



开发人员指南

AMB 访问多边形



AMB 访问多边形: 开发人员指南

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Table of Contents

.....	v
关于 AMB 访问多边形	1
面向首次使用 AMB Access Polygon 用户的资源	1
重要概念	2
注意事项和限制	2
设置	5
使用 AMB 访问多边形的先决条件	5
报名参加 AWS	5
创建具有适当权限的 IAM 用户	5
安装和配置 AWS Command Line Interface	6
开始使用	7
创建 IAM 策略	7
控制台 RPC 示例	8
awscurlRPC 示例	9
Node.js RPC 示例	10
发送交易	15
读取交易	16
基于令牌的访问权限	18
为基于令牌的访问创建访问器令牌	18
查看访问者令牌的详细信息	19
删除访问者令牌	20
JSON-RPC 和 API	22
多边形用例	29
分析多边形 NFT 数据	29
支持 NFT 购买	29
创建 Polygon 钱包	29
钱包即服务	30
代币门控体验	30
教程	31
安全性	32
数据保护	32
数据加密	33
传输中加密	33
Identity and access management	33

受众	34
使用身份进行身份验证	34
使用策略管理访问	35
亚马逊托管区块链 (AMB) Access Polygon 如何与 IAM 配合使用	37
基于身份的策略示例	41
问题排查	45
CloudTrail 日志	48
AMB 访问中的多边形信息 CloudTrail	48
了解 AMB Access Polygon 日志文件条目	49
CloudTrail 用于追踪 Polygon JSON-RPCs	49
文档历史记录	52

Amazon Managed Blockchain (AMB) Access Polygon 处于预览版，可能会发生变化。

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。

什么是亚马逊托管区块链 (AMB) 访问多边形？

亚马逊托管区块链 (AMB) Access Polygon 是一项完全托管的服务，可帮助你在 Polygon 区块链上构建弹性的 Web3 应用程序。AMB Access Polygon 提供对 Polygon 区块链的即时和无服务器访问。

Polygon 是一种以太坊虚拟机 (EVM) 作为基础的扩展解决方案。Polygon 区块链以高交易吞吐量和低交易费用而闻名。Polygon 区块链使用共 proof-of-stake 机制。Polygon 通常用于构建与 Web3 游戏和代币化用例 NFTs 等相关的去中心化应用程序 (dApps)。

本指南介绍如何使用亚马逊托管区块链 (AMB) Access Polygon 创建和管理 Polygon 区块链资源。

面向首次使用 AMB Access Polygon 用户的资源

如果这是你第一次使用 AMB Access Polygon，我们建议你先阅读以下章节：

- [关键概念：亚马逊托管区块链 \(AMB\) Access Polygon](#)
- [亚马逊托管区块链 \(AMB\) Access Polygon 入门](#)
- [托管区块链 API 和 JSON-RPCs 支持 AMB Access Polygon](#)

关键概念：亚马逊托管区块链 (AMB) Access Polygon

Note

本指南假设你熟悉 Polygon 必不可少的概念。这些概念包括质押、去中心化应用程序、交易、钱包、智能合约、Polygon (POL , 前身为 MATIC) 等。在使用亚马逊托管区块链 (AMB) Access Polygon 之前，我们建议您查看 [Polygon 开发文档](#)和 [Polygon 维基](#)。

Amazon Managed Blockchain (AMB) Access Polygon 为您提供对 Polygon 主网和 Polygon 主网网络的无服务器访问，无需您预置和管理任何 Polygon 基础设施，包括节点。网络上的 Polygon 节点共同存储 Polygon 区块链状态、验证交易并参与共识以更改区块链状态。您可以使用此托管服务按需快速访问 Polygon 网络，从而降低总体拥有成本。

使用 AMB Access Polygon，您可以访问 JSON 远程过程 (JSON-RPC) 调用。您可以调用 Polygon JSON-RPCs，通过托管区块链管理的节点与 Polygon 区块链进行通信。您可以使用 AMB Access Polygon 服务来开发和使用与 Polygon 区块链交互的去中心化应用程序 (dApp)。dApps 不可或缺的一部分是智能合约。您可以使用 AMB Access Polygon 创建智能合约并将其部署到 Polygon 区块链中。您还可以通过 RPCs 对 AMB Access Polygon 端点调用 JSON-来检查钱包的余额、交易详情、估算费用等，这些端点在 Polygon 网络的所有对等节点上去中心化方式运行。Polygon 网络的任何节点都可以开发和部署智能合约。

Important

您负责创建、维护、使用和管理您的 Polygon 地址。您还要对您的 Polygon 地址的内容负责。AWS 对使用 Amazon Managed Blockchain 上的 Polygon 节点部署或调用的任何交易概不负责。

使用亚马逊托管区块链 (AMB) Access Polygon 的注意事项和限制

当你使用亚马逊托管区块链 (AMB) Access Polygon 时，请考虑以下几点：

- 支持的多边形网络

AMB Access Polygon 支持以下公共网络：

- 主网 — 通过 proof-of-stake 共识保护的公共 Polygon 区块链，Polygon (POL) 代币是在该区块链上发行和交易的。主网上的交易具有实际价值（也就是说，它们会产生实际成本），并记录在公共区块链上。
- Polygon 不再支持网络
 - 正如 [Polygon Labs 所传达](#) 的那样，孟买测试网网络将于 4 月中旬停用。与此消息一致，AMB Access Polygon 于 2024 年 4 月 15 日终止了对孟买测试网的支持。我们建议使用淘大测试网来处理您的测试工作量。
 - 不支持私有网络。
 - 此外，AMB Access Polygon 不包括对 Polygon zkEVM 网络的支持。
- 与流行的第三方编程库兼容

AMB Access Polygon 与流行的编程库（例如 ethers.js）兼容，允许开发人员使用熟悉的工具与 Polygon 区块链进行交互，从而轻松地与现有实现集成或快速开发新应用程序。

- 支持的区域：

仅美国东部（弗吉尼亚北部）地区支持此服务。

- 服务终端节点

以下是 AMB Access Polygon 的服务端点。要连接服务，您必须使用包含其中一个支持的区域的终端节点。

- `mainnet.polygon.managedblockchain.us-east-1.amazonaws.com`

- 不支持质押

AMB Access Polygon 不支持多边形 (POL) 验证器节点。proof-of-stake

- 签名版本 4 对 Polygon JSON-RPC 请求进行签名


RPCs 在 Amazon Managed Blockchain 上调用 Polygon JSON 时，您可以通过使用 [签名版本 4 签名流程](#) 进行身份验证的 HTTPS 连接进行调用。这意味着只有 AWS 账户中获得授权的 IAM 委托人才可以进行 Polygon JSON-RPC 调用。为此，必须在呼叫中提供 AWS 证书（访问密钥 ID 和私有访问密钥）。

Important

- 不要在面向用户的应用程序中嵌入客户端凭据。
- 您不能使用 IAM 策略来限制对单个 Polygon JSON-RPC 的访问权限。

- Support 支持基于令牌的访问

您还可以使用访问器令牌对 Polygon 网络端点进行 JSON-RPC 调用，以此作为签名版本 4 (Sigv4) 签名过程的便捷替代方案。您必须提供一个BILLING_TOKEN来自您[创建](#)的 Accessor 令牌，并将其作为参数添加到调用中。

 Important

- 如果您将安全性和可审计性置于便利性之上，请改用 Sigv4 签名流程。
- RPCs 您可以使用签名版本 4 (Sigv4) 和基于令牌的访问权限来访问 Polygon JSON。但是，如果您选择同时使用这两种协议，则您的请求将被拒绝。
- 切勿在面向用户的应用程序中嵌入 Accessor 令牌。

- 仅支持提交原始交易

使用 eth_sendrawtransaction JSON-RPC 提交更新 Polygon 区块链状态的交易。

设置亚马逊托管区块链 (AMB) Access Polygon

在您首次使用亚马逊托管区块链 (AMB) Access Polygon 之前，请按照本节中的步骤创建 AWS 账户。下一章讨论如何开始使用 AMB Access Polygon。

