aws

開発者ガイド

AMB アクセスポリゴン

AMB アクセスポリゴン: 開発者ガイド

Table of Contents

Amazon Managed Blockchain (AMB) Access Polygon はプレビューリリースであり、変更される可能性があります。

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。

Amazon Managed Blockchain (AMB) アクセスポリゴンとは

Amazon Managed Blockchain (AMB) Access Polygon は、Polygon ブロックチェーン上に回復力のある Web3 アプリケーションを構築するのに役立つフルマネージドサービスです。AMB Access Polygon は、Polygon ブロックチェーンへの即時かつサーバーレスアクセスを提供します。

Polygon は、Ethereum Virtual Machine (EVM) を基盤として使用するスケーリングソリューションです。ポリゴンブロックチェーンは、トランザクションスループットが高く、トランザクション料金が低いことで知られています。ポリゴンブロックチェーンは proof-of-stake コンセンサスメカニズムを使用します。ポリゴンは、NFT、Web3 ゲーム、トークン化のユースケースなどに関連する分散アプリケーション (dApps) の構築によく使用されます。 NFTs

このガイドでは、Amazon Managed Blockchain (AMB) アクセスポリゴンを使用してポリゴンブロックチェーンリソースを作成および管理する方法について説明します。

AMB Access Polygon を初めて使用するユーザーのリソース

AMB Access Polygon を初めて使用する場合は、まず以下のセクションを読むことをお勧めします。

- 主な概念: Amazon Managed Blockchain (AMB) アクセスポリゴン
- Amazon Managed Blockchain (AMB) Access Polygon の開始方法
- AMB Access Polygon でサポートされている Managed Blockchain API と JSON-RPCs

主な概念: Amazon Managed Blockchain (AMB) アクセスポリゴン

> **Note**
>
> このガイドでは、ポリゴンに不可欠な概念に精通していることを前提としています。これらの概念には、ステーク、dAppsトランザクション、ウォレット、スマートコントラクト、ポリゴン (POL、以前は MATIC) などがあります。Amazon Managed Blockchain (AMB) Access Polygon を使用する前に、Polygon Development Documentation と Polygon Wiki を確認することをお勧めします。

Amazon Managed Blockchain (AMB) Access Polygon では、ノードを含む Polygon インフラストラクチャをプロビジョニングおよび管理することなく、Polygon Mainnet および Polygon Mainnet ネットワークへのサーバーレスアクセスが提供されます。ネットワーク上のポリゴンノードは、ポリゴンブロックチェーンの状態をまとめて保存し、トランザクションを検証し、ブロックチェーンの状態を変更するコンセンサスに参加します。このマネージドサービスを使用して、Polygon ネットワークに迅速かつオンデマンドでアクセスできるため、全体的な所有コストを削減できます。

AMB Access Polygon を使用すると、JSON Remote Procedure (JSON-RPC) 呼び出しにアクセスできます。ポリゴン JSON-RPCs を呼び出して、マネージドブロックチェーンによって管理されるノードを介してポリゴンブロックチェーンと通信できます。AMB Access Polygon サービスを使用して、Polygon ブロックチェーンとやり取りする分散アプリケーション (dApps) を開発および使用できます。dApps の不可欠な部分は、スマートコントラクト です。AMB アクセスポリゴンを使用して、スマートコントラクトを作成してポリゴンブロックチェーンにデプロイできます。また、Polygon ネットワークにピアリングされているすべてのノードで分散的に実行される AMB Access Polygon エンドポイントに対して JSON-RPCs を呼び出すことで、ウォレット、トランザクションの詳細、見積り料金などの残高を確認することもできます。Polygon ネットワークへのピアは、スマートコントラクトを開発およびデプロイできます。

> **Important**
>
> お客様は、Polygon アドレスを作成、保守、使用、管理する責任があります。また、Polygon アドレスの内容についても責任を負います。 AWS は、Amazon Managed Blockchain で Polygon ノードを使用してデプロイまたは呼び出されたトランザクションについては責任を負いません。

