aws

使用者指南

AWS 上的 Claude 平台

AWS 上的 Claude 平台: 使用者指南

Table of Contents

什麼是 AWS 上的 Claude 平台？

透過具有 Anthropic 受管基礎設施的 AWS 存取 Claude 的完整平台功能。

AWS 上的 Claude 平台提供您完整的 Anthropic 平台體驗，包括透過 AWS 帳戶存取的訊息 API、客服人員技能、程式碼執行和 Beta 版功能。與 AWS 操作推論堆疊的 Amazon Bedrock 不同，Anthropic 會在 AWS 上操作 Claude 平台。AWS 透過 AWS Marketplace 提供身分驗證層 (SigV4 或 API 金鑰）、IAM 型存取控制和帳單整合。

> **Note**
>
> Anthropic SDKs 支援 AWS 上的 Claude 平台。如需每個語言用戶端的可用性，請參閱 Anthropic 用戶端 SDKs 文件。

平台整合的運作方式

Claude 模型會在 Anthropic 受管基礎設施上執行。這是用於透過 AWS 計費和存取的商業整合。Anthropic 和 AWS 都是獨立的資料處理者。AWS 服務條款管理 AWS 上的 Claude 平台。Anthropic 的商業服務條款、資料處理增補和用量政策也適用。

請注意下列操作特性。資料可能不會位於 AWS 中。推論可能會路由到 Anthropic 的主要雲端。子服務可能會變更，恕不另行通知。為每個請求設定 `inference_geo` 參數，將推論鎖定在特定地理位置。如需詳細資訊，請參閱資料駐留。

AWS 上的 Claude 平台遵循與第一方 Claude API 相同的資料保留政策。零資料保留 (ZDR) 可應要求提供。請聯絡您的 Anthropic 帳戶代表，為您的帳戶啟用它。

AWS 上 Claude 平台的功能

AWS 上的 Claude 平台提供下列功能：

- 當日模型和功能存取 — 新的 Claude 模型和 API 功能可在第一方 Claude API 上啟動的同一天提供。
- 客服人員技能 — 使用預先建置的技能來產生文件 (PowerPoint、Excel、Word、PDF) 並建立自訂技能。
- 程式碼執行 — 在 Anthropic 的受管沙盒中執行程式碼。

- 延伸思考：為複雜的推理任務啟用延伸思考。
- 串流和批次處理 — 使用即時 SSE 串流或提交高輸送量工作負載的批次請求。
- 提示快取 — 快取工具、系統提示和訊息歷史記錄，以減少延遲和成本。
- 檔案 API — 跨請求上傳和參考檔案。
- AWS IAM 整合 — 使用標準 IAM 政策和 SigV4 身分驗證控制存取。
- 統一 AWS 帳單 — 使用以耗用量為基礎的計量（以商城做為帳單後端）透過現有的 AWS 帳戶付款。

如需支援功能的完整清單，請參閱功能支援。

本指南的編排方式

本指南涵蓋下列主題：

- 入門 — 帳戶設定、先決條件、身分驗證和 SDK 安裝。
- 使用 API — 可用的模型、提出請求和功能支援。
- 管理您的環境：資料駐留、工作區、Claude 主控台、速率限制和帳單。
- 監控和操作 — CloudTrail 記錄和請求 ID 追蹤。
- 遷移 — 從 Amazon Bedrock 移至 AWS 上的 Claude 平台。
- 安全性和存取控制 — IAM 政策和動作參考。

AWS 上的 Claude 平台與 Amazon Bedrock

這兩種方案都可讓您透過 AWS 使用 Claude，但它們在架構、API 表面和功能可用性方面有很大的差異。

	AWS 上的 Claude 平台	Amazon Bedrock
誰操作堆疊	Anthropic	AWS
API 表面	Anthropic Messages API (`/v1/messages`)	Bedrock API (Converse / InvokeModel)
功能可用性	與 Claude API 同一天	取決於 AWS 整合時間表
客服人員技能	Available	無法使用（需要程式碼執行）
Beta 功能	使用`anthropic-beta` 標頭傳遞（請參閱功能支援)	需要 AWS 支援
身分驗證	AWS IAM / SigV4 或 API 金鑰	AWS IAM / SigV4 或承載字符（僅限 C#、Go 和 Java SDKs)
基本 URL	`aws-external-anthropic.{region}.api.aws`	`bedrock-runtime.{region}.amazonaws.com`
SDK 用戶端	Beta 版中的平台特定用戶端類別（例如，`AnthropicAWS` 在 Python 中）	`AnthropicBedrock` / Bedrock SDK
主控台	Claude 主控台 (`platform.claude.com`，透過 AWS 主控台存取）	Bedrock 主控台
速率限制和配額	由 Anthropic 管理	由 AWS 管理
資料處理者	Anthropic 和 AWS	AWS

如果您需要 AWS 操作的 Claude 搭配 Bedrock API，請參閱 Amazon Bedrock。

> **Important**
>
> Anthropic 在 AWS 上提供 Claude 平台做為第三方產品。標準 AWS 合規計劃、認證或稽核報告（例如 SOC、ISO 或 HIPAA 資格）並未涵蓋此項目。客戶全權負責執行自己的盡職調查，以確保此第三方產品符合其法規、法律和合規要求。

何時選擇 Bedrock：如果您的組織需要 AWS 操作的推論、AWS 作為唯一資料處理者，或 AWS 合規計劃（包括 FedRAMP High、IL4、IL5、SOC、ISO 和 HIPAA 資格）涵蓋範圍，請選擇 Amazon Bedrock。Bedrock 完全在 AWS 控制的基礎設施上執行，並以 AWS 做為操作方。

設定您的帳戶

在 AWS 上設定 Claude 平台有四個階段：在 AWS 主控台中註冊、設定您的 Anthropic 組織、記下您的工作區 ID，以及登入 Claude 主控台。

> **Note**
>
> 透過 AWS 主控台註冊會佈建與您的 AWS 帳戶繫結的新 Anthropic 組織。此組織與貴公司具有 Anthropic 的任何現有組織不同，包括透過 AWS Marketplace 採購的 Claude Enterprise 組織。來自第一方 Anthropic 組織的 API 金鑰、工作區和 Claude Console 設定不會轉移。
> 如果您有現有的 Amazon Bedrock 私有優惠，請在註冊之前聯絡您的 Anthropic 或 AWS 客戶主管，以便從第一個請求套用折扣。折扣無法追溯到接受私有優惠之前產生的用量。請參閱 私有優惠。

步驟 1：在 AWS 主控台中註冊

1. 開啟 AWS 主控台並導覽至 AWS 服務上的 Claude 平台頁面。
2. 選擇註冊。
3. 選擇繼續。

此頁面會顯示註冊進行中橫幅。停留在 頁面上。註冊需要幾分鐘的時間，而 AWS 會為您處理 AWS Marketplace 訂閱，然後自動重新引導您。

如果您的組織有來自 Anthropic 的私有優惠，主控台會查詢它，並提示您在 AWS Marketplace 中接受它。如需詳細資訊，請參閱 私有優惠。

> **Note**
>
> 如果您在 AWS 上使用 Claude 平台，您的內容（例如提示和完成）將由 AWS 外部的 Anthropic 處理。如需如何處理和儲存內容和中繼資料的詳細資訊，請參閱 Anthropic 的資料使用政策。

步驟 2：設定您的 Anthropic 組織

註冊完成後，系統會將您重新導向至 `platform.claude.com/partner-signup`。

1. 輸入組織擁有者的電子郵件地址，然後選擇開始使用。
2. 檢查電子郵件收件匣是否有設定連結並遵循該連結。如果您的瀏覽器顯示已登入為不同的帳戶頁面，請選擇登出並繼續。
3. 完成組織詳細資訊表單（組織名稱、實體類型、國家/地區、預期用途），然後選擇完成設定。

完成設定會建立您的 Anthropic 組織。AWS 服務條款管理 AWS 上的 Claude 平台。Anthropic 的商業服務條款、資料處理增補和用量政策也適用。AWS 主控台服務頁面現在會顯示左側導覽，其中包含首頁、API 金鑰、Quickstart 和工作區。

