aws

사용 설명서

# AWS의 Claude 플랫폼

# AWS의 Claude 플랫폼: 사용 설명서

# Table of Contents

# AWS의 Claude 플랫폼이란 무엇입니까?

Anthropic 관리형 인프라를 사용하여 AWS를 통해 Claude의 전체 플랫폼 기능에 액세스합니다.

AWS의 Claude Platform은 AWS 계정을 통해 액세스할 수 있는 Messages API, Agent Skills, 코드 실행 및 베타 기능을 비롯한 전체 Anthropic 플랫폼 환경을 제공합니다. AWS가 추론 스택을 운영하는 Amazon Bedrock과 달리 Anthropic은 AWS에서 Claude 플랫폼을 운영합니다. AWS는 AWS Marketplace를 통해 인증 계층(SigV4 또는 API 키), IAM 기반 액세스 제어 및 결제 통합을 제공합니다.

> **Note**
>
> Anthropic SDKs는 AWS에서 Claude 플랫폼을 지원합니다. 언어별 클라이언트 가용성은 Anthropic Client SDKs.

## 플랫폼 통합 작동 방식

Claude 모델은 Anthropic 관리형 인프라에서 실행됩니다. AWS를 통한 결제 및 액세스를 위한 상용 통합입니다. Anthropic과 AWS 모두 독립 데이터 프로세서 역할을 합니다. AWS 서비스 약관은 AWS의 Claude 플랫폼을 관리합니다. Anthropic의 상용 서비스 약관, 데이터 처리 부록 및 사용 정책도 적용됩니다.

다음 운영 특성에 유의하세요. 데이터는 AWS에 상주하지 않을 수 있습니다. 추론은 Anthropic의 기본 클라우드로 라우팅될 수 있습니다. 하위 서비스는 예고 없이 변경될 수 있습니다. 추론을 특정 리전에 고정하려면 요청당 `inference_geo` 파라미터를 설정합니다. 자세한 내용은 데이터 레지던시를 참조하세요.

AWS의 Claude 플랫폼은 자사 Claude API와 동일한 데이터 보존 정책을 따릅니다. 요청 시 제로 데이터 보존(ZDR)을 사용할 수 있습니다. Anthropic 계정 담당자에게 문의하여 계정에 대해 활성화합니다.

## AWS의 Claude 플랫폼 기능

AWS의 Claude 플랫폼은 다음과 같은 기능을 제공합니다.

- 당일 모델 및 기능 액세스 - 새로운 Claude 모델 및 API 기능은 자사 Claude API에서 시작하는 당일에 사용할 수 있습니다.
- 에이전트 스킬 - 문서 생성(PowerPoint, Excel, Word, PDF)에 사전 구축된 스킬을 사용하고 사용자 지정 스킬을 생성합니다.

- 코드 실행 - Anthropic의 관리형 샌드박스에서 코드를 실행합니다.
- 확장 사고 - 복잡한 추론 작업에 대한 확장 사고를 활성화합니다.
- 스트리밍 및 배치 처리 - 실시간 SSE 스트리밍을 사용하거나 처리량이 많은 워크로드에 대한 배치 요청을 제출합니다.
- 프롬프트 캐싱 - 지연 시간과 비용을 줄이기 위한 캐시 도구, 시스템 프롬프트 및 메시지 기록입니다.
- 파일 API - 요청 간에 파일을 업로드하고 참조합니다.
- AWS IAM 통합 - 표준 IAM 정책 및 SigV4 인증을 사용하여 액세스를 제어합니다.
- 통합 AWS 결제 - 기존 AWS 계정을 통해 소비 기반 측정(Marketplace를 결제 백엔드로 사용)으로 결제합니다.

지원되는 기능의 전체 목록은 기능 지원을 참조하세요.

# 이 가이드의 구성 방식

이 안내서에서는 다음 주제를 다룹니다.

- 시작하기 - 계정 설정, 사전 조건, 인증 및 SDK 설치.
- API 사용 - 사용 가능한 모델, 요청 및 기능 지원.
- 환경 관리 - 데이터 레지던시, 워크스페이스, Claude 콘솔, 요금 제한 및 청구.
- 모니터링 및 작업 - CloudTrail 로깅 및 요청 ID 추적.
- 마이그레이션 - Amazon Bedrock에서 AWS의 Claude 플랫폼으로 이동합니다.
- 보안 및 액세스 제어 - IAM 정책 및 작업 참조입니다.

# AWS 기반 Claude 플랫폼과 Amazon Bedrock 비교

두 제품 모두 AWS를 통해 Claude를 사용할 수 있지만 아키텍처, API 표면 및 기능 가용성이 크게 다릅니다.

| | AWS의 Claude 플랫폼 | Amazon Bedrock |
|---|---|---|
| 스택을 운영하는 사람 | Anthropic | AWS |
| API 표면 | Anthropic Messages API(`/v1/messages`) | Bedrock API(Converse/InvokeModel) |
| 기능 가용성 | Claude API와 같은 날 | AWS 통합 타임라인에 따라 다름 |
| 에이전트 기술 | Available | 사용할 수 없음(코드 실행 필요) |
| 베타 기능 | `anthropic-beta` 헤더를 사용하여 전달(기능 지원 참조) | AWS 지원 필요 |
| Authentication | AWS IAM/SigV4 또는 API 키 | AWS IAM/SigV4 또는 보유자 토큰(C#, Go 및 Java SDKs만 해당) |
| 기본 URL | `aws-external-anthropic.{region}.api.aws` | `bedrock-runtime.{region}.amazonaws.com` |
| SDK 클라이언트 | 베타 형식의 플랫폼별 클라이언트 클래스(예: `AnthropicAWS` Python) | `AnthropicBedrock` / Bedrock SDK |
| 콘솔 | Claude 콘솔(`platform.claude.com` , AWS 콘솔을 통한 액세스) | Bedrock 콘솔 |
| 속도 제한 및 할당량 | Anthropic에서 관리 | AWS에서 관리 |
| 데이터 프로세서 | Anthropic 및 AWS | AWS |

Bedrock API와 함께 AWS에서 운영하는 Claude가 필요한 경우 Amazon Bedrock을 참조하세요.

> **Important**
>
> Anthropic은 AWS의 Claude 플랫폼을 타사 제품으로 제공합니다. 표준 AWS 규정 준수 프로그램, 인증 또는 감사 보고서(예: SOC, ISO 또는 HIPAA 자격)에는 적용되지 않습니다. 고객은 자체 실사를 수행하여이 타사 제품이 규제, 법률 및 규정 준수 요구 사항을 충족하는지 확인할 전적인 책임이 있습니다.

Bedrock을 선택하는 경우: 조직에 AWS에서 운영하는 추론, AWS를 유일한 데이터 프로세서로 사용하거나 AWS 규정 준수 프로그램(FedRAMP High, IL4, IL5, SOC, ISO 및 HIPAA 자격 포함)에 따른 적용 범위가 필요한 경우 Amazon Bedrock을 선택합니다. Bedrock은 AWS를 운영 당사자로 하는 AWS 제어 인프라에서 완전히 실행됩니다.

# 계정 설정

AWS에서 Claude 플랫폼 설정에는 AWS 콘솔에 가입하고, Anthropic 조직을 설정하고, 워크스페이스 ID를 기록하고, Claude 콘솔에 로그인하는 4단계가 있습니다.

> **Note**
>
> AWS 콘솔을 통해 가입하면 AWS 계정에 연결된 새 Anthropic 조직이 프로비저닝됩니다. 이 조직은 AWS Marketplace를 통해 조달된 Claude Enterprise 조직을 포함하여 Anthropic을 통해 회사가 보유한 기존 조직과는 별개입니다. 자사 Anthropic 조직의 API 키, 워크스페이스 및 Claude 콘솔 설정은 이월되지 않습니다.
> 기존 Amazon Bedrock 비공개 제안이 있는 경우 가입하기 전에 Anthropic 또는 AWS 계정 관리자에게 문의하여 첫 요청부터 할인이 적용되도록 하세요. 비공개 제안이 수락되기 전에 발생한 사용량에는 소급하여 할인을 적용할 수 없습니다. 비공개 제안을 참조하세요.

## 1단계: AWS 콘솔에 가입

1. AWS 콘솔을 열고 AWS의 Claude 플랫폼 서비스 페이지로 이동합니다.
2. 가입을 선택합니다.
3. 계속을 선택합니다.

페이지에는 진행 중인 가입 배너가 표시됩니다. 페이지에 머무르십시오. AWS가 AWS Marketplace 구독을 처리하는 동안 가입하는 데 몇 분 정도 걸리고 자동으로 리디렉션됩니다.

조직에 Anthropic의 비공개 제안이 있는 경우 콘솔이 이를 조회하고 AWS Marketplace에서 수락하라는 메시지를 표시합니다. 자세한 내용은 비공개 제안을 참조하세요.

> **Note**
>
> AWS에서 Claude 플랫폼을 사용하는 경우 콘텐츠(예: 프롬프트 및 완료)는 Anthropic에서 AWS 외부에서 처리됩니다. 콘텐츠 및 메타데이터의 처리 및 저장 방법에 대한 자세한 내용은 Anthropic의 데이터 사용 정책을 참조하세요.

# 2단계: Anthropic 조직 설정

가입이 완료되면 로 리디렉션됩니다`platform.claude.com/partner-signup`.

1. 조직 소유자의 이메일 주소를 입력하고 시작하기를 선택합니다.
2. 이메일 받은 편지함에서 설정 링크를 확인하고 따르세요. 브라우저에 다른 계정으로 로그인된 페이지가 표시되면 로그아웃 후 계속을 선택합니다.
3. 조직 세부 정보 양식(조직 이름, 엔터티 유형, 국가, 용도)을 작성하고 설정 완료를 선택합니다.

설정을 완료하면 Anthropic 조직이 생성됩니다. AWS 서비스 약관은 AWS의 Claude 플랫폼을 관리합니다. Anthropic의 상용 서비스 약관, 데이터 처리 부록 및 사용 정책도 적용됩니다. 이제 AWS 콘솔 서비스 페이지에 홈, API 키, Quickstart 및 Workspace가 포함된 왼쪽 탐색이 표시됩니다.

# 3단계: 워크스페이스 ID 기록

가입하면 기본 워크스페이스가 자동으로 프로비저닝됩니다. 리전별로 추가 워크스페이스를 생성할 수 있습니다. 리전 바인딩, 워크스페이스 관리 및 IAM 리소스 크기 조정에 대한 자세한 내용은 워크스페이스를 참조하세요.

AWS 서비스 페이지의 AWS 콘솔 Claude 플랫폼 또는 Claude 콘솔의 Workspaces에서 워크스페이스 ID를 찾습니다(Claude 콘솔 사용 참조). Workspace IDs 형식 `wrkspc_` 뒤에 영숫자 식별자를 사용합니다.

# 4단계: Claude 콘솔에 로그인

Claude 콘솔에 대한 액세스는 AWS IAM을 통해 페더레이션됩니다.

1. `aws-external-anthropic:AssumeConsole` 권한이 있는 IAM 역할을 수임합니다. IAM 작업을 참조하세요.
2. AWS의 Claude 플랫폼 서비스 페이지에서 로그인을 선택합니다. AWS 콘솔은 JWT를 실행하고 사용자를 로 리디렉션합니다`platform.claude.com`.
3. 처음 로그인하면 이메일 주소를 입력하라는 메시지가 표시됩니다. 작업 이메일을 입력합니다. 플랫폼은 Claude 콘솔 사용자를 just-in-time.

AWS 콘솔을 통해 로그인하면 Claude 콘솔이 AWS의 Claude 플랫폼 조직으로 범위가 지정됩니다. AWS에서 관리하는 계정 표시기가 Claude 콘솔 사이드바에 나타납니다.

# 계정 설정 문제 해결

- “로그인 실패: OutboundWebIdentityFederation을 활성화하지 못했습니다.”: 처음 제출할 때이 빨간색 배너가 표시되면 계속을 다시 선택합니다. IAM 활성화가 전파되는 데 시간이 걸릴 수 있습니다.
- 가입 중 진행률 표시기 없음: 가입하는 데 몇 분 정도 걸립니다. 이 페이지에는 AWS가 계정을 프로비저닝하는 동안 진행률 표시줄이 없는 정적 진행 중 가입 배너가 표시됩니다.
- 설정 링크를 따른 후 "다른 계정으로 로그인": 로그아웃 및 계속을 선택합니다. 이 페이지에서는 입력한 이메일 주소로 다시 인증합니다.
- 로그인 중 "찾을 수 없음" 메시지:이 메시지는 리디렉션 중에 잠시 나타날 수 있습니다. 이를 무시할 수 있습니다.
- 사용 페이지에는 첫 번째 API 호출 후 데이터가 표시되지 않습니다. 사용 데이터가 Claude 콘솔에 표시되는 데 몇 분 정도 걸릴 수 있습니다.

