CLAUDE CODE DEEP DIVE WORKSHOP

Chapter 3 – Admin Setup

개인의 도구를 조직의 플랫폼으로

2026.09.01

Song, Minji
Solutions Architect
AWS Korea
ssminji@amazon.com

Chapter 3 Learning Objectives

본 챕터에서 달성할 학습 목표

AT THE END OF THIS CHAPTER

조직 규모의 설치와 인증 체계를 설계하고, managed settings로 정책을 강제하며, OTel과 대시보드로 사용을 관측하고, 컴플라이언스 요구에 답할 수 있습니다.

OBJ 1

배포 설계

설치 채널과 미러, 버전 통제로 대량 배포를 설계.

OBJ 2

인증 체계

공급자 선택과 SSO, 자격증명 회전 체계를 구축.

OBJ 3

정책 강제

managed settings 4채널과 강제 키로 거버넌스 확립.

OBJ 4

관측과 대응

OTel, 비용 귀속, 감사 로그로 운영을 가시화.

Admin이 마주하는 도전

개인 도구와 조직 플랫폼의 간극

개발자의 세계 (Chapter 1, 2)

- 원라이너 설치, 개인 구독 로그인
- 내 settings.json과 내 에이전트
- 자유로운 도구와 웹 접근
- 비용은 내 플랜 한도

관리자의 질문 (Chapter 3)

- 수백 명에게 어떻게 배포하고 갱신하나
- 키를 나눠주지 않고 인증하려면
- 금지 명령과 도메인을 어떻게 강제하나
- 누가 얼마나 쓰는지, 감사는 어떻게

Scale
수백 명 규모

No keys
키 미배포 원칙

Enforce
정책은 강제로

Audit
증적 가능한 운영

03

Admin Setup

배포, 인증, 거버넌스, 관측의 조직 설계

CHAPTER 03 / PART 01

배포 전략

수백 대에 설치하고 갱신하는 법

- 설치 채널 4계열과 조직 관점 선택
- 사내 미러와 Artifactory 운영
- 버전 통제: 채널 고정과 minimumVersion
- 점진 롤아웃과 에어갭 대응

설치 채널 4계열 / 조직 관점

무엇으로 수백 대를 깔 것인가

Native 스크립트	claude.ai/install.sh, 단일 바이너리, 자동 갱신	표준 권장, MDM 스크립트화
OS 패키지 저장소	apt, dnf, apk 공식 저장소, GPG 서명	리눅스 서버 플릿
패키지 매니저	Homebrew cask, WinGet	개발자 셀프서비스
컨테이너	표준 이미지, devcontainer	CI 러너, 통제 환경
npm (레거시)	Node 의존, 신규 배포 비권장	기존 파이프라인 호환용

사내 미러 운영

Artifactory, Nexus로 다운로드 경로 내재화

INTERNAL MIRROR

외부 다운로드가 막힌 망이나 대역폭, 감사 요구가 있는 조직은 공식 저장소를 사내 아티팩트 서버로 미러링합니다.

WHAT

미러 대상

apt, rpm 저장소와 native 설치 자산, 필요 시 npm 패키지.

HOW

프록시 저장소

Artifactory, Nexus의 remote repo로 상류 캐시 구성.

POINT

클라이언트 전환

sources.list와 설치 스크립트 URL을 사내 주소로 치환.

TRUST

서명 유지

GPG 검증은 그대로, 미러도 서명 파일을 함께 제공.

Remote repo
상류 캐시 방식

서명 보존
무결성 원칙

감사 로그
다운로드 이력 확보

Lab 1
실습에서 구축

멀티 플랫폼 전략표

OS별 표준 조합

macOS	Native + MDM 스크립트, plist 정책	Jamf, Kandji 배포
Windows	WinGet + Intune, HKLM 정책	WSL 상속 키 병행
Linux	apt/dnf 저장소, /etc/claude-code 정책	Ansible 코드화
컨테이너	표준 이미지, 정책 내장, 갱신 차단	CI와 devcontainer
검증	전 플랫폼 /status로 정책 소스 확인	(remote/plist/HKLM/file)

버전 통제 두 다이얼

채널과 최소 버전

VERSION GOVERNANCE

autoUpdatesChannel로 갱신 속도를, minimumVersion으로 하한선을 통제합니다. 둘 다 관리 설정으로 강제할 수 있습니다.

DIAL 1

autoUpdatesChannel

stable 고정이 플릿 기본, latest는 얼리어답터 링만.

DIAL 2

minimumVersion

보안 픽스 하한선, 미달 버전은 강제 갱신 유도.

FREEZE

완전 고정

DISABLE_AUTOUPDATER=1 + 버전 지정 설치, CI 전용.