步驟 3：記下您的工作區 ID

註冊時會自動佈建預設工作區。每個區域可以建立其他工作區；如需區域繫結、工作區管理和 IAM 資源範圍的詳細資訊，請參閱工作區。

在 AWS 服務頁面上的 AWS 主控台 Claude 平台的工作區下或在 Claude 主控台中尋找工作區 ID（請參閱使用 Claude 主控台)。工作區 IDs使用 的格式，`wrkspc_`後面接著英數字元識別符。

步驟 4：登入 Claude 主控台

透過 AWS IAM 聯合存取 Claude 主控台：

1. 使用 `aws-external-anthropic:AssumeConsole`許可擔任 IAM 角色。請參閱 IAM 動作。
2. 從 AWS 服務上的 Claude 平台頁面，選擇登入。AWS 主控台會發出 JWT 並將您重新導向至 `platform.claude.com`。
3. 第一次登入時，系統會提示您輸入電子郵件地址。輸入您的工作電子郵件。平台just-in-time佈建您的 Claude Console 使用者。

當您透過 AWS 主控台登入時，Claude 主控台會範圍限定為 AWS 組織的 Claude 平台。由 AWS 指標管理的帳戶會顯示在 Claude Console 側邊欄中。

對帳戶設定進行故障診斷

- 「註冊失敗：無法啟用 OutboundWebIdentityFederation」：如果您在第一次提交時看到此紅色橫幅，請再次選擇繼續。IAM 啟用可能需要一些時間才能傳播。
- 註冊期間沒有進度指標：註冊需要幾分鐘的時間。當 AWS 佈建您的帳戶時，此頁面會顯示沒有進度列的靜態註冊進行中橫幅。

- 在遵循設定連結後「以不同帳戶登入」：選擇登出並繼續。此頁面會使用您輸入的電子郵件地址重新驗證您的身分。
- 登入期間出現「找不到」訊息：此訊息可能會在重新導向期間短暫出現。您可以將其關閉。
- 用量頁面在第一次 API 呼叫後不會顯示任何資料：用量資料可能需要幾分鐘才會出現在 Claude 主控台中。

先決條件

在進行 API 呼叫之前，請確定您有：

1. 在 AWS 上訂閱 Claude 平台的作用中 AWS 帳戶（請參閱設定您的帳戶）。
2. 已安裝並設定 AWS CLI。
3. 在您的 AWS 帳戶上啟用傳出 Web 聯合身分（一次性設定步驟；請參閱啟用傳出 Web 聯合身分）。
4. 您的工作區 ID（請參閱取得您的工作區 ID）。

啟用傳出 Web 聯合身分

AWS 閘道上的 Claude 平台會呼叫`sts:GetWebIdentityToken`伺服器端，以截取轉送至 Anthropic 的 JWT。每個 AWS 帳戶預設都會停用此 STS 功能。每個帳戶啟用一次：

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

如果回應為 `[ERROR] (FeatureEnabled) … already enabled` ，則您帳戶的 設定已開啟，您可以繼續。驗證並擷取您帳戶的發行者 URL：

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

> **Warning**
>
> 如果沒有此步驟，每個請求都會傳回 `"Outbound web identity federation is disabled for your account"`。這是最常見的設定錯誤。

取得您的工作區 ID

註冊時，預設工作區會在每個區域中自動佈建（請參閱設定您的帳戶）。工作區繫結至單一 AWS 區域。您可以在工作區下的 Claude 主控台或 AWS 主控台服務頁面的工作區區段中找到工作區 ID。如需區域繫結和 IAM 資源範圍，請參閱工作區。

設定 `ANTHROPIC_AWS_WORKSPACE_ID`和 `AWS_REGION`環境變數，讓 SDK 用戶端自動讀取它們：

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

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

區域為必要項目。如果未設定區域，則 SDK 用戶端會擲回 。將 aws_region/awsRegion 傳遞至建構函數，或設定 AWS_REGION (或 AWS_DEFAULT_REGION)。支援所有 AWS 商業區域；選擇加入區域要求您先選擇加入該區域的帳戶。

身分驗證

AWS 上的 Claude 平台支援兩種身分驗證方法：AWS IAM 搭配 SigV4 請求簽署（主要）和 API 金鑰身分驗證。兩者都使用相同的基本 URL 和請求格式。

Anthropic SDKs（請參閱安裝 SDK) 實作以下所述的身分驗證流程和登入資料解析。如果您不是使用 Anthropic SDK，則必須針對區域端點 建構 SigV4-signed請求（或將您的 API 金鑰顯示為承載字符）`https://aws-external-anthropic.<region>.api.aws`。如需開發套件特定的用戶端組態和授權憑證解析順序，請參閱 Anthropic 文件中的 AWS 上的 Claude 設定指南。

SigV4 身分驗證

SigV4 是企業原生路徑，可與現有的 AWS IAM 政策、角色和稽核整合。使用 AWS 預設登入資料提供者鏈結支援的任何方法設定 AWS 登入資料：

- 環境變數 (`AWS_ACCESS_KEY_ID`、`AWS_SECRET_ACCESS_KEY`、`AWS_SESSION_TOKEN`)
- 共用登入資料檔案 (`~/.aws/credentials`)
- 共用組態檔案 (`~/.aws/config`)，包括 SSO 和 `credential_process`
- IRSA 和 GitHub 動作的 Web 身分 (`AWS_WEB_IDENTITY_TOKEN_FILE` 和 `AWS_ROLE_ARN`)
- ECS 容器憑證
- EC2 執行個體中繼資料服務 (IMDS)

若要驗證 Anthropic SDK 正在取得您的登入資料並解析為正確的區域端點，請遵循提出請求中的範例提出測試請求。成功的回應會確認登入資料、區域、基本 URL 和工作區 ID 都已正確設定。

API 金鑰身分驗證

對於更簡單的整合路徑（本機開發和指令碼），您可以使用 API 金鑰而不是 SigV4 進行身分驗證。設定`ANTHROPIC_AWS_API_KEY`環境變數或傳遞`apiKey`至 SDK 建構函式。Anthropic SDK 會將金鑰作為承載字符傳送至 AWS 端點上的 Claude 平台；IAM 會透過 `aws-external-anthropic:CallWithBearerToken`動作授權請求（請參閱 IAM 政策)。

在 AWS 上的 Claude 平台 → API 金鑰下的 AWS 主控台中產生 API 金鑰。選擇產生金鑰，然後複製金鑰值。將 `aws-external-anthropic:CallWithBearerToken` IAM 動作授予應允許使用 API 金鑰身分驗證的主體。

> **Note**
>
> 只有在 AWS 上的 Claude 平台下的 AWS 主控台中建立的 API 金鑰才能使用此服務。
>
> - 在標準 Claude 主控台中為第一方 API 存取建立的金鑰不適用於 AWS 端點上的 Claude 平台。
> - Amazon Bedrock API 金鑰也無法運作 — Bedrock 使用單獨的端點、身分驗證流程和 IAM 命名空間。

短期 API 金鑰

對於您想要 API 金鑰身分驗證但沒有長期金鑰生命週期的情況，Anthropic 會發佈權杖產生器程式庫，該程式庫會從 AWS IAM 憑證中挖掘短期 API 金鑰。字符預設為 12 小時的生命週期，並且可以範圍限定在特定工作區和 `aws-external-anthropic:CallWithBearerToken`動作。為您的語言安裝產生器、交換 API 金鑰的 IAM 憑證，並將金鑰傳遞至 Anthropic SDK。

- 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

短期 API 金鑰仍然需要發起人的 IAM 主體保留在目標工作區`aws-external-anthropic:CallWithBearerToken`上。它們會自行過期，不需要明確輪換或撤銷。