# 사전 조건

API를 호출하기 전에 다음 사항을 확인해야 합니다.

1. AWS에서 Claude Platform을 구독하는 활성 AWS 계정(계정 설정 참조).
2. AWS CLI가 설치 및 구성되었습니다.
3. AWS 계정에서 활성화된 아웃바운드 웹 자격 증명 페더레이션(일회성 설정 단계, 아웃바운드 웹 자격 증명 페더레이션 활성화 참조).
4. 워크스페이스 ID(워크스페이스 ID 가져오기 참조).

## 아웃바운드 웹 ID 페더레이션 활성화

AWS 게이트웨이의 Claude 플랫폼은 `sts:GetWebIdentityToken` 서버 측을 호출하여 Anthropic에 전달하는 JWT를 민트합니다. 이 STS 기능은 모든 AWS 계정에서 기본적으로 비활성화되어 있습니다. 계정당 한 번 활성화:

```
aws iam enable-outbound-web-identity-federation
```

응답이 인 경우 계정`[ERROR] (FeatureEnabled) … already enabled` 의 설정이 이미 켜져 있으므로 계속 진행할 수 있습니다. 계정의 발급자 URL을 확인하고 검색합니다.

```
aws iam get-outbound-web-identity-federation-info
```

> **Warning**
>
> 이 단계가 없으면 모든 요청이를 반환합니다`"Outbound web identity federation is disabled for your account"`. 가장 일반적인 설정 오류입니다.

## 워크스페이스 ID 가져오기

기본 워크스페이스는 가입 시 각 리전에서 자동으로 프로비저닝됩니다(계정 설정 참조). Workspace는 단일 AWS 리전에 바인딩됩니다. 워크스페이스 ID는 Claude 콘솔의 워크스페이스 또는 AWS 콘솔 서비스 페이지의 워크스페이스 섹션에서 찾을 수 있습니다. 리전 바인딩 및 IAM 리소스 크기 조정은 Workspaces를 참조하세요.

SDK 클라이언트가 자동으로 읽도록 ANTHROPIC_AWS_WORKSPACE_ID 및 AWS_REGION 환경 변수를 설정합니다.

```
export ANTHROPIC_AWS_WORKSPACE_ID='wrkspc_01AbCdEf23GhIj'
export AWS_REGION='us-west-2'
```

리전은 필수입니다. 리전이 설정되지 않은 경우 SDK 클라이언트에서가 발생합니다. 생성자에 aws_region/awsRegion를 전달하거나 AWS_REGION (또는 )를 설정합니다AWS_DEFAULT_REGION. 모든 AWS 상용 리전이 지원됩니다. 옵트인 리전에서는 먼저 해당 리전에 계정을 옵트인해야 합니다.

# Authentication

AWS의 Claude Platform은 SigV4 요청 서명(기본)을 사용하는 AWS IAM과 API 키 인증이라는 두 가지 인증 방법을 지원합니다. 둘 다 동일한 기본 URL과 요청 형식을 사용합니다.

Anthropic SDKs(SDK 설치 참조)는 아래 설명된 인증 흐름 및 자격 증명 확인을 구현합니다. Anthropic SDK를 사용하지 않는 경우 리전 엔드포인트에 대해 SigV4-signed 요청을 구성(또는 API 키를 보유자 토큰으로 제시)해야 합니다`https://aws-external-anthropic.<region>.api.aws`. SDK별 클라이언트 구성 및 신뢰할 수 있는 자격 증명 해결 순서는 Anthropic 설명서의 Claude on AWS 설정 가이드를 참조하세요.

## SigV4 인증

SigV4는 엔터프라이즈 네이티브 경로이며 기존 AWS IAM 정책, 역할 및 감사와 통합됩니다. AWS 기본 자격 증명 공급자 체인에서 지원하는 모든 방법을 사용하여 AWS 자격 증명을 구성합니다.

- 환경 변수(`AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, `AWS_SESSION_TOKEN`)
- 공유 자격 증명 파일(`~/.aws/credentials`)
- SSO 및를 포함한 공유 구성 파일(`~/.aws/config`) `credential_process`
- IRSA 및 GitHub Actions용 웹 자격 증명(`AWS_WEB_IDENTITY_TOKEN_FILE` 및 `AWS_ROLE_ARN`)
- ECS 컨테이너 자격 증명
- EC2 인스턴스 메타데이터 서비스(IMDS)

Anthropic SDK가 자격 증명을 선택하고 올바른 리전 엔드포인트로 확인되는지 확인하려면 요청하기의 예제에 따라 테스트 요청을 수행합니다. 응답이 성공하면 자격 증명, 리전, 기본 URL 및 워크스페이스 ID가 모두 올바르게 구성되었음을 확인합니다.

## API 키 인증

더 간단한 통합 경로(로컬 개발 및 스크립트)를 위해 SigV4 대신 API 키로 인증할 수 있습니다. `ANTHROPIC_AWS_API_KEY` 환경 변수를 설정하거나 `apiKey` SDK 생성자에 전달합니다. Anthropic SDK는 키를 보유자 토큰으로 AWS 엔드포인트의 Claude 플랫폼에 전송합니다. IAM은 `aws-external-anthropic:CallWithBearerToken` 작업을 통해 요청을 승인합니다(IAM 정책 참조).

AWS 콘솔의 Claude Platform on AWS → API 키에서 API 키를 생성합니다. 키 생성을 선택한 다음 키 값을 복사합니다. API 키 인증을 사용하도록 허용해야 하는 보안 주체에 `aws-external-anthropic:CallWithBearerToken` IAM 작업을 부여합니다.

> **Note**
>
> AWS의 Claude 플랫폼 아래의 AWS 콘솔에서 생성된 API 키만이 서비스에서 작동합니다.
>
> - 자사 API 액세스를 위해 표준 Claude 콘솔에서 생성된 키는 AWS의 Claude 플랫폼 엔드포인트에서 작동하지 않습니다.
> - Amazon Bedrock API 키도 작동하지 않음 - Bedrock은 별도의 엔드포인트, 인증 흐름 및 IAM 네임스페이스를 사용합니다.

## 단기 API 키

API 키 인증을 원하지만 수명이 긴 키의 수명이 없는 경우 Anthropic은 AWS IAM 자격 증명에서 단기 API 키를 민트하는 토큰 생성자 라이브러리를 게시합니다. 토큰은 기본적으로 수명 주기가 12시간이며 특정 워크스페이스 및 `aws-external-anthropic:CallWithBearerToken` 작업으로 범위가 지정될 수 있습니다. 언어에 맞는 생성기를 설치하고, IAM 자격 증명을 API 키와 교환하고, 키를 Anthropic SDK에 전달합니다.

- Python: aws/token-generator-for-aws-external-anthropic-python
- JavaScript / TypeScript: aws/token-generator-for-aws-external-anthropic-js
- Java: aws/token-generator-for-aws-external-anthropic-java

단기 API 키는 여전히 호출자의 IAM 보안 주체가 대상 작업 영역을 보유해야 `aws-external-anthropic:CallWithBearerToken` 합니다. 자체적으로 만료되므로 명시적으로 교체하거나 취소할 필요가 없습니다.

## 자격 증명 우선 순위 및 리전 확인

Anthropic SDK의 Claude Platform on AWS 클라이언트는 정의된 우선 순위 순서를 사용하여 자격 증명과 리전을 확인합니다. 인수 이름은 각 언어의 규칙을 따릅니다. 즉, TypeScript 및 PHP용 camelCase, Python 및 Ruby용 snake_case, Go용 PascalCase, C# 및 Java용 속성 또는 빌더 패턴입니다.

각 SDK에 대한 신뢰할 수 있는 우선 순위 순서, 지원되는 환경 변수 및 생성자 인수는 Anthropic 설명서의 Claude on AWS setup을 참조하세요. 일반적으로 명시적 생성자 인수는 환경 변수보다 우선하며 기본 AWS 자격 증명 공급자 체인보다 `ANTHROPIC_AWS_API_KEY` 우선합니다. 리전이 필요합니다`AnthropicBedrock`. (로 대체됨`us-east-1`)과 달리 생성자 또는 `AWS_REGION` /에서 리전을 제공하지 않으면 AWS 클라이언트에서 Claude가 발생합니다`AWS_DEFAULT_REGION`.

## WorkSpace ID

모든 데이터 영역 요청은 `anthropic-workspace-id` 헤더에 워크스페이스 ID를 포함해야 합니다. Anthropic SDKs는 기본적으로 `ANTHROPIC_AWS_WORKSPACE_ID` 환경 변수에서 읽거나 `workspaceId` /`workspace_id`를 클라이언트 생성자에 전달할 수 있습니다. 기본 `Anthropic` 클라이언트(Claude-on-AWS 클라이언트 아님)를 사용하는 경우 헤더를 명시적으로 전달합니다. 워크스페이스 ID를 찾는 방법은 언어별 예제 요청 및 워크스페이스를 참조하세요.

# SDK 설치

Anthropic의 클라이언트 SDKs AWS에서 Claude 플랫폼을 지원합니다. 7개의 SDKs 제공합니다. 또는 AWS 기반 Claude 플랫폼 기본 URL을 사용하여 기본 클라이언트를 구성할 수 있습니다.

Python

```
pip install -U "anthropic[aws]"
```

> **Tip**
>
> Homebrew Python 또는 기타 외부 관리형 Python 환경이 있는 macOS에서는 PEP 668 `externally-managed-environment` 오류와 함께 실패할 `pip install` 수 있습니다. 먼저 가상 환경을 생성하고 활성화합니다`python3 -m venv .venv && source .venv/bin/activate`.

TypeScript

```
npm install @anthropic-ai/aws-sdk
```

C#

```
dotnet add package Anthropic.Aws
```

Go

```
go get github.com/anthropics/anthropic-sdk-go
```

Java

```
implementation("com.anthropic:anthropic-java-aws:2.27.0")
```

```
<dependency>
  <groupId>com.anthropic</groupId>
  <artifactId>anthropic-java-aws</artifactId>
  <version>2.27.0</version>
```

```
</dependency>
```

### PHP

```
composer require anthropic-ai/sdk aws/aws-sdk-php
```

### Ruby

```
gem install anthropic aws-sdk-core
```

> **Note**
>
> AWS의 Claude Platform용 SDK 클라이언트는 베타 버전입니다.

# 사용 가능한 모델

AWS의 Claude 플랫폼은 자사 Claude API와 동일한 Claude 모델 세트를 제공합니다.

모델, 모델 IDs, 컨텍스트 창 크기 및 기능의 현재 목록은 Anthropic 설명서의 모델 개요를 참조하세요.

## 모델 ID

AWS의 Claude 플랫폼 모델 IDs는 자사 Claude API에 사용되는 모델 ID와 동일합니다(예: , `claude-opus-4-6`claude-sonnet-4-6, `claude-haiku-4-5`). Bedrock 스타일 ARNs 또는 `anthropic.` 접두사는 없습니다. Anthropic이 게시하는 대로 모델 ID를 정확히 사용합니다.

# 요청 만들기

AWS의 Claude 플랫폼은 Claude 자사 플랫폼과 동일한 API 표면인 Anthropic Messages API(`/v1/messages`)를 사용합니다. 차이점은 기본 URL, 인증 방법 및 요청 대상이 되는 워크스페이스를 식별하는 필수 `anthropic-workspace-id` 헤더입니다.

