aws

開發人員指南

AMB 存取多邊形

AMB 存取多邊形: 開發人員指南

Table of Contents

..... v
關於 AMB Access Polygon 1
第一次 AMB Access Polygon 使用者的資源 1
重要概念 2
考量和限制 2
設定 5
使用 AMB Access Polygon 的先決條件 5
註冊 AWS 5
建立具有適當許可的 IAM 使用者 5
安裝和設定 AWS Command Line Interface 6
開始使用 7
建立 IAM 政策 7
主控台 RPC 範例 8
awscurl RPC 範例 9
Node.js RPC 範例 10
傳送交易 15
讀取交易 16
字符型存取 18
建立以字符為基礎的存取權杖 18
檢視 Accessor 字符詳細資訊 19
刪除 Accessor 字符 20
JSON-RPC 和 API 22
多邊形使用案例 29
分析多邊形 NFT 資料 29
支援 NFT 購買 29
建立多邊形錢包 29
錢包即服務 30
權杖門控體驗 30
教學課程 31
安全 32
資料保護 32
資料加密 33
傳輸中加密 33
身分與存取管理 34

目標對象 34
使用身分驗證 34
使用政策管理存取權 36
Amazon Managed Blockchain (AMB) Access Polygon 如何與 IAM 搭配使用 37
身分型政策範例 42
疑難排解 45
CloudTrail 日誌 48
CloudTrail 中的 AMB Access Polygon 資訊 48
了解 AMB Access Polygon 日誌檔案項目 49
使用 CloudTrail 追蹤多邊形 JSON-RPCs 49
文件歷史紀錄 52

Amazon Managed Blockchain (AMB) Access Polygon 處於預覽版本，可能會有所變更。

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。

什麼是 Amazon Managed Blockchain (AMB) Access Polygon？

Amazon Managed Blockchain (AMB) Access Polygon 是一項全受管服務，可協助您在 Polygon 區塊鏈上建置彈性 Web3 應用程式。AMB Access Polygon 提供 Polygon 區塊鏈的即時無伺服器存取。

Polygon 是一種擴展解決方案，使用 Ethereum Virtual Machine (EVM) 作為基礎。Polygon 區塊鏈以高交易輸送量和低交易費用著稱。Polygon 區塊鏈使用 proof-of-stake共識機制。多邊形常用於建置與 NFTs、Web3 遊戲和字符化使用案例等相關的分散式應用程式 (dApps。

本指南說明如何使用 Amazon Managed Blockchain (AMB) Access Polygon 建立和管理 Polygon 區塊鏈資源。

第一次 AMB Access Polygon 使用者的資源

如果這是您第一次使用 AMB Access Polygon，建議您先閱讀下列章節：

- 重要概念：Amazon Managed Blockchain (AMB) Access Polygon
- Amazon Managed Blockchain (AMB) Access Polygon 入門
- AMB Access Polygon 支援的受管區塊鏈 API 和 JSON-RPCs

重要概念：Amazon Managed Blockchain (AMB) Access Polygon

> **Note**
>
> 本指南假設您熟悉多邊形不可或缺的概念。這些概念包括 staking、dApps、交易、錢包、智慧合約、多邊形 (POL、先前為 MATIC) 等。在使用 Amazon Managed Blockchain (AMB) Access Polygon 之前，建議您檢閱多邊形開發文件和多邊形 wiki。

Amazon Managed Blockchain (AMB) Access Polygon 可讓您無伺服器存取 Polygon Mainnet 和 Polygon Mainnet 網路，而不需要您佈建和管理任何 Polygon 基礎設施，包括節點。網路上的多邊形節點會共同存放多邊形區塊鏈狀態、驗證交易，並參與共識以變更區塊鏈狀態。您可以使用此受管服務快速且隨需地存取多邊形網路，進而降低整體擁有成本。

使用 AMB Access Polygon，您可以存取 JSON 遠端程序 (JSON-RPC) 呼叫。您可以叫用多邊形 JSON-RPCs，透過受管區塊鏈管理的節點與多邊形區塊鏈通訊。您可以使用 AMB Access Polygon 服務來開發和使用與 Polygon 區塊鏈互動的分散式應用程式 (dApps。dApps 不可或缺的一部分是智慧合約。您可以使用 AMB Access Polygon，在多邊形區塊鏈中建立和部署智慧合約。您也可以透過針對 AMB Access Polygon 端點調用 JSON-RPCs 來檢查錢包的餘額、交易詳細資訊、預估費用等，這些端點在多邊形網路對等的所有節點之間以分散式方式執行。多邊形網路的任何對等都可以開發和部署智慧型合約。

> **Important**
>
> 您負責建立、維護、使用和管理多邊形地址。您也需負責 Polygon 地址的內容。對於在 Amazon Managed Blockchain 上使用 Polygon 節點部署或呼叫的任何交易 AWS ， 概不負責。

使用 Amazon Managed Blockchain (AMB) Access Polygon 的考量和限制

當您使用 Amazon Managed Blockchain (AMB) Access Polygon 時，請考慮下列事項：

- 支援的多邊形網路

AMB Access Polygon 支援下列公有網路：

- Mainnet - 公有多邊形區塊鏈，受到proof-of-stake共識的保護，以及在其中發行和交易多邊形 (POL) 字符。Mainnet 上的交易具有實際值（即會產生實際成本），並會記錄在公有區塊鏈中。

- Polygon 不再支援的網路
  - 如 Polygon Labs 所傳達，孟買 Testnet 網路將於 4 月中日落下。根據此新聞，AMB Access Polygon 已於 2024 年 4 月 15 日結束對孟買 Testnet 的支援。建議您將 Amoy Testnet 用於測試工作負載。
  - 不支援私有網路。
  - 此外，AMB Access Polygon 不包含對 Polygon zkEVM 網路的支援。
- 與熱門第三方程式設計程式庫的相容性

  AMB Access Polygon 與熱門的程式設計程式庫相容，例如 ethers.js，可讓開發人員使用熟悉的工具與 Polygon 區塊鏈互動，輕鬆整合現有實作或快速開發新應用程式。
- 支援的區域

  此服務僅支援美國東部（維吉尼亞北部）區域。
- 服務端點

  以下是 AMB Access Polygon 的服務端點。若要與服務連線，您必須使用包含其中一個支援區域的端點。
  - `mainnet.polygon.managedblockchain.us-east-1.amazonaws.com`
- 不支援儲存

  AMB Access Polygon 不支援 Polygon (POL) 驗證器節點來提供proof-of-stake。
- Signature 第 4 版簽署多邊形 JSON-RPC 請求

  在 Amazon Managed Blockchain 上呼叫 Polygon JSON-RPCs 時，您可以透過使用 Signature 第 4 版簽署程序驗證的 HTTPS 連線來執行此操作。這表示只有帳戶中的授權 IAM AWS 主體才能進行多邊形 JSON-RPC 呼叫。若要這樣做，必須隨呼叫提供 AWS 憑證（存取金鑰 ID 和私密存取金鑰）。

  > **Important**
  > - 請勿在面向使用者的應用程式中嵌入用戶端登入資料。

- 您無法使用 IAM 政策來限制對個別多邊形 JSON-RPCs存取。

- 支援以字符為基礎的存取

  您也可以使用 Accessor 權杖對多邊形網路端點進行 JSON-RPC 呼叫，做為 Signature 第 4 版 (SigV4) 簽署程序的便利替代方案。您必須`BILLING_TOKEN`從您建立的其中一個配件字符提供 ，並在呼叫中新增 做為參數。

  > **Important**
  > - 如果您將安全性和可稽核性優先於便利性，請改用 SigV4 簽署程序。
  > - 您可以使用 Signature 第 4 版 (SigV4) 和字符型存取來存取 Polygon JSON-RPCs。不過，如果您選擇使用這兩個通訊協定，您的請求會遭到拒絕。
  > - 您絕對不能在面向使用者的應用程式中嵌入存取器字符。

- 僅支援提交原始交易

  使用 `eth_sendrawtransaction` JSON-RPC 提交更新 Polygon 區塊鏈狀態的交易。

設定 Amazon Managed Blockchain (AMB) 存取多邊形

第一次使用 Amazon Managed Blockchain (AMB) Access Polygon 之前，請遵循本節中的步驟建立 AWS 帳戶。下一章討論如何開始使用 AMB Access Polygon。

使用 AMB Access Polygon 的先決條件

AWS 第一次使用 之前，您必須擁有 AWS 帳戶。

註冊 AWS

當您註冊 時 AWS，您的 AWS 帳戶 會自動註冊所有 AWS 服務，包括 Amazon Managed Blockchain (AMB) Access Polygon。您只需針對所使用的服務付費。

如果您已有 AWS 帳戶 ，請前往下一個步驟。如果您還沒有 AWS 帳戶，請使用下列程序建立新帳戶。

建立 AWS 帳戶

1. 開啟 https://portal.aws.amazon.com/billing/signup。
2. 請遵循線上指示進行。

   部分註冊程序需接收來電或簡訊，並在電話鍵盤輸入驗證碼。

   當您註冊 時 AWS 帳戶，AWS 帳戶根使用者會建立 。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。作為安全最佳實務，請將管理存取權指派給使用者，並且僅使用根使用者來執行需要根使用者存取權的任務。

建立具有適當許可的 IAM 使用者

若要建立和使用 AMB Access Polygon，您必須具有允許必要受管區塊鏈動作的 AWS Identity and Access Management (IAM) 委託人 （使用者或群組）。

在 Amazon Managed Blockchain 上呼叫 Polygon JSON-RPCs 時，您可以透過使用 Signature 第 4 版簽署程序驗證的 HTTPS 連線來執行此操作。這表示只有帳戶中的授權 IAM AWS 主體才能進行多邊形 JSON-RPC 呼叫。若要這樣做，必須隨呼叫提供 AWS 憑證 （存取金鑰 ID 和私密存取金鑰）。