登入資料優先順序和區域解析

Anthropic SDK 在 AWS 用戶端上的 Claude 平台會使用定義的優先順序來解析登入資料和區域。引數名稱遵循每種語言的慣例：適用於 TypeScript 和 PHP 的 camelCase、適用於 Python 和 Ruby 的 snake_case、適用於 Go 的 PascalCase，以及適用於 C# 和 Java 的屬性或建置器模式。

如需每個 SDK 的授權優先順序、支援的環境變數和建構函數引數，請參閱 Anthropic 文件中的 AWS 設定上的 Claude。一般而言，明確的建構函數引數優先於環境變數，並`ANTHROPIC_AWS_API_KEY`優先於預設的 AWS 憑證提供者鏈結。區域為必要項目；與 `AnthropicBedrock`不同，如果建構函數或 / 沒有提供區域，則 AWS 用戶端上的 `us-east-1`Claude `AWS_REGION` 會擲回 `AWS_DEFAULT_REGION`。

工作空間 ID

每個資料平面請求都必須在 `anthropic-workspace-id`標頭中包含工作區 ID。依預設， Anthropic SDKs 會從`ANTHROPIC_AWS_WORKSPACE_ID`環境變數讀取此項目，您也可以將 `workspaceId` / 傳遞`workspace_id`給用戶端建構函式。如果您使用基本`Anthropic`用戶端（而非 Claude-on-AWS 用戶端），請明確傳遞 標頭。如需如何尋找工作區 ID，請參閱針對每個語言範例和工作區提出請求。

安裝 開發套件

Anthropic 的用戶端 SDKs支援 AWS 上的 Claude 平台。所有七個 SDKs 都提供平台特定的用戶端類別；或者，您可以使用 AWS 基礎 URL 上的 Claude 平台來設定基礎用戶端。

Python

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

> **Tip**
>
> 在具有 Homebrew Python 或其他外部受管 Python 環境的 macOS 上，`pip install`可能會因為 PEP 668 `externally-managed-environment`錯誤而失敗。首先建立和啟用虛擬環境：`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**
>
> AWS 上適用於 Claude 平台的 SDK 用戶端為 Beta 版。

可用的模型

AWS 上的 Claude 平台提供與第一方 Claude API 相同的一組 Claude 模型。

如需目前模型、模型 IDs、內容視窗大小和功能的清單，請參閱 Anthropic 文件中的模型概觀。

模型 ID

AWS 上 Claude 平台上的模型 IDs 與第一方 Claude API 上所使用的模型 ID 相同（例如，、`claude-opus-4-6`、`claude-sonnet-4-6`、`claude-haiku-4-5`）。沒有 Bedrock 樣式ARNs 或字首 `anthropic.` - 完全按照 Anthropic 發佈模型 ID 的方式使用模型 ID。

提出要求

AWS 上的 Claude 平台使用 Anthropic Messages API (`/v1/messages`)，與 Claude 第一方平台相同的 API 表面。差異在於基本 URL、身分驗證方法，以及識別請求目標之工作區的必要`anthropic-workspace-id`標頭。

Shell

```
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
```

用戶端ANTHROPIC_AWS_WORKSPACE_ID會從環境讀取 AWS_REGION (或 AWS_DEFAULT_REGION) 和 。您可以透過將 aws_region / awsRegion或 workspace_id / 傳遞workspaceId至建構函數來覆寫 。區域和工作區 ID 都是必要的；如果無法從這些來源解析，建構函數會擲回 。

> **Note**
>
> 只有臨時登入資料需要 x-amz-security-token標頭 (curl)，例如 IAM 角色、SSO 或 STS。使用長期 IAM 使用者登入資料時請將其省略。SDK 用戶端會根據登入資料來源自動處理此問題。

此--aws-sigv4值遵循格式 aws:amz:<region>:<service>。SigV4 服務名稱為 aws-external-anthropic，且區域必須與端點 URL 中的區域相符。兩者中的不相符會產生一般簽章拒絕錯誤，而不是特定的診斷。

使用基本 Anthropic 用戶端

如果您不想新增 AWS 開發套件相依性，您可以使用 基礎Anthropic用戶端。此路徑使用 vanilla SDK 環境變數名稱，而不是專用用戶端讀取ANTHROPIC_AWS_*的名稱：

```
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 和 Go SDKsANTHROPIC_BASE_URL會自動讀取ANTHROPIC_API_KEY和；對於其他語言，請將它們傳遞給建構函式。在每個語言中，傳遞 anthropic-workspace-id標頭中的工作區 ID。

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);
```

功能支援

AWS 上的 Claude 平台會直接使用 Anthropic Messages API。您可以使用第一方 Claude API 取得完整的功能同位，除非功能目前不可用：

- Beta 功能：傳遞標準`anthropic-beta`標頭以存取 Beta 功能，就像使用 Claude API 一樣。
- 客服人員技能：使用預先建置和自訂的客服人員技能，搭配與 Claude API 相同的`container.skills`參數和 Beta 版標頭。所有預先建置的技能 (PowerPoint、Excel、Word、PDF) 都可以立即使用。
- 程式碼執行：使用程式碼執行工具在 Anthropic 的受管沙盒中執行程式碼。
- 工具使用：提供電腦使用和所有其他工具使用功能。
- 延伸思考：使用與 Claude API 相同的參數啟用延伸思考。
- 串流：即時回應的完整 SSE 串流支援。
- 批次處理：提交高輸送量工作負載的批次請求。
- 提示快取：快取工具、系統提示和訊息歷史記錄，以減少延遲和成本。所有提示快取功能 (5 分鐘 TTL、1 小時 TTL 和自動快取） 都可使用。
- 檔案 API：跨請求上傳和參考檔案。
- 具有 IAM 整合的工作區標籤：工作區可以加上標籤，標籤會流經 IAM 以進行屬性型存取控制，並流經 AWS 成本和用量報告以進行成本分配。工作區標記是 AWS 上 Claude 平台的 GA 功能 （這是第一方 Claude API 上的 Beta 功能）。

Claude 受管代理程式

Claude Managed Agents 可在 AWS 上的 Claude 平台上使用。代理程式、工作階段、環境、登入資料保存庫和記憶體存放區都是一流的 IAM 資源。如需動作參考，請參閱 IAM 動作；如需對應的頁面，請參閱使用 Claude 主控台。Anthropic Agent SDK 可在 AWS 上使用 Claude 平台，無需額外的整合。將其指向 AWS 基礎 URL 上的 Claude 平台，並使用 SigV4 或 API 金鑰進行驗證。

AWS 上 Claude 平台上的自治工作階段需要每 6 小時重新驗證一次。長時間執行的代理程式執行必須在該視窗中重新整理其 SigV4 登入資料或 API 金鑰，否則工作階段會結束。

下列 Claude Managed Agents 功能目前不適用於 AWS 上的 Claude 平台：

- 結果：客服人員工作階段的結果追蹤不可用。

- 多客服人員工作階段：無法使用具有多個互動客服人員的工作階段。

合規

> **Important**
>
> Anthropic 提供此服務做為第三方產品。標準 AWS 合規計劃、認證或稽核報告（例如 SOC、ISO 或 HIPAA 資格）並未涵蓋此項目。客戶全權負責執行自己的盡職調查，以確保此第三方產品符合其法規、法律和合規要求。在處理敏感或管制的資料之前，您應該透過 Anthropic Trust Center 檢閱 Anthropic 的官方合規文件。如需詳細資訊，請參閱 AWS 服務條款。

如何建立工作區成員資格的模型

在第一方 Claude API 上，存取是由users-to-workspaces成員資格管理，使用者會新增至工作區，並繼承工作區的存取權限。AWS 上的 Claude 平台會以 IAM 授權取代該模型：沒有每個工作區的使用者清單。相反地，IAM 主體（使用者或角色）會保留對特定工作區 ARNs 授予`aws-external-anthropic`動作的政策。IAM 政策評估會決定每個主體在每個工作區中可以做什麼。

透過修改 IAM 政策 (SCPs、許可界限、身分連接政策或資源層級條件），而不是在 Claude Console 中管理每個工作區成員資格，來授予和撤銷工作區存取權。如需範例，請參閱 IAM 政策。