使用 AMB 访问多边形的先决条件

在你 AWS 第一次使用之前，你必须有一个 AWS 账户。

报名参加 AWS

当您注册时，系统会自动注册所有 AWS 服务用户 AWS 账户，包括亚马逊托管区块链 (AMB) Access Polygon。您只需为使用的服务付费。

如果您 AWS 账户 已经有，请转到下一步。如果您还没有 AWS 账户，请使用以下步骤创建。

要创建 AWS 账户

1. 打开<https://portal.aws.amazon.com/billing/>注册。
2. 按照屏幕上的说明操作。

在注册时，将接到电话或收到短信，要求使用电话键盘输入一个验证码。

当您注册时 AWS 账户，就会创建AWS 账户根用户一个。根用户有权访问该账户中的所有 AWS 服务和资源。作为最佳安全实践，请为用户分配管理访问权限，并且只使用根用户来执行[需要根用户访问权限的任务](#)。

创建具有适当权限的 IAM 用户

要创建和使用 AMB Access Polygon，您必须拥有一个 AWS Identity and Access Management (IAM) 委托人（用户或群组），该委托人必须具有允许进行必要的托管区块链操作的权限。

RPCs 在 Amazon Managed Blockchain 上调用 Polygon JSON 时，您可以通过使用[签名版本 4 签名流程](#)进行身份验证的 HTTPS 连接进行调用。这意味着只有 AWS 账户中获得授权的 IAM 委托人才能进行 Polygon JSON-RPC 调用。为此，必须在呼叫中提供 AWS 证书（访问密钥 ID 和私有访问密钥）。

您还可以使用访问令牌对 Polygon 网络端点进行 JSON-RPC 调用，以此作为签名版本 4 (Sigv4) 签名过程的便捷替代方案。您必须提供一个BILLING_TOKEN来自您[创建](#)的 Accessor 令牌，并将其作为

参数添加到调用中。但是，您仍然需要 IAM 访问权限才能获得使用 AWS 管理控制台 AWS CLI、和 SDK 创建访问器令牌的权限。

有关如何创建 IAM 用户的信息，请参阅[在您的 AWS 账户中创建 IAM 用户](#)。有关如何向用户关联权限策略的更多信息，请参阅[更改 IAM 用户的权限](#)。有关可用于授予用户使用 AMB Access Polygon 的权限策略示例，请参阅[亚马逊托管区块链 \(AMB\) Access Polygon 的基于身份的策略示例](#)。

安装和配置 AWS Command Line Interface

如果您还没有这样做，请安装最新的 AWS Command Line Interface (AWS CLI) 以使用终端上的 AWS 资源。有关更多信息，请参阅[安装或更新 AWS CLI 的最新版本](#)。

Note

要进行 CLI 访问，您需要访问密钥 ID 和秘密访问密钥。如果可能，请使用临时凭证代替长期访问密钥。临时凭证包括访问密钥 ID、秘密访问密钥，以及一个指示凭证何时到期的安全令牌。有关更多信息，请参阅 IAM 用户指南中的[将临时证书与 AWS 资源配合使用](#)。

亚马逊托管区块链 (AMB) Access Polygon 入门

使用本节中的信息和程序，开始使用 Amazon Managed Blockchain (AMB) Access Polygon。

主题

- [创建用于访问 Polygon 区块链网络的 IAM 策略](#)
- [使用 AMB Access RPC 编辑器发出 Polygon 远程过程调用 \(RPC\) 请求 AWS 管理控制台](#)
- [使用 AMB Access Polygon JSON-RPC 请求进来 awscurl AWS CLI](#)
- [在 Node.js 中发出 Polygon JSON-RPC 请求](#)

创建用于访问 Polygon 区块链网络的 IAM 策略

要访问 Polygon 主网的公共终端节点以进行 JSON-RPC 调用，您必须拥有对亚马逊托管区块链 (AMB AWS_SECRET_ACCESS_KEY) Access Polygon 具有相应 IAM 权限的用户证书 (AWS_ACCESS_KEY_ID 和)。在 AWS CLI 安装了终端中，运行以下命令创建用于访问两个 Polygon 终端节点的 IAM 策略：

```
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

前面的示例允许您访问所有可用的 Polygon 网络。要访问特定端点，请使用以下 Action 命令：

- "managedblockchain:InvokeRpcPolygonMainnet"

创建策略后，将该策略附加到您的 IAM 用户的角色以使其生效。在中 AWS 管理控制台，导航到 IAM 服务，并将策略附加 AmazonManagedBlockchainPolygonAccess 到分配给您的 IAM 用户的角色。

使用 AMB Access RPC 编辑器发出 Polygon 远程过程调用 (RPC) 请求 AWS 管理控制台

您可以 AWS 管理控制台 使用 AMB Access Polygon 在上编辑、配置和提交远程过程调用 (RPCs)。有了这些 RPCs，您可以在 Polygon 网络上读取数据和写入交易，包括检索数据和向 Polygon 网络提交交易。

Example

以下示例说明如何使用 `eth_getBlockByNumber` RPC 获取有关最新区块的信息。将突出显示的变量更改为您自己的输入，或者选择列出的 RPC 方法之一，然后输入所需的相关输入。

1. 打开托管区块链控制台，网址为 <https://console.aws.amazon.com/managedblockchain/>。
2. 选择 RPC 编辑器。
3. 在“请求”部分中，选择“*POLYGON_MAINNET*”作为“**Blockchain Network**”。
4. 选择 *eth_getBlockByNumber* 作为 RPC 方法。
5. 输入 `latest` 为“**Block number**”然后选择 *False* 作为“全部交易”标志。
6. 然后，选择提交 RPC。
7. 您可以在“响应”部分中看到 `latest` 区块的结果。然后，您可以复制完整的原始交易以供进一步分析或在应用程序的业务逻辑中使用。

有关更多信息，请参阅 [AMB Access Polygon RPCs 支持的](#)

使用 AMB Access Polygon JSON-RPC 请求进来 `awscurl` AWS CLI

Example

使用[签名版本 4 \(Sigv4\) 使用您的 IAM 用户证书签署](#)请求，以便向 AMB Access Polygon 终端节点发出 Polygon JSON-RPC 请求。`awscurl` 命令行工具可以帮助您使用 Sigv4 签署对 AWS 服务的请求。有关更多信息，请参阅 [awscurl](#) README.md。

`awscurl` 使用适合您的操作系统的方法进行安装。在 macOS 上，推荐 HomeBrew 使用以下应用程序：

```
brew install awscurl
```

如果您已经安装并配置了 AWS CLI，则您的 IAM 用户证书和默认 AWS 区域证书将在您的环境中设置并有权访问 `awscurl`。使用 `awscurl`，通过调用 RPC 向 Polygon 主网提交请求。`eth_getBlockByNumber` 此调用接受与您要检索其信息的区块号相对应的字符串参数。

以下命令使用 `params` 数组中的区块编号来选择要检索标头的特定块，从而从 Polygon 主网检索区块数据。

```
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

您也可以使用令牌 `curl` 和基于 AMB Access 令牌的访问功能发出同样的请求。Accessor 有关更多信息，请参阅 [为基于令牌的访问权限创建和管理访问令牌以发出 AMB Access Polygon 请求](#)。

```
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'
```

任一命令的响应都会返回有关最新区块的信息。出于说明目的，请参见以下示例：

```

{"error":null,"id":"eth_getBlockByNumber-curltest","jsonrpc":"1.0",
  "result":{"baseFeePerGas":"0x873bf591e","difficulty":"0x18",
    "extraData":"0xd78301000683626f7288676f312e32312e32856c696e7578000000000000000009a
  \
    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 中发出 Polygon JSON-RPC 请求

您可以使用 HTTPS 提交签名的请求RPCs 来调用 Polygon JSON，使用 [Node.js 中的原生 https 模块](#) 访问 Polygon 主网网络，也可以使用第三方库，例如 [A XIOS](#)。以下 [Node.js 示例向您展示了如何使用签名版本 4 \(Sigv4\) 和基于令牌的访问向 AMB Access Polygon 端点发出 Polygon JSON-RPC 请求](#)。第一个示例将交易从一个地址发送到另一个地址，以下示例从区块链请求交易详情和余额信息。

Example

要运行此示例 Node.js 脚本，请应用以下先决条件：

1. 您的计算机上必须安装节点版本管理器 (nvm) 和 Node.js。您可以[在此处](#)找到操作系统的安装说明。
2. 使用 `node --version` 命令并确认您使用的是 Node 版本 18 或更高版本。如果需要，您可以使用 `命令和 nvm install v18.12.0 命令安装版本 18，即 LTS 版本的 Node。 nvm use v18.12.0`
3. 环境变量 `AWS_ACCESS_KEY_ID` 和 `AWS_SECRET_ACCESS_KEY` 必须包含与您的账户关联的证书。

使用以下命令将这些变量作为字符串导出到客户端。将以下字符串中的红色值替换为您的 IAM 用户账户中的相应值。

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

完成所有先决条件后，使用首选的代码编辑器将以下文件复制到本地环境的目录中：

package.j