쉘

```
curl "https://aws-external-anthropic.us-west-2.api.aws/v1/messages" \
  --aws-sigv4 "aws:amz:us-west-2:aws-external-anthropic" \
  --user "$AWS_ACCESS_KEY_ID:$AWS_SECRET_ACCESS_KEY" \
  -H "x-amz-security-token: $AWS_SESSION_TOKEN" \
  -H "content-type: application/json" \
  -H "anthropic-version: 2023-06-01" \
  -H "anthropic-workspace-id: $ANTHROPIC_AWS_WORKSPACE_ID" \
  -d '{
    "model": "claude-sonnet-4-6",
    "max_tokens": 1024,
    "messages": [
      {"role": "user", "content": "Hello!"}
    ]
  }'
```

Python

```
from anthropic import AnthropicAWS

client = AnthropicAWS()

message = client.messages.create(
    model="claude-sonnet-4-6",
    max_tokens=1024,
    messages=[{"role": "user", "content": "Hello!"}],
)
print(message)
```

TypeScript

```
import AnthropicAws from "@anthropic-ai/aws-sdk";

const client = new AnthropicAws();
```

```
const message = await client.messages.create({
  model: "claude-sonnet-4-6",
  max_tokens: 1024,
  messages: [{ role: "user", content: "Hello!" }]
});
console.log(message);
```

C#

```
using Anthropic;
using Anthropic.Aws;

var client = new AnthropicAwsClient();

var message = await client.Messages.Create(new()
{
    Model = "claude-sonnet-4-6",
    MaxTokens = 1024,
    Messages = [new() { Role = "user", Content = "Hello!" }]
});

Console.WriteLine(message);
```

Go

```
client, err := anthropicaws.NewClient(context.Background(),
 anthropicaws.ClientConfig{})
if err != nil {
    panic(err)
}

message, err := client.Messages.New(context.Background(), anthropic.MessageNewParams{
    Model:     anthropic.ModelClaudeSonnet4_6,
    MaxTokens: 1024,
    Messages: []anthropic.MessageParam{
        anthropic.NewUserMessage(anthropic.NewTextBlock("Hello!")),
    },
})
if err != nil {
    panic(err)
}
```

```
fmt.Println(message.Content[0].Text)
```

### Java

```
import com.anthropic.aws.backends.AwsBackend;
import com.anthropic.client.AnthropicClient;
import com.anthropic.client.okhttp.AnthropicOkHttpClient;
import com.anthropic.models.messages.Message;
import com.anthropic.models.messages.MessageCreateParams;
import com.anthropic.models.messages.Model;

AnthropicClient client = AnthropicOkHttpClient.builder()
    .backend(AwsBackend.fromEnv())
    .build();

Message message = client.messages().create(
    MessageCreateParams.builder()
        .model(Model.CLAUDE_SONNET_4_6)
        .maxTokens(1024)
        .addUserMessage("Hello!")
        .build()
);

IO.println(message);
```

### PHP

```
use Anthropic\Aws\Client;

$client = new Client();

$message = $client->messages->create(
    model: 'claude-sonnet-4-6',
    maxTokens: 1024,
    messages: [['role' => 'user', 'content' => 'Hello!']],
);

echo $message;
```

### Ruby

```
require "anthropic"
```

```
client = Anthropic::AWSClient.new

message = client.messages.create(
  model: "claude-sonnet-4-6",
  max_tokens: 1024,
  messages: [{ role: "user", content: "Hello!" }]
)

puts message
```

클라이언트는 환경에서 AWS_REGION (또는 AWS_DEFAULT_REGION) 및 ANTHROPIC_AWS_WORKSPACE_ID를 읽습니다. 생성자에 aws_region / awsRegion 또는 workspace_id /를 전달하여 재정의workspaceId할 수 있습니다. 리전 및 작업 영역 ID가 모두 필요합니다. 이러한 소스에서 둘 중 하나를 확인할 수 없는 경우 생성자가를 발생시킵니다.

> **Note**
>
> x-amz-security-token 헤더(curl)는 IAM 역할, SSO 또는 STS와 같은 임시 자격 증명에만 필요합니다. 장기 IAM 사용자 자격 증명을 사용할 때는 생략합니다. SDK 클라이언트는 자격 증명 소스를 기반으로 이를 자동으로 처리합니다.

--aws-sigv4 값은 형식을 따릅니다aws:amz:<region>:<service>. SigV4 서비스 이름은 이며 aws-external-anthropic리전은 엔드포인트 URL의 리전과 일치해야 합니다. 둘 중 하나가 일치하지 않으면 특정 진단이 아닌 일반적인 서명 거부 오류가 발생합니다.

## 기본 Anthropic 클라이언트 사용

AWS SDK 종속성을 추가하지 않으려면 기본 Anthropic 클라이언트를 사용할 수 있습니다. 이 경로는 전용 클라이언트가 읽는 이름 대신 바닐라 SDK 환경 변수 ANTHROPIC_AWS_* 이름을 사용합니다.

```
export ANTHROPIC_API_KEY='your-api-key'
export ANTHROPIC_BASE_URL='https://aws-external-anthropic.us-west-2.api.aws'
export ANTHROPIC_WORKSPACE_ID='your-workspace-id'
```

Python, TypeScript 및 Go SDKs는 ANTHROPIC_API_KEY 자동으로 읽고 ANTHROPIC_BASE_URL 읽습니다. 다른 언어의 경우 생성자에게 전달합니다. 모든 언어에서 anthropic-workspace-id 헤더에 워크스페이스 ID를 전달합니다.

### Python

```
import os
from anthropic import Anthropic

client = Anthropic(
    # ANTHROPIC_API_KEY and ANTHROPIC_BASE_URL are read automatically
    default_headers={"anthropic-workspace-id": os.environ["ANTHROPIC_WORKSPACE_ID"]},
)

message = client.messages.create(
    model="claude-sonnet-4-6",
    max_tokens=1024,
    messages=[{"role": "user", "content": "Hello!"}],
)
print(message.content[0].text)
```

### TypeScript

```
import Anthropic from "@anthropic-ai/sdk";

const client = new Anthropic({
  // ANTHROPIC_API_KEY and ANTHROPIC_BASE_URL are read automatically
  defaultHeaders: { "anthropic-workspace-id": process.env.ANTHROPIC_WORKSPACE_ID }
});

const message = await client.messages.create({
  model: "claude-sonnet-4-6",
  max_tokens: 1024,
  messages: [{ role: "user", content: "Hello!" }]
});
console.log(message.content);
```

# 기능 지원

AWS의 Claude 플랫폼은 Anthropic Messages API를 직접 사용합니다. 현재 사용할 수 없는 기능에 명시된 경우를 제외하고 자사 Claude API와 전체 기능 패리티를 얻을 수 있습니다.

- 베타 기능: Claude API와 마찬가지로 표준 `anthropic-beta` 헤더를 전달하여 베타 기능에 액세스합니다.
- 에이전트 스킬: Claude API와 동일한 `container.skills` 파라미터 및 베타 헤더로 사전 빌드된 사용자 지정 에이전트 스킬을 사용합니다. 사전 구축된 모든 기술(PowerPoint, Excel, Word, PDF)은 즉시 사용할 수 있습니다.
- 코드 실행: 코드 실행 도구를 사용하여 Anthropic의 관리형 샌드박스에서 코드를 실행합니다.
- 도구 사용: 컴퓨터 사용 및 기타 모든 도구 사용 기능을 사용할 수 있습니다.
- 확장 사고: Claude API와 동일한 파라미터로 확장 사고를 활성화합니다.
- 스트리밍: 실시간 응답에 대한 전체 SSE 스트리밍 지원.
- 배치 처리: 처리량이 많은 워크로드에 대한 배치 요청을 제출합니다.
- 프롬프트 캐싱: 지연 시간과 비용을 줄이기 위한 캐시 도구, 시스템 프롬프트 및 메시지 기록입니다. 모든 프롬프트 캐싱 기능(5분 TTL, 1시간 TTL, 자동 캐싱)을 사용할 수 있습니다.
- 파일 API: 요청 간에 파일을 업로드하고 참조합니다.
- IAM 통합이 포함된 Workspace 태그: Workspace에 태그를 지정할 수 있으며 태그는 속성 기반 액세스 제어를 위해 IAM으로, 비용 할당을 위해 AWS Cost and Usage Reports로 전달됩니다. Workspace 태깅은 AWS의 Claude 플랫폼에서 제공하는 GA 기능입니다(일방 Claude API의 베타 기능).

## Claude Managed Agents

Claude Managed Agents는 AWS의 Claude 플랫폼에서 사용할 수 있습니다. 에이전트, 세션, 환경, 자격 증명 볼트 및 메모리 스토어는 일류 IAM 리소스입니다. 작업 참조에 대한 IAM 작업과 해당 페이지에 대한 Claude 콘솔 사용을 참조하세요. Anthropic Agent SDK는 추가 통합 없이 AWS의 Claude 플랫폼에서 작동합니다. AWS 기반 Claude 플랫폼 기본 URL을 가리키고 SigV4 또는 API 키로 인증합니다.

AWS의 Claude 플랫폼에서의 자율 세션은 6시간마다 재인증이 필요합니다. 장기 실행 에이전트 실행은 해당 기간 내에 SigV4 자격 증명 또는 API 키를 새로 고쳐야 합니다. 그렇지 않으면 세션이 종료됩니다.

다음 Claude Managed Agents 기능은 현재 AWS의 Claude 플랫폼에서 사용할 수 없습니다.

- 결과: 에이전트 세션에 대한 결과 추적을 사용할 수 없습니다.
- 다중 에이전트 세션: 여러 상호 작용 에이전트가 있는 세션을 사용할 수 없습니다.

# 규정 준수

> **Important**
>
> Anthropic은이 서비스를 타사 제품으로 제공합니다. 표준 AWS 규정 준수 프로그램, 인증 또는 감사 보고서(예: SOC, ISO 또는 HIPAA 자격)에는 적용되지 않습니다. 고객은 자체 실사를 수행하여이 타사 제품이 규제, 법률 및 규정 준수 요구 사항을 충족하는지 확인할 전적인 책임이 있습니다. 민감하거나 규제되는 데이터를 처리하기 전에 Anthropic Trust Center를 통해 Anthropic의 공식 규정 준수 설명서를 검토해야 합니다. 자세한 내용은 AWS 서비스 약관을 참조하세요.

# 워크스페이스 멤버십 모델링 방법

자사 Claude API에서 액세스는 users-to-workspaces 멤버십에 의해 관리됩니다. 사용자는 워크스페이스에 추가되고 워크스페이스의 액세스 권한을 상속받습니다. AWS의 Claude Platform은 해당 모델을 IAM 권한 부여로 대체합니다. 워크스페이스별 사용자 목록은 없습니다. 대신 IAM 보안 주체(사용자 또는 역할)는 특정 워크스페이스 ARNs에 대한 `aws-external-anthropic` 작업을 부여하는 정책을 보유합니다. IAM 정책 평가는 각 보안 주체가 각 워크스페이스에서 수행할 수 있는 작업을 결정합니다.

Claude 콘솔에서 워크스페이스별 멤버십을 관리하는 대신 IAM 정책(SCPs, 권한 경계, 자격 증명 연결 정책 또는 리소스 수준 조건)을 수정하여 워크스페이스 액세스 권한을 부여하고 취소합니다. 예제는 IAM 정책을 참조하세요.

# 현재 사용할 수 없는 기능

AWS의 Claude 플랫폼에서는 현재 다음 기능을 사용할 수 없습니다.

- HIPAA 준비: Anthropic의 HIPAA 지원 프로그램은 AWS의 Claude 플랫폼에서 사용할 수 없습니다. HIPAA 요구 사항이 있는 고객은 대신 Amazon Bedrock에서 Claude를 평가해야 합니다.
- 표준 및 배치 이외의 서비스 계층: Priority Tier를 사용할 수 없습니다.
- 관리자 API - 조직 구성원, 워크스페이스 구성원, 초대, API 키, 사용 보고서, 비용 보고서 및 속도 제한 보고서 엔드포인트: 이러한 관리자 API 엔드포인트는 현재 사용할 수 없습니다. Workspace

CRUD 및 엔드포인트 이름 바꾸기(`/v1/organizations/workspaces`)는 `aws-external-anthropic` 네임스페이스를 통해 사용할 수 있습니다. IAM 작업을 참조하세요. 멤버십은 워크스페이스 멤버십 목록이 아닌 AWS IAM을 통해 모델링됩니다(이전 섹션 참조). API 키 수명 주기는 AWS 콘솔의 Claude Platform on AWS → API 키에서 관리됩니다. 사용량, 비용 및 속도 제한 데이터의 경우 Claude 콘솔 보기(모니터링 및 로깅 참조) 또는 Anthropic Usage and Cost API를 사용합니다.

- 지출 한도: 사용할 수 없습니다. 대신 AWS 결제 제어에 의존합니다.
- Claude Code 워크스페이스 및 분석 API: 자동 속도 제한이 있는 Claude Code 워크스페이스를 사용할 수 없습니다. Claude Code 사용량은 전용 화면이 아닌 일반 사용량 보기에 표시됩니다.
- OAuth 인증: 지원되지 않습니다. SigV4 또는 API 키 인증을 사용합니다.
- OpenAI 호환 API 엔드포인트: AWS의 Claude 플랫폼에서는 사용할 수 없습니다.
- Workspace 수준 추론 지리 제어: `allowed_inference_geos` 및 `default_inference_geo`를 사용할 수 없습니다. 대신 각 요청에 `inference_geo`를 설정합니다.