目前無法使用的功能

下列功能目前不適用於 AWS 上的 Claude 平台：

- HIPAA 整備：Anthropic 的 HIPAA 就緒計劃不適用於 AWS 上的 Claude 平台。具有 HIPAA 要求的客戶應該改為在 Amazon Bedrock 中評估 Claude。
- 標準和批次以外的服務方案：無法使用優先方案。
- Admin API：組織成員、工作區成員、邀請、API 金鑰、用量報告、成本報告和速率限制報告端點：目前無法使用這些 Admin API 端點。工作區 CRUD 和重新命名端點 (`/v1/organizations/workspaces`) 可透過 `aws-external-anthropic` 命名空間取得；請參閱 IAM 動作。成員是透過 AWS IAM 建立模型，而不是透過工作區成員資格清單建立模型（請參閱上一節）。API 金鑰生命週期是在 AWS 上的 Claude 平台 → API 金鑰下的 AWS 主控台中管理。如需用量、成本和速率限制資料，請使用 Claude Console 檢視（請參閱監控和記錄) 或 Anthropic Usage and Cost API。

- 花費限制：無法使用。請改用 AWS 帳單控制。
- Claude Code 工作區和分析 API：無法使用具有自動速率限制的 Claude Code 工作區。Claude Code 用量會顯示在一般用量檢視中，而不是專用畫面。
- OAuth 身分驗證：不支援。使用 SigV4 或 API 金鑰身分驗證。
- OpenAI 相容 API 端點：不適用於 AWS 上的 Claude 平台。
- 工作區層級推論地理控制： `allowed_inference_geos`和 `default_inference_geo` 無法使用。請改為在每個請求`inference_geo`上設定 。

資料落地

AWS 上的 Claude 平台支援下列推論地理位置：

- 美國：推論會保留在美國資料中心內。適用 1.1 倍的定價乘數。
- 全球：推論可能會路由到全球任何由 Anthropic 操作的資料中心。適用標準定價。

使用 `inference_geo` 參數設定每個請求的推論地理：

> **Note**
>
> Claude Opus 4.6、Claude Sonnet 4.6 和更新版本支援 `inference_geo` 參數。在 Claude Opus 4.5、Claude Sonnet 4.5 或 Claude Haiku 4.5 `inference_geo`上使用 的請求會傳回 400 錯誤。如需模型可用性詳細資訊，請參閱資料駐留。

Shell

```
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
```

如果您省略 inference_geo，則請求預設為 global。

AWS 上的 Claude 平台無法使用工作區層級推論地理控制 (allowed_inference_geos 和 default_inference_geo)。請改為在每個請求inference_geo上設定 。

工作區

AWS 上 Claude 平台上的推論和資源請求以工作區為目標。您可以在這些 API 呼叫的 anthropic-workspace-id標頭中傳遞工作區的 ID。工作區 IDs使用標記格式，wrkspc_後面接著英數字元識別符 （ 例如 wrkspc_01AbCdEf23GhIj)。如果您還沒有工作區 ID，請參閱取得工作區 ID。

工作區範圍

工作區繫結至單一 AWS 區域。在 中建立的工作區us-west-2只能透過us-west-2端點存取。每個工作區的使用量、配額、成本、檔案、批次和技能都會彙總，為您提供 Claude 主控台中的每個區域明細。

Workspaces 也做為 AWS 上 Claude 平台的主要 IAM 資源。您可以使用工作區 ARN，透過 AWS IAM 政策授予或拒絕存取特定工作區。ARN 的資源區段與您在 anthropic-workspace-id標頭中傳遞的字首 ID wrkspc_相同：

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

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

如需政策範例，請參閱 IAM 政策。

建立工作區

預設工作區會在註冊時自動佈建。其他工作區可以透過/v1/organizations/workspaces端點建立、更新、重新命名和封存，並由對應的aws-external-anthropic動作 (CreateWorkspace、UpdateWorkspace、ArchiveWorkspace) 授權；請參閱 IAM 動作。

在用戶端中設定工作區 ID

每個資料平面請求都必須在 anthropic-workspace-id標頭中包含工作區 ID。當您使用 Anthropic SDK 的 Claude-on-AWS 用戶端時，有三種方式可以提供：

1. 環境變數 — 設定 ANTHROPIC_AWS_WORKSPACE_ID，用戶端會自動讀取它。

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

2. 建構器引數 — 在執行個體化用戶端時傳遞 workspaceId / workspace_id（名稱遵循 SDK 的語言慣例）。
3. 每個請求覆寫 — 如果您需要從相同的用戶端處理多個工作區，請在個別請求上直接傳遞 標頭。

如果您使用基本Anthropic用戶端（沒有 AWS 後端），請針對每個請求明確設定 anthropic-workspace-id標頭，請參閱對每個語言範例提出請求。如需開發套件特定的引數名稱，請參閱 Anthropic Client SDKs。

標記工作區

工作區標籤是連接到工作區的鍵/值對。它們與屬性型存取控制的 AWS IAM 和成本分配的 AWS 成本和用量報告整合（請參閱 CUR 詳細資訊的監控和記錄)。AWS 上的 Claude 平台上的標記是 GA。

使用 aws-external-anthropic 命名空間UntagResource中的 TagResource和 更新現有工作區上的標籤。相同的標籤索引鍵可以驅動 IAM 條件和成本分配，而無需額外的組態。