```
{  
  "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

以下代码使用硬编码私钥生成钱包 Signer，仅Ethers.js供演示之用。请勿在生产环境中使用此代码，因为它有真实资金并存在安全风险。

如有必要，请联系您的客户团队，就钱包和签名者的最佳实践提供建议。

```
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("recipient-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");
```

将这些文件保存到您的目录后，使用以下命令安装运行代码所需的依赖项：

```
npm install
```

在 Node.js 中发送交易

前面的示例通过签署交易并使用 AMB Access Polygon 将其广播到 Polygon 主网，将原生 Polygon 主网令牌 (POL) 从一个地址发送到另一个地址。为此，请使用该脚本，该 `sendTx.js` 脚本使用一个流行的库 `Ethers.js`，用于与以太坊和 Polygon 等兼容以太坊的区块链进行交互。您需要替换代码中以红色突出显示的三个变量，包括 `billingToken` 用于 [基于令牌访问的 Accessor 令牌](#)、用于签署交易的私钥以及接收 POL 的收件人地址。

Tip

我们建议您为此目的创建一个新的私钥（钱包），而不是重复使用现有的钱包，以消除资金损失的风险。你可以使用 Ethers 库的钱包类方法 `createRandom()` 生成一个要测试的钱包。此外，如果您需要从 Polygon 主网申请 POL，则可以使用公共 POL 水龙头申请少量用于测试。

将资金充值钱包的私钥和收款人的地址添加到代码中后，您就可以运行以下代码来签署一笔交易，将 .0001 POL 从您的 `billingToken` 地址发送到另一个地址，然后使用 AMB Access Polygon 调用 JSON-RPC 将其广播到 Polygon 主网，使用 AMB Access Polygon 调用 `eth_sendRawTransaction` JSON-RPC。

```
node sendTx.js
```

收到的回复类似于以下内容：

```
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: []  
}
```

回复构成交易收据。保存属性的值hash。这是您刚刚提交给区块链的交易的标识符。您可以在读取事务示例中使用此属性从 Polygon 主网获取有关此事务的更多详细信息。

请注意blockNumber，响应null中blockHash有 and。这是因为交易尚未记录在 Polygon 网络上的区块中。请注意，这些值是稍后定义的，当您在下一节中请求交易详细信息时，您可能会看到它们。

在 Node.js 中读取一笔交易

在本节中，您将使用AMB Access Polygon向Polygon主网发送的读取请求来请求先前提提交的交易的交易详细信息，并使用AMB Access Polygon的读取请求来检索收件人地址的POL余额。在readTx.js文件中，替换标`your-transaction-id`有hash您在上一节中运行代码的响应中保存的变量。

[此代码使用一个实用程序dispatch-evm-rpc.js，该实用程序使用 AWS 软件开发工具包中必需的签名版本 4 \(Sigv4\) 模块签署向 AMB Access Polygon 的 HTTPS 请求，并使用广泛使用的 HTTP 客户端 AXIOS 发送请求。](#)

收到的回复类似于以下内容：

```
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
```

响应表示交易详情。请注意，现在可能blockNumber已经定义了blockHash和。这表示交易已记录在区块中。如果这些值仍然存在null，请等待几分钟，然后再次运行代码以检查您的交易是否已包含在区块中。最后，使用Ethers的方法将收件人地址余额（0x110d9316ec000）的十六进制表示形式转换为十进制，该formatEther()方法将十六进制转换为十进制，并将小数位移18位（ 10^{18} ），以得出POL中的真实余额。

Tip

虽然前面的代码示例说明了如何使用 Node.js、Ethers 和 Axios RPCs 在 AMB Access Polygon 上使用一些支持的 JSON，但你可以修改示例并编写其他代码，使用此服务在 Polygon 上构建应用程序。有关 AMB Access Polygon RPCs 上支持的 JSON 的完整列表，请参阅[托管区块链 API 和 JSON-RPCs 支持 AMB Access Polygon](#)。

为基于令牌的访问权限创建和管理访问令牌以发出 AMB Access Polygon 请求

您还可以使用访问器令牌对 Polygon 网络端点进行 JSON-RPC 调用，以此作为签名版本 4 (Sigv4) 签名过程的便捷替代方案。您必须提供一个BILLING_TOKEN来自您[创建](#)的 Accessor 令牌，并将其作为参数添加到调用中。

Important

- 如果您将安全性和可审计性置于便利性之上，请改用 Sigv4 签名流程。
- RPCs 您可以使用签名版本 4 (Sigv4) 和基于令牌的访问权限来访问 Polygon JSON。但是，如果您选择同时使用这两种协议，则您的请求将被拒绝。
- 切勿在面向用户的应用程序中嵌入 Accessor 令牌。

在控制台中，令牌访问器页面显示了所有访问器令牌的列表，您可以使用这些令牌从客户端上的源代码发出 AMB Access Polygon JSON-RPC 调用。AWS 账户

有关 AMB Access Polygon JSON-RPC 请求的更多信息，请参阅。[托管区块链 API 和 JSON-RPCs 支持 AMB Access Polygon](#)

您可以使用创建和管理访问器令牌。AWS 管理控制台您还可以使用以下 API 操作创建和管理 Accessor 令牌：[CreateAccessor](#) [GetAccessor](#)、[ListAccessors](#)、和 [DeleteAccessor](#)。A BILLING_TOKEN 是访问器的属性。此BILLING_TOKEN属性用于跟踪你的 Accessor，也用于向你发出的 AMB Access Polygon JSON-RPC 请求计费。AWS 账户

与创建和管理 Accessor 令牌相关的所有API操作也可通过 AWS 管理控制台 AWS CLI、和 SDKs获取。

为基于令牌的访问创建访问器令牌

你可以创建 Accessor 令牌然后用它在你的任何 AMB Access Polygon 节点上调用 AMB Access Polygon API。AWS 账户

使用创建访问器令牌以发出 AMB Access Polygon JSON-RPC 请求 AWS 管理控制台

1. 打开托管区块链控制台，网址为<https://console.aws.amazon.com/managedblockchain/>。

2. 选择令牌访问器。
3. 选择“创建访问器”。
4. 选择一个有效的 Polygon 区块链网络。
5. (可选) 为您的访问器添加标签。
6. 选择“创建访问器”以创建新的访问者令牌。

使用创建访问器令牌以发出 AMB Access Polygon JSON-RPC 请求 AWS CLI

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

如以下示例所示BillingToken，前一个命令将与AccessorId一起返回。

```
{
  "AccessorId": "ac-NGQ6QNKXLNEBXD3UI6*****",
  "NetworkType": "POLYGON_MAINNET",
  "BillingToken": "jZlP80UI-PcQSKINyX9euJJDC5-IcW9e-n*****"
}
```

您回复中的关键要素是BillingToken. 您可以使用此属性进行 AMB Access Polygon JSON-RPC 调用。出于安全考虑，示例中的某些值已被模糊处理，但会完全出现在实际响应中。

Note

操作运行后，托管区块链将为您预置和配置令牌。此过程的长度取决于许多变量。

查看访问者令牌的详细信息

您可以查看自己 AWS 账户 拥有的每个 Accessor 令牌的属性。例如，您可以查看访问者的访问者 ID 或 Amazon 资源名称 (ARN)。您还可以查看状态、类型、创建日期和BillingToken。

要查看访问者令牌的信息，请使用 AWS 管理控制台

1. 打开托管区块链控制台，网址为<https://console.aws.amazon.com/managedblockchain/>。
2. 在导航窗格中，选择令牌访问器。
3. 从列表中选择令牌的访问器 ID。

随即弹出代币详情页面。在此页面上，您可以查看令牌的属性。

要查看访问者令牌的信息，请使用 AWS CLI

运行以下命令查看访问器令牌的详细信息。将的--accessor-id值替换为您的访问者 ID。

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

返回BillingToken和其他密钥属性，如以下示例所示。出于安全考虑，示例中的某些值已被模糊处理，但完全出现在实际响应中。

```
{
  "Accessor": {
    "Id": "ac-NGQ6QNKXLNEBXD3UI6*****",
    "Type": "BILLING_TOKEN",
    "BillingToken": "jZlP80UI-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*****"
  }
}
```