Amazon Managed Blockchain (AMB) Access Polygon を使用する際の考慮事項と制限事項

Amazon Managed Blockchain (AMB) アクセスポリゴンを使用する場合は、次の点を考慮してください。

- サポートされているポリゴンネットワーク

  AMB Access Polygon は、次のパブリックネットワークをサポートしています。

  - Mainnet — proof-of-stake コンセンサスによって保護され、ポリゴン (POL) トークンが発行されて処理されるパブリックポリゴンブロックチェーン。Mainnet のトランザクションは実際の値 (つまり、実際のコストが発生します) を持ち、パブリックブロックチェーンに記録されます。

- Polygon でサポートされなくなったネットワーク
  - Polygon Labs から伝えられているように、ムンバイテストネットネットワークは 4 月中旬にサンセットされます。このニュースに従って、AMB Access Polygon は 2024 年 4 月 15 日にムンバイテストネットのサポートを終了しました。テストワークロードには Amoy Testnet を使用することをお勧めします。
  - プライベートネットワークはサポートされていません。
  - さらに、AMB Access Polygon には、Polygon zkEVM ネットワークのサポートは含まれていません。
- 一般的なサードパーティー製プログラミングライブラリとの互換性

  AMB Access Polygon は ethers.js などの一般的なプログラミングライブラリと互換性があり、デベロッパーは使い慣れたツールを使用して Polygon ブロックチェーンとやり取りし、既存の実装と簡単に統合したり、新しいアプリケーションを迅速に開発したりできます。
- サポートされるリージョン

  このサービスは、米国東部 (バージニア北部) リージョンでのみサポートされています。
- サービスエンドポイント

  AMB Access Polygon のサービスエンドポイントは次のとおりです。サービスと接続するには、サポートされているリージョンのいずれかを含むエンドポイントを使用する必要があります。

  - `mainnet.polygon.managedblockchain.us-east-1.amazonaws.com`
- ステーキングはサポートされていません

AMB アクセスポリゴンは、 のポリゴン (POL) 検証ノードをサポートしていません proof-of-stake。

- Polygon JSON-RPC リクエストの署名バージョン 4 の署名

  Amazon Managed Blockchain で Polygon JSON-RPCsを呼び出す場合、署名バージョン 4 の署名プロセス を使用して認証された HTTPS 接続を介して呼び出すことができます。つまり、アカウント内の AWS 認可された IAM プリンシパルのみがポリゴン JSON-RPC 呼び出しを行うことができます。これを行うには、 AWS 認証情報 (アクセスキー ID とシークレットアクセスキー) を 呼び出しで提供する必要があります。

  > **Important**
  > - ユーザー向けアプリケーションにクライアント認証情報を埋め込まないでください。
  > - IAM ポリシーを使用して、個々のポリゴン JSON-RPCs へのアクセスを制限することはできません。

- トークンベースのアクセスのサポート

  また、署名バージョン 4 (SigV4) 署名プロセスの便利な代替として、アクセサトークンを使用してポリゴンネットワークエンドポイントに JSON-RPC 呼び出しを行うこともできます。呼び出しでは、作成してパラメータとして追加する Accessor トークンの 1 つ`BILLING_TOKEN`から を指定する必要があります。

  > **Important**
  > - 利便性よりもセキュリティと監査可能性を優先する場合は、代わりに SigV4 署名プロセスを使用してください。
  > - 署名バージョン 4 (SigV4) とトークンベースのアクセスを使用して、Polygon JSON-RPCs にアクセスできます。ただし、両方のプロトコルを使用することを選択した場合、リクエストは拒否されます。
  > - ユーザー向けアプリケーションに Accessor トークンを埋め込まないでください。

- raw トランザクションの送信のみがサポートされます

  `eth_sendrawtransaction` JSON-RPC を使用して、Polygon ブロックチェーンの状態を更新するトランザクションを送信します。

Amazon Managed Blockchain (AMB) アクセスポリゴンのセットアップ

Amazon Managed Blockchain (AMB) Access Polygon を初めて使用する前に、このセクションの手順に従って を作成します AWS アカウント。次の章では、AMB アクセスポリゴンの使用を開始する方法について説明します。