標籤型存取控制

您可以使用aws:ResourceTag/<key>條件內容索引鍵在工作區標籤值上切換aws-external-anthropic動作。下列政策僅允許標記 的工作區進行推論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"
        }
      }
    }
  ]
}
```

在 上使用 aws:RequestTag/<key>和 aws:TagKeys條件索引鍵TagResource來強制執行標記政策 （例如，要求工作區攜帶 CostCenter和 Owner標籤）。如需更多範例，請參閱 IAM 政策。

使用 Claude 主控台

AWS 上的 Claude 平台使用標準 Claude 主控台，網址為 https：//platform.claude.com。當您從 AWS 主控台登入時，由 AWS 管理的帳戶會顯示在 Claude 主控台側邊欄中。主控台範圍適用於 AWS 組織的 Claude 平台。它提供用量分析、成本明細、速率限制可見性、工作區可見性，以及用於管理檔案、客服人員技能、批次任務、客服人員、工作階段、環境、登入資料保存庫和記憶體存放區的頁面。

登入

透過 AWS IAM 聯合存取 Claude 主控台。如需完整的第一次登入流程，請參閱設定您的帳戶。簡而言之：

1. 使用 `aws-external-anthropic:AssumeConsole`許可擔任 IAM 角色。請參閱 IAM 動作。
2. 在 AWS 主控台中導覽至 AWS 上的 Claude 平台頁面。
3. 選擇開啟 Claude 主控台。AWS 主控台會發出 JWT 並將您重新導向至 `platform.claude.com`。
4. 第一次登入時，系統會提示您輸入電子郵件地址；輸入您的工作電子郵件。平台just-in-time佈建您的 Claude Console 使用者。

有兩個 Claude Console 角色可用：管理員和開發人員。管理員角色會授予 AWS 上 Claude 平台可用的所有 Claude 主控台頁面和設定的存取權。開發人員角色授予對用量、成本、速率限制和工作區資訊的讀取存取權。請聯絡您的 Anthropic 帳戶代表，將管理員或開發人員角色指派給委託人。

可用的頁面

Via AWS 閘道欄指出頁面是否透過 AWS 閘道讀取和寫入資料（因此由 IAM 動作管理）。標記為「無」的頁面 直接透過 Anthropic APIs並略過 IAM 動作檢查。

頁面	Available	透過 AWS 閘道	備註
用途	是	否	依模型、工作區和維度檢視字符用量。資料可能需要幾分鐘的時間才會在請求之後出現。
成本	是	否	依模型和工作區檢視成本明細。AWS Cost Explorer 會顯示彙總的 CCU 明細項目。

頁面	Available	透過 AWS 閘道	備註
限制	是	否	檢視速率限制（唯讀）。
工作區	是	否	檢視每個區域的工作區（唯讀）。
檔案	是	是	檢視和管理上傳的檔案。
技能	是	是	檢視和管理客服人員技能。
批次	是	是	檢視和管理批次處理任務。
客服人員	是	是	檢視和管理客服人員定義。
工作階段	是	是	檢視客服人員工作階段和事件歷史記錄。
環境	是	是	檢視和管理工作階段的雲端容器組態。
登入資料保存庫	是	是	檢視和管理工作階段身分驗證的登入資料保存庫。
記憶體存放區	是	是	檢視和管理持久性代理程式記憶體。
API 金鑰	否	N/A	在 AWS 主控台中管理 API 金鑰 (AWS 上的 Claude 平台 → API 金鑰)。請參閱身分驗證。
成員	否	N/A	不適用。AWS IAM 會管理存取。
帳單	否	N/A	不適用。AWS Marketplace 會管理帳單和發票。在成本頁面上檢視成本明細。
Claude 程式碼	否	N/A	在用量頁面上檢視 Claude Code 用量。

切換組織

Claude 主控台不支援在 AWS 上切換 Claude 平台的組織。若要存取不同的組織，請使用該組織的 AWS 帳戶的 IAM 角色，透過 AWS 主控台登出並重新驗證。

速率限制和配額

AWS 上的 Claude 平台會在您訂閱時指派第 1 層速率限制。Anthropic 會直接管理速率限制，而不是透過 AWS 配額系統。

預設限制

AWS 上的 Claude 平台使用 Anthropic 的標準方案排程，與第一方 Claude API 相同。每個工作區適用第 1 層限制。限制會依模型系列進行集區。例如，所有 Opus 模型共用一個合併限制，所有 Sonnet 模型共用另一個限制，所有 Haiku 模型共用第三個限制。

如需目前的層級 1 值 (RPM、IMC、OTPM) 和更高層級閾值，請參閱 Anthropic 文件網站上的速率限制。Anthropic 頁面是事實的來源，並在限制變更時更新。

速率限制標頭

每個回應都包含報告目前速率限制狀態的標頭。金鑰標頭：

- `anthropic-ratelimit-requests-limit` — 每分鐘請求數上限
- `anthropic-ratelimit-requests-remaining` — 仍在目前視窗中的請求
- `anthropic-ratelimit-requests-reset` — 請求限制重設的時間 (RFC 3339)
- `anthropic-ratelimit-tokens-limit` — 每分鐘合併字符（輸入 + 輸出）上限
- `anthropic-ratelimit-tokens-remaining` — 目前視窗中剩餘的合併字符
- `anthropic-ratelimit-tokens-reset` — 合併字符限制重設的時間 (RFC 3339)
- `anthropic-ratelimit-input-tokens-limit` / `-remaining` / `-reset` — Input-token-specific標頭
- `anthropic-ratelimit-output-tokens-limit` / `-remaining` / `-reset` — Output-token-specific標頭
- `retry-after` — 在 429 回應上，重試前要等待的秒數

如需完整集，請參閱 Anthropic 文件網站上的回應標頭。

請求更高的限制

與第一方 Claude API 不同，自動方案提升不適用於 AWS 上的 Claude 平台。若要請求更高的限制，請聯絡您的 Anthropic 客戶代表，並提供工作區 ID 和所需的輸送量。如需層閾值和其他詳細資訊，請參閱 Anthropic 文件網站上的速率限制。

速率限制錯誤

當您超過速率限制時，API 會傳回具有 `rate_limit_error`類型的 HTTP 429。在重試邏輯中實作具有抖動的指數退避。`retry-after` 標頭指出在重試之前要等待多少秒。

帳單

AWS 上的 Claude 平台使用 AWS Marketplace 做為以耗用量為基礎的計量計費後端。Anthropic 會每小時透過 AWS Marketplace 計量用量，AWS 會在您現有的 AWS 帳單上發出每月發票。

帳單是僅限臨時付款 （您支付所使用的費用） 和稅前付款 (AWS 會套用您帳戶的稅務設定）。

用量以 Claude 用量單位 (CCUs) 計算，每個 CCU 為 0.01 USD。CCU 價格是固定且永遠不會折扣。Anthropic 會以標準每個模型、每個功能費率計算您的字符用量，套用任何交涉折扣，然後將結果轉換為 CCUs每個 CCU 為 0.01 美元。折扣會導致計量的 CCUs較少，而不是較低的 CCU 價格。CCUs不是預付點數；沒有 CCU 餘額或承諾。請參閱 CCU 定義和每個模型字符費率的定價。

監控和記錄

AWS CloudTrail 可以擷取對 AWS 上 Claude 平台的所有請求。根據預設，工作區和保存庫操作會記錄為管理事件。所有其他操作（推論、批次、檔案、技能、模型、使用者設定檔和保存庫以外的 Claude 受管代理程式）都是資料事件。記錄它們需要明確的組態，並產生額外的 CloudTrail 費用。如需完整的事件類型分類和組態詳細資訊，請參閱 AWS CloudTrail 文件的 IAM 動作。

設定 CloudTrail 資料事件記錄時，請選取資源類型 `AWS::AWSExternalAnthropic::Workspace`。這是 AWS 工作區上 Claude 平台的 CloudTrail 資源類型，需要此類型才能擷取資料平面事件（推論、批次、檔案和其他每個工作區的操作）。如果您想要記錄一部分工作區的活動，而非整個帳戶，請將資料事件選擇器範圍限定為特定工作區 ARNs。

請求 IDs

每個回應在回應標頭中包含兩個請求 IDs：

- AWS 請求 ID (**`x-amzn-requestid`**)：在 CloudTrail 中編製索引的主要 ID。透過 AWS 工具調查請求或聯絡 AWS 支援時，請使用此選項。
- 人類請求 ID (**`request-id`**)：次要 ID。聯絡 Anthropic Support 時使用此選項。

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 SDK 目前不會公開原始回應存取器，因此無法從 Ruby 中的回應標頭讀取請求 IDs。如果您需要 request-ID 記錄，請使用上述任何其他語言，或在 HTTP 代理層擷取 IDs。

Anthropic 建議至少連續 30 天記錄您的活動，以了解使用模式並調查任何潛在問題。

> **Note**
>
> AWS CloudTrail 已在您的 AWS 帳戶中設定。啟用記錄功能並不會提供 AWS 或 Anthropic 存取您的內容，超出帳單和服務操作所需的範圍。

用量指標和儀表板

AWS 上的 Claude 平台不會將每個工作區的使用量指標發佈至 Amazon CloudWatch。符記計數、請求量和每個模型的用量，皆可透過 Anthropic 的用量表面取得：

- Claude Console 用量檢視 - 當主體與 Claude Console 聯合時 `aws-external-anthropic:AssumeConsole`（請參閱 IAM 政策），用量儀表板會顯示請求量、字符計數和可存取工作區的每個模型明細。整個帳戶的用量檢視需要管理員主控台功能。
- Anthropic Usage and Cost API — 如需以程式設計方式存取用量和成本資料，包括每個工作空間和每個模型彙總，請參閱 Anthropic 文件中的 Usage and Cost API。

對於操作監控（錯誤率、延遲、速率限制標頭），請使用每個請求傳回的回應標頭（請參閱速率限制和配額），以及 CloudTrail 中擷取的 AWS 請求 IDs。

成本分配和監控

AWS 上的 Claude 平台費用會顯示在 AWS 服務的 Claude 平台下，具有每個模型的使用維度。使用 AWS 成本和用量報告 (CUR) 結合工作區標籤，進行精細的成本分配：

1. 使用您關心的配置維度標記您的工作區（團隊、應用程式、環境、成本中心）。如需標記詳細資訊，請參閱工作區。
2. 在 AWS Billing 主控台中，將每個標籤金鑰啟用為成本分配標籤。啟用後，啟用日期當天或之後產生的用量會在 CUR 中依標籤分割。
3. 在 CUR 或 AWS Cost Explorer 中，依資料`resourceTags/user:<tag-key>`欄分組以依工作區標籤細分支出。

工作區標籤會流經 AWS 用量上所有 Claude 平台的 CUR 明細項目。相同的標籤索引鍵可同時驅動 IAM 型存取控制和成本分配。如需設定步驟和啟用延遲，請參閱 AWS Billing 文件中的成本分配標籤。

從 Amazon Bedrock 遷移

如果您目前在 Bedrock 上使用 Claude，遷移到 AWS 上的 Claude 平台需要在整個整合過程中進行變更。仍然支援 SigV4 簽署，但簽署內容、基本 URL、API 格式、模型 IDs、SDK 用戶端和套件、串流格式、請求標頭和區域可用性都會變更。下表摘要說明差異。

有哪些變更

	Amazon Bedrock	AWS 上的 Claude 平台
基本 URL	`bedrock-runtime.{region}.amazonaws.com`	`aws-external-anthropic.{region}.api.aws`
API 格式	Bedrock Converse/InvokeModel	Anthropic Messages API (`/v1/messages`)
模型 IDs	`anthropic.claude-opus-4-7-v1`	`claude-opus-4-7`
SDK 用戶端	`AnthropicBedrock` / Bedrock SDK	Beta `AnthropicAWS` 版中的平台特定用戶端 `AnthropicAws` (`AnthropicAwsClient`、、等)
SDK 套件	`anthropic[bedrock]`、`@anthropic-ai/bedrock-sdk`等。	`anthropic[aws]`、`@anthropic-ai/aws-sdk` 等（請參閱安裝 SDK)
SigV4 服務名稱	`bedrock`	`aws-external-anthropic`
串流格式	AWS EventStream	SSE（與 Claude API 相同）
工作區標頭	不適用	`anthropic-workspace-id` 必要
區域可用性	多個商業區域	所有 AWS 商業區域（選擇加入區域需要帳戶選擇加入）

您獲得什麼

- 通常可以當日存取新模型和功能，無需單獨的合作夥伴整合步驟
- 產生文件的客服人員技能 (PowerPoint、Excel、Word、PDF)
- Anthropic 受管沙盒中的程式碼執行
- 透過 `anthropic-beta`標頭的 Beta 功能 （ 請參閱目前無法使用的功能)
- 用於配額可見性和用量分析的 Claude Console
- Direct Anthropic 支援
- 做為 SigV4 替代方案的 API 金鑰身分驗證 （ 請參閱身分驗證)

保持不變的項目

- AWS IAM 身分驗證 (SigV4)
- 透過您的 AWS 帳戶計費
- AWS 承諾淘汰

遷移陷阱

> **Warning**
>
> 首先啟用傳出 Web 聯合身分。如果您的 AWS 帳戶先前未在 AWS 上使用 Claude 平台，您必須在提出請求之前，為每個帳戶啟用傳出 Web 聯合身分一次。如果沒有此步驟，所有請求都會失敗並出現聯合錯誤。Bedrock 不需要此步驟。

> **Warning**
>
> AWS 上的 Claude 平台上選擇加入零資料保留 (ZDR)。在 Bedrock 上，AWS 是資料處理者，而 Anthropic 不會保留推論輸入或輸出；Anthropic 的 ZDR 程式不適用於該處。在 AWS 上的 Claude 平台上，Anthropic 是資料處理者，ZDR 遵循第一方 Claude API 模型：可透過您的 Anthropic 帳戶代表請求取得。在遷移取決於資料保留保證的生產工作負載之前，請先確認 ZDR 註冊。

商業考量

- 服務條款：AWS 服務條款適用於 AWS 上的 Claude 平台。Anthropic 的商業服務條款、資料處理增補和用量政策也適用。
- 折扣和私有優惠：協議折扣和 AWS Marketplace 私有優惠不會在 Bedrock 和 AWS 上的 Claude 平台之間自動轉移。與您的 Anthropic 帳戶代表合作，為 AWS 上的 Claude 平台設定商業條款。

IAM 政策

AWS 上的 Claude 平台與 AWS IAM 整合以進行存取控制。您可以使用標準 IAM 政策語法，授予或拒絕特定工作區上特定 API 動作的存取權。

SigV4 服務名稱和 IAM 動作命名空間為 `aws-external-anthropic`。動作遵循 模式 `aws-external-anthropic:<Action>` (例如，`aws-external-anthropic:CreateInference`)。

範例：拒絕批次推論

下列政策允許即時推論，同時封鎖批次處理，這是 ZDR 敏感工作負載的常見需求：