您也可以使用 Accessor 權杖對多邊形網路端點進行 JSON-RPC 呼叫，做為 Signature 第 4 版 (SigV4) 簽署程序的便利替代方案。您必須`BILLING_TOKEN`從您建立的其中一個配件字符提供 ，並在呼叫中

新增 做為參數。不過，您仍然需要 IAM 存取權，才能取得使用 AWS 管理主控台 AWS CLI和 SDK 建立配件字符的許可。

如需如何建立 IAM 使用者的資訊，請參閱在 AWS 帳戶中建立 IAM 使用者。如需如何將許可政策連接至使用者的詳細資訊，請參閱變更 IAM 使用者的許可。如需可用於授予使用者許可以使用 AMB Access Polygon 的許可政策範例，請參閱 Amazon Managed Blockchain (AMB) Access Polygon 的身分型政策範例。

安裝和設定 AWS Command Line Interface

如果您尚未這麼做，請安裝 latest AWS Command Line Interface (AWS CLI) 以使用來自終端機 AWS 的資源。如需詳細資訊，請參閱安裝或更新最新版本的 AWS CLI。

> **Note**
>
> 對於 CLI 存取，您需要存取金鑰 ID 和私密存取金鑰。盡可能使用臨時憑證，而不是長期存取金鑰。臨時憑證包含存取金鑰 ID、私密存取金鑰，以及指出憑證何時到期的安全符記。如需詳細資訊，請參閱《IAM 使用者指南》中的將臨時登入資料與 AWS 資源搭配使用。

Amazon Managed Blockchain (AMB) Access Polygon 入門

使用本節中的資訊和程序，開始使用 Amazon Managed Blockchain (AMB) Access Polygon。

主題

- 建立 IAM 政策以存取 Polygon 區塊鏈網路
- 使用 在 AMB Access RPC 編輯器上提出多邊形遠端程序呼叫 (RPC) 請求 AWS 管理主控台
- awscurl 使用 在 中提出 AMB Access Polygon JSON-RPC 請求 AWS CLI
- 在 Node.js 中提出多邊形 JSON-RPC 請求

建立 IAM 政策以存取 Polygon 區塊鏈網路

若要存取 Polygon Mainnet 的公有端點以進行 JSON-RPC 呼叫，您必須擁有具有 Amazon Managed Blockchain (AWS_ACCESS_KEY_IDAMBAWS_SECRET_ACCESS_KEY) Access Polygon 適當 IAM 許可的使用者登入資料 (和)。在已安裝 AWS CLI 的終端機中，執行下列命令來建立 IAM 政策，以存取兩個多邊形端點：

```
cat <<EOT > ~/amb-polygon-access-policy.json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid" : "AMBPolygonAccessPolicy",
            "Effect": "Allow",
            "Action": [
                "managedblockchain:InvokeRpcPolygon*"
            ],
            "Resource": "*"
        }
    ]
}
EOT
aws iam create-policy --policy-name AmazonManagedBlockchainPolygonAccess --policy-
document file://$HOME/amb-polygon-access-policy.json
```

> **Note**
>
> 上一個範例可讓您存取所有可用的多邊形網路。若要存取特定端點，請使用下列Action命令：
>
> - "managedblockchain:InvokeRpcPolygonMainnet"

建立政策後，將該政策連接至 IAM 使用者的角色，讓政策生效。在 中 AWS 管理主控台，導覽至 IAM 服務，並將政策連接至指派給 IAM 使用者AmazonManagedBlockchainPolygonAccess的角色。

使用 在 AMB Access RPC 編輯器上提出多邊形遠端程序呼叫 (RPC) 請求 AWS 管理主控台

您可以使用 AWS 管理主控台 AMB Access Polygon 在 上編輯、設定和提交遠端程序呼叫 (RPCs)。透過這些 RPCs，您可以在多邊形網路上讀取資料和寫入交易，包括擷取資料和將交易提交至多邊形網路。

Example

下列範例示範如何使用 `eth_getBlockByNumber` RPC 取得最新區塊的相關資訊。將反白顯示的變數變更為您自己的輸入，或選擇列出的其中一個 RPC 方法，然後輸入所需的相關輸入。

1. 開啟位於 https：//https://console.aws.amazon.com/managedblockchain/ 的 Managed Blockchain 主控台。
2. 選擇 RPC 編輯器。
3. 在請求區段中，選擇 *POLYGON_MAINNET*做為*#####*。
4. 選擇 *eth_getBlockByNumber*作為 RPC 方法。
5. 輸入 **latest**做為*####*，然後選擇 *False*做為完整交易旗標。
6. 然後，選擇提交 RPC。
7. 您可以在回應區段中取得 `latest`區塊的結果。然後，您可以複製完整的原始交易以進行進一步分析，或在應用程式的商業邏輯中使用 。

如需詳細資訊，請參閱 AMB Access Polygon 支援的 RPCs

awscurl 使用 在 中提出 AMB Access Polygon JSON-RPC 請求 AWS CLI

Example

使用 Signature 第 4 版 (SigV4) 使用您的 IAM 使用者憑證簽署請求，以便對 AMB Access Polygon 端點發出多邊形 JSON-RPC 請求。awscurl 命令列工具可協助您使用 SigV4 簽署對 AWS 服務的請求。如需詳細資訊，請參閱 awscurl https：//README.md。

使用適用於您作業系統的 方法awscurl安裝 。在 macOS 上，建議使用 HomeBrew 應用程式：

```
brew install awscurl
```

如果您已安裝並設定 AWS CLI，您的 IAM 使用者登入資料和預設值 AWS 區域 會在您的環境中設定，並可存取 awscurl。使用 awscurl，透過叫用 eth_getBlockByNumber RPC 向 Polygon Mainnet 提交請求。此呼叫接受對應至您要擷取資訊的區塊號碼的字串參數。

下列命令會使用params陣列中的區塊號碼，從 Polygon Mainnet 擷取區塊資料，以選取要擷取標頭的特定區塊。

```
awscurl -X POST -d '{ "jsonrpc": "2.0", "id": "eth_getBlockByNumber-curltest",
 "method":"eth_getBlockByNumber", "params":["latest", false] }' --service
 managedblockchain https://mainnet.polygon.managedblockchain.us-east-1.amazonaws.com -k
```

> **Tip**
>
> 您也可以使用 curl和使用字符的 AMB Access Token Accessor 型存取功能提出相同的請求。如需詳細資訊，請參閱建立和管理字符型存取的存取器字符，以發出 AMB Access Polygon 請求。
>
> ```
> curl -X POST -d '{"jsonrpc":"2.0", "id": "eth_getBlockByNumber-curltest",
>  "method":"eth_getBlockByNumber", "params":["latest", false] }'
>  'https://mainnet.polygon.managedblockchain.us-east-1.amazonaws.com?
> billingtoken=your-billing-token'
> ```

任一命令的回應都會傳回最新區塊的相關資訊。如需說明，請參閱下列範例：

```
{"error":null,"id":"eth_getBlockByNumber-curltest","jsonrpc":"1.0",
      "result":{"baseFeePerGas":"0x873bf591e","difficulty":"0x18",
      "extraData":"0xd78301000683626f7288676f312e32312e32856c696e757800000000000000000009a
\
      423a58511085d90eaf15201a612af21ccbf1e9f8350455adaba0d27eff0ecc4133e8cd255888304cc
\
      67176a33b451277c2c3c1a6a6482d2ec25ee1573e8ba000",
      "gasLimit":"0x1c9c380","gasUsed":"0x14ca04d",
      "hash":"0x1ee390533a3abc3c8e1306cc1690a1d28d913d27b437c74c761e1a49********;",
      "nonce":"0x0000000000000000","number":"0x2f0ec4d",

 "parentHash":"0x27d47bc2c47a6d329eb8aa62c1353f60e138fb0c596e3e8e9425de163afd6dec",

 "receiptsRoot":"0x394da96025e51cc69bbe3644bc4e1302942c2a6ca6bf0cf241a5724c74c063fd",

 "sha3Uncles":"0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347",
      "size":"0xbd6b",
      "stateRoot":"0x7ca9363cfe9baf4d1c0dca3159461b2cca8604394e69b30af05d7d5c1beea6c3",
      "timestamp":"0x653ff542",
      "totalDifficulty":"0x33eb01dd","transactions":[...],

 "transactionsRoot":"0xda1602c66ffd746dd470e90a47488114a9d00f600ab598466ecc0f3340b24e0c",
      "uncles":[]}}
```

在 Node.js 中提出多邊形 JSON-RPC 請求

您可以使用 HTTPS 提交簽署的請求，以使用 Node.js 中的原生 https 模組存取 Polygon Mainnet 網路，或使用第三方程式庫，例如 AXIOS，來叫用 Polygon JSON-RPCs。 https://nodejs.org/api/https.html下列 Node.js 範例示範如何使用 Signature 第 4 版 (SigV4) 和字符型存取，對 AMB Access Polygon 端點提出 Polygon JSON-RPC 請求。第一個範例會將交易從一個地址傳送至另一個地址，而下列範例會從區塊鏈請求交易詳細資訊和平衡資訊。

Example

若要執行此範例 Node.js 指令碼，請套用下列先決條件：

1. 您必須在機器上安裝節點版本管理員 (nvm) 和 Node.js。您可以在此處找到作業系統的安裝說明。
2. 使用 `node --version`命令並確認您正在使用 Node 版本 18 或更新版本。如有需要，您可以使用 `nvm install v18.12.0`命令，後面接著 `nvm use v18.12.0`命令，安裝 Node 的 LTS 版本 18 版。

3. 環境變數 AWS_ACCESS_KEY_ID和 AWS_SECRET_ACCESS_KEY必須包含與您 帳戶相關聯的登入資料。

   使用下列命令，將這些變數匯出為用戶端上的字串。將下列字串中的紅色值取代為 IAM 使用者帳戶的適當值。

```
export AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"
export AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
```