# 데이터 레지던시

AWS의 Claude Platform은 다음과 같은 추론 지역을 지원합니다.

- 미국: 추론은 미국 데이터 센터 내에 있습니다. 1.1x 요금 승수가 적용됩니다.
- 글로벌: 추론은 Anthropic에서 운영하는 전 세계 데이터 센터로 라우팅될 수 있습니다. 표준 요금이 적용됩니다.

`inference_geo` 파라미터를 사용하여 요청당 추론 지오그래피를 설정합니다.

> **Note**
>
> `inference_geo` 파라미터는 Claude Opus 4.6, Claude Sonnet 4.6 이상 모델에서 지원됩니다. Claude Opus 4.5, Claude Sonnet 4.5 또는 Claude Haiku 4.5`inference_geo`에서를 사용하는 요청은 400 오류를 반환합니다. 모델 가용성 세부 정보는 데이터 레지던시를 참조하세요.

쉘

```
curl "https://aws-external-anthropic.us-west-2.api.aws/v1/messages" \
  --aws-sigv4 "aws:amz:us-west-2:aws-external-anthropic" \
  --user "$AWS_ACCESS_KEY_ID:$AWS_SECRET_ACCESS_KEY" \
  -H "x-amz-security-token: $AWS_SESSION_TOKEN" \
  -H "content-type: application/json" \
  -H "anthropic-version: 2023-06-01" \
  -H "anthropic-workspace-id: $ANTHROPIC_AWS_WORKSPACE_ID" \
  -d '{
    "model": "claude-sonnet-4-6",
    "max_tokens": 1024,
    "inference_geo": "us",
    "messages": [
      {"role": "user", "content": "Hello!"}
    ]
  }'
```

Python

```
from anthropic import AnthropicAWS
```

```
client = AnthropicAWS(aws_region="us-west-2")
message = client.messages.create(
    model="claude-sonnet-4-6",
    max_tokens=1024,
    inference_geo="us",
    messages=[{"role": "user", "content": "Hello!"}],
)
print(message)
```

### TypeScript

```
import AnthropicAws from "@anthropic-ai/aws-sdk";
const client = new AnthropicAws({ awsRegion: "us-west-2" });
const message = await client.messages.create({
  model: "claude-sonnet-4-6",
  max_tokens: 1024,
  inference_geo: "us",
  messages: [{ role: "user", content: "Hello!" }]
});
console.log(message);
```

### C#

```
using Anthropic;
using Anthropic.Aws;

var client = new AnthropicAwsClient(new() { AwsRegion = "us-west-2" });

var message = await client.Messages.Create(new()
{
    Model = "claude-sonnet-4-6",
    MaxTokens = 1024,
    InferenceGeo = "us",
    Messages = [new() { Role = "user", Content = "Hello!" }]
});

Console.WriteLine(message);
```

### Go

```
client, err := anthropicaws.NewClient(context.Background(),
 anthropicaws.ClientConfig{})
```

```
if err != nil {
    panic(err)
}

message, err := client.Messages.New(context.Background(), anthropic.MessageNewParams{
    Model:        anthropic.ModelClaudeSonnet4_6,
    MaxTokens:    1024,
    InferenceGeo: anthropic.String("us"),
    Messages: []anthropic.MessageParam{
        anthropic.NewUserMessage(anthropic.NewTextBlock("Hello!")),
    },
})
if err != nil {
    panic(err)
}

fmt.Println(message)
```

### Java

```
import com.anthropic.aws.backends.AwsBackend;
import com.anthropic.client.AnthropicClient;
import com.anthropic.client.okhttp.AnthropicOkHttpClient;
import com.anthropic.models.messages.Message;
import com.anthropic.models.messages.MessageCreateParams;
import com.anthropic.models.messages.Model;
import software.amazon.awssdk.regions.Region;

AnthropicClient client = AnthropicOkHttpClient.builder()
    .backend(AwsBackend.builder().region(Region.of("us-west-2")).build())
    .build();

Message message = client.messages().create(
    MessageCreateParams.builder()
        .model(Model.CLAUDE_SONNET_4_6)
        .maxTokens(1024)
        .inferenceGeo("us")
        .addUserMessage("Hello!")
        .build()
);

IO.println(message);
```

PHP

```
use Anthropic\Aws\Client;

$client = new Client(awsRegion: 'us-west-2');

$message = $client->messages->create(
    model: 'claude-sonnet-4-6',
    maxTokens: 1024,
    inferenceGeo: 'us',
    messages: [['role' => 'user', 'content' => 'Hello!']],
);

echo $message;
```

Ruby

```
require "anthropic"

client = Anthropic::AWSClient.new(aws_region: "us-west-2")

message = client.messages.create(
  model: "claude-sonnet-4-6",
  max_tokens: 1024,
  inference_geo: "us",
  messages: [{ role: "user", content: "Hello!" }]
)

puts message
```

`inference_geo`를 생략하면 요청의 기본값은 입니다`global`.

Workspace 수준 추론 지오그래피 제어(`allowed_inference_geos` 및 `default_inference_geo`)는 AWS의 Claude 플랫폼에서 사용할 수 없습니다. 대신 각 요청에 `inference_geo`를 설정합니다.

# 워크스페이스

AWS의 Claude 플랫폼에 대한 추론 및 리소스 요청은 워크스페이스를 대상으로 합니다. 이러한 API 호출의 `anthropic-workspace-id` 헤더에 워크스페이스의 ID를 전달합니다. Workspace IDs 태그가 지정된 형식 `wrkspc_` 뒤에 영숫자 식별자(예: `wrkspc_01AbCdEf23GhIj`)를 사용합니다. 워크스페이스 ID가 아직 없는 경우 획득을 참조하세요.

## Workspace 범위 지정

Workspace는 단일 AWS 리전에 바인딩됩니다. 에서 생성된 워크스페이스는 `us-west-2` 엔드포인트를 통해서만 액세스할 `us-west-2` 수 있습니다. 사용량, 할당량, 비용, 파일, 배치 및 스킬은 모두 워크스페이스별로 롤업되므로 Claude 콘솔에서 리전별로 분류할 수 있습니다.

Workspace는 AWS의 Claude Platform에 대한 기본 IAM 리소스 역할도 합니다. 워크스페이스 ARN을 사용하여 AWS IAM 정책을 통해 특정 워크스페이스에 대한 액세스 권한을 부여하거나 거부합니다. ARN의 리소스 세그먼트는 `anthropic-workspace-id` 헤더에 전달하는 `wrkspc_`접두사 ID와 동일합니다.

```
arn:aws:aws-external-anthropic:{region}:{account-id}:workspace/{workspace-id}
```

예: `arn:aws:aws-external-anthropic:us-west-2:123456789012:workspace/wrkspc_01AbCdEf23GhIj`

정책 예제는 IAM 정책을 참조하세요.

## 워크스페이스 생성

기본 워크스페이스는 가입 시 자동으로 프로비저닝됩니다. `/v1/organizations/workspaces` 엔드포인트를 통해 추가 워크스페이스를 생성, 업데이트, 이름 변경 및 보관할 수 있으며, 해당 `aws-external-anthropic` 작업(, `CreateWorkspace`, `UpdateWorkspaceArchiveWorkspace`)에 의해 승인됩니다. IAM 작업을 참조하세요.

## 클라이언트에서 워크스페이스 ID 설정

모든 데이터 영역 요청은 `anthropic-workspace-id` 헤더에 워크스페이스 ID를 포함해야 합니다. Anthropic SDK의 Claude-on-AWS 클라이언트를 사용하는 경우 다음 세 가지 방법으로 제공할 수 있습니다.

1. 환경 변수 -를 설정ANTHROPIC_AWS_WORKSPACE_ID하면 클라이언트가 자동으로 읽습니다.

```
export ANTHROPIC_AWS_WORKSPACE_ID='wrkspc_01AbCdEf23GhIj'
```

2. 생성자 인수 - 클라이언트를 인스턴스화할 때 workspaceId /workspace_id(이름은 SDK의 언어 규칙을 따릅니다)를 전달합니다.
3. 요청별 재정의 - 동일한 클라이언트에서 여러 워크스페이스를 처리해야 하는 경우 개별 요청에 직접 헤더를 전달합니다.

기본 Anthropic 클라이언트(AWS 백엔드 제외)를 사용하는 경우 모든 요청에 anthropic-workspace-id 대해 헤더를 명시적으로 설정합니다. 언어별 요청 예제를 참조하세요. SDK별 인수 이름은 Anthropic Client SDKs.

# 워크스페이스 태그 지정

Workspace 태그는 워크스페이스에 연결된 키-값 페어입니다. 속성 기반 액세스 제어를 위해 AWS IAM과 통합되고 비용 할당을 위해 AWS 비용 및 사용 보고서와 통합됩니다(CUR 세부 정보는 모니터링 및 로깅 참조). 태그 지정은 AWS의 Claude 플랫폼에서 GA입니다.

aws-external-anthropic 네임스페이스에서 TagResource 및를 사용하여 기존 워크스페이스UntagResource의 태그를 업데이트합니다. 동일한 태그 키는 추가 구성 없이 IAM 조건 및 비용 할당을 구동할 수 있습니다.

## 태그 기반 액세스 제어

aws:ResourceTag/<key> 조건 컨텍스트 키를 사용하여 워크스페이스 태그 값에 대한 aws-external-anthropic 작업을 게이트할 수 있습니다. 다음 정책은 태그가 지정된 워크스페이스에서만 추론을 허용합니다. Environment=production

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-external-anthropic:CreateInference",
        "aws-external-anthropic:CreateBatchInference",
        "aws-external-anthropic:CountTokens"
      ],
```