AMB Access Polygon を使用するための前提条件

AWS を初めて使用する場合は、事前に が必要です AWS アカウント。

にサインアップする AWS

にサインアップすると AWS、Amazon Managed Blockchain (AMB) アクセスポリゴンを含むすべての に AWS のサービスが自動的にサインアップ AWS アカウント されます。サービスを実際に使用した分の料金のみが請求されます。

を AWS アカウント すでにお持ちの場合は、次のステップに進みます。 AWS アカウントをお持ちでない場合は、次に説明する手順に従ってアカウントを作成してください。

を作成するには AWS アカウント

1. https://portal.aws.amazon.com/billing/signup を開きます。
2. オンラインの手順に従います。

   サインアップ手順の一環として、通話呼び出しを受け取り、電話キーパッドで検証コードを入力するように求められます。

   にサインアップすると AWS アカウント、 AWS アカウントのルートユーザーが作成されます。ルートユーザーには、アカウントのすべての AWS のサービス とリソースへのアクセス権があります。セキュリティのベストプラクティスとして、ユーザーに管理アクセスを割り当て、ルートユーザーのみを使用してルートユーザーアクセスが必要なタスクを実行してください。

適切なアクセス許可を持つ IAM ユーザーを作成する

AMB Access Polygon を作成して使用するには、必要な Managed Blockchain アクションを許可するアクセス許可を持つ AWS Identity and Access Management (IAM) プリンシパル (ユーザーまたはグループ) が必要です。

Amazon Managed Blockchain でポリゴン JSON-RPCs を呼び出す場合、署名バージョン 4 の署名プロセス を使用して認証された HTTPS 接続を介して呼び出すことができます。つまり、アカウント内の AWS 認可された IAM プリンシパルのみがポリゴン JSON-RPC 呼び出しを行うことができます。これを行うには、 AWS 認証情報 (アクセスキー ID とシークレットアクセスキー) を 呼び出しで提供する必要があります。

また、署名バージョン 4 (SigV4) 署名プロセスの便利な代替として、アクセサートークンを使用してポリゴンネットワークエンドポイントに JSON-RPC 呼び出しを行うこともできます。呼び出しでは、作成してパラメータとして追加する Accessor トークンの 1 つBILLING_TOKENから を指定する必要があります。ただし、、、および SDK を使用してアクセサートークンを作成するアクセス許可を取得するには AWS Management Console AWS CLI、IAM アクセスが必要です。

IAM ユーザーを作成する方法については、「アカウントでの IAM ユーザーの作成 AWS」を参照してください。アクセス許可ポリシーをユーザーにアタッチする方法の詳細については、「IAM ユーザーのアクセス許可の変更」を参照してください。AMB Access Polygon を操作するアクセス許可をユーザーに付与するために使用できるアクセス許可ポリシーの例については、「」を参照してくださいAmazon Managed Blockchain (AMB) Access Polygon のアイデンティティベースのポリシーの例。

AWS Command Line Interfaceのインストールと設定

まだインストールしていない場合は、最新の AWS Command Line Interface （ AWS CLI) をインストールして、ターミナルの AWS リソースを操作します。詳細については、「Installing or updating the latest version of the AWS CLI」を参照してください。

> **Note**
>
> CLI アクセスには、アクセスキー ID とシークレットアクセスキーが必要です。長期のアクセスキーの代わりに一時的な認証情報をできるだけ使用します。一時的な認証情報には、アクセスキー ID、シークレットアクセスキー、および認証情報の失効を示すセキュリティトークンが含まれています。詳細については、「IAM ユーザーガイド」の「 AWS リソースでの一時的な認証情報の使用」を参照してください。