完成所有先決條件後，請使用您偏好的程式碼編輯器，將下列檔案複製到本機環境中的目錄：

package.json

```
{
  "name": "polygon-rpc",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "ethers": "^6.8.1",
    "@aws-crypto/sha256-js": "^5.2.0",
    "@aws-sdk/credential-provider-node": "^3.360.0",
    "@aws-sdk/protocol-http": "^3.357.0",
    "@aws-sdk/signature-v4": "^3.357.0",
    "axios": "^1.6.2"
  }
}
```

dispatch-evm-rpc.js

```
const axios = require("axios");
const SHA256 = require("@aws-crypto/sha256-js").Sha256;
const defaultProvider = require("@aws-sdk/credential-provider-node").defaultProvider;
const HttpRequest = require("@aws-sdk/protocol-http").HttpRequest;
const SignatureV4 = require("@aws-sdk/signature-v4").SignatureV4;
```

```
// define a signer object with AWS service name, credentials, and region
const signer = new SignatureV4({
  credentials: defaultProvider(),
  service: "managedblockchain",
  region: "us-east-1",
  sha256: SHA256,
});
const rpcRequest = async (rpcEndpoint, rpc) => {

  // parse the URL into its component parts (e.g. host, path)
  let url = new URL(rpcEndpoint);

  // create an HTTP Request object
  const req = new HttpRequest({
    hostname: url.hostname.toString(),
    path: url.pathname.toString(),
    body: JSON.stringify(rpc),
    method: "POST",
    headers: {
      "Content-Type": "application/json",
      "Accept-Encoding": "gzip",
      host: url.hostname,
    },
  });

  // use AWS SignatureV4 utility to sign the request, extract headers and body
  const signedRequest = await signer.sign(req, { signingDate: new Date() });

  try {
    //make the request using axios
    const response = await axios({
      ...signedRequest,
      url: url,
      data: req.body,
    });
    return response.data;
  } catch (error) {
    console.error("Something went wrong: ", error);
  }
};

module.exports = { rpcRequest: rpcRequest };
```

sendTx.js

> **Warning**
>
> 下列程式碼使用硬式編碼私有金鑰來產生錢包 Signer，使用 Ethers.js僅是為了示範。請勿在生產環境中使用此程式碼，因為它具有實際資金並構成安全風險。
> 如有需要，請聯絡您的客戶團隊，以提供錢包和 Signer 最佳實務的建議。

```
const ethers = require("ethers");

//set AMB Access Polygon endpoint using token based access (TBA)
let token = "your-billing-token"
let url = `https://mainnet.polygon.managedblockchain.us-east-1.amazonaws.com?
billingtoken=${token}`;

//prevent batch RPCs
let options = {
  batchMaxCount: 1,
};

//create JSON RPC provider with AMB Access endpoint and options
let provider = new ethers.JsonRpcProvider(url, null, options);

let sendTx = async (to) => {
  //create an instance of the Wallet class with a private key
  //DO NOT USE A WALLET YOU USE ON MAINNET, NEVER USE A RAW PRIVATE KEY IN PROD
  let pk = "wallet-private-key";
  let signer = new ethers.Wallet(pk, provider);

  //use this wallet to send a transaction of POL from one address to another
  const tx = await signer.sendTransaction({
    to: to,
    value: ethers.parseUnits("0.0001", "ether"),
  });

  console.log(tx);
};

sendTx("recipent-address");
```

readTx.js

```
let rpcRequest = require("./dispatch-evm-rpc").rpcRequest;
let ethers = require("ethers");

let getTxDetails = async (txHash) => {
  //set url to a Signature Version 4 endpoint for AMB Access
  let url = "https://mainnet.polygon.managedblockchain.us-east-1.amazonaws.com";

  //set RPC request body to get transaction details
  let getTransactionByHash = {
    id: "1",
    jsonrpc: "2.0",
    method: "eth_getTransactionByHash",
    params: [txHash],
  };

  //make RPC request for transaction details
  let txDetails = await rpcRequest(url, getTransactionByHash);

  //set RPC request body to get recipient user balance
  let getBalance = {
    id: "2",
    jsonrpc: "2.0",
    method: "eth_getBalance",
    params: [txDetails.result.to, "latest"],
  };

  //make RPC request for recipient user balance
  let recipientBalance = await rpcRequest(url, getBalance);

  console.log("TX DETAILS: ", txDetails.result, "BALANCE: ",
 ethers.formatEther(recipientBalance.result));
};

getTxDetails("your-transaction-id");
```

將這些檔案儲存至您的目錄後，請使用下列命令安裝執行程式碼所需的相依性：

```
npm install
```

在 Node.js 中傳送交易

上述範例透過簽署交易並使用 AMB Access Polygon 將其廣播至 Polygon Mainnet，將原生 Polygon Mainnet 字符 (POL) 從一個地址傳送至另一個地址。若要這樣做，請使用 指令碼，該`sendTx.js`指令碼使用 `Ethers.js`，這是與多邊形等 Ethereum 和 Ethereum 相容區塊鏈互動的熱門程式庫。您需要在以紅色反白的程式碼中取代三個變數，包括`billingToken`用於字符型存取之配件字符的 、您用來簽署交易的私有金鑰，以及接收 POL 的收件人地址。

> **Tip**
>
> 我們建議您為此目的建立新的私有金鑰 （錢包），而不是重複使用現有的錢包，以避免遺失資金的風險。您可以使用 Ethers 程式庫的錢包類別方法 createRandom() 來產生錢包以供測試。此外，如果您需要從 Polygon Mainnet 請求 POL，您可以使用公有 POL 水龍頭請求少量用於測試。

將 `billingToken`、有資金錢包的私有金鑰和收件人的地址新增至程式碼後，您可以執行下列程式碼來簽署交易，讓 .0001 POL 從您的地址傳送到另一個地址，並使用 AMB Access Polygon 將其廣播至 Polygon Mainnet 叫用 `eth_sendRawTransaction` JSON-RPC。

```
node sendTx.js
```

收到的回應類似以下內容：

```
TransactionResponse {
provider: JsonRpcProvider {},
blockNumber: null,
blockHash: null,
index: undefined,
hash: '0x8d7538b4841261c5120c0a4dd66359e8ee189e7d1d34ac646a1d9923********',
type: 2,
to: '0xd2bb4f4f1BdC4CB54f715C249Fc5a991********',
from: '0xcf2C679AC6cb7de09Bf6BB6042ecCF05********',
nonce: 2,
gasLimit: 21000n,
gasPrice: undefined,
maxPriorityFeePerGas: 16569518669n,
maxFeePerGas: 16569518685n,
data: '0x',
value: 100000000000000n,
```

```
chainId: 80001n,
signature: Signature {
r: "0x1b90ad9e9e4e005904562d50e904f9db10430a18b45931c059960ede337238ee",
s: "0x7df3c930a964fd07fed4a59f60b4ee896ffc7df4ea41b0facfe82b470db448b7",
yParity: 0,
networkV: null
},
accessList: []
}
```

回應構成交易接收。儲存 屬性 的值hash。這是您剛提交至區塊鏈之交易的識別符。您可以在讀取交易範例中使用此屬性，從 Polygon Mainnet 取得此交易的其他詳細資訊。

請注意， blockNumber和 blockHash位於回應null中。這是因為交易尚未記錄在多邊形網路上的區塊中。請注意，這些值會在稍後定義，當您請求下一節的交易詳細資訊時，您可能會看到這些值。

在 Node.js 中讀取交易

在本節中，您會為先前提交的交易請求交易詳細資訊，並使用 AMB Access Polygon 對 Polygon Mainnet 提出讀取請求，擷取收件人地址的 POL 餘額。在 `readTx.js` 檔案中，使用hash您在上一節中執行程式碼的回應中儲存的 *`your-transaction-id`* 取代標記為 的變數。

此程式碼使用公用程式 ，dispatch-evm-rpc.js它會使用 AWS SDK 中必要的 Signature 第 4 版 (SigV4) 模組向 AMB Access Polygon 簽署 HTTPS 請求，並使用廣泛使用的 HTTP 用戶端 AXIOS 傳送請求。

收到的回應類似以下內容：

```
TX DETAILS: {
blockHash: '0x59433e0096c783acab0659175460bb3c919545ac14e737d7465b3ddc********',
blockNumber: '0x28b4059',
from: '0xcf2c679ac6cb7de09bf6bb6042eccf05b7fa1394',
gas: '0x5208',
gasPrice: '0x3db9eca5d',
maxPriorityFeePerGas: '0x3db9eca4d',
maxFeePerGas: '0x3db9eca5d',
hash: '0x8d7538b4841261c5120c0a4dd66359e8ee189e7d1d34ac646a1d9923********',
input: '0x',
nonce: '0x2',
to: '0xd2bb4f4f1bdc4cb54f715c249fc5a991********',
transactionIndex: '0x0',
value: '0x5af3107a4000',
```

```
type: '0x2',
accessList: [],
chainId: '0x13881',
v: '0x0',
r: '0x1b90ad9e9e4e005904562d50e904f9db10430a18b45931c059960ede337238ee',
s: '0x7df3c930a964fd07fed4a59f60b4ee896ffc7df4ea41b0facfe82b470db448b7'
} BALANCE: 0.0003
```

回應代表交易詳細資訊。請注意，現在可能已定義 `blockHash`和 `blockNumber` 。這表示交易已記錄在區塊中。如果這些值仍然為 `null`，請等待幾分鐘，然後再次執行程式碼，以檢查您的交易是否已包含在區塊中。最後，收件人地址餘額的十六進位表示法 (0x110d9316ec000) 會使用 Ethers 的 `formatEther()`方法轉換為十進位，將十六進位轉換為十進位，並將十進位移 18 (10^18)，以在 POL 中提供真正的平衡。