删除访问者令牌

当您删除访问器令牌时，该令牌的状态将从变AVAILABLE为PENDING_DELETION状态。您不能将访问者令牌与PENDING_DELETION状态一起使用。

要删除访问者令牌，请使用 AWS 管理控制台

1. 打开托管区块链控制台，网址为<https://console.aws.amazon.com/managedblockchain/>。
2. 在导航窗格中，选择令牌访问器。
3. 从列表中选择所需的访问器令牌。
4. 选择删除。
5. 确认您的选择。

您将使用已删除的访问者令牌返回到令牌访问者页面。该页面显示PENDING_DELETION状态。

要删除访问者令牌，请使用 AWS CLI

以下示例说明如何删除令牌。使用 `delete-accessor` 命令删除令牌。 `--accessor-id` 使用您的访问者 ID 设置的值。

使用 CLI 删除访问器令 AWS 牌

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

如果此命令成功运行，则不会返回任何消息。

托管区块链 API 和 JSON-RPCs 支持 AMB Access Polygon

Amazon Managed Blockchain 提供了 API 操作，用于[创建和管理 AMB Access Polygon 的令牌访问器](#)。有关更多信息，请参阅[托管区块链 API 参考指南](#)。

以下主题提供了 AMB Access Polygon 支持的 Polygon JSON 列表和参考资料。RPCs 每个支持的 JSON-RPC 都有其用法的简要描述。您可以使用 Polygon JSON-RPCs 来查询和获取智能合约数据、获取交易详情、提交交易以及其他实用工具，例如对交易运行跟踪和估算费用。

AMB Access Polygon 支持以下 JSON-RPC 方法。每个支持的 JSON-RPC 都有一个类别，并对其实用程序和默认请求配额进行了简要描述。在适用的情况下，说明了在亚马逊托管区块链中使用 JSON-RPC 方法的独特注意事项。

Note

- 不支持任何未列出的方法。
- RPCs 在 Amazon Managed Blockchain 上调用 Polygon JSON 时，您可以通过使用[签名版本 4 签名流程](#)进行身份验证的 HTTPS 连接进行调用。这意味着只有 AWS 账户中获得授权的 IAM 委托人才能进行 Polygon JSON-RPC 调用。为此，必须在呼叫中提供 AWS 证书（访问密钥 ID 和私有访问密钥）。
- 您还可以使用基于令牌的访问作为签名版本 4 (Sigv4) 签名过程的便捷替代方案。如果您将安全性和可审计性置于便利性之上，请改用 Sigv4 签名流程。但是，如果您同时使用 Sigv4 和基于令牌的访问权限，则您的请求将无法运行。
- 此预览版的亚马逊托管区块链 (AMB) Access Polygon 不支持 JSON-RPC 批量请求。
- 下表中的配额列出了每个 JSON-RPC 的配额。配额以每个 JSON-RPC 的每个区域每个多边形网络（主网）的每秒请求数 (RPS) 为单位设置。

要增加配额，您必须联系支持。要联系支持，请登录[AWS Support Center Console](#)。选择创建案例。选择“技术”。选择托管区块链作为您的服务。选择 Access: Polygon 作为您的类别，选择一般指导作为严重性。输入 RPC 配额作为主题，并在描述文本框中列出 JSON-RPC 以及适用于您的需求的配额限制（按每个区域每个多边形网络的 RPS 表示）。提交您的案例。

类别	JSON-RPC	描述	注意事项
以太坊	eth_blockNu	返回最近区块的数字。	
	eth_call	立即运行新的消息调用，而无需在区块链上创建交易。	eth_call消耗 0 gas，但对于需要它的消息，它有一个 gas 参数。
	eth_chainid	返回 EIP-155 中引入的当前配置Chain Id值的整数值。None如果没有可用Chain Id，则返回。	
	eth_EstimateGas	在不将交易添加到区块链的情况下，估算并返回交易所需的汽油。	
	eth_fee历史记录	返回历史天然气信息的集合。	
	eth_gasPrice	返回以 Wei 为单位的当前每种天然气价格。	
	eth_getBalanc	返回指定账户地址和区块标识符的账户余额。	
	eth_ 哈希 getBlockBy	返回有关使用区块哈希指定的区块的信息。	

类别	JSON-RPC	描述	注意事项
	eth_ 数字 getBlockBy	返回有关使用区块号指定的区块的信息。	
	eth_ getBlockReceipts	返回有关使用区块号指定的区块的收据。	
	eth_ getBlockTransaction CountByHash	返回使用区块哈希指定的区块中的交易数量。	
	eth_ getBlockTransaction CountByNumber	返回使用区块号指定的区块中的交易数量。	
	eth_getCode	返回指定账户地址和区块标识符处的代码。	
	eth_getLogs	返回指定过滤器对象的所有日志的数组。	如果提供了合约地址，则可以在默认区块范围为 1K 的任何区块范围内 eth_getlogs 发出请求。活跃度高的合约可能仅限于较小的区块范围。如果未提供合约地址，则区块范围将为 8。

类别	JSON-RPC	描述	注意事项
	eth_getRawTransaction ByHash	返回由指定的交易的原始形式transaction_hash。	
	eth_getStorageAt	返回指定账户地址和区块标识符的指定存储位置的值。	
	eth_getTransactionBy BlockHash AndIndex	使用指定的区块哈希值和交易索引位置返回有关交易的信息。	
	eth_getTransactionBy BlockNumberAndIndex	使用指定的区块号和交易索引位置返回有关交易的信息。	
	eth_ 哈希 getTransactionBy	返回有关具有指定交易哈希值的交易的信息。	
	eth_getTransactionCount	返回从指定地址和区块标识符发送的交易数量。	
	eth_getTransactionReceipt	使用指定的交易哈希返回交易收据。	
	eth_getUncleBy BlockHashAndIndex	返回有关使用区块哈希值和叔叔索引位置指定的叔区块的信息。	

类别	JSON-RPC	描述	注意事项
	eth_getUncleBy BlockNumberAndIndex	返回有关使用区块编号和叔叔索引位置指定的叔块的信息。	
	eth_getUncleCount ByBlockHash	返回使用叔哈希指定的叔叔中的计数。	
	eth_getUncleCount ByBlockNumber	返回使用叔叔编号指定的叔叔中的计数。	
	eth_maxPriorityFee PerGas	返回每笔汽油的费用，该费用是您为将交易包含在当前区块中而可以支付的优先费或“小费”的估计金额。	通常，您使用从此方法返回的值maxFeePer Gas 在您提交的后续事务中进行设置。
	eth_协议版本	返回当前的以太坊协议版本。	
	eth_sendRawTransaction	创建新的消息调用交易或为已签名的交易创建合约。	托管区块链仅支持原始交易。在发送交易之前，您必须创建并签署交易。
Debug	debug_ 哈希 traceBlockBy	通过使用跟踪器执行区块哈希指定的区块中的所有交易，返回可能的跟踪结果号（需要跟踪模式）。	

类别	JSON-RPC	描述	注意事项
	debug_数字 traceBlockBy	通过使用跟踪器执行由数字指定的区块中的所有交易来返回跟踪结果（需要跟踪模式）。	
	debug_traceCall	在给定区块执行的上下文中执行 eth 调用，返回可能的跟踪结果数量（需要跟踪模式）。	
	debug_traceTr	返回给定交易的所有跟踪（需要跟踪模式）。	
净值	网络版本	返回当前的网络 ID。	
跟踪	痕迹块	返回区块中包含的所有交易的所有调用操作码的完整堆栈跟踪。	
	追踪通话	在给定区块执行的上下文中执行 eth 调用，返回可能的跟踪结果数量（需要跟踪模式）。	
	追踪交易	返回给定交易的所有跟踪（需要跟踪模式）。	

类别	JSON-RPC	描述	注意事项
Tx Pool	txpool_content	返回所有待处理和排队的交易。	
	txpool_status	提供当前待包含在下一个区块中的所有交易以及已排队的交易的计数（仅计划在将来执行）。	
Web	web3_clientVersion	返回当前的客户端版本。	

亚马逊托管区块链 (AMB) 的 Polygon 用例 Access Polygon

Polygon 区块链通常用于构建与 Web3 游戏和代币化用例 NFTs 等相关的去中心化应用程序 (dApps)。本主题列出了您可以使用 Amazon Managed Blockchain (AMB) Access Polygon 实现的一些用例。