Amazon Managed Blockchain (AMB) Access Polygon の開始方法

このセクションの情報と手順を使用して、Amazon Managed Blockchain (AMB) Access Polygon の使用を開始します。

トピック

Polygon ブロックチェーンネットワークにアクセスするための IAM ポリシーを作成する

Polygon Mainnet のパブリックエンドポイントにアクセスして JSON-RPC 呼び出しを行うには、Amazon Managed Blockchain (AWS_ACCESS_KEY_IDAMBAWS_SECRET_ACCESS_KEY) アクセスポリゴンに適切な IAM アクセス許可を持つユーザー認証情報 (および) が必要です。がインストールされているターミナルで AWS CLI 、次のコマンドを実行して、両方のポリゴンエンドポイントにアクセスするための IAM ポリシーを作成します。

```
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**
>
> 前の例では、使用可能なすべてのポリゴンネットワークにアクセスできます。特定のエンドポイントにアクセスするには、次のActionコマンドを使用します。
>
> - `"managedblockchain:InvokeRpcPolygonMainnet"`

ポリシーを作成したら、そのポリシーを IAM ユーザーのロールにアタッチして有効にします。で AWS Management Console、IAM サービスに移動し、IAM ユーザーに割り当てられたロールにポリシーAmazonManagedBlockchainPolygonAccessをアタッチします。

を使用して AMB Access RPC エディタで Polygon リモートプロシージャコール (RPC) リクエストを行う AWS Management Console

AMB Access Polygon AWS Management Console を使用して、 でリモートプロシージャコール (RPCsを編集、設定、送信できます。これらの RPCs を使用すると、データの取得や、Polygon ネットワークへのトランザクションの送信など、Polygon ネットワーク上のデータの読み取りとトランザクションの書き込みを行うことができます。

Example

次の例は、eth_getBlockByNumberRPC を使用して最新のブロックに関する情報を取得する方法を示しています。強調表示された変数を独自の入力に変更するか、リストされている RPC メソッドのいずれかを選択し、必要な関連する入力を入力します。

1. https://console.aws.amazon.com/managedblockchain/ で Managed Blockchain コンソールを開きます。
2. RPC エディタ を選択します。
3. リクエストセクションで、ブロックチェーンネットワーク *POLYGON_MAINNET*として を選択します。

4. RPC メソッド *eth_getBlockByNumber*として を選択します。
5. ******latest**として を入力し、フルトランザクションフラグ*False*として を選択します。
6. 次に、RPC の送信 を選択します。
7. latest ブロックの結果は、「レスポンス」セクションで取得できます。その後、詳細な分析やアプリケーションのビジネスロジックでの使用のために、完全な raw トランザクションをコピーできます。

詳細については、RPCs」を参照してください。

を使用して **awscurl**で AMB Access Polygon JSON-RPC リクエストを行う AWS CLI

Example

AMB Access Polygon エンドポイントに Polygon JSON-RPC リクエストを行うには、署名バージョン 4 (SigV4) を使用して IAM ユーザー認証情報でリクエストに署名します。awscurl コマンドラインツールは、SigV4 を使用して AWS サービスへのリクエストに署名するのに役立ちます。詳細については、awscurl README.md を参照してください。

ご使用のオペレーティングシステムに適した方法awscurlを使用して をインストールします。macOS では、 が推奨アプリケーション HomeBrew です。

```
brew install awscurl
```

を既にインストールして設定している場合は AWS CLI、IAM ユーザー認証情報とデフォルト AWS リージョン が環境で設定され、 にアクセスできますawscurl。を使用してawscurl、RPC を呼び出してリクエストを Polygon Mainnet eth_getBlockByNumber に送信します。この呼び出しは、情報を取得するブロック番号に対応する文字列パラメータを受け入れます。

次のコマンドは、 params配列のブロック番号を使用して、ヘッダーを取得する特定のブロックを選択して、Polygon Mainnet からブロックデータを取得します。

```
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**
>
> を使用して同じリクエストを行いcurl、トークンを使用して AMB アクセストークンベースのアクセス機能Accessorを実行することもできます。詳細については、「AMB Access Polygon リクエストを作成するためのトークンベースのアクセス用の Accessor トークンの作成と管理」を参照してください。
>
> ```
> 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'
> ```

いずれかのコマンドからのレスポンスは、最新のブロックに関する情報を返します。説明のために次の例を参照してください。

```
{"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":[]}}
```