> **Tip**
>
> 雖然上述程式碼範例說明如何使用 Node.js、Ethers 和 Axios 在 AMB Access Polygon 上使用一些支援的 JSON-RPCs，但您可以使用此服務修改範例並撰寫其他程式碼以在 Polygon 上建置應用程式。如需 AMB Access Polygon 上支援 JSON-RPCs 的完整清單，請參閱 AMB Access Polygon 支援的受管區塊鏈 API 和 JSON-RPCs。

建立和管理字符型存取的存取器字符，以發出 AMB Access Polygon 請求

您也可以使用 Accessor 權杖對多邊形網路端點進行 JSON-RPC 呼叫，作為 Signature 第 4 版 (SigV4) 簽署程序的便利替代方案。您必須BILLING_TOKEN從您建立的其中一個配件字符提供 ，並在呼叫中新增 做為參數。

> **Important**
>
> - 如果您將安全性和可稽核性優先於便利性，請改用 SigV4 簽署程序。
> - 您可以使用 Signature 第 4 版 (SigV4) 和字符型存取來存取 Polygon JSON-RPCs。不過，如果您選擇使用這兩種通訊協定，您的請求會遭到拒絕。
> - 您絕不能在面向使用者的應用程式中嵌入存取器字符。

在 主控台中，權杖存取器頁面會顯示所有存取器權杖的清單，您可以使用這些權杖從用戶端上的程式碼 AWS 帳戶 從您的 進行 AMB Access Polygon JSON-RPC 呼叫。

如需 AMB Access Polygon JSON-RPC 請求的詳細資訊，請參閱 AMB Access Polygon 支援的受管區塊鏈 API 和 JSON-RPCs。

您可以使用 建立和管理 Accessor 權杖 AWS 管理主控台。您也可以使用下列 API 操作來建立和管理 Accessor 權杖：`CreateAccessor`、`ListAccessors`、 `GetAccessor`和`DeleteAccessor`。BILLING_TOKEN 是 配件的屬性。此BILLING_TOKEN屬性用於追蹤您的 配件，以及針對從 發出的 AMB Access Polygon JSON-RPC 請求計費 AWS 帳戶。

與建立和管理配件權杖相關的所有 API 動作，也可以透過 AWS 管理主控台 AWS CLI、 和 SDKs 取得。

建立以字符為基礎的存取權杖

您可以建立配件字符，並使用它在 中的任何 AMB Access Polygon 節點上進行 AMB Access Polygon API 呼叫 AWS 帳戶。

建立配件字符，以使用 提出 AMB Access Polygon JSON-RPC 請求 AWS 管理主控台

1. 開啟位於 https：//https://console.aws.amazon.com/managedblockchain/ 的 Managed Blockchain 主控台。
2. 選擇權杖存取器。
3. 選擇建立存取點。
4. 選擇有效的多邊形區塊鏈網路。
5. 選用，為您的配件新增標籤。
6. 選擇建立存取器以建立新的存取器字符。

建立存取器字符，以使用 提出 AMB Access Polygon JSON-RPC 請求 AWS CLI

```
aws managedblockchain create-accessor --accessor-type BILLING_TOKEN --network-type
 POLYGON_MAINNET
```

先前的命令會傳回 `AccessorId`與 `BillingToken`，如下列範例所示。

```
{
"AccessorId": "ac-NGQ6QNKXLNEBXD3UI6********",
"NetworkType": "POLYGON_MAINNET",
"BillingToken": "jZlP8OUI-PcQSKINyX9euJJDC5-IcW9e-n********"
}
```

回應中的關鍵元素是 `BillingToken`。您可以使用此屬性進行 AMB Access Polygon JSON-RPC 呼叫。基於安全考量，範例中的某些值已混淆，但會在實際回應中完全顯示。

> **Note**
>
> 操作執行後，受管區塊鏈會為您佈建和設定字符。此程序的長度取決於許多變數。

檢視 Accessor 字符詳細資訊

您可以檢視您 AWS 帳戶 擁有之每個 Accessor 權杖的屬性。例如，您可以檢視配件 ID 或配件的 Amazon Resource Name (ARN)。您也可以檢視狀態、類型、建立日期和 `BillingToken`。

使用 檢視配件字符的資訊 AWS 管理主控台

1. 開啟位於 https：//https://console.aws.amazon.com/managedblockchain/ 的 Managed Blockchain 主控台。
2. 在導覽窗格中，選擇權杖存取器。
3. 從清單中選擇權杖的存取器 ID。

快顯的字符詳細資訊頁面。從此頁面，您可以檢視字符的屬性。

使用 檢視配件字符的資訊 AWS CLI

執行下列命令來檢視配件字符的詳細資訊。將 的值取代`--accessor-id`為您的 配件 ID。

```
aws managedblockchain get-accessor --accessor-id ac-NGQ6QNKXLNEBXD3UI6********
```

傳回 `BillingToken`和其他金鑰屬性，如下列範例所示。基於安全考量，範例中的某些值已混淆，但完全出現在實際回應中。

```
{
  "Accessor": {
  "Id": "ac-NGQ6QNKXLNEBXD3UI6********",
  "Type": "BILLING_TOKEN",
  "BillingToken": "jZlP8OUI-PcQSKINyX9euJJDC5-IcW9e-n********",
  "Status": "AVAILABLE",
  "NetworkType": "POLYGON_MAINNET"
  "CreationDate": "2022-01-04T23:09:47.750Z",
  "Arn": "arn:aws:managedblockchain:us-east-1:666666666666:accessors/ac-
NGQ6QNKXLNEBXD3UI6********"
  }
}
```

刪除 Accessor 字符

當您刪除配件字符時，字符會從 `AVAILABLE`變更為 `PENDING_DELETION` 狀態。您無法使用具有 `PENDING_DELETION` 狀態的 Accessor 權杖。

使用 刪除 配件字符 AWS 管理主控台

1. 開啟位於 https：//https://console.aws.amazon.com/managedblockchain/ 的 Managed Blockchain 主控台。
2. 在導覽窗格中，選擇權杖配件。
3. 從清單中選取您想要的配件字符。
4. 選擇 刪除。
5. 確認您的選擇。

系統會使用已刪除的附加元件權杖返回權杖存取器頁面。頁面會顯示PENDING_DELETION狀態。

使用 刪除配件字符 AWS CLI

下列範例示範如何刪除字符。使用 delete-accessor命令來刪除字符。使用 --accessor-id 配件 ID 設定 的值。

使用 CLI AWS 刪除配件字符

```
aws managedblockchain delete-accessor --accessor-id ac-NGQ6QNKXLNEBXD3UI6********
```

如果此命令成功執行，則不會傳回任何訊息。

AMB Access Polygon 支援的受管區塊鏈 API 和 JSON-RPCs

Amazon Managed Blockchain 提供 API 操作，用於建立和管理 AMB Access Polygon 的字符存取器。如需詳細資訊，請參閱 Managed Blockchain API 參考指南 。

下列主題提供 AMB Access Polygon 支援的 Polygon JSON-RPCs 清單和參考。每個支援的 JSON-RPC 都有其使用的簡短描述。您可以使用 Polygon JSON-RPCs來查詢和取得智慧型合約資料、取得交易詳細資訊、提交交易，以及對交易執行追蹤等其他公用程式，以及預估費用。

AMB Access Polygon 支援下列 JSON-RPC 方法。每個支援的 JSON-RPC 都有一個類別，以及其公用程式及其預設請求配額的簡短描述。在適用的情況下，會指出搭配 Amazon Managed Blockchain 使用 JSON-RPC 方法的唯一考量。

Note

- 不支援未列出的任何方法。
- 在 Amazon Managed Blockchain 上呼叫 Polygon JSON-RPCs 時，您可以透過使用 Signature 第 4 版簽署程序驗證的 HTTPS 連線來執行此操作。這表示只有帳戶中的授權 IAM AWS 主體可以進行 Polygon JSON-RPC 呼叫。若要這樣做，必須隨呼叫提供 AWS 憑證 （ 存取金鑰 ID 和私密存取金鑰 ）。
- 您也可以使用字符型存取做為 Signature 第 4 版 (SigV4) 簽署程序的便利替代方案。如果您將安全性和可稽核性排定在便利性上，請改用 SigV4 簽署程序。不過，如果您同時使用 SigV4 和字符型存取，您的請求將無法運作。
- Amazon Managed Blockchain (AMB) Access Polygon 不支援此預覽版的 JSON-RPC 批次請求。
- 下表中的配額欄列出每個 JSON-RPC 的配額。配額是以每個 JSON-RPC 每個多邊形網路 (Mainnet) 區域每秒請求數 (RPS) 設定。

  若要提高配額，您必須聯絡 支援。若要聯絡 支援，請登入 AWS Support Center Console。選擇建立案例。選擇技術。選擇 Managed Blockchain 做為您的服務。選擇 Access：Polygon 作為您的類別和一般指引作為您的嚴重性。輸入 RPC Quota 做為主旨，並在描述文字方塊中列出 JSON-RPC 和配額限制，這些限制適用於您在每個區域每個多邊形網路 RPS 中的需求。提交您的案例。

類別	JSON-RPC	描述	考量事項
Ethereum	eth_blockNumber	傳回最近區塊的數量。	
	eth_call	立即執行新的訊息呼叫，而不在區塊鏈上建立交易。	eth_call 會耗用 0 瓦斯，但對於需要該瓦斯參數的訊息，則具有一個瓦斯參數。
	eth_chainId	傳回 EIP-155 中引入之目前設定Chain Id值的整數值。None 如果沒有可用的 Chain Id ，則傳回 。	
	eth_estimateGas	估算並傳回交易所需的氣體，而不將交易新增至區塊鏈。	
	eth_feeHistory	傳回歷史瓦斯資訊的集合。	
	eth_gasPrice	傳回 Wei 中每瓦斯的目前價格。	
	eth_getBalance	傳回指定帳戶地址和區塊識別符的帳戶餘額。	
	eth_getBlockByHash	傳回使用區塊雜湊所指定區塊的相關資訊。	