```
      "Resource": "arn:aws:aws-external-anthropic:*:*:workspace/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Environment": "production"
        }
      }
    }
  ]
}
```

의 aws:RequestTag/<key> 및 aws:TagKeys 조건 키를 사용하여 태그 지정 정책을 TagResource 적용합니다(예: 워크스페이스에 CostCenter 및 Owner 태그가 있어야 함). 자세한 예는 IAM 정책을 참조하세요.

# Claude 콘솔 사용

AWS의 Claude 플랫폼은 platform.claude.com 표준 Claude 콘솔을 사용합니다. AWS 콘솔에서 로그인하면 AWS에서 관리하는 계정 표시기가 Claude 콘솔 사이드바에 나타납니다. 콘솔은 AWS의 Claude 플랫폼 조직으로 범위가 지정됩니다. 파일, 에이전트 기술, 배치 작업, 에이전트, 세션, 환경, 자격 증명 볼트 및 메모리 스토어를 관리하기 위한 사용 분석, 비용 분석, 속도 제한 가시성, 워크스페이스 가시성 및 페이지를 제공합니다.

## 로그인

Claude 콘솔에 대한 액세스는 AWS IAM을 통해 페더레이션됩니다. 전체 최초 로그인 흐름은 계정 설정을 참조하세요. 간단히 설명하면 다음과 같습니다.

1. `aws-external-anthropic:AssumeConsole` 권한이 있는 IAM 역할을 수임합니다. IAM 작업을 참조하세요.
2. AWS 콘솔에서 AWS의 Claude 플랫폼 페이지로 이동합니다.
3. Claude 콘솔 열기를 선택합니다. AWS 콘솔은 JWT를 실행하고 사용자를 로 리디렉션합니다`platform.claude.com`.
4. 처음 로그인할 때 이메일 주소를 입력하라는 메시지가 표시되고 작업 이메일을 입력합니다. 플랫폼은 Claude 콘솔 사용자를 just-in-time.

관리자 및 개발자라는 두 가지 Claude 콘솔 역할을 사용할 수 있습니다. 관리자 역할은 AWS의 Claude 플랫폼에 사용할 수 있는 모든 Claude 콘솔 페이지 및 설정에 대한 액세스 권한을 부여합니다. 개발자 역할은 사용량, 비용, 속도 제한 및 워크스페이스 정보에 대한 읽기 액세스 권한을 부여합니다. Anthropic 계정 담당자에게 문의하여 보안 주체에 관리자 또는 개발자 역할을 할당합니다.

## 사용 가능한 페이지

AWS 게이트웨이를 통해 열은 페이지가 AWS 게이트웨이를 통해 데이터를 읽고 쓰는지(따라서 IAM 작업에 의해 관리되는지) 여부를 나타냅니다. Anthropic APIs를 통해 조직 수준 메타데이터를 직접 읽고 IAM 작업 확인을 우회하지 않음으로 표시된 페이지.

| Page | Available | AWS 게이트웨이를 통해 | 참고 |
|---|---|---|---|
| 사용량 | 예 | 아니요 | 모델, 워크스페이스 및 차원별로 토큰 사용량을 확인합니다. 요청 후 데이터가 표시되는 데 몇 분 정도 걸릴 수 있습니다. |
| 비용 | 예 | 아니요 | 모델 및 워크스페이스별로 비용 내역을 봅니다. AWS Cost Explorer는 집계된 CCU 항목을 표시합니다. |
| Limits | 예 | 아니요 | 속도 제한을 봅니다(읽기 전용). |
| 워크스페이스 | 예 | 아니요 | 리전별 워크스페이스 보기(읽기 전용). |
| 파일 | 예 | 예 | 업로드된 파일을 보고 관리합니다. |
| 기술 | 예 | 예 | 에이전트 스킬을 보고 관리합니다. |
| 배치 | 예 | 예 | 배치 처리 작업을 보고 관리합니다. |
| 에이전트 | 예 | 예 | 에이전트 정의를 보고 관리합니다. |
| 세션 | 예 | 예 | 에이전트 세션 및 이벤트 기록을 봅니다. |
| 환경 | 예 | 예 | 세션에 대한 클라우드 컨테이너 구성을 보고 관리합니다. |
| 자격 증명 볼트 | 예 | 예 | 세션 인증을 위한 자격 증명 저장소를 보고 관리합니다. |
| 메모리 스토어 | 예 | 예 | 영구 에이전트 메모리를 보고 관리합니다. |
| API 키 | 아니요 | 해당 사항 없음 | AWS 콘솔에서 API 키를 관리합니다(AWS의 Claude 플랫폼 → API 키). 인증을 참조하세요. |
| 회원 | 아니요 | 해당 사항 없음 | 해당 사항 없음. AWS IAM은 액세스를 관리합니다. |

| Page | Available | AWS 게이트웨이를 통해 | 참고 |
|---|---|---|---|
| 결제 | 아니요 | 해당 사항 없음 | 해당 사항 없음. AWS Marketplace는 결제 및 인보이스 발행을 관리합니다. 비용 페이지에서 비용 내역을 봅니다. |
| Claude 코드 | 아니요 | 해당 사항 없음 | 사용량 페이지에서 Claude 코드 사용량을 봅니다. |

# 조직 전환

Claude 콘솔은 AWS의 Claude 플랫폼에 대한 조직 전환을 지원하지 않습니다. 다른 조직에 액세스하려면 해당 조직의 AWS 계정에 대한 IAM 역할을 사용하여 AWS 콘솔을 통해 로그아웃하고 다시 인증합니다.

# 속도 제한 및 할당량

구독 시 AWS의 Claude 플랫폼은 티어 1 속도 제한을 할당합니다. Anthropic은 AWS 할당량 시스템이 아닌 속도 제한을 직접 관리합니다.

## 기본 제한

AWS의 Claude 플랫폼은 자사 Claude API와 동일한 Anthropic의 표준 티어 일정을 사용합니다. 워크스페이스당 티어 1 제한이 적용됩니다. 제한은 모델 패밀리별로 풀링됩니다. 예를 들어 모든 Opus 모델은 하나의 결합된 제한을 공유하고, 모든 Sonnet 모델은 다른 제한을 공유하며, 모든 Haiku 모델은 1/3을 공유합니다.

현재 티어 1 값(RPM, ITPM, OTPM) 및 상위 티어 임계값은 Anthropic 설명서 웹 사이트의 속도 제한을 참조하세요. Anthropic 페이지는 사실의 소스이며 제한이 변경될 때 업데이트됩니다.

## 속도 제한 헤더

모든 응답에는 현재 속도 제한 상태를 보고하는 헤더가 포함됩니다. 키 헤더:

- `anthropic-ratelimit-requests-limit` - 분당 최대 요청 수
- `anthropic-ratelimit-requests-remaining` - 현재 창에 남아 있는 요청
- `anthropic-ratelimit-requests-reset` - 요청 제한이 재설정되는 시간(RFC 3339)
- `anthropic-ratelimit-tokens-limit` - 분당 최대 결합된 토큰(입력 + 출력)
- `anthropic-ratelimit-tokens-remaining` - 현재 창에 남아 있는 결합된 토큰
- `anthropic-ratelimit-tokens-reset` - 결합된 토큰 제한이 재설정되는 시간(RFC 3339)
- `anthropic-ratelimit-input-tokens-limit` / `-remaining` `-reset`- Input-token-specific 헤더
- `anthropic-ratelimit-output-tokens-limit` / `-remaining` / `-reset` Output-token-specific 헤더
- `retry-after` - 429 응답에서 재시도하기 전에 대기할 초 수

전체 세트는 Anthropic 설명서 웹 사이트의 응답 헤더를 참조하세요.

# 더 높은 한도 요청

자사 Claude API와 달리 AWS의 Claude 플랫폼에서는 자동 티어 승격이 적용되지 않습니다. 더 높은 한도를 요청하려면 워크스페이스 ID와 원하는 처리량을 사용하여 Anthropic 계정 담당자에게 문의하세요. 티어 임계값 및 기타 세부 정보는 Anthropic 설명서 웹 사이트의 속도 제한을 참조하세요.

# 속도 제한 오류

속도 제한을 초과하면 API는 `rate_limit_error` 유형과 함께 HTTP 429를 반환합니다. 재시도 로직에서 지터를 사용하여 지수 백오프를 구현합니다. `retry-after` 헤더는 재시도하기 전에 기다려야 하는 초 수를 나타냅니다.

# 결제

AWS의 Claude 플랫폼은 AWS Marketplace를 소비 기반 측정의 결제 백엔드로 사용합니다. Anthropic은 AWS Marketplace를 통해 시간당 사용량을 측정하며, AWS는 기존 AWS 청구서에 월별 인보이스를 발행합니다.

결제는 요금(사용한 만큼만 지불) 및 세금 전(AWS가 계정의 세금 설정 적용)입니다.

사용량은 CCUs당 0.01 USD의 Claude 소비 단위(CCU)로 표시됩니다. CCU 가격은 고정되며 할인되지 않습니다. Anthropic은 표준 모델별, 기능별 요금으로 토큰 사용량을 USD로 평가하고 협상된 할인을 적용한 다음 결과를 CCUs당 0.01 USD의 CCU로 변환합니다. 할인은 더 낮은 CCUs 가격이 아닌 더 적은 CCU 측정으로 이어집니다. CCUs는 선불 크레딧이 아니며 CCU 잔액이나 약정이 없습니다. CCU 정의 요금 및 모델별 토큰 요금을 참조하세요.

# 모니터링 및 로깅

AWS CloudTrail은 AWS의 Claude 플랫폼에 대한 모든 요청을 캡처할 수 있습니다. Workspace 및 볼트 작업은 기본적으로 관리 이벤트로 기록됩니다. 다른 모든 작업(예: 볼트를 제외한 추론, 배치, 파일, 스킬, 모델, 사용자 프로필 및 Claude Managed Agents)은 데이터 이벤트입니다. 로깅하려면 명시적 구성이 필요하며 추가 CloudTrail 요금이 발생합니다. 전체 이벤트 유형 분류는 IAM 작업을 참조하고 구성 세부 정보는 AWS CloudTrail 설명서를 참조하세요.

CloudTrail 데이터 이벤트 로깅을 구성할 때 리소스 유형을 선택합니다`AWS::AWSExternalAnthropic::Workspace`. 이는 AWS 워크스페이스의 Claude Platform에 대한 CloudTrail 리소스 유형이며 데이터 영역 이벤트(추론, 배치, 파일 및 기타 워크스페이스별 작업)를 캡처하는 데 필요합니다. 전체 계정이 아닌 워크스페이스 하위 집합에 대한 활동을 로깅하려는 경우 데이터 이벤트 선택기를 특정 워크스페이스 ARNs으로 범위를 지정합니다.

## 요청 IDs

각 응답에는 응답 헤더에 두 개의 요청 IDs가 포함됩니다.

- AWS 요청 ID(**`x-amzn-requestid`**): CloudTrail에서 인덱싱된 기본 ID입니다. AWS 도구를 통해 요청을 조사하거나 AWS 지원팀에 문의할 때 사용합니다.
- Anthropic 요청 ID(**`request-id`**): 보조 ID입니다. Anthropic 지원팀에 문의할 때 사용합니다.

Python

```
from anthropic import AnthropicAWS

client = AnthropicAWS(aws_region="us-west-2")

response = client.messages.with_raw_response.create(
    model="claude-sonnet-4-6",
    max_tokens=1024,
    messages=[{"role": "user", "content": "Hello!"}],
)

print(response.headers.get("x-amzn-requestid"))  # AWS request ID
print(response.headers.get("request-id"))  # Anthropic request ID

message = response.parse()
```

```
print(message.content)
```

### TypeScript

```
import AnthropicAws from "@anthropic-ai/aws-sdk";

const client = new AnthropicAws({ awsRegion: "us-west-2" });

const { data: message, response } = await client.messages
  .create({
    model: "claude-sonnet-4-6",
    max_tokens: 1024,
    messages: [{ role: "user", content: "Hello!" }]
  })
  .withResponse();

console.log(response.headers.get("x-amzn-requestid")); // AWS request ID
console.log(response.headers.get("request-id")); // Anthropic request ID
console.log(message.content);
```

### C#

```
using Anthropic;
using Anthropic.Aws;

var client = new AnthropicAwsClient(new() { AwsRegion = "us-west-2" });

var response = await client.WithRawResponse.Messages.Create(new()
{
    Model = "claude-sonnet-4-6",
    MaxTokens = 1024,
    Messages = [new() { Role = "user", Content = "Hello!" }]
});

Console.WriteLine(response.Headers.GetValues("x-amzn-requestid").First()); // AWS
 request ID
Console.WriteLine(response.Headers.GetValues("request-id").First()); // Anthropic
 request ID
Console.WriteLine(response.Value.Content);
```

### Go

```
client, err := anthropicaws.NewClient(context.Background(),
 anthropicaws.ClientConfig{})
if err != nil {
    panic(err)
}

var response *http.Response
message, err := client.Messages.New(
    context.Background(),
    anthropic.MessageNewParams{
        Model:     anthropic.ModelClaudeSonnet4_6,
        MaxTokens: 1024,
        Messages: []anthropic.MessageParam{
            anthropic.NewUserMessage(anthropic.NewTextBlock("Hello!")),
        },
    },
    option.WithResponseInto(&response),
)
if err != nil {
    panic(err)
}

fmt.Println(response.Header.Get("x-amzn-requestid")) // AWS request ID
fmt.Println(response.Header.Get("request-id"))       // Anthropic request ID
fmt.Println(message.Content[0].Text)
```

### Java

```
import com.anthropic.aws.backends.AwsBackend;
import com.anthropic.client.AnthropicClient;
import com.anthropic.client.okhttp.AnthropicOkHttpClient;
import com.anthropic.core.http.HttpResponseFor;
import com.anthropic.models.messages.Message;
import com.anthropic.models.messages.MessageCreateParams;
import com.anthropic.models.messages.Model;

AnthropicClient client = AnthropicOkHttpClient.builder()
    .backend(AwsBackend.fromEnv())
    .build();

HttpResponseFor<Message> response = client.messages().withRawResponse().create(
    MessageCreateParams.builder()
        .model(Model.CLAUDE_SONNET_4_6)
```

```
        .maxTokens(1024)
        .addUserMessage("Hello!")
        .build()
);

IO.println(response.headers().values("x-amzn-requestid")); // AWS request ID
IO.println(response.requestId()); // Anthropic request ID
IO.println(response.parse().content());
```

PHP

```
use Anthropic\Aws\Client;

$client = new Client();