WHERE

강제 위치

managed-settings에 넣어 개인이 못 바꾸게.

stable
플릿 채널 기본

minimumVersion
하한 강제 키

CI만 고정
완전 고정 범위

managed
정책 배포 채널

점진 롤아웃 전략

파일럿에서 전사까지 4링

RING DEPLOYMENT

링 단위로 넓히며 각 관문에서 채택 지표와 사고 여부를 확인합니다. 정책과 버전은 링별로 독립 조정합니다.

RING 0

챔피언 10명

latest 채널, 피드백 루프 구축

RING 1

파일럿 팀 50

stable, managed 정책 초판 적용

RING 2

부문 200

교육 자료 배포, 지표 관측

RING 3

전사

표준 온보딩, 지원 체계 정착

CHAPTER 03 / PART 02

Provider와 자격증명

키를 나눠주지 않는 인증 설계

- 공급자 결정표와 인증 우선순위
- Bedrock: IAM 정책과 Identity Center SSO
- Claude Platform on AWS (Marketplace)
- 회전, 볼트 통합, PrivateLink

Provider 결정표

공식 admin-setup 권고 기준

Teams / Enterprise	좌석제, 인프라 불필요, 기본 권장	claude.ai와 통합 관리
Claude Platform	API 우선, 종량 과금	파이프라인 중심 조직
Amazon Bedrock	AWS 컴플라이언스와 과금 상속	AWS 표준 기업
혼합	LLM gateway로 단일 엔드포인트	중앙 로깅 요구 시

Bedrock IAM 정책

최소 권한의 표준형

iam policy (claude-code-bedrock)

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "ClaudeCodeInvoke",
    "Effect": "Allow",
    "Action": ["bedrock:InvokeModel",
      "bedrock:InvokeModelWithResponseStream"],
    "Resource":
      "arn:aws:bedrock:*::foundation-model/anthropic.*"
  }]
}
```

Invoke 2종
필요 액션 전부

anthropic.*
리소스 스코프

리전 축소
와일드카드 대체 가능

Deny 조합
특정 모델 차단

Bedrock + Identity Center SSO

키 없는 인증의 전체 흐름

SSO FLOW

개발자는 브라우저 로그인만, 자격증명은 단기 토큰으로 자동 발급됩니다. 퇴사자는 IdP에서 끊는 순간 접근이 사라집니다.

STEP 1

IdP 로그인

aws sso login, 브라우저 인증

STEP 2

권한 매핑

그룹 -> Permission Set (IAM 정책)

STEP 3

단기 자격증명

STS 토큰 자동 발급, 갱신

STEP 4

Bedrock 호출

Claude Code가 SDK 체인으로 사용

PrivateLink와 VPC Endpoint

추론 트래픽의 프라이빗 경로

PRIVATE INFERENCE

Bedrock 호출을 인터넷 없이 VPC 내부에서 처리합니다.
폐쇄망 요건과 데이터 경로 통제 요구의 표준 답안입니다.

SETUP

엔드포인트

bedrock-runtime 인터페이스 엔드포인트 생성.

DNS

사설 해석

Private DNS 활성으로 SDK 무수정 전환.

CTRL

정책 결합

엔드포인트 정책으로 호출 가능 모델, 주체 제한.

AUDIT

경로 증명

VPC Flow Logs로 트래픽 경로 증적.

bedrock-runtime
대상 서비스

무수정
Private DNS 효과

EP 정책
추가 통제 계층

Flow Logs
경로 증적

API Key 분배의 위험과 대안

정적 키를 사람 손에 주지 않는다

NO STATIC KEYS

키 분배는 유출, 이직, 회전의 3중 부담을 만듭니다.

사람에게는 SSO, 기계에는 볼트와 헬퍼가 원칙입니다.

정적 키 분배의 부담

- 커밋, 로그, 채팅으로 유출 경로 다수
- 퇴사 시 회수 불가, 전수 회전 필요
- 누가 썼는지 귀속 불가
- 감사 대응 시 증적 공백

대안 체계

- 사람: SSO 단기 토큰 (Bedrock, Entra)
- 기계: Secrets Manager + apiKeyHelper
- CI: OIDC 페더레이션, setup-token 최소화
- 환경변수 하드코딩 전면 금지 정책

자격증명 의사결정

주체별 표준 경로 확정

개발자 (일상)	Bedrock + Identity Center SSO	정적 키 0
CI 파이프라인	OIDC 역할 인수, 필요 시 setup-token	장기 시크릿 최소화
공유 서비스	Secrets Manager + apiKeyHelper	TTL 회전 자동화
폐쇄망	PrivateLink + SSO	경로와 인증 동시 해결
예외 승인	기간 한정, 사유 기록, 자동 만료	감사 대장 유지

CHAPTER 03 / PART 03

Claude apps gateway

자체 호스팅 SSO 게이트웨이

- 왜 게이트웨이인가: 클라우드 경로의 빈 조각
- gateway.yaml 아키텍처와 SSO 사인인
- 그룹별 모델 라우팅과 지출 한도 API
- 배포, 운영, OTLP 텔레메트리

왜 게이트웨이인가

클라우드 경로에 남던 빈 조각

THE MISSING PIECE

Bedrock 직결은 인증과 과금을 상속하지만, 조직 로그인 경험과 그룹별 모델 통제, 개인 지출 한도는 비어 있었습니다.
게이트웨이가 그 조각을 채웁니다.

Bedrock 직결에서 비던 것

- 개발자별 지출 한도 실시간 강제
- IdP 그룹별 모델 접근 차등
- 조직 표준 로그인 UX
- 요청 단위 중앙 텔레메트리

apps gateway가 채우는 것

- 일, 주, 월 한도를 Admin API로 설정, 즉시 강제
- 그룹 -> 모델 라우팅 규칙
- OIDC SSO 사인인 플로우
- OTLP로 전 요청 관측

apps gateway vs 일반 LLM gateway

무엇이 다른가

PURPOSE-BUILT

일반 LLM 게이트웨이는 범용 프록시, apps gateway는 Claude Code 조직 운영에 특화된 완제품입니다.

일반 LLM gateway (LiteLLM 등)

- 다중 벤더 추상화가 목적
- Claude Code 정책 모델 미인지
- 한도, 라우팅을 직접 구현
- 범용 프록시 유지보수 부담

Claude apps gateway

- Claude Code 전용 설계
- managed 정책과 자연 결합
- SSO, 라우팅, 한도가 내장 완제품
- 공식 배포, 운영 문서 제공

CHAPTER 03 / PART 04

거버넌스와 정책

개인이 못 바꾸는 것을 설계한다

- managed settings 4채널과 우선순위
- 병합 규칙과 잠금 키
- MCP, 마켓플레이스, 훅 통제
- 조직 CLAUDE.md와 auto-mode 구성

managed settings 4채널

정책이 기기에 도달하는 네 경로 (우선순위순)

1 Server-managed	claude.ai 어드민 콘솔에서 배포	전 플랫폼, Teams/Ent 전용
2 plist / HKLM	com.anthropic.claudecode, HKLM₩...₩ClaudeCode	macOS, Windows, 변조 저항
3 파일 기반	/etc/claude-code/managed-settings.json 등	전 플랫폼, 형상 관리 배포
4 HKCU	사용자 레지스트리, 권한 상승 불필요	Windows, 편의용 (비강제)
규칙	기기에서 처음 발견되는 채널 하나 사용	/status가 소스 표기

managed-settings.json 예시

조직 기준선 정책 한 장

/etc/claude-code/managed-settings.json