```
{
  "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` 動作會同時授權批次中繼資料路由和批次結果路由。拒絕它，以及 `ListBatchInferences`，會同時封鎖讀取和批次列舉。

Allow 陳述式會列舉特定 Get*和 List*動作，而不是使用萬用字元。萬用字元會授予 GetFile（下載檔案位元組）和其他您不想要的讀取；Allow無論如何都會Deny覆寫，但明確形式是建立模型的安全模式。

範例：單一工作區上的同步推論

授予對一個生產工作區執行推論的 IAM 主體最低許可：

```
{
  "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**
>
> 此政策中的List*萬用字元也符合ListWorkspaces帳戶範圍的 。工作區 ARN 限制條件會無提示地篩選掉，因此此政策不會授權列出工作區。如果您的服務帳戶需要列舉工作區，請使用新增 ListWorkspaces的個別Allow陳述式Resource: "*"。
> 此政策假設 AWS SigV4 身分驗證。如果委託人使用 API 金鑰進行身分驗證，請同時授予 aws-external-anthropic:CallWithBearerToken（請參閱身分驗證）。

範例：每個客戶工作區隔離

將角色限制為單一工作區：

```
{
  "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**
>
> 第一個陳述式中的`aws-external-anthropic:*`萬用字元包含帳戶範圍動作 (`CreateWorkspace`、`ListWorkspaces`)，工作區 ARN 限制會以無提示方式篩選掉。這與「隔離」意圖一致 — 角色無法建立或列舉工作區 — 但政策包含沒有效果的許可。請參閱佈建自動化以取得帳戶範圍模式。
>
> 第二個陳述式會在所有資源`AssumeConsole`上授予 `CallWithBearerToken`和 ，因為兩者都是未繫結至工作區 ARN 的無路由動作。如果角色僅使用 SigV4，且永遠不會與 Claude 主控台聯合，請省略第二個陳述式。

範例：對 ZDR 敏感工作區的功能鎖定

封鎖特定工作區上的批次處理和檔案上傳，同時保持同步推論可用。當工作區處理不得保留伺服器端的零資料保留 (ZDR) 資料時很有用。將此政策與允許政策連接，例如 `AnthropicLimitedAccess`或上述的單一工作區範例；僅拒絕政策本身不會授予任何許可：

```
{
  "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**
>
> 此拒絕只會封鎖建立。除非您也列出其他檔案和批次動作，否則不會拒絕這些動作。如需工作區絕不能保留檔案或批次的完整鎖定，請同時拒絕 aws-external-anthropic:GetFile、aws-external-anthropic:ListFiles、aws-external-anthropic:DeleteFile、aws-external-anthropic:GetBatchInferenceaws-external-anthropic:ListBatchInferences、 aws-external-anthropic:CancelBatchInference和 aws-external-anthropic:DeleteBatchInference。

範例：佈建自動化

授予 CI/CD 角色建立和管理工作區所需的動作，無需任何推論許可：

```
{
  "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 和 ListWorkspaces是帳戶範圍的操作。在這些動作上指定工作區 ARN 沒有效果；請使用 Resource: "*"。