$response = $client->messages->raw->create(
    model: 'claude-sonnet-4-6',
    maxTokens: 1024,
    messages: [['role' => 'user', 'content' => 'Hello!']],
);

echo $response->getHeaderLine('x-amzn-requestid') . "\n"; // AWS request ID
echo $response->getHeaderLine('request-id') . "\n"; // Anthropic request ID
echo $response->parse();
```

Ruby

Ruby SDK는 현재 원시 응답 액세스 관리자를 노출하지 않으므로 Ruby의 응답 헤더에서 요청 IDs를 읽을 수 없습니다. 요청 ID 로깅이 필요한 경우 위의 다른 언어를 사용하거나 HTTP 프록시 계층에서 IDs를 캡처합니다.

Anthropic은 사용 패턴을 이해하고 잠재적 문제를 조사하기 위해 최소 30일 연속으로 활동을 로깅할 것을 권장합니다.

> **Note**
>
> AWS CloudTrail은 AWS 계정 내에서 구성됩니다. 로깅을 활성화해도 결제 및 서비스 작업에 필요한 것 이상으로 콘텐츠에 대한 AWS 또는 Anthropic 액세스는 제공되지 않습니다.

# 사용량 지표 및 대시보드

AWS의 Claude Platform은 Amazon CloudWatch에 워크스페이스당 사용량 지표를 게시하지 않습니다. 토큰 수, 요청 볼륨 및 모델별 사용량은 Anthropic의 사용 표면을 통해 대신 사용할 수 있습니다.

- Claude 콘솔 사용 보기 - 보안 주체가를 사용하여 Claude 콘솔에 페더레이션하면`aws-external-anthropic:AssumeConsole`(IAM 정책 참조) 사용 대시보드에 액세스 가능한 워크스페이스에 대한 요청 볼륨, 토큰 수 및 모델별 분류가 표시됩니다. 계정 전체 사용량 보기에는 관리자 콘솔 기능이 필요합니다.
- Anthropic Usage and Cost API - 워크스페이스별 및 모델별 집계를 포함하여 사용량 및 비용 데이터에 프로그래밍 방식으로 액세스하려면 Anthropic 설명서의 Usage and Cost API를 참조하세요.

운영 모니터링(오류율, 지연 시간, 속도 제한 헤더)의 경우 CloudTrail에 캡처된 AWS 요청 IDs와 함께 모든 요청에 대해 반환된 응답 헤더(속도 제한 및 할당량 참조)를 사용합니다.

# 비용 할당 및 모니터링

AWS의 Claude 플랫폼 요금은 AWS의 Claude 플랫폼 서비스 아래의 AWS 청구서에 모델별 사용 차원과 함께 표시됩니다. 세분화된 비용 할당을 위해 워크스페이스 태그와 결합된 AWS 비용 및 사용 보고서(CUR)를 사용합니다.

1. 관심 있는 할당 차원(팀, 애플리케이션, 환경, 비용 센터)으로 워크스페이스에 태그를 지정합니다. 태그 지정 세부 정보는 Workspaces를 참조하세요.
2. AWS Billing 콘솔에서 각 태그 키를 비용 할당 태그로 활성화합니다. 활성화 후 활성화 날짜 또는 그 이후에 발생한 사용량은 CUR의 태그로 분할됩니다.
3. CUR 또는 AWS Cost Explorer에서 `resourceTags/user:<tag-key>` 열을 기준으로 그룹화하여 워크스페이스 태그별로 지출을 분류합니다.

Workspace 태그는 AWS의 모든 Claude Platform 사용에 대해 CUR 라인 항목으로 흐릅니다. 동일한 태그 키는 IAM 기반 액세스 제어와 비용 할당을 모두 구동합니다. 설정 단계 및 활성화 지연은 AWS Billing 설명서의 비용 할당 태그를 참조하세요.

# Amazon Bedrock에서 마이그레이션

현재 Bedrock에서 Claude를 사용하는 경우 AWS에서 Claude 플랫폼으로 마이그레이션하려면 통합 전반에 걸쳐 변경이 필요합니다. SigV4 서명은 계속 지원되지만 서명 컨텍스트, 기본 URL, API 형식, 모델 IDs, SDK 클라이언트 및 패키지, 스트리밍 형식, 요청 헤더 및 리전 가용성은 모두 변경됩니다. 다음 표에는 차이점이 요약되어 있습니다.

## 변경 사항

| | Amazon Bedrock | AWS의 Claude 플랫폼 |
|---|---|---|
| 기본 URL | `bedrock-runtime.{region}.amazonaws.com` | `aws-external-anthropic.{region}.api.aws` |
| API 형식 | Bedrock Converse/InvokeModel | Anthropic Messages API(`/v1/messages`) |
| 모델 IDs | `anthropic.claude-opus-4-7-v1` | `claude-opus-4-7` |
| SDK 클라이언트 | `AnthropicBedrock` / Bedrock SDK | 베타 기반 플랫폼별 클라이언트(`AnthropicAWS`, `AnthropicAws` `AnthropicAwsClient`, 등) |
| SDK 패키지 | `anthropic[bedrock]`, `@anthropic-ai/bedrock-sdk`등 | `anthropic[aws]`, `@anthropic-ai/aws-sdk` 등(SDK 설치 참조) |
| SigV4 서비스 이름 | `bedrock` | `aws-external-anthropic` |
| 스트리밍 형식 | AWS EventStream | SSE(Claude API와 동일) |
| Workspace 헤더 | 해당 사항 없음 | `anthropic-workspace-id` 필수 |
| 리전 가용성 | 여러 상용 리전 | 모든 AWS 상용 리전(옵트인 리전에는 계정 옵트인 필요) |

## 얻는 이점

- 일반적으로 별도의 파트너 통합 단계 없이 새 모델 및 기능에 대한 당일 액세스
- 문서 생성을 위한 에이전트 기술(PowerPoint, Excel, Word, PDF)
- Anthropic의 관리형 샌드박스에서 코드 실행
- `anthropic-beta` 헤더를 통한 베타 기능(현재 사용할 수 없는 기능 참조)
- 할당량 가시성 및 사용량 분석을 위한 Claude 콘솔
- 직접 Anthropic 지원
- SigV4의 대안으로 API 키 인증(인증 참조)

## 동일하게 유지되는 항목

- AWS IAM 인증(SigV4)
- AWS 계정을 통한 결제
- AWS 약정 사용 중지

## 마이그레이션의 함정

> **Warning**
>
> 먼저 아웃바운드 웹 자격 증명 페더레이션을 활성화합니다. AWS 계정이 이전에 AWS에서 Claude 플랫폼을 사용한 적이 없는 경우 요청하기 전에 계정당 한 번 아웃바운드 웹 자격 증명 페더레이션을 활성화해야 합니다. 이 단계가 없으면 페더레이션 오류와 함께 모든 요청이 실패합니다. Bedrock에는이 단계가 필요하지 않습니다.

> **Warning**
>
> 제로 데이터 보존(ZDR)은 AWS의 Claude 플랫폼에서 옵트인됩니다. Bedrock에서 AWS는 데이터 프로세서이며 Anthropic은 추론 입력 또는 출력을 유지하지 않습니다. Anthropic의 ZDR 프로그램은 여기에 적용되지 않습니다. AWS의 Claude 플랫폼에서 Anthropic은 데이터 프로세서이며 ZDR은 자사 Claude API 모델을 따릅니다. 이는 Anthropic 계정 담당자를 통해 요청 시

제공됩니다. 데이터 보존 보장에 의존하는 프로덕션 워크로드를 마이그레이션하기 전에 ZDR 등록을 확인합니다.

# 상업적 고려 사항

- 서비스 약관: AWS 서비스 약관https://aws.amazon.com/service-terms/은 AWS의 Claude 플랫폼을 관리합니다. Anthropic의 상용 서비스 약관, 데이터 처리 부록 및 사용 정책도 적용됩니다.
- 할인 및 비공개 제안: 협상된 할인 및 AWS Marketplace 비공개 제안은 AWS의 Bedrock과 Claude 플랫폼 간에 자동으로 전송되지 않습니다. Anthropic 계정 담당자와 협력하여 AWS의 Claude Platform에 대한 상용 조건을 설정합니다.

# IAM 정책

AWS의 Claude 플랫폼은 액세스 제어를 위해 AWS IAM과 통합됩니다. 표준 IAM 정책 구문을 사용하여 특정 워크스페이스의 특정 API 작업에 대한 액세스 권한을 부여하거나 거부합니다.

SigV4 서비스 이름과 IAM 작업 네임스페이스는 입니다`aws-external-anthropic`. 작업은 패턴을 따릅니다`aws-external-anthropic:<Action>`(예: `aws-external-anthropic:CreateInference`).

## 예: 배치 추론 거부

다음 정책은 ZDR에 민감한 워크로드의 일반적인 요구 사항인 배치 처리를 차단하면서 실시간 추론을 허용합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-external-anthropic:CreateInference",
        "aws-external-anthropic:CountTokens",
        "aws-external-anthropic:GetModel",
        "aws-external-anthropic:ListModels",
        "aws-external-anthropic:GetWorkspace",
        "aws-external-anthropic:ListWorkspaces"
      ],
      "Resource": "arn:aws:aws-external-anthropic:*:*:workspace/*"
    },
    {
      "Effect": "Deny",
      "Action": [
        "aws-external-anthropic:CreateBatchInference",
        "aws-external-anthropic:GetBatchInference",
        "aws-external-anthropic:ListBatchInferences"
      ],
      "Resource": "*"
    }
  ]
}
```

GetBatchInference 작업은 배치 메타데이터 경로와 배치 결과 경로를 모두 승인합니다. 와 함께 거부하면 읽기와 배치 열거가 모두 ListBatchInferences차단됩니다.

Allow 문은 와일드카드를 사용하는 대신 특정 Get* 및 List* 작업을 열거합니다. 와일드카드는 GetFile (파일 바이트를 다운로드하는) 및 의도하지 않은 기타 읽기를 부여하고,는에 Allow 관계없이 Deny 재정의하지만 명시적 형식은 모델에 더 안전한 패턴입니다.

## 예: 단일 워크스페이스에 대한 동기 추론

하나의 프로덕션 워크스페이스에 대해 추론을 실행하는 IAM 보안 주체에 대한 최소 권한을 부여합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-external-anthropic:CreateInference",
        "aws-external-anthropic:CountTokens",
        "aws-external-anthropic:Get*",
        "aws-external-anthropic:List*"
      ],
      "Resource": "arn:aws:aws-external-anthropic:us-west-2:123456789012:workspace/
wrkspc_01AbCdEf23GhIj"
    }
  ]
}
```

> **Note**
>
> 이 정책의 List* 와일드카드는 계정 범위ListWorkspaces인 와도 일치합니다. 워크스페이스 ARN 제약 조건은 자동으로 필터링하므로이 정책은 워크스페이스 나열을 승인하지 않습니다. 서비스 계정이 워크스페이스를 열거해야 하는 경우를 ListWorkspaces 사용하여에 대한 별도의 Allow 문을 추가합니다Resource: "*".
> 이 정책은 AWS SigV4 인증을 가정합니다. 보안 주체가 API 키로 인증하는 경우 도 부여합니다aws-external-anthropic:CallWithBearerToken(인증 참조).

# 예: 고객별 워크스페이스 격리

역할을 단일 워크스페이스로 제한합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "aws-external-anthropic:*",
      "Resource": "arn:aws:aws-external-anthropic:us-west-2:123456789012:workspace/
wrkspc_01AbCdEf23GhIj"
    },
    {
      "Effect": "Allow",
      "Action": [
        "aws-external-anthropic:CallWithBearerToken",
        "aws-external-anthropic:AssumeConsole"
      ],
      "Resource": "*"
    }
  ]
}
```

> **Note**
>
> 첫 번째 문의 `aws-external-anthropic:*` 와일드카드에는 워크스페이스 ARN 제약 조건이 자동으로 필터링하는 계정 범위 작업(`CreateWorkspace`, `ListWorkspaces`)이 포함됩니다. 이는 "격리" 의도와 일치하며, 역할은 워크스페이스를 생성하거나 열거할 수 없지만 정책에는 효과가 없는 권한이 포함되어 있습니다. 계정 범위 패턴은 자동화 프로비저닝을 참조하세요.
>
> 두 번째 문은 워크스페이스 ARN`AssumeConsole`에 바인딩되지 않는 라우팅 없는 작업이므로 모든 리소스에 `CallWithBearerToken` 및를 부여합니다. 역할이 SigV4만 사용하고 Claude 콘솔에 페더레이션하지 않는 경우 두 번째 문을 생략합니다.