主题

- [分析多边形 NFT 数据](#)
- [支持 NFT 购买](#)
- [创建 Polygon 钱包](#)
- [钱包即服务](#)
- [代币门控体验](#)

分析多边形 NFT 数据

您可以收集有关 Polygon 的数据 NFTs，包括指定时间段内的传输事件和 NFT 元数据等信息。然后，您可以分析这些数据，以得出一些见解，例如 NFTs 哪些是趋势，或者哪些用户最常与给定集合进行交互。

有关更多信息，请参阅 [托管区块链 API 和 JSON-RPCs 支持 AMB Access Polygon](#)。

支持 NFT 购买

您可以使用 AMB Access Polygon 使用初始铸币厂、许可名单或二级市场提交购买 NFT 的交易。然后，结合使用其他 AWS 服务，您可以允许使用信用卡进行购买，接受法定货币或加密货币，并对所有相关利益相关者进行快速结算。

有关更多信息，请参阅 [托管区块链 API 和 JSON-RPCs 支持 AMB Access Polygon](#)。

创建 Polygon 钱包

您可以使用 AMB Access Polygon 来提供数字资产钱包的关键功能，例如从区块链上的智能合约中读取用户代币余额或将已签名的交易广播到区块链。

有关更多信息，请参阅 [托管区块链 API 和 JSON-RPCs 支持 AMB Access Polygon](#)。

钱包即服务

您可以使用支持的 Polygon JSON-RPC，使用 AMB Access Polygon 开发支持常见钱包交易 wallet-as-a-service 所需的操作，例如查看余额、资产转移、资产发送和费用估算。RPCs

有关更多信息，请参阅 [托管区块链 API 和 JSON-RPCs 支持 AMB Access Polygon](#)。

代币门控体验

您可以使用 AMB Access Polygon 为用户打造代币门控体验。例如，您可以有条件地仅向特定 NFT 的所有者提供对某一内容的访问权限。为此，您必须阅读区块链以确定用户地址的 NFT 所有权。

有关更多信息，请参阅 [托管区块链 API 和 JSON-RPCs 支持 AMB Access Polygon](#)。

亚马逊托管区块链 (AMB) Access Polygon 教程

本节重点介绍的以下教程是社区文章，这些文章提供了演练 AWS re:Post ，可帮助您学习如何使用 AMB Access Polygon 在 Polygon 区块链上执行一些常见任务。

- [使用 AMB Access Polygon 和 web3.js 发送交易](#)
- [使用 AMB 访问多边形和 Hardhat Ignition 部署智能合约](#)
- [与智能合约互动](#)
- [使用 AMB Access Polygon 和 Chainlink 数据源在链下检索当前价格数据](#)
- [使用 AMB 访问权限在 Polygon 主网上分析 ERC-20 代币数据](#)

亚马逊托管区块链 (AMB) Access Polygon 中的安全

云安全 AWS 是重中之重。作为 AWS 客户，您可以受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。

安全是双方共同承担 AWS 的责任。[责任共担模型](#)将其描述为既是云端的安全性，又是云端的安全：

- 云安全 — AWS 负责保护在云中运行 AWS 服务的基础架构 AWS 云。AWS 还为您提供可以安全使用的服务。作为 [AWS 合规性计划](#) 的一部分，第三方审核人员将定期测试和验证安全性的有效性。要了解适用于亚马逊托管区块链 (AMB) Access Polygon 的合规计划，请参阅 [合规计划范围内的 AWS 服务](#)。
- 云端安全-您的责任由您使用的 AWS 服务决定。您还需要对其他因素负责，包括您的数据的敏感性、您公司的要求以及适用的法律法规。

为了提供数据保护、身份验证和访问控制，Amazon Managed Blockchain 使用了在托管区块链中运行的开源框架的功能和 AWS 功能。

本文档可帮助您了解在使用 AMB Access Polygon 时如何应用分担责任模型。以下主题向您展示了如何配置 AMB Access Polygon 以满足您的安全和合规性目标。您还将学习如何使用其他 AWS 服务来帮助您监控和保护您的 AMB Access Polygon 资源。

主题

- [亚马逊托管区块链 \(AMB\) Access Polygon 中的数据保护](#)
- [亚马逊托管区块链 \(AMB\) Access Polygon 的身份和访问管理](#)

亚马逊托管区块链 (AMB) Access Polygon 中的数据保护

AWS [分担责任模型](#)适用于亚马逊托管区块链 (AMB) Access Polygon 中的数据保护。如本模型所述 AWS，负责保护运行所有内容的全球基础架构 AWS 云。您负责维护对托管在此基础结构上的内容的控制。您还负责您所使用的 AWS 服务的安全配置和管理任务。有关数据隐私的更多信息，请参阅[数据隐私常见问题](#)。有关欧洲数据保护的信息，请参阅 AWS Security Blog 上的 [AWS Shared Responsibility Model and GDPR](#) 博客文章。

出于数据保护目的，我们建议您保护 AWS 账户凭证并使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 设置个人用户。这样，每个用户只获得履行其工作职责所需的权限。还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证 (MFA)。
- 使用 SSL/TLS 与资源通信。AWS 我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 使用设置 API 和用户活动日志 AWS CloudTrail。有关使用 CloudTrail 跟踪捕获 AWS 活动的信息，请参阅《AWS CloudTrail 用户指南》中的[使用跟 CloudTrail 跟踪](#)。
- 使用 AWS 加密解决方案以及其中的所有默认安全控件 AWS 服务。
- 使用高级托管安全服务 (例如 Amazon Macie)，它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果您在 AWS 通过命令行界面或 API 进行访问时需要经过 FIPS 140-3 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅[《美国联邦信息处理标准 \(FIPS \) 第 140-3 版》](#)。

强烈建议您切勿将机密信息或敏感信息 (如您客户的电子邮件地址) 放入标签或自由格式文本字段 (如名称字段)。这包括你 AWS 服务 使用控制台、API 或与 AMB Access Polygon 或其他人合作时。AWS CLI AWS SDKs在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供网址，强烈建议您不要在网址中包含凭证信息来验证对该服务器的请求。

数据加密

数据加密有助于防止未经授权的用户从区块链网络和相关的数据存储系统读取数据。这包括在网络中传输时可能被拦截的数据，即传输中的数据。

传输中加密

默认情况下，托管区块链使用 HTTPS/TLS 连接来加密从运行的客户端计算机传输到服务端点的所有数据。AWS CLI AWS

您无需执行任何操作即可使用 HTTPS/TLS。除非您使用命令为单个 AWS CLI 命令明确禁用它，否则它始终处于启用状态。`--no-verify-ssl`

亚马逊托管区块链 (AMB) Access Polygon 的身份和访问管理

AWS Identity and Access Management (IAM) AWS 服务 可帮助管理员安全地控制对 AWS 资源的访问权限。IAM 管理员控制谁可以进行身份验证 (登录) 和授权 (有权限) 使用 AMB Access Polygon 资源。您可以使用 IAM AWS 服务，无需支付额外费用。

主题

- [受众](#)
- [使用身份进行身份验证](#)
- [使用策略管理访问](#)
- [亚马逊托管区块链 \(AMB\) Access Polygon 如何与 IAM 配合使用](#)
- [亚马逊托管区块链 \(AMB\) Access Polygon 的基于身份的策略示例](#)
- [对亚马逊托管区块链 \(AMB\) Access Polygon 身份和访问进行故障排除](#)

受众

您的使用方式 AWS Identity and Access Management (IAM) 因您的角色而异：

- 服务用户：如果您无法访问功能，请从管理员处请求权限（请参阅[对亚马逊托管区块链 \(AMB\) Access Polygon 身份和访问进行故障排除](#)）
- 服务管理员：确定用户访问权限并提交权限请求（请参阅[亚马逊托管区块链 \(AMB\) Access Polygon 如何与 IAM 配合使用](#)）
- IAM 管理员：编写用于管理访问权限的策略（请参阅[亚马逊托管区块链 \(AMB\) Access Polygon 的基于身份的策略示例](#)）