類別	JSON-RPC	描述	考量事項
	eth_getBlockByNumber	傳回使用 區塊編號所指定區塊的相關資訊。	
	eth_getBlockReceipts	傳回有關使用區塊編號所指定區塊的收據。	
	eth_getBlockTransactionCountByHash	傳回使用區塊雜湊所指定區塊中的交易數量。	
	eth_getBlockTransactionCountByNumber	傳回使用 區塊編號所指定區塊中的交易數量。	
	eth_getCode	在指定的帳戶地址和區塊識別符傳回程式碼。	
	eth_getLogs	傳回指定篩選條件物件的所有日誌陣列。	提供合約地址時，您可以對具有 1K 區塊範圍的任何區塊範圍提出`eth_getlogs` 請求。具有高活動的合約可能僅限於較小的區塊範圍。如果未提供合約地址，區塊範圍將為 8。
	eth_getRawTransactionByHash	傳回 指定的交易原始形式`transaction_hash` 。	

類別	JSON-RPC	描述	考量事項
	eth_getStorageAt	傳回指定帳戶地址和區塊識別符之指定儲存位置的值。	
	eth_getTransactionByBlockHashAndIndex	傳回使用指定區塊雜湊和交易索引位置的交易相關資訊。	
	eth_getTransactionByBlockNumberAndIndex	傳回使用指定區塊號碼和交易索引位置的交易相關資訊。	
	eth_getTransactionByHash	傳回具有指定交易雜湊之交易的相關資訊。	
	eth_getTransactionCount	傳回從指定地址和區塊識別符傳送的交易數量。	
	eth_getTransactionReceipt	使用指定的交易雜湊傳回交易的接收。	
	eth_getUncleByBlockHashAndIndex	傳回使用區塊雜湊和叔叔索引位置指定之叔叔區塊的相關資訊。	
	eth_getUncleByBlockNumberAndIndex	傳回使用區塊編號和叔叔索引位置所指定之叔叔區塊的相關資訊。	

類別	JSON-RPC	描述	考量事項
	eth_getUncleCountByBlockHash	傳回使用叔叔雜湊指定之叔叔中的計數數目。	
	eth_getUncleCountByBlockNumber	傳回使用叔叔編號指定的叔叔中的計數。	
	eth_maxPriorityFeePerGas	傳回每瓦斯的費用，這是您可以支付多少作為優先順序費用，或「提示」以取得目前區塊中包含的交易。	一般而言，您可以使用從此方法傳回的值，在您提交的後續交易maxFeePerGas 中設定。
	eth_protocolVersion	傳回目前的 Ethereum 通訊協定版本。	
	eth_sendRawTransaction	為已簽章的交易建立新的訊息呼叫交易或建立合約。	受管區塊鏈僅支援原始交易。您必須先建立和簽署交易，才能傳送交易。
偵錯	debug_traceBlockByHash	使用追蹤器執行區塊雜湊所指定區塊中的所有交易，以傳回可能的追蹤結果編號（需要追蹤模式）。	

類別	JSON-RPC	描述	考量事項
	debug_traceBlockByNumber	使用追蹤器執行 編號指定的區塊中的所有交易，以傳回追蹤結果 （需要追蹤模式）。	
	debug_traceCall	在指定的區塊執行內容中執行 eth 呼叫，傳回可能追蹤結果的數量 （需要追蹤模式）。	
	debug_traceTransaction	傳回指定交易的所有追蹤 （需要追蹤模式）。	
淨值	net_version	傳回目前的網路 ID。	
追蹤	trace_block	傳回區塊中包含之所有交易的所有調用 opcode 的完整堆疊追蹤。	
	trace_call	在指定的區塊執行內容中執行 eth 呼叫，傳回可能追蹤結果的數量 （需要追蹤模式）。	
	trace_transaction	傳回指定交易的所有追蹤 （需要追蹤模式）。	

類別	JSON-RPC	描述	考量事項
Tx 集區	txpool_content	傳回所有待處理和已排入佇列的交易。	
	txpool_status	提供目前待包含在下一個區塊中的所有交易，以及已排入佇列的交易計數（排程為未來執行）。	
Web	web3_clientVersion	傳回目前的用戶端版本。	

具有 Amazon Managed Blockchain (AMB) Access Polygon 的多邊形使用案例

Polygon 區塊鏈常用於建置與 NFTs、Web3 遊戲和字符化使用案例等相關的分散式應用程式 (dApps。本主題提供您可以使用 Amazon Managed Blockchain (AMB) Access Polygon 實作的一些使用案例清單。

主題

- 分析多邊形 NFT 資料
- 支援 NFT 購買
- 建立多邊形錢包
- 錢包即服務
- 權杖門控體驗

分析多邊形 NFT 資料

您可以收集 Polygon NFTs的資料，包括指定期間內的傳輸事件和 NFT 中繼資料等資訊。然後，您可以分析此資料來繪製洞見，例如哪些 NFTs 正在趨勢，哪些使用者最常與指定的集合互動。

如需詳細資訊，請參閱AMB Access Polygon 支援的受管區塊鏈 API 和 JSON-RPCs。

支援 NFT 購買

您可以使用 AMB Access Polygon 來提交交易，以便在次要市場使用初始的 Mint、允許清單或 NFT 購買。然後 AWS ，您可以使用其他服務的組合來允許使用信用卡進行購買，接受 Fiat 或加密貨幣，並為所有涉及的利益相關者進行快速和解。

如需詳細資訊，請參閱AMB Access Polygon 支援的受管區塊鏈 API 和 JSON-RPCs。

建立多邊形錢包

您可以使用 AMB Access Polygon 來提供數位資產錢包的重要功能，例如從區塊鏈上的智慧合約讀取使用者權杖餘額，或將簽署的交易廣播至區塊鏈。

如需詳細資訊，請參閱AMB Access Polygon 支援的受管區塊鏈 API 和 JSON-RPCs。

錢包即服務

您可以使用 AMB Access Polygon 來開發所需的操作wallet-as-a-service，以支援常見的錢包交易，例如使用支援的 Polygon JSON-RPCs 檢查餘額、資產轉移、資產傳送和費用估算。

如需詳細資訊，請參閱AMB Access Polygon 支援的受管區塊鏈 API 和 JSON-RPCs。

權杖門控體驗

您可以使用 AMB Access Polygon 為您的使用者建置字符門控體驗。例如，您可以有條件地僅將存取某段內容的權限提供給特定 NFT 的擁有者。若要達成此目的，您必須讀取區塊鏈，以判斷使用者地址的 NFT 擁有權。

如需詳細資訊，請參閱AMB Access Polygon 支援的受管區塊鏈 API 和 JSON-RPCs。

Amazon Managed Blockchain (AMB) Access Polygon 教學課程

本節中強調的下列教學課程是來自 AWS re:Post 的社群文章，提供逐步解說，協助您了解如何使用 AMB Access Polygon 在 Polygon 區塊鏈上執行一些常見任務。

- 使用 AMB Access Polygon 和 web3.js 傳送交易
- 使用 AMB Access Polygon 和 Hardhat Ignition 部署智慧合約
- 與智慧合約互動
- 使用 AMB Access Polygon 和 Chainlink 資料饋送擷取目前的非鏈價格資料
- 使用 AMB Access 分析 Polygon Mainnet 上的 ERC-20 字符資料

Amazon Managed Blockchain (AMB) Access Polygon 中的安全性

的雲端安全性 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，這些架構專為滿足最安全敏感組織的需求而建置。

安全性是 AWS 與您之間的共同責任。共同責任模型將此描述為雲端安全性和雲端安全性：

- 雲端的安全性 – AWS 負責保護在 中執行 AWS 服務的基礎設施 AWS 雲端。 AWS 也為您提供可安全使用的服務。在 AWS 合規計劃中，第三方稽核員會定期測試並驗證我們的安全功效。若要了解適用於 Amazon Managed Blockchain (AMB) Access Polygon 的合規計劃，請參閱AWS 合規計劃範圍內的服務。
- 雲端的安全性 – 您的責任取決於您使用 AWS 的服務。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規。

為了提供資料保護、身分驗證和存取控制，Amazon Managed Blockchain 會使用 AWS Managed Blockchain 中執行的開放原始碼架構的功能。

本文件可協助您了解如何在使用 AMB Access Polygon 時套用共同責任模型。下列主題說明如何設定 AMB Access Polygon 以符合您的安全與合規目標。您也會了解如何使用其他 AWS 服務來協助您監控和保護 AMB Access Polygon 資源。

主題

- Amazon Managed Blockchain (AMB) Access Polygon 中的資料保護
- Amazon Managed Blockchain (AMB) Access Polygon 的身分和存取管理

Amazon Managed Blockchain (AMB) Access Polygon 中的資料保護

AWS 共同責任模型適用於 Amazon Managed Blockchain (AMB) Access Polygon 中的資料保護。如此模型所述， AWS 負責保護執行所有 的全域基礎設施 AWS 雲端。您負責維護在此基礎設施上託管內容的控制權。您也同時負責所使用 AWS 服務 的安全組態和管理任務。如需資料隱私權的詳細資訊，請參閱資料隱私權常見問答集。如需有關歐洲資料保護的相關資訊，請參閱 AWS 安全性部落格上的 AWS 共同的責任模型和 GDPR 部落格文章。