受管政策

AWS 提供常見存取模式的受管政策：

- **AnthropicFullAccess**：aws-external-anthropic:*授予所有資源。
- **AnthropicReadOnlyAccess**：在所有資源CallWithBearerToken上授予 List*、 Get*和 。
- **AnthropicInferenceAccess**：授予所有資源上的 ReadOnly 動作加上推論動作 (CreateInference、CreateBatchInferenceCancelBatchInference、、DeleteBatchInferen
- **AnthropicLimitedAccess**：授予所有資源上的AnthropicInferenceAccess動作加上所有 Claude Managed Agents 動作（代理程式、工作階段、環境、保存庫、記憶體存放區）。
- **AnthropicSelfHostedEnvironmentAccess**：授予自我託管沙盒工作者輪詢和處理環境工作項目、讀取相關環境、工作階段和技能資源，以及更新工作階段狀態所需的許可。將此政策連接到您自我託管環境工作者擔任的 IAM 角色。 AnthropicSelfHostedEnvironmentAccess是建議的單一角色預設值；如果您在不同的 IAM 主體下執行工作輪詢器和每個工作階段沙盒，您可以將這些許可分割為兩個更窄的自訂政策，以實現最低權限。

AnthropicInferenceAccess 是足以執行推論的最窄受管政策。透過 Get*和 List* 萬用字元，它授予 命名空間中每個 API 資源的讀取存取權，包括透過 下載的檔案內容GetFile，以及透過 下載的記憶體內容GetMemoryStore。它不會授予檔案或技能建立或刪除、使用者設定檔管理、工作區變動或主控台聯合。

> **Note**
>
> AnthropicReadOnlyAccess、 AnthropicInferenceAccess和 AnthropicLimitedAccess 不授予 AssumeConsole。需要聯合到 Claude Console 的委託人需要單獨授予 aws-external-anthropic:AssumeConsole - 透過 AnthropicFullAccess或自訂政策。請參閱聯合到 Claude 主控台。

> **Note**
>
> `CreateInference` 和 `CreateBatchInference` 是獨立的動作。拒絕一個不會封鎖另一個。如果您想要防止所有模型呼叫，請同時拒絕兩者。

聯合到 Claude 主控台

`aws-external-anthropic:AssumeConsole` 可讓 IAM 主體聯合到 Anthropic 操作的 Claude 主控台。對於大多數操作，主控台內的存取仍然由 IAM 管理，但管理操作的子集，主要是沒有對應 IAM API 的使用情況檢視，取決於主體聯合的功能。

有兩個功能：

- **`developer`** — 允許 AWS 開發人員在day-to-day上所需的 Claude 平台操作：從主控台執行推論、讀取工作區資料、檢視個人用量。
- **`admin`** — 還允許僅管理員主控台操作，包括非透過 IAM 動作呈現的整個帳戶用量檢視和管理設定。