使用身份进行身份验证

身份验证是您 AWS 使用身份凭证登录的方式。您必须以 IAM 用户身份进行身份验证 AWS 账户根用户，或者通过担任 IAM 角色进行身份验证。

您可以使用来自身份源的证书 AWS IAM Identity Center（例如（IAM Identity Center）、单点登录身份验证或 Google/Facebook 证书，以联合身份登录。有关登录的更多信息，请参阅《AWS 登录 用户指南》中的[如何登录您的 AWS 账户](#)。

对于编程访问，AWS 提供 SDK 和 CLI 来对请求进行加密签名。有关更多信息，请参阅《IAM 用户指南》中的[适用于 API 请求的 AWS 签名版本 4](#)。

AWS 账户 root 用户

创建时 AWS 账户，首先会有一个名为 AWS 账户 root 用户的登录身份，该身份可以完全访问所有资源 AWS 服务和资源。我们强烈建议不要使用根用户进行日常任务。有关需要根用户凭证的任务，请参阅《IAM 用户指南》中的[需要根用户凭证的任务](#)。

联合身份

作为最佳实践，要求人类用户使用与身份提供商的联合身份验证才能 AWS 服务 使用临时证书进行访问。

联合身份是指来自您的企业目录、Web 身份提供商的用户 Directory Service ，或者 AWS 服务 使用来自身份源的凭据进行访问的用户。联合身份代入可提供临时凭证的角色。

要集中管理访问权限，建议使用。AWS IAM Identity Center 有关更多信息，请参阅《AWS IAM Identity Center 用户指南》中的[什么是 IAM Identity Center ?](#)。

IAM 用户和群组

[IAM 用户](#)是对某个人员或应用程序具有特定权限的一个身份。建议使用临时凭证，而非具有长期凭证的 IAM 用户。有关更多信息，请参阅 IAM 用户指南中的[要求人类用户使用身份提供商的联合身份验证才能 AWS 使用临时证书进行访问](#)。

[IAM 组](#)指定一组 IAM 用户，便于更轻松地对大量用户进行权限管理。有关更多信息，请参阅《IAM 用户指南》中的[IAM 用户使用案例](#)。

IAM 角色

[IAM 角色](#)是具有特定权限的身份，可提供临时凭证。您可以通过[从用户切换到 IAM 角色 \(控制台\)](#) 或调用 AWS CLI 或 AWS API 操作来代入角色。有关更多信息，请参阅《IAM 用户指南》中的[担任角色的方法](#)。

IAM 角色对于联合用户访问、临时 IAM 用户权限、跨账户访问、跨服务访问以及在 Amazon EC2 上运行的应用程序非常有用。有关更多信息，请参阅《IAM 用户指南》中的[IAM 中的跨账户资源访问](#)。

使用策略管理访问

您可以 AWS 通过创建策略并将其附加到 AWS 身份或资源来控制中的访问权限。策略定义了与身份或资源关联时的权限。AWS 在委托人提出请求时评估这些政策。大多数策略都以 JSON 文档的 AWS 形式存储在中。有关 JSON 策略文档的更多信息，请参阅《IAM 用户指南》中的[JSON 策略概述](#)。

管理员使用策略，通过定义哪个主体可以在什么条件下对哪些资源执行哪些操作来指定谁有权访问什么。

默认情况下，用户和角色没有权限。IAM 管理员创建 IAM 策略并将其添加到角色中，然后用户可以担任这些角色。IAM 策略定义权限，与执行操作所用的方法无关。

基于身份的策略

基于身份的策略是您附加到身份（用户、组或角色）的 JSON 权限策略文档。这些策略控制身份可以执行什么操作、对哪些资源执行以及在什么条件下执行。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的[使用客户管理型策略定义自定义 IAM 权限](#)。

基于身份的策略可以是内联策略（直接嵌入到单个身份中）或托管策略（附加到多个身份的独立策略）。要了解如何在托管策略和内联策略之间进行选择，请参阅《IAM 用户指南》中的[在托管策略与内联策略之间进行选择](#)。

基于资源的策略

基于资源的策略是附加到资源的 JSON 策略文档。示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。您必须在基于资源的策略中[指定主体](#)。

基于资源的策略是位于该服务中的内联策略。您不能在基于资源的策略中使用 IAM 中的 AWS 托管策略。

其他策略类型

AWS 支持其他策略类型，这些策略类型可以设置更常见的策略类型授予的最大权限：

- 权限边界 – 设置基于身份的策略可以授予 IAM 实体的最大权限。有关更多信息，请参阅《IAM 用户指南》中的[IAM 实体的权限边界](#)。
- 服务控制策略 (SCPs)-在中指定组织或组织单位的最大权限 AWS Organizations。有关更多信息，请参阅《AWS Organizations 用户指南》中的[服务控制策略](#)。
- 资源控制策略 (RCPs)-设置账户中资源的最大可用权限。有关更多信息，请参阅《AWS Organizations 用户指南》中的[资源控制策略 \(RCPs\)](#)。
- 会话策略 – 在为角色或联合用户创建临时会话时，作为参数传递的高级策略。有关更多信息，请参阅《IAM 用户指南》中的[会话策略](#)。

多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解在涉及多种策略类型时如何 AWS 确定是否允许请求，请参阅 IAM 用户指南中的[策略评估逻辑](#)。

亚马逊托管区块链 (AMB) Access Polygon 如何与 IAM 配合使用

在使用 IAM 管理对 AMB Access Polygon 的访问权限之前，请先了解有哪些 IAM 功能可用于 AMB Access Polygon。

您可以与亚马逊托管区块链 (AMB) Access Polygon 一起使用的 IAM 功能

IAM 功能	AMB 访问多边形支持
基于身份的策略	是
基于资源的策略	否
策略操作	是
策略资源	否
策略条件密钥	否
ACLs	否
ABAC (策略中的标签)	否
临时凭证	否
主体权限	否
服务角色	否
服务关联角色	否

要全面了解 AMB Access Polygon 和其他功能如何 AWS 服务 与大多数 IAM 功能配合使用，请参阅 IAM 用户指南中的与 IAM [配合使用的AWS 服务](#)。

AMB Access Polygon 的基于身份的策略

支持基于身份的策略：是

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的[使用客户管理型策略定义自定义 IAM 权限](#)。

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。要了解可在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的[IAM JSON 策略元素引用](#)。

AMB Access Polygon 的基于身份的策略示例

要查看 AMB Access Polygon 基于身份的策略的示例，请参阅。[亚马逊托管区块链 \(AMB\) Access Polygon 的基于身份的策略示例](#)

AMB Access Polygon 中基于资源的策略

支持基于资源的策略：否

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中[指定主体](#)。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

要启用跨账户访问，您可以将整个账户或其他账户中的 IAM 实体指定为基于资源的策略中的主体。有关更多信息，请参阅《IAM 用户指南》中的[IAM 中的跨账户资源访问](#)。

AMB Access Polygon 的策略操作

支持策略操作：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

JSON 策略的 Action 元素描述可用于在策略中允许或拒绝访问的操作。在策略中包含操作以授予执行关联操作的权限。

要查看 AMB Access Polygon 操作列表，请参阅《服务授权参考》中的[Amazon Managed Blockchain \(AMB\) Access Polygon 定义的操作](#)。

AMB Access Polygon 中的策略操作在操作前使用以下前缀：

```
managedblockchain:
```

要在单个语句中指定多项操作，请使用逗号将它们隔开。

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

您也可以使用通配符 (*) 指定多个操作。例如，要指定以单词 InvokeRpcPolygon 开头的所有操作，包括以下操作：

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

要查看 AMB Access Polygon 基于身份的策略的示例，请参阅 [亚马逊托管区块链 \(AMB\) Access Polygon 的基于身份的策略示例](#)

AMB Access Polygon 的策略资源

支持策略资源：否

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

Resource JSON 策略元素指定要向其应用操作的一个或多个对象。作为最佳实践，请使用其 [Amazon 资源名称 \(ARN \)](#) 指定资源。对于不支持资源级权限的操作，请使用通配符 (*) 指示语句应用于所有资源。

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

要查看 AMB Access Polygon 资源类型及其列表 ARNs，请参阅《服务授权参考》中的 [Amazon Managed Blockchain \(AMB\) Access Polygon 定义的资源](#)。要了解您可以使用哪些操作来指定每种资源的 ARN，请参阅 [Amazon Managed Blockchain \(AMB\) Access Polygon 定义的操作](#)。

要查看 AMB Access Polygon 基于身份的策略的示例，请参阅 [亚马逊托管区块链 \(AMB\) Access Polygon 的基于身份的策略示例](#)