基於資料保護目的，我們建議您保護 AWS 帳戶 登入資料，並使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 設定個別使用者。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用 設定 API 和使用者活動記錄 AWS CloudTrail。如需使用 CloudTrail 追蹤擷取 AWS 活動的資訊，請參閱AWS CloudTrail 《 使用者指南》中的使用 CloudTrail 追蹤。
- 使用 AWS 加密解決方案，以及其中的所有預設安全控制 AWS 服務。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在 AWS 透過命令列界面或 API 存取 時需要 FIPS 140-3 驗證的密碼編譯模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的更多相關資訊，請參閱聯邦資訊處理標準 (FIPS) 140-3。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如名稱欄位。這包括當您使用 AMB Access Polygon 或其他 AWS 服務 使用主控台 AWS CLI、API 或 AWS SDKs。您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

資料加密

資料加密有助於防止未經授權的使用者從區塊鏈網路和相關聯的資料儲存系統讀取資料。這包括在網路移動時可能攔截的資料，稱為傳輸中的資料。

傳輸中加密

根據預設，受管區塊鏈會使用 HTTPS/TLS 連線來加密從執行 AWS CLI 至 AWS 服務端點的用戶端電腦傳輸的所有資料。

您不須採取任何行動即可啟用 HTTPS/TLS。除非您使用 AWS CLI 命令明確停用個別命令的`--no-verify-ssl`命令，否則一律會啟用。

Amazon Managed Blockchain (AMB) Access Polygon 的身分和存取管理

AWS Identity and Access Management (IAM) 是一種 AWS 服務 ，可協助管理員安全地控制對 AWS 資源的存取。IAM 管理員可控制誰可以進行身分驗證 （ 登入 ） 和授權 （ 具有許可 ），以使用 AMB Access Polygon 資源。IAM 是 AWS 服務 您可以免費使用的 。

主題

- 目標對象
- 使用身分驗證
- 使用政策管理存取權
- Amazon Managed Blockchain (AMB) Access Polygon 如何與 IAM 搭配使用
- Amazon Managed Blockchain (AMB) Access Polygon 的身分型政策範例
- 對 Amazon Managed Blockchain (AMB) Access Polygon 身分和存取進行故障診斷

目標對象

使用方式 AWS Identity and Access Management (IAM) 會根據您的角色而有所不同：

- 服務使用者 — 若無法存取某些功能，請向管理員申請所需許可 (請參閱 對 Amazon Managed Blockchain (AMB) Access Polygon 身分和存取進行故障診斷)
- 服務管理員 — 負責設定使用者存取權並提交相關許可請求 (請參閱 Amazon Managed Blockchain (AMB) Access Polygon 如何與 IAM 搭配使用)
- IAM 管理員 — 撰寫政策以管理存取控制 (請參閱 Amazon Managed Blockchain (AMB) Access Polygon 的身分型政策範例)

使用身分驗證

身分驗證是您 AWS 使用身分憑證登入 的方式。您必須以 AWS 帳戶根使用者、IAM 使用者或擔任 IAM 角色身分進行身分驗證。

您可以使用身分來源的登入資料，例如 AWS IAM Identity Center (IAM Identity Center)、單一登入身分驗證或 Google/Facebook 登入資料，以聯合身分的形式登入。如需有關登入的詳細資訊，請參閱《AWS 登入 使用者指南》中的如何登入您的 AWS 帳戶。

對於程式設計存取， AWS 提供 SDK 和 CLI 以密碼編譯方式簽署請求。如需詳細資訊，請參閱《IAM 使用者指南》中的 API 請求的AWS 第 4 版簽署程序。

AWS 帳戶 根使用者

當您建立 時 AWS 帳戶，您會從一個名為 AWS 帳戶 theroot 使用者的登入身分開始，該身分具有對所有 AWS 服務 和 資源的完整存取權。強烈建議不要使用根使用者來執行日常任務。有關需要根使用者憑證的任務，請參閱《IAM 使用者指南》中的需要根使用者憑證的任務。

聯合身分

最佳實務是要求人類使用者使用聯合身分提供者，以 AWS 服務 使用臨時憑證存取 。

聯合身分是您企業目錄、Web 身分提供者的使用者，或使用來自身分來源的 AWS 服務 憑證存取 Directory Service 。聯合身分會擔任角色，而該角色會提供臨時憑證。

若需集中化管理存取權限，建議使用 AWS IAM Identity Center。如需詳細資訊，請參閱 AWS IAM Identity Center 使用者指南中的什麼是 IAM Identity Center？。

IAM 使用者和群組

IAM 使用者https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users.html是一種身分具備單人或應用程式的特定許可權。建議以臨時憑證取代具備長期憑證的 IAM 使用者。如需詳細資訊，請參閱《IAM 使用者指南》中的要求人類使用者使用聯合身分提供者來 AWS 使用臨時憑證存取 。

IAM 群組會指定 IAM 使用者集合，使管理大量使用者的許可權更加輕鬆。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 使用者的使用案例。

IAM 角色

IAM 角色https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html的身分具有特定許可權，其可以提供臨時憑證。您可以透過從使用者切換到 IAM 角色 （主控台） 或呼叫 AWS CLI 或 AWS API 操作來擔任角色。如需詳細資訊，請參閱《IAM 使用者指南》中的擔任角色的方法。

IAM 角色適用於聯合身分使用者存取、臨時 IAM 使用者許可、跨帳戶存取權與跨服務存取，以及在 Amazon EC2 執行的應用程式。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 中的快帳戶資源存取。

使用政策管理存取權

您可以透過建立政策並將其連接到身分或資源 AWS 來控制 AWS 中的存取。政策定義與身分或資源相關聯的許可。當委託人提出請求時 AWS ， 會評估這些政策。大多數政策會以 JSON 文件 AWS 形式存放在 中。如需進一步了解 JSON 政策文件，請參閱《IAM 使用者指南》中的 JSON 政策概觀。

管理員會使用政策，透過定義哪些主體可在哪些條件下對哪些資源執行動作，以指定可存取的範圍。

預設情況下，使用者和角色沒有許可。IAM 管理員會建立 IAM 政策並將其新增至角色，供使用者後續擔任。IAM 政策定義動作的許可，無論採用何種方式執行。

身分型政策

身分型政策是附加至身分 (使用者、使用者群組或角色) 的 JSON 許可政策文件。這類政策控制身分可對哪些資源執行哪些動作，以及適用的條件。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的透過客戶管理政策定義自訂 IAM 許可。

身分型政策可分為內嵌政策 (直接內嵌於單一身分) 與受管政策 (可附加至多個身分的獨立政策)。如需了解如何在受管政策及內嵌政策之間做選擇，請參閱《IAM 使用者指南》中的在受管政策與內嵌政策之間選擇。

資源型政策

資源型政策是附加到資源的 JSON 政策文件。範例包括 IAM 角色信任政策與 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。您必須在資源型政策中指定主體。

資源型政策是位於該服務中的內嵌政策。您無法在以資源為基礎的政策中使用來自 IAM 的 AWS 受管政策。

其他政策類型

AWS 支援其他政策類型，可設定更多常見政策類型授予的最大許可：

- 許可界限 — 設定身分型政策可授與 IAM 實體的最大許可。如需詳細資訊，請參閱《 IAM 使用者指南》中的 IAM 實體許可界限。
- 服務控制政策 (SCP) — 為 AWS Organizations中的組織或組織單位指定最大許可。如需詳細資訊，請參閱《AWS Organizations 使用者指南》中的服務控制政策。
- 資源控制政策 (RCP) — 設定您帳戶中資源可用許可的上限。如需詳細資訊，請參閱《AWS Organizations 使用者指南》中的資源控制政策 (RCP)。

- 工作階段政策 — 在以程式設計方式為角色或聯合身分使用者建立臨時工作階段時，以參數形式傳遞的進階政策。如需詳細資訊，請參《IAM 使用者指南》中的工作階段政策。

多種政策類型

當多種類型的政策套用到請求時，產生的許可會更複雜而無法理解。若要了解如何 AWS 在涉及多種政策類型時決定是否允許請求，請參閱《IAM 使用者指南》中的政策評估邏輯。

Amazon Managed Blockchain (AMB) Access Polygon 如何與 IAM 搭配使用

在您使用 IAM 管理 AMB Access Polygon 的存取權之前，請先了解哪些 IAM 功能可與 AMB Access Polygon 搭配使用。

您可以搭配 Amazon Managed Blockchain (AMB) Access Polygon 使用的 IAM 功能

IAM 功能	AMB Access Polygon 支援
身分型政策	是
資源型政策	否
政策動作	是
政策資源	否
政策條件索引鍵	否
ACL	否
ABAC(政策中的標籤)	否
臨時憑證	否
主體許可	否
服務角色	否
服務連結角色	否

若要全面了解 AMB Access Polygon 和其他 如何與大多數 IAM 功能 AWS 服務 搭配使用，請參閱《IAM 使用者指南》中的與 IAM AWS 搭配使用的 服務。

AMB Access Polygon 的身分型政策

支援身分型政策：是

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的透過客戶管理政策定義自訂 IAM 許可。

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及在何種條件下允許或拒絕動作。如要了解您在 JSON 政策中使用的所有元素，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素參考。

AMB Access Polygon 的身分型政策範例

若要檢視 AMB Access Polygon 身分型政策的範例，請參閱 Amazon Managed Blockchain (AMB) Access Polygon 的身分型政策範例。

AMB Access Polygon 內的資源型政策

支援資源型政策：否

資源型政策是附加到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中指定主體。委託人可以包含帳戶、使用者、角色、聯合身分使用者或 AWS 服務。

如需啟用跨帳戶存取權，您可以在其他帳戶內指定所有帳戶或 IAM 實體作為資源型政策的主體。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 中的快帳戶資源存取。

AMB Access Polygon 的政策動作

支援政策動作：是

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

JSON 政策的 `Action` 元素描述您可以用來允許或拒絕政策中存取的動作。政策會使用動作來授予執行相關聯動作的許可。

若要查看 AMB Access Polygon 動作清單，請參閱《服務授權參考》中的 Amazon Managed Blockchain (AMB) Access Polygon 定義的動作。