Node.js でポリゴン JSON-RPC リクエストを作成する

Node.RPCs を呼び出すことができます。または、AXIOS などのサードパーティーライブラリを使用できます。 https://nodejs.org/api/https.html次の Node.js の例は、署名バージョン 4 (SigV4) とトークンベースのアクセス の両方を使用して、AMB アクセスポリゴンエンドポイントにポリゴン JSON-RPC リクエストを行う方法を示しています。最初の例では、あるアドレスから別のアドレスにトランザクションを送信し、次の例では、ブロックチェーンからトランザクションの詳細と残高情報をリクエストします。

Example

このサンプル Node.js スクリプトを実行するには、次の前提条件を適用します。

1. マシンにノードバージョンマネージャー (nvm) と Node.js がインストールされている必要があります。OS のインストール手順については、「」を参照してください。
2. `node --version` コマンドを使用して、Node バージョン 18 以降を使用していることを確認します。必要に応じて、 `nvm install v18.12.0` コマンドの後に `nvm use v18.12.0` コマンドを使用して、Node の LTS バージョンであるバージョン 18 をインストールできます。
3. 環境変数 AWS_ACCESS_KEY_IDおよび には、アカウントに関連付けられている認証情報が含まれているAWS_SECRET_ACCESS_KEY必要があります。

   次のコマンドを使用して、これらの変数をクライアントの文字列としてエクスポートします。次の文字列の赤の値を、IAM ユーザーアカウントの適切な値に置き換えます。

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

すべての前提条件を満たしたら、任意のコードエディタを使用して、次のファイルをローカル環境のディレクトリにコピーします。

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**
>
> 次のコードでは、ハードコードされたプライベートキーを使用して、デモンストレーションのみEthers.jsを目的として を使用するウォレット Signer を生成します。このコードには実際の資金があり、セキュリティ上のリスクがあるため、本番環境では使用しないでください。
> 必要に応じて、アカウントチームに連絡して、ウォレットと署名者のベストプラクティスについてアドバイスしてください。

```
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");
```

これらのファイルがディレクトリに保存されたら、次のコマンドを使用してコードの実行に必要な依存関係をインストールします。

```
npm install
```

Node.js でトランザクションを送信する

前述の例では、トランザクションに署名し、AMB アクセスポリゴンを使用してポリゴンメインネットにブロードキャストすることで、あるアドレスから別のアドレスにネイティブポリゴンメインネットトークン (POL) を送信します。これを行うには、 `sendTx.js`スクリプトを使用します。これは`Ethers.js`、Polygon などの Ethereum および Ethereum 互換ブロックチェーンとやり取りするための一般的なライブラリです。トークンベースのアクセス`billingToken`用のアクセサートークンの 、トランザクションに署名するプライベートキー、POL を受け取る受信者のアドレスなど、赤で強調表示されているコード内の 3 つの変数を置き換える必要があります。 https://docs.aws.amazon.com/managed-blockchain/latest/ambp-dg/polygon-tokens.html

> **Tip**
>
> 資金を失うリスクを排除するために、既存のウォレットを再利用するのではなく、この目的のために新しいプライベートキー (ウォレット) を作成することをお勧めします。Ethers ラ

イブラリの Wallet クラスメソッド createRandom () を使用して、テストするウォレットを生成できます。さらに、Polygon Mainnet から POL をリクエストする必要がある場合は、パブリック POL 蛇口を使用して、テストに使用する少量をリクエストできます。

billingToken、ウォレットのプライベートキー 、受信者のアドレスをコードに追加したら、次のコードを実行して、アドレスから別のアドレスに送信される .0001 POL のトランザクションに署名し、AMB アクセスポリゴンを使用して eth_sendRawTransactionJSON-RPC を呼び出すポリゴンメインネットにブロードキャストします。

```
node sendTx.js
```

返されるレスポンスは次のようになります。

```
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: []
}
```

レスポンスはトランザクションの受信を構成します。プロパティ の値を保存しますhash。これは、ブロックチェーンに送信したトランザクションの識別子です。読み取りトランザクションの例でこのプロパティを使用して、このトランザクションに関する追加の詳細をポリゴンメインネットから取得します。

blockNumber と blockHashはレスポンスnullに含まれていることに注意してください。これは、トランザクションがポリゴンネットワークのブロックにまだ記録されていないためです。これらの値は後で定義され、次のセクションでトランザクションの詳細をリクエストすると表示される場合があることに注意してください。