```
{
  "permissions": {
    "deny": ["Read(./.env*)", "Bash(rm -rf:*)",
             "Bash(curl * | bash:*)"],
    "disableBypassPermissionsMode": "disable"
  },
  "sandbox": { "enabled": true, "network":
    { "allowedDomains": ["api.anthropic.com", "*.corp.example"] } },
  "allowedMcpServers": ["github", "corp-wiki"],
  "minimumVersion": "2.1.190",
  "env": { "CLAUDE_CODE_USE_BEDROCK": "1" }
}
```

deny 핵심
위험 최소 집합

bypass 차단
우회 모드 봉쇄

sandbox
도메인 통제 결합

env 강제
공급자 스위치

잠금 키 일람

우회로를 닫는 스위치들

allowManagedPermissionRulesOnly	managed 규칙만 유효, 개인 allow 무시	고통제 환경용
disableBypassPermissionsMode	--dangerously-skip 우회 봉쇄	전사 기본 권장
allowManagedHooksOnly	managed 훅만 로드	훅 주입 차단
allowedHttpHookUrls	HTTP 훅 목적지 허용 목록	웹훅 유출 통제
strict/blockedMarketplaces	플러그인 마켓 소스 통제	공급망 방어
minimumVersion	구버전 강제 상향	보안 픽스 하한

적용 검증 / /status

정책이 정말 걸렸는지 한 줄로

~/proj $ claude