AMB Access Polygon 的策略条件密钥

支持特定于服务的策略条件键：否

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

Condition 元素根据定义的条件指定语句何时执行。您可以创建使用[条件运算符](#)（例如，等于或小于）的条件表达式，以使策略中的条件与请求中的值相匹配。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的[AWS 全局条件上下文密钥](#)。

要查看 AMB Access Polygon 条件密钥列表，请参阅《服务授权参考》中的 [Amazon Managed Blockchain \(AMB\) 访问多边形的条件](#) 密钥。要了解您可以使用哪些操作和资源使用条件密钥，请参阅 [Amazon Managed Blockchain \(AMB\) Access Polygon 定义的操作](#)。

要查看 AMB Access Polygon 基于身份的策略的示例，请参阅 [亚马逊托管区块链 \(AMB\) Access Polygon 的基于身份的策略示例](#)

ACLs 在 AMB 访问多边形中

支持 ACLs：否

访问控制列表 (ACLs) 控制哪些委托人（账户成员、用户或角色）有权访问资源。ACLs 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

带有 AMB 接入多边形的 ABAC

支持 ABAC（策略中的标签）：否

基于属性的访问权限控制（ABAC）是一种授权策略，该策略基于称为标签的属性来定义权限。您可以将标签附加到 IAM 实体和 AWS 资源，然后设计 ABAC 策略以允许在委托人的标签与资源上的标签匹配时进行操作。

要基于标签控制访问，您需要使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 条件键在策略的[条件元素](#)中提供标签信息。

如果某个服务对于每种资源类型都支持所有这三个条件键，则对于该服务，该值为是。如果某个服务仅对于部分资源类型支持所有这三个条件键，则该值为部分。

有关 ABAC 的更多信息，请参阅《IAM 用户指南》中的[使用 ABAC 授权定义权限](#)。要查看设置 ABAC 步骤的教程，请参阅《IAM 用户指南》中的[使用基于属性的访问权限控制 \(ABAC\)](#)。

在 AMB Access Polygon 中使用临时证书

支持临时凭证：否

临时证书提供对 AWS 资源的短期访问权限，并且是在您使用联合身份或切换角色时自动创建的。AWS 建议您动态生成临时证书，而不是使用长期访问密钥。有关更多信息，请参阅《IAM 用户指南》中的 [IAM 中的临时安全凭证](#) 和 [使用 IAM 的 AWS 服务](#)

AMB Access Polygon 的跨服务主体权限

支持转发访问会话 (FAS) : 否

转发访问会话 (FAS) 使用调用主体的权限 AWS 服务，再加上 AWS 服务 向下游服务发出请求的请求。有关发出 FAS 请求时的策略详情，请参阅 [转发访问会话](#)。

AMB Access Polygon 的服务角色

支持服务角色 : 否

服务角色是由一项服务担任、代表您执行操作的 [IAM 角色](#)。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的 [创建向 AWS 服务委派权限的角色](#)。

Warning

更改服务角色的权限可能会中断 AMB Access Polygon 的功能。仅当 AMB Access Polygon 提供相关指导时才编辑服务角色。

AMB Access Polygon 的服务相关角色

支持服务相关角色 : 否

服务相关角色是一种与服务相关联的 AWS 服务角色。服务可以代入代表您执行操作的角色。服务相关角色出现在您的 AWS 账户，并且归服务所有。IAM 管理员可以查看但不能编辑服务关联角色的权限。

有关创建或管理服务相关角色的详细信息，请参阅 [能够与 IAM 搭配使用的 AWS 服务](#)。在表中查找服务相关角色列中包含 Yes 的表。选择是链接以查看该服务的服务相关角色文档。

亚马逊托管区块链 (AMB) Access Polygon 的基于身份的策略示例

默认情况下，用户和角色无权创建或修改 AMB Access Polygon 资源。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。

要了解如何使用这些示例 JSON 策略文档创建基于 IAM 身份的策略，请参阅《IAM 用户指南》中的[创建 IAM 策略 \(控制台\)](#)。

有关 AMB Access Polygon 定义的操作和资源类型 (包括每种资源类型的格式) 的详细信息，请参阅《服务授权参考》中的[Amazon Managed Blockchain \(AMB\) 访问多边形的操作、资源和条件密钥](#)。

ARNs

主题

- [策略最佳实践](#)
- [使用 AMB Access Polygon 控制台](#)
- [允许用户查看他们自己的权限](#)
- [访问多边形网络](#)

策略最佳实践

基于身份的策略决定了某人是否可以在您的账户中创建、访问或删除 AMB Access Polygon 资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下指南和建议：

- 开始使用 AWS 托管策略并转向最低权限权限 — 要开始向用户和工作负载授予权限，请使用为许多常见用例授予权限的 AWS 托管策略。它们在你的版本中可用 AWS 账户。我们建议您通过定义针对您的用例的 AWS 客户托管策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的[AWS 托管策略或工作职能的 AWS 托管策略](#)。
- 应用最低权限：在使用 IAM 策略设置权限时，请仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的[IAM 中的策略和权限](#)。
- 使用 IAM 策略中的条件进一步限制访问权限：您可以向策略添加条件来限制对操作和资源的访问。例如，您可以编写策略条件来指定必须使用 SSL 发送所有请求。如果服务操作是通过特定 AWS 服务的 (例如) 使用的，则也可以使用条件来授予对服务操作的访问权限 CloudFormation。有关更多信息，请参阅《IAM 用户指南》中的[IAM JSON 策略元素：条件](#)。
- 使用 IAM Access Analyzer 验证您的 IAM 策略，以确保权限的安全性和功能性：IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM 策略语言 (JSON) 和 IAM 最佳实践。IAM Access Analyzer 提供 100 多项策略检查和可操作的建议，以帮助您制定安全且功能性强的策略。有关更多信息，请参阅《IAM 用户指南》中的[使用 IAM Access Analyzer 验证策略](#)。
- 需要多重身份验证 (MFA)-如果 AWS 账户您的场景需要 IAM 用户或根用户，请启用 MFA 以提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅《IAM 用户指南》中的[使用 MFA 保护 API 访问](#)。

有关 IAM 中的最佳实操的更多信息，请参阅《IAM 用户指南》中的 [IAM 中的安全最佳实践](#)。

使用 AMB Access Polygon 控制台

要访问亚马逊托管区块链 (AMB) Access Polygon 控制台，您必须拥有一组最低权限。这些权限必须允许您列出和查看有关您 AWS 账户的 AMB Access Polygon 资源的详细信息。如果创建比必需的最低权限更为严格的基于身份的策略，对于附加了该策略的实体（用户或角色），控制台将无法按预期正常运行。

对于仅调用 AWS CLI 或 AWS API 的用户，您无需为其设置最低控制台权限。相反，只允许访问与其尝试执行的 API 操作相匹配的操作。

为确保用户和角色仍然可以使用 AMB Access Polygon 控制台，还要将 AMB Access Polygon *ConsoleAccess* 或 *ReadOnly* AWS 托管策略附加到实体。有关更多信息，请参阅《IAM 用户指南》中的 [为用户添加权限](#)。

允许用户查看他们自己的权限

该示例说明了您如何创建策略，以允许 IAM 用户查看附加到其用户身份的内联和托管式策略。此策略包括在控制台上或使用 AWS CLI 或 AWS API 以编程方式完成此操作的权限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "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": "*"
}
]
}

```

访问多边形网络

Note

要访问 Polygon 的公共终端节点mainnet并mainnet进行 JSON-RPC 调用，您需要具有 AMB Access Polygon 相应 IAM 权限的用户证书（AWS_ACCESS_KEY_ID和AWS_SECRET_ACCESS_KEY）。

Example用于访问所有 Polygon 网络的 IAM 策略

此示例授予您中的 IAM 用户 AWS 账户 访问所有 Polygon 网络的权限。

JSON