Node.js でトランザクションを読み取る

このセクションでは、以前に送信されたトランザクションのトランザクション詳細をリクエストし、AMB Access Polygon を使用して Polygon Mainnet への読み取りリクエストを使用して受信者アドレスの POL 残高を取得します。readTx.js ファイルで、 というラベル*your-transaction-id*の付いた変数を、前のセクションでコードを実行したレスポンスからhash保存した に置き換えます。

このコードでは、ユーティリティ を使用します。ユーティリティ はdispatch-evm-rpc.js、AWS SDK から必要な Signature Version 4 (SigV4) モジュールを使用して AMB Access Polygon への HTTPS リクエストに署名し、広く使用されている HTTP クライアント AXIOS を使用してリクエストを送信します。

返されるレスポンスは次のようになります。

```
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
```

レスポンスはトランザクションの詳細を表します。これで、 `blockHash`と `blockNumber` が定義されている可能性が高いことに注意してください。これは、トランザクションがブロックに記録されたことを示します。これらの値が のままの場合は`null`、数分待ってからコードを再度実行し、トランザクションがブロックに含まれているかどうかを確認します。最後に、受信者のアドレス残高 (0x110d9316ec000) の 16 進表現は、Ethers の メソッドを使用して 10 進数に変換されます。このメソッドは、16 進数を 10 進数に変換し、18 進数を 18 (10^18) シフトして POL の真の残高を提供します。 `formatEther()`

> **Tip**
>
> 上記のコード例は、Node.js、Ethers、Axios を使用して AMB Access Polygon でサポートされている JSON-RPCs の一部を利用する方法を示していますが、このサービスを使用してサンプルを変更したり、Polygon でアプリケーションを構築するための他のコードを記述したりできます。AMB Access Polygon でサポートされている JSON-RPCs「」を参照してくださいAMB Access Polygon でサポートされている Managed Blockchain API と JSON-RPCs 。

AMB Access Polygon リクエストを作成するためのトークンベースのアクセス用の Accessor トークンの作成と管理

また、署名バージョン 4 (SigV4) 署名プロセスの便利な代替手段として、アクセサートークンを使用してポリゴンネットワークエンドポイントに JSON-RPC 呼び出しを行うこともできます。呼び出しでは、作成してパラメータとして追加する Accessor トークンの 1 つBILLING_TOKENから を指定する必要があります。

> **Important**
>
> - 利便性よりもセキュリティと監査可能性を優先する場合は、代わりに SigV4 署名プロセスを使用してください。
> - 署名バージョン 4 (SigV4) とトークンベースのアクセスを使用して、Polygon JSON-RPCs にアクセスできます。ただし、両方のプロトコルを使用することを選択した場合、リクエストは拒否されます。
> - ユーザー向けアプリケーションに Accessor トークンを埋め込まないでください。

コンソールのトークンアクセサーページには、クライアント上のコード AWS アカウント から から AMB Access Polygon JSON-RPC 呼び出しを行うために使用できるすべてのアクセサートークンのリストが表示されます。

AMB Access Polygon JSON-RPC リクエストの詳細については、「」を参照してくださいAMB Access Polygon でサポートされている Managed Blockchain API と JSON-RPCs 。

を使用して Accessor トークンを作成および管理できます AWS Management Console。
また、、、、および の API オペレーションを使用して `CreateAccessor`Accessor トークンを作成および管理することもできます `GetAccessorListAccessors` `DeleteAccessor`。BILLING_TOKEN はアクセサーのプロパティです。このBILLING_TOKENプロパティは、アクセサーを追跡し、 から行われた AMB Access Polygon JSON-RPC リクエストの請求に使用されます AWS アカウント。

Accessor トークンの作成と管理に関連するすべての API アクションは AWS Management Console、、 AWS CLI、 SDKsからも利用できます。

トークンベースのアクセス用のアクセサートークンの作成

Accessor トークンを作成し、それを使用して、 内の任意の AMB Access Polygon ノードで AMB Access Polygon API コールを行うことができます AWS アカウント。