```
> /status

# 출력 중 확인 라인
Enterprise managed settings (file)
#  괄호 안 소스: remote ¦ plist ¦ HKLM ¦ HKCU ¦ file

# 함께 확인
#  Provider: Bedrock (env 강제 반영 여부)
#  Model: 조직 기본 모델
#  Sandbox: enabled

# 파일럿 기기 전수에서 소스 표기 스크린샷 수집
```

한 줄 검증
managed 표기

5 소스
괄호 안 채널명

전수 확인
링별 검증 절차

Lab 1
실습 판정 기준

세션과 오프보딩

떠나는 순간 접근도 끝난다

OFFBOARDING BY DESIGN

SSO 구조에서는 IdP 비활성화가 곧 접근 종료입니다. 남는 것은 세션 수명과 잔여 토큰의 시간창뿐이며, 그 창을 짧게 설계합니다.

CUT

원천 절단

IdP 계정 비활성 즉시 신규 인증 불가.

TTL

세션 수명

SSO 세션과 STS 만료를 업무 리듬에 맞게 단축.

SWEEP

잔여 정리

setup-token 등 장기 토큰 발급 대장 기반 회수.

PROOF

증적

비활성 시각과 마지막 호출 시각의 대조 리포트.

IdP 절단
단일 지점

짧은 TTL
잔여 창 축소

토큰 대장
예외 관리

대조 리포트
감사 증적

데이터 정책

감사관의 첫 질문에 대한 답

학습 사용	Team, Enterprise, API, 클라우드 경로 미사용	공식 데이터 정책
보존	경로별 retention, 공급자 정책 상속	data-usage 문서
ZDR	요청 완료 후 무저장, Enterprise 제공	zero-data-retention
요청 단위 감사	필요 시 LLM gateway로 중앙 로깅	민감도 라우팅 포함
로컬 데이터	세션 트랜스크립트 30일 기본 정리	cleanupPeriodDays

CHAPTER 03 / PART 05

모니터링과 비용

쓰임을 보고, 비용을 귀속시키기

- 관측 3축: OTel, Analytics, Costs
- 메트릭 카탈로그와 SIEM 연계
- 사용자별 비용 귀속과 예산 알람
- 최적화 전략과 이상 감지

관측 3축

무엇을 보고 싶은가로 고른다 (admin-setup)

Usage monitoring	OTel로 세션, 도구, 토큰 송출	전 공급자 지원
Analytics 대시보드	사용자별 지표, 기여 추적	Anthropic 경로 전용
Cost tracking	지출 한도, 사용 귀속	Anthropic 경로 전용
클라우드 경로	Cost Explorer, GCP Billing, Azure CM	과금 데이터 직접 활용
대시보드 주소	claude.ai/analytics/claude-code	Teams, Enterprise 포함

사용자별 비용 귀속

누가 얼마나 쓰는가

ATTRIBUTION

OTel의 사용자 차원과 Bedrock 호출 신원을 결합하면 개인, 팀 단위 귀속이 완성됩니다.
게이트웨이 경유라면 이미 태그가 붙어 옵니다.

OTEL

클라이언트 차원

user, 부서 리소스 속성으로 토큰, 비용 절단.

AWS

호출 신원

SSO 역할 세션 이름이 CloudTrail에 사용자 단서로.

GW

게이트웨이

apps gateway는 사용자, 그룹 태그를 기본 제공.

REPORT

월간 절단

부서, 팀, 개인 3단 리포트를 자동 생성.

user.id
OTel 차원

role session
CloudTrail 단서

기본 태그
gateway 경유 시

3단 절단
부서, 팀, 개인

예산 알람

AWS Budgets로 초과 전에 알기

aws budgets (cli)

```
$ aws budgets create-budget \
  --account-id 111122223333 \
  --budget '{ "BudgetName": "claude-code-monthly",
    "BudgetLimit": {"Amount":"5000","Unit":"USD"},
    "TimeUnit": "MONTHLY", "BudgetType": "COST",
    "CostFilters": {"Service":["Amazon Bedrock"]} }' \
  --notifications-with-subscribers '[{
    "Notification": {"NotificationType":"ACTUAL",
      "ComparisonOperator":"GREATER_THAN","Threshold":80},
    "Subscribers": [{"SubscriptionType":"SNS",
      "Address":"arn:aws:sns:...:cost-alerts"}] }]'
```

80%
1차 경보 문턱

Bedrock 필터
서비스 한정

SNS
채널 연동 지점

FORECASTED
예측 경보 병행

Recap

모니터링과 비용 핵심 정리

KEY TAKEAWAY

OTel을 관리 설정으로 전사 배포하고, 태그로 귀속시키며, 예산 알람과 이상 감지로 루프를 닫습니다.

숫자가 정책의 다음 판을 결정합니다.

핵심 정리

- 3축: OTel, Analytics, Costs
- managed env로 전사 계측
- 귀속: 속성 태그 + 역할 세션
- Budgets 80% 경보 + 예측
- 분기 리뷰 5축 템플릿

참고: AWS에서의 Claude 구매 3가지 옵션

데이터 처리자와 네트워크 경계 기준

Amazon Bedrock	AWS가 데이터 처리자, 리전 내 처리	규제, 데이터 잔류 요건
Claude Platform on AWS	Anthropic 처리, IAM 인증, CCU 종량	최신 기능 Day-1
Claude Enterprise	전 직원 데스크톱 구독, SAML/SCIM	전사 배포
공통	Marketplace 과금, 같은 계정 병행 가능	PrivateLink는 Bedrock만