AMB Access Polygon 中的政策動作在動作之前使用下列字首：

```
managedblockchain:
```

如需在單一陳述式中指定多個動作，請用逗號分隔。

```
"Action": [
      "managedblockchain::action1",
      "managedblockchain::action2"
         ]
```

您也可以使用萬用字元 (*) 來指定多個動作。例如，若要指定開頭是 InvokeRpcPolygon 文字的所有動作，請包含以下動作：

```
"Action": "managedblockchain::InvokeRpcPolygon*"
```

若要檢視 AMB Access Polygon 身分型政策的範例，請參閱 Amazon Managed Blockchain (AMB) Access Polygon 的身分型政策範例。

AMB Access Polygon 的政策資源

支援政策資源：否

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Resource JSON 政策元素可指定要套用動作的物件。最佳實務是使用其 Amazon Resource Name (ARN) 來指定資源。若動作不支援資源層級許可，使用萬用字元 (*) 表示該陳述式適用於所有資源。

```
"Resource": "*"
```

若要查看 AMB Access Polygon 資源類型及其 ARNs 的清單，請參閱《服務授權參考》中的 Amazon Managed Blockchain (AMB) Access Polygon 定義的資源。若要了解您可以使用哪些動作指定每個資源的 ARN，請參閱 Amazon Managed Blockchain (AMB) Access Polygon 定義的動作。

若要檢視 AMB Access Polygon 身分型政策的範例，請參閱 Amazon Managed Blockchain (AMB) Access Polygon 的身分型政策範例。

AMB Access Polygon 的政策條件索引鍵

支援服務特定政策條件金鑰：否

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

`Condition` 元素會根據定義的條件，指定陳述式的執行時機。您可以建立使用條件運算子的條件運算式 (例如等於或小於)，來比對政策中的條件和請求中的值。若要查看所有 AWS 全域條件索引鍵，請參閱《IAM 使用者指南》中的AWS 全域條件內容索引鍵。

若要查看 AMB Access Polygon 條件金鑰清單，請參閱《服務授權參考》中的 Amazon Managed Blockchain (AMB) Access Polygon 的條件金鑰。若要了解您可以使用條件金鑰的動作和資源，請參閱 Amazon Managed Blockchain (AMB) Access Polygon 定義的動作。

若要檢視 AMB Access Polygon 身分型政策的範例，請參閱 Amazon Managed Blockchain (AMB) Access Polygon 的身分型政策範例。

AMB Access Polygon 中的 ACLs

支援 ACL：否

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

ABAC 搭配 AMB Access Polygon

支援 ABAC (政策中的標籤)：否

屬性型存取控制 (ABAC) 是一種授權策略，依據稱為標籤的屬性來定義許可。您可以將標籤連接至 IAM 實體 AWS 和資源，然後設計 ABAC 政策，以便在委託人的標籤符合資源上的標籤時允許操作。

如需根據標籤控制存取，請使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 條件索引鍵，在政策的條件元素中，提供標籤資訊。

如果服務支援每個資源類型的全部三個條件金鑰，則對該服務而言，值為 Yes。如果服務僅支援某些資源類型的全部三個條件金鑰，則值為 Partial。

如需 ABAC 的詳細資訊，請參閱《IAM 使用者指南》中的使用 ABAC 授權定義許可。如要查看含有設定 ABAC 步驟的教學課程，請參閱《IAM 使用者指南》中的使用屬性型存取控制 (ABAC)。

搭配 AMB Access Polygon 使用臨時登入資料

支援臨時登入資料：否

臨時登入資料提供 AWS 資源的短期存取權，並在您使用聯合或切換角色時自動建立。 AWS 建議您動態產生臨時登入資料，而不是使用長期存取金鑰。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 中的臨時安全憑證與可與 IAM 搭配運作的AWS 服務。

AMB Access Polygon 的跨服務主體許可

支援轉寄存取工作階段 (FAS)：否

轉送存取工作階段 (FAS) 使用呼叫 的委託人許可 AWS 服務，並結合 請求 AWS 服務 向下游服務提出請求。如需提出 FAS 請求時的政策詳細資訊，請參閱轉發存取工作階段。

AMB Access Polygon 的服務角色

支援服務角色：否

服務角色是服務擔任的 IAM 角色，可代您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱《IAM 使用者指南》中的建立角色以委派許可給 AWS 服務。

> **Warning**
>
> 變更服務角色的許可可能會中斷 AMB Access Polygon 功能。只有在 AMB Access Polygon 提供指引時，才能編輯服務角色。

AMB Access Polygon 的服務連結角色

支援服務連結角色：否

服務連結角色是連結至 的一種服務角色 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 中 AWS 帳戶 ，並由服務擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

如需建立或管理服務連結角色的詳細資訊，請參閱可搭配 IAM 運作的AWS 服務。在資料表中尋找服務，其中包含服務連結角色欄中的 Yes。選擇是連結，以檢視該服務的服務連結角色文件。

Amazon Managed Blockchain (AMB) Access Polygon 的身分型政策範例

根據預設，使用者和角色沒有建立或修改 AMB Access Polygon 資源的許可。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。

如需了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的建立 IAM 政策 (主控台)。

如需 AMB Access Polygon 定義的動作和資源類型的詳細資訊，包括每種資源類型的 ARNs 格式，請參閱《服務授權參考》中的 Amazon Managed Blockchain (AMB) Access Polygon 的動作、資源和條件金鑰。

主題

- 政策最佳實務
- 使用 AMB Access Polygon 主控台
- 允許使用者檢視他們自己的許可
- 存取多邊形網路

政策最佳實務

身分型政策會判斷您帳戶中的某個人員是否可以建立、存取或刪除 AMB Access Polygon 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管政策並邁向最低權限許可 – 若要開始將許可授予您的使用者和工作負載，請使用將許可授予許多常見使用案例的 AWS 受管政策。它們可在您的 中使用 AWS 帳戶。我們建議您定義特定於使用案例 AWS 的客戶受管政策，進一步減少許可。如需更多資訊，請參閱《IAM 使用者指南》中的 AWS 受管政策或任務職能的AWS 受管政策。
- 套用最低權限許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的許可。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱《IAM 使用者指南》中的 IAM 中的政策和許可。
- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。如果透過特定 例如 使用服務動作 AWS 服務，您也可以使用條件來授予其存取權 CloudFormation。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素：條件。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access

Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您撰寫安全且實用的政策。如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM Access Analyzer 驗證政策。

- 需要多重要素驗證 (MFA) – 如果您的案例需要 IAM 使用者或 中的根使用者 AWS 帳戶，請開啟 MFA 以提高安全性。如需在呼叫 API 操作時請求 MFA，請將 MFA 條件新增至您的政策。如需詳細資訊，請參閱《IAM 使用者指南》中的透過 MFA 的安全 API 存取。

如需 IAM 中最佳實務的相關資訊，請參閱《IAM 使用者指南》中的 IAM 安全最佳實務。

使用 AMB Access Polygon 主控台

若要存取 Amazon Managed Blockchain (AMB) Access Polygon 主控台，您必須擁有一組最低許可。這些許可必須允許您列出和檢視 中 AMB Access Polygon 資源的詳細資訊 AWS 帳戶。如果您建立比最基本必要許可更嚴格的身分型政策，則對於具有該政策的實體 (使用者或角色) 而言，主控台就無法如預期運作。

對於僅呼叫 AWS CLI 或 AWS API 的使用者，您不需要允許最低主控台許可。反之，只需允許存取符合他們嘗試執行之 API 操作的動作就可以了。

為了確保使用者和角色仍然可以使用 AMB Access Polygon 主控台，請將 AMB Access Polygon *ConsoleAccess*或 *ReadOnly* AWS 受管政策連接至實體。如需詳細資訊，請參閱《IAM 使用者指南》中的新增許可到使用者。

允許使用者檢視他們自己的許可

此範例會示範如何建立政策，允許 IAM 使用者檢視附加到他們使用者身分的內嵌及受管政策。此政策包含在主控台或使用 或 AWS CLI AWS API 以程式設計方式完成此動作的許可。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
```

```
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

存取多邊形網路

> **Note**
>
> 為了存取 Polygon 的公mainnet有端點mainnet和進行 JSON-RPC 呼叫，您需要具有 AMB Access Polygon 適當 IAM 許可的使用者憑證 (AWS_ACCESS_KEY_ID 和 AWS_SECRET_ACCESS_KEY)。

Example存取所有多邊形網路的 IAM 政策

此範例會授予 IAM 使用者 AWS 帳戶 存取所有多邊形網路的權限。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AccessAllPolygonNetworks",
            "Effect": "Allow",
```

```
            "Action": [
                "managedblockchain:InvokeRpcPolygon*"
            ],
            "Resource": "*"
        }
    ]
}
```

Example存取多邊形 Mainnet 網路的 IAM 政策

此範例會授予 IAM 使用者 AWS 帳戶 存取 Polygon Mainnet 網路的權限。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AccessPolygonTestnet",
            "Effect": "Allow",
            "Action": [
                "managedblockchain:InvokeRpcPolygonMainnet"
            ],
            "Resource": "*"
        }
    ]
}
```