を使用して AMB Access Polygon JSON-RPC リクエストを行うアクセサートークンを作成する AWS Management Console

1. https://console.aws.amazon.com/managedblockchain/ で Managed Blockchain コンソールを開きます。
2. トークンアクセサー を選択します。
3. 「アクセサーの作成」を選択します。
4. 有効なポリゴンブロックチェーンネットワーク を選択します。
5. オプションで、アクセサーのタグを追加します。
6. 「アクセサーの作成」を選択して、新しいアクセサートークンを作成します。

を使用して AMB Access Polygon JSON-RPC リクエストを行うアクセサートークンを作成する AWS CLI

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

前のコマンドは、次の例に示すようにBillingToken、 AccessorIdとともに を返します。

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

レスポンスのキー要素は ですBillingToken。このプロパティを使用して、AMB Access Polygon JSON-RPC 呼び出しを行うことができます。この例の一部の値はセキュリティ上の理由から難読化されていますが、実際のレスポンスでは完全に表示されます。

> **Note**
>
> オペレーションが実行されると、 Managed Blockchain はトークンをプロビジョニングして設定します。このプロセスの長さは、多くの変数によって異なります。

Accessor トークンの詳細の表示

AWS アカウント 所有する各 Accessor トークンのプロパティを表示できます。例えば、アクセサー ID またはアクセサーの Amazon リソースネーム (ARN) を表示できます。ステータス、タイプ、作成日、および を表示することもできますBillingToken。

を使用して Accessor トークンの情報を表示するには AWS Management Console

1. https://console.aws.amazon.com/managedblockchain/ で Managed Blockchain コンソールを開きます。
2. ナビゲーションペインで、トークンアクセサー を選択します。
3. リストからトークンのアクセサー ID を選択します。

ポップアップするトークンの詳細ページ。このページから、トークンのプロパティを表示できます。

を使用して Accessor トークンの情報を表示するには AWS CLI

次のコマンドを実行して、Accessor トークンの詳細を表示します。の値をアクセサー ID `--accessor-id`に置き換えます。

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

BillingToken およびその他のキープロパティは、次の例に示すように返されます。この例の一部の値は、セキュリティ上の理由から難読化されていますが、実際のレスポンスでは完全に表示されます。

```
{
  "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********"
  }
}
```

Accessor トークンの削除

Accessor トークンを削除すると、トークンは から PENDING_DELETIONステータスAVAILABLEに変わります。PENDING_DELETION ステータスでアクセサートークンを使用することはできません。

を使用して Accessor トークンを削除するには AWS Management Console

1. https://console.aws.amazon.com/managedblockchain/ で Managed Blockchain コンソールを開きます。
2. ナビゲーションペインで、トークンアクセサー を選択します。
3. リストから目的のアクセサートークンを選択します。
4. [削除] を選択します。
5. 選択内容を確認します。

削除した Accessor トークンを含む Tokens Accessors ページに戻ります。このページにPENDING_DELETIONステータスが表示されます。

を使用して Accessor トークンを削除するには AWS CLI

次の例は、トークンを削除する方法を示しています。delete-accessor コマンドを使用してトークンを削除します。の値をアクセサー ID --accessor-idで設定します。

AWS CLI を使用した Accessor トークンの削除

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

このコマンドが正常に実行されると、メッセージは返されません。

AMB Access Polygon でサポートされている Managed Blockchain API と JSON-RPCs

Amazon Managed Blockchain は、AMB Access Polygon のトークンアクセサーを作成および管理するための API オペレーションを提供します。詳細については、「 Managed Blockchain API リファレンスガイド」を参照してください。

次のトピックでは、AMB アクセスポリゴンがサポートするポリゴン JSON-RPCs のリストとリファレンスを示します。サポートされている各 JSON-RPC には、その使用に関する簡単な説明があります。Polygon JSON-RPCs を使用して、スマートコントラクトデータのクエリと取得、トランザクションの詳細の取得、トランザクションの送信、トランザクションのトレースの実行などのその他のユーティリティ、および料金の見積もりを行います。

