기둥으로 답을 제시하고, 조직에 맞는 도입 옵션 비교와 월요일 아침에 바로 실행할 수 있는 로드맵으로 마무리합니다.

학습 목표

- 도구가 바뀌어도 변하지 않는 보안 4대 경계(Privilege·Data·Audit·Cost) 이해
- LLM Gateway 구현 3가지 옵션(AWS Native·상용OSS·자체 구축) 비교 및 선택 기준 습득
- 6영역 18문항 체크리스트로 조직 요구사항을 정의하고 적합한 옵션 선택
- 활용 교육 4개 축(비용 효율·하네스·워크플로우·전사 확장)과 도입 로드맵 수립

[▶ Skill Builder](#)[▶ YouTube](#)[📄 발표자료](#)

지금 바로 시작할 준비가 되셨나요?

M0부터 순서대로, 혹은 현재 팀의 필요에 맞는 모듈부터 시작하세요.
CCB와 함께하는 개발의 새로운 기준을 경험해 보시기 바랍니다.

- 총 10개 모듈 구성
- 4단계 학습 로드맵
- Skill Builder & YouTube 제공
- 기초부터 실전까지