對 Amazon Managed Blockchain (AMB) Access Polygon 身分和存取進行故障診斷

使用以下資訊來協助您診斷和修正使用 AMB Access Polygon 和 IAM 時可能遇到的常見問題。

主題

- 我無權在 AMB Access Polygon 中執行動作
- 我未獲得執行 iam:PassRole 的授權

- 我想要允許 以外的人員 AWS 帳戶 存取我的 AMB Access Polygon 資源

我無權在 AMB Access Polygon 中執行動作

如果您收到錯誤，告知您未獲授權執行動作，您的政策必須更新，允許您執行動作。

下列範例錯誤會在mateojackson IAM 使用者嘗試使用主控台檢視一個虛構 *my-example-widget* 資源的詳細資訊，但卻無虛構 managedblockchain::*GetWidget* 許可時發生。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 managedblockchain::GetWidget on resource: my-example-widget
```

在此情況下，必須更新 mateojackson 使用者的政策，允許使用 managedblockchain::*GetWidget* 動作存取 *my-example-widget* 資源。

如果您需要協助，請聯絡您的 AWS 管理員。您的管理員提供您的簽署憑證。

我未獲得執行 iam:PassRole 的授權

如果您收到錯誤，告知您無權執行 iam:PassRole動作，您的政策必須更新，以允許您將角色傳遞給 AMB Access Polygon。

有些 AWS 服務 可讓您將現有角色傳遞給該服務，而不是建立新的服務角色或服務連結角色。如需執行此作業，您必須擁有將角色傳遞至該服務的許可。

當名為 的 IAM marymajor 使用者嘗試使用主控台在 AMB Access Polygon 中執行動作時，會發生下列範例錯誤。但是，動作請求服務具備服務角色授予的許可。Mary 沒有將角色傳遞給服務的許可。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

在這種情況下，Mary 的政策必須更新，允許她執行 iam:PassRole 動作。

如果您需要協助，請聯絡您的 AWS 管理員。您的管理員提供您的簽署憑證。

我想要允許 以外的人員 AWS 帳戶 存取我的 AMB Access Polygon 資源

您可以建立一個角色，讓其他帳戶中的使用者或您組織外部的人員存取您的資源。您可以指定要允許哪些信任物件取得該角色。針對支援基於資源的政策或存取控制清單 (ACL) 的服務，您可以使用那些政策來授予人員存取您的資源的許可。

如需進一步了解，請參閱以下內容：

- 若要了解 AMB Access Polygon 是否支援這些功能，請參閱 Amazon Managed Blockchain (AMB) Access Polygon 如何與 IAM 搭配使用。
- 若要了解如何 AWS 帳戶 在您擁有的 資源之間提供存取權，請參閱《IAM 使用者指南》中的在您擁有 AWS 帳戶 的另一個 IAM 使用者中提供存取權。
- 若要了解如何將資源的存取權提供給第三方 AWS 帳戶，請參閱《IAM 使用者指南》中的將存取權提供給第三方 AWS 帳戶 擁有。
- 如需了解如何透過聯合身分提供存取權，請參閱《IAM 使用者指南》中的將存取權提供給在外部進行身分驗證的使用者 (聯合身分)。
- 如需了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱《IAM 使用者指南》中的 IAM 中的跨帳戶資源存取。

使用 記錄 Amazon Managed Blockchain (AMB) 存取多邊形事件 AWS CloudTrail

> **Note**
>
> Amazon Managed Blockchain (AMB) Access Polygon 不支援管理事件。

Amazon Managed Blockchain 會在 上執行 AWS CloudTrail，此服務會提供使用者、角色或 Managed Blockchain 中 AWS 服務所採取之動作的記錄。CloudTrail 會將受管區塊鏈的 AMB Access Polygon 端點叫用者擷取為資料平面事件。

如果您建立已正確設定且訂閱接收所需資料平面事件的線索，您可以接收 AMB Access Polygon 相關 CloudTrail 事件持續交付至 S3 儲存貯體。您可以使用 CloudTrail 收集的資訊，判斷已向其中一個 AMB Access Polygon 端點提出請求、請求的 IP 地址、提出請求的人員、提出請求的時間，以及其他詳細資訊。

若要進一步了解 CloudTrail，請參閱「AWS CloudTrail 使用者指南」。

CloudTrail 中的 AMB Access Polygon 資訊

建立 CloudTrail AWS 帳戶 時，會在您的 上啟用 CloudTrail。不過，您必須設定資料平面事件，以檢視誰叫用 AMB Access Polygon 端點。

若要持續記錄 中的事件 AWS 帳戶，包括 AMB Access Polygon 的事件，請建立追蹤。線索可讓 CloudTrail 將日誌檔案傳遞到 S3 儲存貯體。依預設，當您在主控台中建立追蹤時，該追蹤會套用至所有的 AWS 區域。線索會記錄 AWS 分割區中所有支援區域的事件，並將日誌檔案交付至您指定的 S3 儲存貯體。此外，您可以設定其他 AWS 服務 進一步分析，並對 CloudTrail 日誌中收集的事件資料採取行動。如需詳細資訊，請參閱下列內容：

- 使用 CloudTrail 追蹤多邊形 JSON-RPCs
- 建立追蹤的概觀
- CloudTrail 支援的服務和整合
- 設定 CloudTrail 的 Amazon SNS 通知
- 接收多個區域的 CloudTrail 日誌檔案及接收多個帳戶的 CloudTrail 日誌檔案

透過分析 CloudTrail 資料事件，您可以監控誰叫用 AMB Access Polygon 端點。

每一筆事件或日誌專案都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 是否使用根或 AWS Identity and Access Management (IAM) 使用者登入資料提出請求
- 提出該請求時，是否使用了特定角色或聯合身分使用者的臨時安全憑證
- 請求是否由另一個 提出 AWS 服務

如需詳細資訊，請參閱 CloudTrail userIdentity 元素。

了解 AMB Access Polygon 日誌檔案項目

對於資料平面事件，線索是一種組態，可讓事件做為日誌檔案交付至指定的 S3 儲存貯體。每個 CloudTrail 日誌檔案都包含一或多個日誌項目，代表來自任何來源的單一請求。這些項目提供所請求動作的詳細資訊，包括動作的日期和時間，以及任何相關聯的請求參數。

> **Note**
>
> 日誌檔案中的 CloudTrail 資料事件不是 AMB Access Polygon API 呼叫的排序堆疊追蹤，因此不會以任何特定順序顯示。

使用 CloudTrail 追蹤多邊形 JSON-RPCs

您可以使用 CloudTrail 來追蹤帳戶中誰叫用 AMB Access Polygon 端點，以及叫用哪些 JSON-RPC 做為資料事件。根據預設，當您建立線索時，不會記錄資料事件。若要將叫用 AMB Access Polygon 端點的人員記錄為 CloudTrail 資料事件，您必須明確新增要收集活動的資源或資源類型至線索。AMB Access Polygon 支援使用 AWS 管理主控台 AWS CLI、 和 SDK 新增資料事件。如需詳細資訊，請參閱AWS CloudTrail 《 使用者指南》中的使用進階選擇器記錄事件。

若要在線索中記錄資料事件，請在建立線索後使用 put-event-selectors 操作。使用 `--advanced-event-selectors`選項來指定`AWS::ManagedBlockchain::Network`資源類型，以開始記錄資料事件，以判斷誰叫用 AMB Access Polygon 端點。

Example您帳戶的所有 AMB Access Polygon 端點請求的資料事件日誌項目

下列範例示範如何使用 `put-event-selectors`操作記錄您帳戶的所有 AMB Access Polygon 端點`my-polygon-trail`對 `us-east-1` 區域中線索的請求。

```
aws cloudtrail put-event-selectors \

--region us-east-1 \
--trail-name my-polygon-trail  \
--advanced-event-selectors '[{
    "Name": "Test",
    "FieldSelectors": [
      { "Field": "eventCategory", "Equals": ["Data"] },
      { "Field": "resources.type", "Equals": ["AWS::ManagedBlockchain::Network"] } ]}]'
```

訂閱後，您可以追蹤連線到先前範例中指定之線索的 S3 儲存貯體中的用量。

下列結果顯示 CloudTrail 所收集的資訊的 CloudTrail 資料事件日誌項目。您可以判斷多邊形 JSON-RPC 請求是對其中一個 AMB Access Polygon 端點提出的請求、請求的 IP 地址、提出請求的人員、提出請求的時間，以及其他詳細資訊。基於安全考量，下列範例中的某些值已混淆，但完全出現在實際的日誌項目中。

```
{
        "eventVersion": "1.09",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "AROA554UO62RJ7KSB7FAX:777777777777",
            "arn": "arn:aws:sts::111122223333:assumed-role/Admin/777777777777",
            "accountId": "111122223333"
        },
        "eventTime": "2023-04-12T19:00:22Z",
        "eventSource": "managedblockchain.amazonaws.com",
        "eventName": "gettxout",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "111.222.333.444",
        "userAgent": "python-requests/2.28.1",
        "errorCode": "-",
        "errorMessage": "-",
        "requestParameters": {
            "jsonrpc": "2.0",
            "method": "gettxout",
            "params": [],
            "id": 1
        },
        "responseElements": null,
        "requestID": "DRznHHEj********",
        "eventID": "baeb232d-2c6b-46cd-992c-0e40********",
```

```
        "readOnly": true,
        "resources": [{
            "type": "AWS::ManagedBlockchain::Network",
            "ARN": "arn:aws:managedblockchain:::networks/n-polygon-mainnet"
        }],
        "eventType": "AwsApiCall",
        "managementEvent": false,
        "recipientAccountId": "111122223333",
        "eventCategory": "Data"
}
```

AMB Access Polygon 使用者指南的文件歷史記錄

下表說明 AMB Access Polygon 的文件版本。

變更	描述	日期
更新 JSON-RPC 的配額	AMB Access Polygon 針對每個支援的 JSON-RPC 支援的配額會更新。	2024 年 4 月 12 日
孟買測試網路的支援結束	AMB Access Polygon 已於 2024 年 4 月 15 日終止對孟買睪丸的支援。	2024 年 4 月 10 日
新增教學課程主題	AWS re：Post 社群文章區段中的 AMB Access Polygon 教學課程。	2024 年 4 月 9 日
公有預覽	Amazon Managed Blockchain (AMB) Access Polygon 服務的公開預覽版本。	2023 年 11 月 24 日