# 예: ZDR에 민감한 워크스페이스에 대한 기능 잠금

동기 추론을 사용할 수 있는 상태로 유지하면서 특정 워크스페이스에서 배치 처리 및 파일 업로드를 차단합니다. 워크스페이스가 서버 측을 유지해서는 안 되는 제로 데이터 보존(ZDR) 데이터를 처리할 때

유용합니다. 이 정책을 AnthropicLimitedAccess 또는 위의 단일 워크스페이스 예제와 같은 허용 정책과 함께 연결합니다. 거부 전용 정책은 자체적으로 권한을 부여하지 않습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "aws-external-anthropic:CreateBatchInference",
        "aws-external-anthropic:CreateFile"
      ],
      "Resource": "arn:aws:aws-external-anthropic:us-west-2:123456789012:workspace/
wrkspc_01AbCdEf23GhIj"
    }
  ]
}
```

> **Note**
>
> 이 거부는 생성만 차단합니다. 다른 파일 및 배치 작업도 나열하지 않는 한 거부되지 않습니다. 워크스페이스에 파일이나 배치가 없어야 하는 완전한 잠금의 경우 , aws-external-anthropic:GetFile, aws-external-anthropic:ListFiles, aws-external-anthropic:DeleteFile, aws-external-anthropic:GetBatchInference, aws-external-anthropic:ListBatchInferencesaws-external-anthropic:CancelBatchInference, 도 거부합니다aws-external-anthropic:DeleteBatchInference.

## 예: 프로비저닝 자동화

추론 권한 없이 워크스페이스를 생성하고 관리하는 데 필요한 작업을 CI/CD 역할에 부여합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-external-anthropic:CreateWorkspace",
```

```
        "aws-external-anthropic:GetWorkspace",
        "aws-external-anthropic:ListWorkspaces",
        "aws-external-anthropic:UpdateWorkspace",
        "aws-external-anthropic:ArchiveWorkspace"
      ],
      "Resource": "*"
    }
  ]
}
```

CreateWorkspace 및 ListWorkspaces는 계정 범위 작업입니다. 이러한 작업에 워크스페이스 ARN을 지정해도 아무런 효과가 없습니다.를 사용합니다Resource: "*".

## 관리형 정책

AWS는 일반적인 액세스 패턴에 대한 관리형 정책을 제공합니다.

- **AnthropicFullAccess**: 모든 리소스aws-external-anthropic:*에 대한 권한을 부여합니다.
- **AnthropicReadOnlyAccess**: 모든 리소스CallWithBearerToken에 List*, 및 Get*를 부여합니다.
- **AnthropicInferenceAccess**: 모든 리소스에 대해 ReadOnly 작업과 추론 작업(CreateInference, CreateBatchInference, CancelBatchInference, DeleteBatchInference, CountTokens)을 부여합니다.
- **AnthropicLimitedAccess**: 모든 리소스에 대한 AnthropicInferenceAccess 작업과 모든 Claude Managed Agents 작업(에이전트, 세션, 환경, 볼트, 메모리 스토어)을 부여합니다.
- **AnthropicSelfHostedEnvironmentAccess**: 자체 호스팅된 샌드박스 작업자가 환경 작업 항목을 폴링 및 처리하고, 연결된 환경, 세션 및 스킬 리소스를 읽고, 세션 상태를 업데이트하는 데 필요한 권한을 부여합니다. 자체 호스팅 환경 작업자가 수임하는 IAM 역할에이 정책을 연결합니다. AnthropicSelfHostedEnvironmentAccess는 권장되는 단일 역할 기본값입니다. 별도의 IAM 보안 주체에서 작업 폴러와 세션당 샌드박스를 실행하는 경우 최소 권한을 위해 이러한 권한을 더 좁은 두 사용자 지정 정책으로 분할할 수 있습니다.

AnthropicInferenceAccess는 추론을 실행하기에 충분한 가장 좁은 관리형 정책입니다. Get* 및 List* 와일드카드를 통해를 통한 파일 콘텐츠 다운로드 GetFile 및를 통한 메모리 콘텐츠를 포함하여 네임스페이스의 모든 API 리소스에 대한 읽기 액세스 권한을 부여합니다GetMemoryStore. 파일 또는 스킬 생성 또는 삭제, 사용자 프로필 관리, 워크스페이스 변형 또는 콘솔 페더레이션은 부여하지 않습니다.

> **Note**
>
> `AnthropicReadOnlyAccess`, `AnthropicInferenceAccess`및 `AnthropicLimitedAccess`는를 부여하지 않습니다`AssumeConsole`. Claude 콘솔에 연동해야 하는 보안 주체는 `AnthropicFullAccess` 또는 사용자 지정 정책을 `aws-external-anthropic:AssumeConsole` 통해 별도의 권한 부여가 필요합니다. Claude 콘솔에 페더레이션을 참조하세요.

> **Note**
>
> `CreateInference` 및 `CreateBatchInference`는 별도의 작업입니다. 하나를 거부해도 다른 하나는 차단되지 않습니다. 모든 모델 호출을 방지하려면 둘 다 거부합니다.

## Claude 콘솔에 페더레이션

`aws-external-anthropic:AssumeConsole`를 사용하면 IAM 보안 주체가 Anthropic에서 작동하는 Claude 콘솔에 페더레이션할 수 있습니다. 콘솔 내 액세스는 여전히 대부분의 작업에 대해 IAM에 의해 관리되지만, 주로 해당 IAM API가 없는 사용 보기인 관리 작업의 하위 집합은 보안 주체가 페더레이션한 기능에 따라 게이트됩니다.

두 가지 기능이 있습니다.

- **`developer`** - 콘솔에서 추론 실행, 워크스페이스 데이터 읽기, 개인 사용량 보기 등 day-to-day 작업에 필요한 Claude Platform on AWS 개발자의 작업을 허용합니다.
- **`admin`** - 계정 전체 사용량 보기 및 IAM 작업을 통해 표시되지 않는 관리 설정을 포함하여 관리자 전용 콘솔 작업을 추가로 허용합니다.

`AssumeConsole` 작업에서 `aws-external-anthropic:Capability` 조건 키를 사용하여 보안 주체가 요청할 수 있는 기능을 제어합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "aws-external-anthropic:AssumeConsole",
```