```

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

```

Example用于访问 Polygon 主网网络的 IAM 策略

此示例授予您中的 IAM 用户 AWS 账户 访问 Polygon 主网网络的权限。

JSON

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

对亚马逊托管区块链 (AMB) Access Polygon 身份和访问进行故障排除

使用以下信息来帮助您诊断和修复在使用 AMB Access Polygon 和 IAM 时可能遇到的常见问题。

主题

- [我无权在 AMB Access Polygon 中执行操作](#)
- [我无权执行 iam : PassRole](#)
- [我想允许我以外的人访问我的 AMB Access Polygon 资源 AWS 账户](#)

我无权在 AMB Access Polygon 中执行操作

如果您收到错误提示，指明您无权执行某个操作，则必须更新策略以允许执行该操作。

当 mateojackson IAM 用户尝试使用控制台查看有关虚构 *my-example-widget* 资源的详细信息，但不拥有虚构 `managedblockchain::GetWidget` 权限时，会发生以下示例错误。

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

在此情况下，必须更新 mateojackson 用户的策略，以允许使用 managedblockchain::GetWidget 操作访问 my-example-widget 资源。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我无权执行 iam : PassRole

如果您收到错误消息，提示您无权执行 iam:PassRole 操作，则必须更新您的策略以允许您将角色传递给 AMB Access Polygon。

有些 AWS 服务 允许您将现有角色传递给该服务，而不是创建新的服务角色或服务相关角色。为此，您必须具有将角色传递到服务的权限。

当名为的 IAM 用户 marymajor 尝试使用控制台在 AMB Access Polygon 中执行操作时，会出现以下示例错误。但是，服务必须具有服务角色所授予的权限才可执行此操作。Mary 不具有将角色传递到服务的权限。

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

在这种情况下，必须更新 Mary 的策略以允许她执行 iam:PassRole 操作。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我想允许我以外的人访问我的 AMB Access Polygon 资源 AWS 账户

您可以创建一个角色，以便其他账户中的用户或您组织外的人员可以使用该角色来访问您的资源。您可以指定谁值得信赖，可以代入角色。对于支持基于资源的策略或访问控制列表 (ACLs) 的服务，您可以使用这些策略向人们授予访问您的资源的权限。

要了解更多信息，请参阅以下内容：

- 要了解 AMB Access Polygon 是否支持这些功能，请参阅 [亚马逊托管区块链 \(AMB\) Access Polygon 如何与 IAM 配合使用](#)。
- 要了解如何提供对您拥有的资源的访问权限 AWS 账户，请参阅 [IAM 用户指南中的向您拥有 AWS 账户的另一个 IAM 用户提供访问权限](#)。

- 要了解如何向第三方提供对您的资源的访问[权限 AWS 账户](#)，请参阅 IAM 用户指南中的[向第三方提供访问权限](#)。AWS 账户
- 要了解如何通过身份联合验证提供访问权限，请参阅《IAM 用户指南》中的[为经过外部身份验证的用户 \(身份联合验证\) 提供访问权限](#)。
- 要了解使用角色和基于资源的策略进行跨账户访问之间的差别，请参阅《IAM 用户指南》中的[IAM 中的跨账户资源访问](#)。

使用记录亚马逊托管区块链 (AMB) 访问 Polygon 事件 AWS CloudTrail

Note

亚马逊托管区块链 (AMB) Access Polygon 不支持管理事件。

Amazon Manage AWS CloudTrail d Blockchain 运行在该服务上，该服务提供托管区块链中用户、角色或 AWS 服务所采取的操作的记录。CloudTrail 捕获谁将托管区块链的 AMB Access Polygon 端点作为数据平面事件调用。

如果您创建了经过正确配置的跟踪，该跟踪已订阅以接收所需的数据平面事件，则可以接收持续向 S3 存储桶传送的 AMB Access Polygon 相关 CloudTrail 事件。使用收集的信息 CloudTrail，您可以确定是否向其中一个 AMB Access Polygon 终端节点发出了请求、请求来自哪个 IP 地址、谁发出了请求、发出请求的时间以及其他详细信息。

要了解更多信息 CloudTrail，请参阅 [《AWS CloudTrail 用户指南》](#)。

AMB 访问中的多边形信息 CloudTrail

CloudTrail 在你创建 AWS 账户 时已在你上启用。但是，您必须配置数据平面事件以查看谁调用了 AMB Access Polygon 端点。

要持续记录您的事件 AWS 账户，包括 AMB Access Polygon 的事件，请创建跟踪。跟踪允许 CloudTrail 将日志文件传送到 S3 存储桶。默认情况下，在控制台中创建跟踪记录时，此跟踪记录应用于所有 AWS 区域。跟踪记录 AWS 分区中所有受支持区域的事件，并将日志文件传送到您指定的 S3 存储桶。此外，您可以配置其他，AWS 服务 以进一步分析并根据 CloudTrail 日志中收集的事件数据采取行动。有关更多信息，请参阅下列内容：

- [CloudTrail 用于追踪 Polygon JSON-RPCs](#)
- [创建跟踪记录概述](#)
- [CloudTrail 支持的服务和集成](#)
- [配置 Amazon SNS 通知 CloudTrail](#)
- [接收来自多个区域的 CloudTrail 日志文件和接收来自多个账户的 CloudTrail 日志文件](#)

通过分析 CloudTrail 数据事件，您可以监控谁调用了 AMB Access Polygon 端点。

每个事件或日志条目都包含有关生成请求的人员信息。身份信息有助于您确定以下内容：

- 请求是使用根用户证书还是 AWS Identity and Access Management (IAM) 用户凭证发出
- 请求是使用角色还是联合用户的临时安全凭证发出的
- 请求是否由其他人提出 AWS 服务

有关更多信息，请参阅 [CloudTrail userIdentity 元素](#)。

了解 AMB Access Polygon 日志文件条目

对于数据平面事件，跟踪是一种配置，允许将事件作为日志文件传送到指定的 S3 存储桶。每个 CloudTrail 日志文件都包含一个或多个日志条目，这些条目代表来自任何来源的单个请求。这些条目提供有关请求操作的详细信息，包括操作的日期和时间以及任何相关的请求参数。

Note

CloudTrail 日志文件中的数据事件不是 AMB Access Polygon API 调用的有序堆栈跟踪，因此它们不会按任何特定的顺序出现。

CloudTrail 用于追踪 Polygon JSON-RPCs

您可以使用 CloudTrail 来跟踪您的账户中谁调用了 AMB Access Polygon 端点，以及哪个 JSON-RPC 被调用为数据事件。默认情况下，当您创建跟踪时，不会记录数据事件。要记录谁将 AMB Access Polygon 终端节点调用为 CloudTrail 数据事件，您必须将要为其收集活动的支持的资源或资源类型明确添加到跟踪中。AMB Access Polygon 支持使用 AWS 管理控制台 AWS CLI、和 SDK 添加数据事件。有关更多信息，请参阅 [《AWS CloudTrail 用户指南》中的使用高级选择器记录事件](#)。

要在跟踪中记录数据事件，请在创建跟踪后使用 [put-event-selectors](#) 操作。使用该 `--advanced-event-selectors` 选项指定 `AWS::ManagedBlockchain::Network` 资源类型，以便开始记录数据事件，从而确定谁调用了 AMB Access Polygon 端点。

Example 所有账户的 AMB Access Polygon 终端节点请求的数据事件日志条目

以下示例演示如何使用该 `put-event-selectors` 操作来记录您的账户针对该 `us-east-1` 区域跟踪 `my-polygon-trail` 的所有 AMB Access Polygon 终端节点请求。

```
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"] } ] ]'
```

订阅后，您可以跟踪连接到上一个示例中指定的跟踪的 S3 存储桶中的使用情况。

以下结果显示了由收集的信息 CloudTrail 的数据事件日志条目 CloudTrail。您可以确定 Polygon JSON-RPC 请求是向其中一个 AMB Access Polygon 端点发出的、请求来自的 IP 地址、谁发出了请求、发出请求的时间以及其他详细信息。出于安全考虑，以下示例中的某些值已被模糊处理，但完全出现在实际日志条目中。

```
{
  "eventVersion": "1.09",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROA554U062RJ7KSB7FAX: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 用户指南的文档历史记录

下表描述了 AMB Access Polygon 的文档版本。

变更	说明	日期
更新了 JSON-RPC 的配额	AMB Access Polygon 为每个支持的 JSON-RPC 支持的配额已更新。	2024 年 4 月 12 日
终止对孟买测试网网络的支持	AMB Access Polygon 于 2024 年 4 月 15 日终止了对孟买测试网的支持。	2024 年 4 月 10 日
新增教程主题	来自 AWS re: Post 社区文章部分的 AMB Access Polygon 教程。	2024 年 4 月 9 日
公开预览	亚马逊托管区块链 (AMB) Access Polygon 服务的公开预览版。	2023 年 11 月 24 日