使用 `AssumeConsole` 動作的條件`aws-external-anthropic:Capability`索引鍵控制主體可以請求的功能：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "aws-external-anthropic:AssumeConsole",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws-external-anthropic:Capability": "admin"
        }
      }
    }
  ]
}
```

AnthropicFullAccess 授予AssumeConsole沒有功能限制。對於任何較窄的授予 - 僅限開發人員的主控台存取或僅限管理員的存取 - 連接自訂政策，其範圍AssumeConsole與上述aws-external-anthropic:Capability條件相同。

使用承載字符呼叫

aws-external-anthropic:CallWithBearerToken 當 API 金鑰顯示為承載字符時， 會授權使用的SigV4-free 請求路徑。使用 API 金鑰進行身分驗證的任何委託人都需要在目標工作區執行此動作。無論您直接使用或ANTHROPIC_AWS_API_KEY設定 Authorization: Bearer 標頭，這都會套用。

除了推論動作 (CreateInference、 等） 之外CreateBatchInference，API 金鑰發起人也需要此項目。如果沒有 CallWithBearerToken，API 金鑰請求會在達到推論授權檢查之前遭到拒絕。SigV4 呼叫者不需要此動作。

AnthropicReadOnlyAccess、AnthropicLimitedAccess、 AnthropicInferenceAccess和AnthropicFullAccess全都包含 CallWithBearerToken。如果您撰寫 API 金鑰存取的自訂政策，請明確新增它。

傳出 Web 聯合身分 （主控台存取時需要）

Claude 主控台會在 Anthropic 基礎設施中執行，而非 AWS。當 IAM 主體呼叫時AssumeConsole，AWS STS 會發出範圍限定為 Anthropic 對象的 Web 身分字符；然後，Anthropic 主控台會接受該字符並建立聯合工作階段。為此，AWS 帳戶必須允許aws-external-anthropic對象使用傳出 Web 聯合身分。

除了 aws-external-anthropic:AssumeConsole動作之外，呼叫 的委託人AssumeConsole還需要下列 STS 許可：

- **sts:GetWebIdentityToken** — 允許發出 Anthropic 主控台使用的 Web 身分字符。
- **sts:TagGetWebIdentityToken** — 允許將工作階段標籤連接到 Web 身分字符。AWS 上的 Claude 平台使用這些標籤，將主體的功能和工作區內容傳遞至主控台。

在主控台使用者擔任的任何角色的信任政策中包含兩者，或在連接到AssumeConsole直接呼叫之使用者身分的內嵌/受管政策中AnthropicFullAccess包含兩者。 包含這兩個 STS 動作。在 sts:* SCP 或許可界限層級拒絕的環境必須明確允許這兩個動作，主控台聯合才能成功。

AWS 上 Claude 平台的 IAM 動作

透過 AWS 政策控制對 AWS 上 Claude 平台存取的 IAM 動作參考。

AWS 上的 Claude 平台使用 AWS IAM 進行存取控制。每個 API 路由都會映射到 `aws-external-anthropic` 命名空間中的 IAM 動作。此頁面列出所有動作、每個動作授權的路由，以及可用於常見存取模式的受管政策。如需平台設定和身分驗證，請參閱什麼是 AWS 上的 Claude 平台？。

服務詳細資訊

IAM 服務字首	`aws-external-anthropic`
資源類型	`workspace`
工作區 ARN	`arn:aws:aws-external-anthropic:{region}:{account-id}:workspace/{workspace-id}`

ARN 區域一律會填入，並與工作區繫結的區域相符。資源區段是已標記的工作區 ID (`wrkspc_…`)，與您在 `anthropic-workspace-id` 標頭中傳遞的值相同。

動作

此服務在 15 個群組中定義 65 個動作。動作遵循 AWS `VerbNoun`慣例並使用動詞紀律，讓 `Get*`和 `List*` 萬用字元產生乾淨的唯讀界限。

Inference

Action	已授權的路由
`CreateInference`	`POST /v1/messages`
`CountTokens`	`POST /v1/messages/count_tokens`

批次處理

Action	已授權的路由
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}

模型

Action	已授權的路由
GetModel	GET /v1/models/{id}
ListModels	GET /v1/models

檔案

Action	已授權的路由
CreateFile	POST /v1/files
GetFile	GET /v1/files/{id} GET /v1/files/{id}/content
ListFiles	GET /v1/files
DeleteFile	DELETE /v1/files/{id}

> **Note**
>
> GetFile 同時授權中繼資料和內容下載。具有唯讀存取權的主體可以下載檔案位元組，而不只是列出檔案。

技能

Action	已授權的路由
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**
>
> 刪除個別技能版本會映射至 UpdateSkill，而非 DeleteSkill。拒絕的政策aws-external-anthropic:Delete*仍然允許版本刪除。如果您需要防止任何技能變動，請拒絕 CreateSkill UpdateSkill和 。

使用者設定檔

Action	已授權的路由
CreateUserProfile	POST /v1/user_profiles

Action	已授權的路由
GetUserProfile	GET /v1/user_profiles/{id}
ListUserProfiles	GET /v1/user_profiles
UpdateUserProfile	POST /v1/user_profiles/{id}

> **Warning**
>
> IAM 動作比對不區分大小寫。萬用字元aws-external-anthropic:*File符合 CreateFile、 GetFile和 DeleteFile，但不符合 ListFiles（以「檔案」結尾，而非「檔案」）。它也會過度比對 CreateUserProfile、和 GetUserProfile，UpdateUserProfile因為 "Profile" 結尾為 "file"。如果您打算僅授予或拒絕檔案 API 動作，請明確列舉它們 (CreateFile、GetFile、ListFiles、DeleteFile)，而不是使用*File尾碼模式。

客服人員

Claude Managed Agents 的代理程式定義。

Action	已授權的路由
CreateAgent	客服人員建立路由
GetAgent	客服人員讀取路由
ListAgents	客服人員清單路由
UpdateAgent	客服人員更新路由
ArchiveAgent	代理程式封存路由

工作階段

客服人員工作階段和事件歷史記錄。

Action	已授權的路由
CreateSession	工作階段建立路由
GetSession	工作階段讀取路由
ListSessions	工作階段清單路由
UpdateSession	工作階段更新路由
ArchiveSession	工作階段封存路由
DeleteSession	工作階段刪除路由

環境

工作階段的雲端容器組態。

Action	已授權的路由
CreateEnvironment	環境建立路由
GetEnvironment	環境讀取路由
ListEnvironments	環境清單路由
UpdateEnvironment	環境更新路由
ArchiveEnvironment	環境封存路由
DeleteEnvironment	環境刪除路由
ProcessEnvironmentWork	自我託管環境工作者路由

保存庫

工作階段身分驗證的登入資料保存庫。

Action	已授權的路由
`CreateVault`	保存庫建立路由
`GetVault`	保存庫讀取路由
`ListVaults`	保存庫清單路由
`UpdateVault`	保存庫更新路由
`ArchiveVault`	保存庫封存路由
`DeleteVault`	保存庫刪除路由

> **Note**
>
> 保存庫操作會分類為 CloudTrail 管理事件（而非資料事件），因為保存庫會保留登入資料，並受益於預設的稽核記錄。其他 Claude Managed Agents 操作（代理程式、工作階段、環境、記憶體存放區）是資料事件。

記憶體存放區

持久性代理程式記憶體。

Action	已授權的路由
`CreateMemoryStore`	記憶體存放區建立路由
`GetMemoryStore`	記憶體存放區讀取路由
`ListMemoryStores`	記憶體存放區清單路由
`UpdateMemoryStore`	記憶體存放區更新路由
`ArchiveMemoryStore`	記憶體存放區封存路由
`DeleteMemoryStore`	記憶體存放區刪除路由

> **Note**
>
> GetMemoryStore 會讀取記憶體內容。因此，受管或自訂政策中的Get*萬用字元會授予記憶體讀取存取權。如果必須限制記憶體內容，請明確範圍政策。

Webhooks

工作階段的生命週期事件通知。

Action	已授權的路由
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

工作區

Action	已授權的路由
CreateWorkspace	POST /v1/organizations/workspaces
GetWorkspace	GET /v1/organizations/workspaces/{id}
ListWorkspaces	GET /v1/organizations/workspaces
UpdateWorkspace	POST /v1/organizations/workspaces/{id}

Action	已授權的路由
`ArchiveWorkspace`	`POST /v1/organizations/workspaces/{id}/archive`

預設工作區是在註冊時佈建；請參閱工作區。

身分驗證

Action	已授權的路由
`CallWithBearerToken`	(無)

`CallWithBearerToken` 是一種身分驗證層許可，可授權委託人透過 API 金鑰（承載字符）而非 AWS SigV4 進行身分驗證。它不會映射到路由。將其與您希望 API 金鑰持有者執行的路由映射動作一起授予。

主控台存取

Action	已授權的路由
`AssumeConsole`	(無)

`AssumeConsole` 授權委託人透過 AWS 主控台聯合流程，為 AWS 工作區上的 Claude 平台開啟 Claude 主控台。它不會映射到路由。將其授予應能夠在 AWS 主控台的 Claude 平台上按一下 Open Claude Console 的主體。`AssumeConsole` 動作上的`aws-external-anthropic:Capability`條件索引鍵可控制主體可以請求的主控台功能（開發人員或管理員）。如需詳細資訊，請參閱 IAM 政策和使用 Claude 主控台進行登入流程。

> **Warning**
>
> `aws-external-anthropic:AssumeConsole` 會發出持續長達 12 小時的 Claude Console 工作階段，與發起人自己的工作階段持續時間無關。IAM 工作階段少於 12 小時的發起人仍然可以取得其來源登入資料過期的主控台工作階段。將此許可限制為需要 Claude Console 存取的主體，並在不再需要時立即將其撤銷。

> **Note**
>
> 聯合之後在 Claude 主控台內執行的動作不會記錄在 AWS CloudTrail 中或透過 IAM 歸因。這包括全域工作區範圍的活動，例如檢視用量報告。如果您需要主控台活動的稽核線索，請使用 Claude Console 中提供的稽核日誌，而不是 CloudTrail。

Route-to-action映射

下表列出 AWS 上 Claude 平台上的每個路由，以及呼叫它所需的 IAM 動作。穩定路由的 IAM 動作也會授權使用 `anthropic-beta`標頭的請求。CloudTrail 會將每個動作分類為資料事件 （大量、資料平面操作） 或管理事件 （控制平面操作）。

Method	路由	IAM 動作	CloudTrail 事件類型
POST	/v1/messages	CreateInference	資料
POST	/v1/messages/count_tokens	CountTokens	資料
POST	/v1/messages/batches	CreateBatchInference	資料
GET	/v1/messages/batches	ListBatchInferences	資料
GET	/v1/messages/batches/{id}	GetBatchInference	資料
GET	/v1/messages/batches/{id}/results	GetBatchInference	資料
POST	/v1/messages/batches/{id}/cancel	CancelBatchInference	資料
DELETE	/v1/messages/batches/{id}	DeleteBatchInference	資料
GET	/v1/models	ListModels	資料

Method	路由	IAM 動作	CloudTrail 事件類型
GET	/v1/models/{id}	GetModel	資料
POST	/v1/files	CreateFile	資料
GET	/v1/files	ListFiles	資料
GET	/v1/files/{id}	GetFile	資料
GET	/v1/files/{id}/content	GetFile	資料
DELETE	/v1/files/{id}	DeleteFile	資料
POST	/v1/skills	CreateSkill	資料
GET	/v1/skills	ListSkills	資料
GET	/v1/skills/{id}	GetSkill	資料
DELETE	/v1/skills/{id}	DeleteSkill	資料
POST	/v1/skills/{id}/versions	UpdateSkill	資料
GET	/v1/skills/{id}/versions	GetSkill	資料
GET	/v1/skills/{id}/versions/{version}	GetSkill	資料
GET	/v1/skills/{id}/versions/{version}/content	GetSkill	資料
DELETE	/v1/skills/{id}/versions/{version}	UpdateSkill	資料
POST	/v1/user_profiles	CreateUserProfile	資料
GET	/v1/user_profiles	ListUserProfiles	資料
GET	/v1/user_profiles/{id}	GetUserProfile	資料

Method	路由	IAM 動作	CloudTrail 事件類型
POST	/v1/user_profiles/{id}	UpdateUserProfile	資料
POST	/v1/organizations/workspaces	CreateWorkspace	管理
GET	/v1/organizations/workspaces	ListWorkspaces	管理
GET	/v1/organizations/workspaces/{id}	GetWorkspace	管理
POST	/v1/organizations/workspaces/{id}	UpdateWorkspace	管理
POST	/v1/organizations/workspaces/{id}/archive	ArchiveWorkspace	管理

Claude Managed Agents 路由（代理程式、工作階段、環境、保存庫、記憶體存放區）遵循相同的 route-to-action模式；如需完整的每個路由映射，請參閱 Anthropic IAM 動作參考。保存庫路由是管理事件；客服人員、工作階段、環境和記憶體存放區路由是資料事件。

閘道預設會拒絕 AWS 上不在 Claude 平台上的路由。

另請參閱

- IAM 政策，例如 政策和 受管政策
- IAM 政策語法和評估邏輯的 AWS IAM 使用者指南
- 稽核記錄組態的 AWS CloudTrail 使用者指南

文件歷史紀錄

下表說明 AWS 使用者指南上 Claude 平台的重要變更。

變更	描述	日期
初次出版	在 AWS 使用者指南上發佈 Claude 平台。	2026 年 5 月 7 日

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。