```
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws-external-anthropic:Capability": "admin"
        }
      }
    }
  ]
}
```

AnthropicFullAccess는 기능 제한 없이 AssumeConsole를 부여합니다. 개발자 전용 콘솔 액세스 또는 관리자 전용 액세스와 같은 더 좁은 권한 부여의 경우 위AssumeConsole와 같이 aws-external-anthropic:Capability 조건의 범위를 지정하는 사용자 지정 정책을 연결합니다.

## 보유자 토큰을 사용한 호출

aws-external-anthropic:CallWithBearerToken는 API 키가 보유자 토큰으로 표시될 때 사용되는 SigV4-free 요청 경로를 승인합니다. API 키로 인증하는 모든 보안 주체는 대상 워크스페이스에서이 작업이 필요합니다. 이는 Authorization: Bearer 헤더를 직접 사용하거나 ANTHROPIC_AWS_API_KEY 설정하든 관계없이 적용됩니다.

이는 추론 작업(CreateInference, CreateBatchInference등) 외에도 API 키 호출자에게 필요합니다. 이 없으면 추론 권한 부여 검사에 도달하기 전에 CallWithBearerTokenAPI 키 요청이 거부됩니다. SigV4 호출자는이 작업이 필요하지 않습니다.

AnthropicReadOnlyAccess, AnthropicInferenceAccessAnthropicLimitedAccess, 및 AnthropicFullAccess 모두를 포함합니다CallWithBearerToken. API 키 액세스에 대한 사용자 지정 정책을 작성하는 경우 명시적으로 추가합니다.

## 아웃바운드 웹 자격 증명 연동(콘솔 액세스에 필요)

Claude 콘솔은 AWS가 아닌 Anthropic 인프라에서 실행됩니다. IAM 보안 주체가를 호출하면 AssumeConsoleAWS STS는 Anthropic 대상에 범위가 지정된 웹 자격 증명 토큰을 발급합니다. 그러면 Anthropic 콘솔이 해당 토큰을 수락하고 페더레이션 세션을 설정합니다. 이렇게 하려면 AWS 계정이 aws-external-anthropic 대상에 대한 아웃바운드 웹 자격 증명 페더레이션을 허용해야 합니다.

를 호출하는 보안 주체는 aws-external-anthropic:AssumeConsole 작업 외에도 다음과 같은 STS 권한이 AssumeConsole 필요합니다.

- **`sts:GetWebIdentityToken`** - Anthropic 콘솔이 사용하는 웹 자격 증명 토큰을 발급할 수 있습니다.
- **`sts:TagGetWebIdentityToken`** - 웹 자격 증명 토큰에 세션 태그를 연결할 수 있습니다. AWS의 Claude 플랫폼은 이러한 태그를 사용하여 보안 주체의 기능과 워크스페이스 컨텍스트를 콘솔에 전달합니다.

콘솔 사용자가 수임하는 역할의 신뢰 정책 또는 `AssumeConsole` 직접 호출하는 사용자 자격 증명에 연결된 인라인/관리형 정책 모두에 포함합니다. `AnthropicFullAccess`에는 두 STS 작업이 모두 포함됩니다. SCP 또는 권한 경계 수준에서 거부하는 환경은 콘솔 페더레이션`sts:*`이 성공하려면 이러한 두 작업을 명시적으로 허용해야 합니다.

# AWS의 Claude 플랫폼에 대한 IAM 작업

AWS 정책을 통해 AWS의 Claude 플랫폼에 대한 액세스를 제어하기 위한 IAM 작업 참조입니다.

AWS의 Claude Platform은 액세스 제어를 위해 AWS IAM을 사용합니다. 모든 API 라우팅은 `aws-external-anthropic` 네임스페이스의 IAM 작업에 매핑됩니다. 이 페이지에는 모든 작업, 각 작업이 권한을 부여하는 라우팅, 일반적인 액세스 패턴에 사용할 수 있는 관리형 정책이 나열됩니다. 플랫폼 설정 및 인증은 AWS의 Claude 플랫폼이란 무엇입니까?를 참조하세요.

## 서비스 세부 정보

| | |
|---|---|
| IAM 서비스 접두사 | `aws-external-anthropic` |
| Resource types | `workspace` |
| Workspace ARN | `arn:aws:aws-external-anthropic:{region}:{account-id}:workspace/{workspace-id}` |

ARN 리전은 항상 채워지며 워크스페이스가 바인딩된 리전과 일치합니다. 리소스 세그먼트는 `anthropic-workspace-id` 헤더에 전달하는 것과 동일한 값인 태그가 지정된 워크스페이스 ID(`wrkspc_…` )입니다.

## 작업

이 서비스는 15개 그룹에서 65개의 작업을 정의합니다. 작업은 AWS `VerbNoun` 규칙을 따르며 `Get*` 및 `List*` 와일드카드가 명확한 읽기 전용 경계를 생성하도록 동사 원칙을 사용합니다.

### Inference

| 작업 | 권한이 부여된 경로 |
|---|---|
| `CreateInference` | `POST /v1/messages` |
| `CountTokens` | `POST /v1/messages/count_tokens` |

## 배치 처리

| 작업 | 권한이 부여된 경로 |
| --- | --- |
| CreateBatchInference | POST /v1/messages/batches |
| GetBatchInference | GET /v1/messages/batches/{id} GET /v1/messages/batches/{id}/results |
| ListBatchInferences | GET /v1/messages/batches |
| CancelBatchInference | POST /v1/messages/batches/{id}/cancel |
| DeleteBatchInference | DELETE /v1/messages/batches/{id} |

## 모델

| 작업 | 권한이 부여된 경로 |
| --- | --- |
| GetModel | GET /v1/models/{id} |
| ListModels | GET /v1/models |

## 파일

| 작업 | 권한이 부여된 경로 |
| --- | --- |
| CreateFile | POST /v1/files |
| GetFile | GET /v1/files/{id} GET /v1/files/{id}/content |
| ListFiles | GET /v1/files |
| DeleteFile | DELETE /v1/files/{id} |

> **Note**
>
> GetFile는 메타데이터 및 콘텐츠 다운로드를 모두 승인합니다. 읽기 전용 액세스 권한이 있는 보안 주체는 파일 나열뿐만 아니라 파일 바이트를 다운로드할 수 있습니다.

## Skills

| 작업 | 권한이 부여된 경로 |
| --- | --- |
| CreateSkill | POST /v1/skills |
| GetSkill | GET /v1/skills/{id} GET /v1/skills/{id}/versions GET /v1/skills/{id}/versions/{version} GET /v1/skills/{id}/versions/{version}/content |
| ListSkills | GET /v1/skills |
| UpdateSkill | POST /v1/skills/{id}/versions DELETE /v1/skills/{id}/versions/{version} |
| DeleteSkill | DELETE /v1/skills/{id} |

> **Note**
>
> 개별 스킬 버전을 삭제하면이 UpdateSkill아닌에 매핑됩니다DeleteSkill. 거부하는 정책은 aws-external-anthropic:Delete* 여전히 버전 삭제를 허용합니다. 스킬 변형을 방지해야 하는 경우 UpdateSkill 및 CreateSkill 도 거부합니다.

## 사용자 프로필

| 작업 | 권한이 부여된 경로 |
| --- | --- |
| CreateUserProfile | POST /v1/user_profiles |

| 작업 | 권한이 부여된 경로 |
| --- | --- |
| `GetUserProfile` | `GET /v1/user_profiles/{id}` |
| `ListUserProfiles` | `GET /v1/user_profiles` |
| `UpdateUserProfile` | `POST /v1/user_profiles/{id}` |

> **Warning**
>
> IAM 작업 일치는 대/소문자를 구분하지 않습니다. 와일드카드는 `CreateFile`, 및 `GetFile`와 `aws-external-anthropic:*File` 일치하지`DeleteFile`만 일치하지 않습니다`ListFiles`('파일'이 아닌 '파일'로 끝남). 또한 `CreateUserProfile`"Profile"이 "file"로 끝나기 `UpdateUserProfile` 때문에 `GetUserProfile`, 및 과도 일치합니다. 파일 API 작업만 부여하거나 거부하려는 경우 `*File` 접미사 패턴을 사용하는 대신 명시적으로 열거합니다(`CreateFile`, `GetFile`, `ListFiles`, `DeleteFile`).

## 에이전트

Claude Managed Agents에 대한 에이전트 정의입니다.

| 작업 | 권한이 부여된 경로 |
| --- | --- |
| `CreateAgent` | 에이전트 생성 경로 |
| `GetAgent` | 에이전트 읽기 경로 |
| `ListAgents` | 에이전트 목록 경로 |
| `UpdateAgent` | 에이전트 업데이트 경로 |
| `ArchiveAgent` | 에이전트 아카이브 경로 |

## 세션

에이전트 세션 및 이벤트 기록.

| 작업 | 권한이 부여된 경로 |
| --- | --- |
| CreateSession | 세션 생성 경로 |
| GetSession | 세션 읽기 경로 |
| ListSessions | 세션 목록 경로 |
| UpdateSession | 세션 업데이트 경로 |
| ArchiveSession | 세션 아카이브 경로 |
| DeleteSession | 세션 삭제 경로 |

## 환경

세션에 대한 클라우드 컨테이너 구성입니다.

| 작업 | 권한이 부여된 경로 |
| --- | --- |
| CreateEnvironment | 환경 생성 경로 |
| GetEnvironment | 환경 읽기 경로 |
| ListEnvironments | 환경 목록 경로 |
| UpdateEnvironment | 환경 업데이트 경로 |
| ArchiveEnvironment | 환경 아카이브 경로 |
| DeleteEnvironment | 환경 삭제 경로 |
| ProcessEnvironmentWork | 자체 호스팅 환경 작업자 경로 |

## 저장소

세션 인증을 위한 자격 증명 저장소입니다.

| 작업 | 권한이 부여된 경로 |
| --- | --- |
| `CreateVault` | 볼트 생성 경로 |
| `GetVault` | 볼트 읽기 경로 |
| `ListVaults` | 볼트 목록 경로 |
| `UpdateVault` | 볼트 업데이트 경로 |
| `ArchiveVault` | 볼트 아카이브 경로 |
| `DeleteVault` | 볼트 삭제 경로 |

> **Note**
>
> 저장소 작업은 (데이터 이벤트가 아닌) CloudTrail 관리 이벤트로 분류됩니다. 저장소에는 자격 증명이 저장되고 기본 감사 로깅의 이점이 있기 때문입니다. 다른 Claude Managed Agents 작업(에이전트, 세션, 환경, 메모리 스토어)은 데이터 이벤트입니다.

## 메모리 스토어

영구 에이전트 메모리.

| 작업 | 권한이 부여된 경로 |
| --- | --- |
| `CreateMemoryStore` | 메모리 스토어 생성 경로 |
| `GetMemoryStore` | 메모리 스토어 읽기 경로 |
| `ListMemoryStores` | 메모리 스토어 목록 경로 |
| `UpdateMemoryStore` | 메모리 스토어 업데이트 경로 |
| `ArchiveMemoryStore` | 메모리 스토어 아카이브 경로 |
| `DeleteMemoryStore` | 메모리 스토어 삭제 경로 |

> **Note**
>
> GetMemoryStore는 메모리 콘텐츠를 읽습니다. 따라서 관리형 또는 사용자 지정 정책의 Get* 와일드카드는 메모리 읽기 액세스 권한을 부여합니다. 메모리 콘텐츠를 제한해야 하는 경우 정책 범위를 명시적으로 지정합니다.

## Webhook

세션에 대한 수명 주기 이벤트 알림입니다.

| 작업 | 권한이 부여된 경로 |
| --- | --- |
| ListWebhooks | GET /v1/webhooks |
| GetWebhook | GET /v1/webhooks/{webhook_endpoint_id} |
| CreateWebhook | POST /v1/webhooks |
| UpdateWebhook | POST /v1/webhooks/{webhook_endpoint_id} |
| DeleteWebhook | DELETE /v1/webhooks/{webhook_endpoint_id} |
| RotateWebhookSecret | POST /v1/webhooks/{webhook_endpoint_id}/regenerate_signing_secret |

## 워크스페이스

| 작업 | 권한이 부여된 경로 |
| --- | --- |
| CreateWorkspace | POST /v1/organizations/workspaces |
| GetWorkspace | GET /v1/organizations/workspaces/{id} |
| ListWorkspaces | GET /v1/organizations/workspaces |
| UpdateWorkspace | POST /v1/organizations/workspaces/{id} |

| 작업 | 권한이 부여된 경로 |
| --- | --- |
| `ArchiveWorkspace` | `POST /v1/organizations/workspaces/{id}/archive` |

기본 워크스페이스는 가입 시 프로비저닝됩니다. 워크스페이스를 참조하세요.

## Authentication

| 작업 | 권한이 부여된 경로 |
| --- | --- |
| `CallWithBearerToken` | (none) |

`CallWithBearerToken`는 보안 주체가 AWS SigV4가 아닌 API 키(베어러 토큰)를 통해 인증할 수 있는 권한을 부여하는 인증 계층 권한입니다. 경로에 매핑되지 않습니다. API 키 소유자가 수행할 라우팅 매핑 작업과 함께 부여합니다.

## 콘솔 액세스

| 작업 | 권한이 부여된 경로 |
| --- | --- |
| `AssumeConsole` | (none) |

`AssumeConsole`는 보안 주체에게 AWS 콘솔 페더레이션 흐름을 통해 AWS 워크스페이스의 Claude 플랫폼용 Claude 콘솔을 열 수 있는 권한을 부여합니다. 경로에 매핑되지 않습니다. AWS 콘솔의 Claude Platform on AWS 서비스 페이지에서 Claude 콘솔 열기를 클릭할 수 있어야 하는 보안 주체에게 부여합니다. `AssumeConsole` 작업의 `aws-external-anthropic:Capability` 조건 키는 보안 주체가 요청할 수 있는 콘솔 기능(개발자 또는 관리자)을 제어합니다. 자세한 내용은 IAM 정책 및 로그인 흐름에 Claude 콘솔 사용을 참조하세요.

> **Warning**
>
> `aws-external-anthropic:AssumeConsole`는 호출자의 세션 기간과 관계없이 최대 12시간 동안 지속되는 Claude 콘솔 세션을 발행합니다. IAM 세션이 12시간 미만인 호출자는 소스

자격 증명을 초과하는 콘솔 세션을 계속 얻을 수 있습니다. 이 권한을 Claude 콘솔 액세스가 필요한 보안 주체로 제한하고 더 이상 필요하지 않은 경우 즉시 취소합니다.

> **Note**
>
> 페더레이션 후 Claude 콘솔 내에서 수행된 작업은 AWS CloudTrail에 기록되지 않거나 IAM을 통해 기인하지 않습니다. 여기에는 사용 보고서 보기와 같은 글로벌 워크스페이스 범위 활동이 포함됩니다. 콘솔 활동에 대한 감사 추적이 필요한 경우 CloudTrail 대신 Claude 콘솔에서 사용할 수 있는 감사 로그를 사용합니다.

## Route-to-action 매핑

다음 표에는 AWS의 Claude 플랫폼의 모든 경로와 이를 호출하는 데 필요한 IAM 작업이 나열되어 있습니다. 안정적인 라우팅의 IAM 작업은 `anthropic-beta` 헤더를 사용하는 요청도 승인합니다. CloudTrail은 각 작업을 데이터 이벤트(대량, 데이터 영역 작업) 또는 관리 이벤트(컨트롤 영역 작업)로 분류합니다.

| 방법 | 경로 | IAM 작업 | CloudTrail 이벤트 유형 |
|---|---|---|---|
| `POST` | `/v1/messages` | `CreateInference` | 데이터 |
| `POST` | `/v1/messages/count_tokens` | `CountTokens` | 데이터 |
| `POST` | `/v1/messages/batches` | `CreateBatchInference` | 데이터 |
| `GET` | `/v1/messages/batches` | `ListBatchInferences` | 데이터 |
| `GET` | `/v1/messages/batches/{id}` | `GetBatchInference` | 데이터 |
| `GET` | `/v1/messages/batches/{id}/results` | `GetBatchInference` | 데이터 |

| 방법 | 경로 | IAM 작업 | CloudTrail 이벤트 유형 |
|---|---|---|---|
| POST | /v1/messages/batches/{id}/cancel | CancelBatchInference | 데이터 |
| DELETE | /v1/messages/batches/{id} | DeleteBatchInference | 데이터 |
| GET | /v1/models | ListModels | 데이터 |
| GET | /v1/models/{id} | GetModel | 데이터 |
| POST | /v1/files | CreateFile | 데이터 |
| GET | /v1/files | ListFiles | 데이터 |
| GET | /v1/files/{id} | GetFile | 데이터 |
| GET | /v1/files/{id}/content | GetFile | 데이터 |
| DELETE | /v1/files/{id} | DeleteFile | 데이터 |
| POST | /v1/skills | CreateSkill | 데이터 |
| GET | /v1/skills | ListSkills | 데이터 |
| GET | /v1/skills/{id} | GetSkill | 데이터 |
| DELETE | /v1/skills/{id} | DeleteSkill | 데이터 |
| POST | /v1/skills/{id}/versions | UpdateSkill | 데이터 |
| GET | /v1/skills/{id}/versions | GetSkill | 데이터 |
| GET | /v1/skills/{id}/versions/{version} | GetSkill | 데이터 |
| GET | /v1/skills/{id}/versions/{version}/content | GetSkill | 데이터 |

| 방법 | 경로 | IAM 작업 | CloudTrail 이벤트 유형 |
|---|---|---|---|
| DELETE | /v1/skills/{id}/versions/{version} | UpdateSkill | 데이터 |
| POST | /v1/user_profiles | CreateUserProfile | 데이터 |
| GET | /v1/user_profiles | ListUserProfiles | 데이터 |
| GET | /v1/user_profiles/{id} | GetUserProfile | 데이터 |
| POST | /v1/user_profiles/{id} | UpdateUserProfile | 데이터 |
| POST | /v1/organizations/workspaces | CreateWorkspace | 관리 |
| GET | /v1/organizations/workspaces | ListWorkspaces | 관리 |
| GET | /v1/organizations/workspaces/{id} | GetWorkspace | 관리 |
| POST | /v1/organizations/workspaces/{id} | UpdateWorkspace | 관리 |
| POST | /v1/organizations/workspaces/{id}/archive | ArchiveWorkspace | 관리 |

Claude Managed Agents 경로(에이전트, 세션, 환경, 볼트, 메모리 스토어)는 동일한 route-to-action 패턴을 따릅니다. 전체 라우팅별 매핑은 Anthropic IAM 작업 참조를 참조하세요. 볼트 경로는 관리 이벤트이고, 에이전트, 세션, 환경 및 메모리 스토어 경로는 데이터 이벤트입니다.

AWS의 Claude 플랫폼에 있지 않은 경로는 기본적으로 게이트웨이에서 거부됩니다.

## 다음 사항도 참조하세요.

- 정책 및 관리형 정책에 대한 IAM 정책

- IAM 정책 구문 및 평가 로직에 대한 AWS IAM 사용 설명서
- 감사 로깅 구성을 위한 AWS CloudTrail 사용 설명서

# 문서 기록

다음 표에서는 AWS의 Claude 플랫폼 사용 설명서의 중요한 변경 사항을 설명합니다.

| 변경 | 설명 | Date |
| --- | --- | --- |
| 최초 게시 | AWS의 Claude 플랫폼 사용 설명서를 게시했습니다. | 2026년 5월 7일 |

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.