AMB Access Polygon は、次の JSON-RPC メソッドをサポートしています。サポートされている各 JSON-RPC には、そのユーティリティとそのデフォルトのリクエストクォータに関するカテゴリと簡単な説明があります。Amazon Managed Blockchain で JSON-RPC メソッドを使用するための固有の考慮事項は、該当する場合に表示されます。

Note

- リストにないメソッドはサポートされていません。
- Amazon Managed Blockchain でポリゴン JSON-RPCs を呼び出す場合、署名バージョン 4 の署名プロセス を使用して認証された HTTPS 接続を介して呼び出すことができます。つまり、アカウント内の AWS 認可された IAM プリンシパルのみが、Polygon JSON-RPC 呼び出しを行うことができます。これを行うには、 AWS 認証情報 (アクセスキー ID とシークレットアクセスキー) を 呼び出しで提供する必要があります。
- 署名バージョン 4 (SigV4) 署名プロセスの便利な代替として、トークンベースのアクセスを使用することもできます。利便性よりもセキュリティと監査可能性を優先する場合は、代わりに SigV4 署名プロセスを使用してください。ただし、SigV4 とトークンベースのアクセスの両方を使用する場合、リクエストは機能しません。
- JSON-RPC バッチリクエストは、このプレビューでは Amazon Managed Blockchain (AMB) Access Polygon ではサポートされていません。
- 次の表のクォータ列には、各 JSON-RPC のクォータが一覧表示されます。クォータは、各 JSON-RPC のポリゴンネットワーク (メインネット) ごとに、リージョンごとに 1 秒あたりのリクエスト (RPS) で設定されます。

クォータを増やすには、 に連絡する必要があります AWS Support。に連絡するには AWS Support、 にサインインしますAWS Support Center Console。[ケースを作成] を選択します。[技術] を選択します。サービスとして Managed Blockchain を選択します。 アクセス：ポリゴンをカテゴリ として選択し、一般的なガイダンスを重要度 として選択します。RPC クォータをサブジェクトとして入力し、説明テキストボックスに JSON-RPC と、リージョン ごとのポリゴンネットワーク あたりの RPS のニーズに適用されるクォータ制限を一覧表示します。ケースを送信します。

カテゴリ	JSON-RPC	説明	考慮事項
イーサリアム	eth_blockNumber	最新のブロックの数を返します。	
	eth_call	ブロックチェーンでトランザクションを作成せずに、新しいメッセージ呼び出しをすぐに実行します。	`eth_call` は 0 個のガスを消費しますが、それを必要とするメッセージのガスパラメータがあります。
	eth_chainId	EIP-155 で導入された現在設定されている`Chain Id`値の整数値を返します。が使用`None`できない場合は`Chain Id`、を返します。	
	eth_estimateGas	トランザクションをブロックチェーンに追加せずに、トランザクション	

カテゴリ	JSON-RPC	説明	考慮事項
		に必要なガスを推定して返します。	
	eth_feeHistory	過去のガス情報のコレクションを返します。	
	eth_gasPrice	Wei でガスあたりの現在の価格を返します。	
	eth_getBalance	指定されたアカウントアドレスとブロック識別子のアカウントの残高を返します。	
	eth_get BlockByハッシュ	ブロックハッシュを使用して指定されたブロックに関する情報を返します。	
	eth_get BlockBy番号	ブロック番号を使用して指定されたブロックに関する情報を返します。	
	eth_getBlockReceipts	ブロック番号を使用して指定されたブロックに関する受信を返します。	

カテゴリ	JSON-RPC	説明	考慮事項
	eth_get BlockTransactionCountByハッシュ	ブロックハッシュを使用して指定されたブロック内のトランザクションの数を返します。	
	eth_get BlockTransactionCountBy番号	ブロック番号を使用して指定されたブロック内のトランザクションの数を返します。	
	eth_getCode	指定されたアカウントアドレスとブロック識別子のコードを返します。	

カテゴリ	JSON-RPC	説明	考慮事項
	eth_getLogs	指定されたフィルターオブジェクトのすべてのログの配列を返します。	契約アドレスが指定されている場合、デフォルトで 1K ブロック範囲の任意のブロック範囲に対してeth_getlogs リクエストを行うことができます。アクティビティの高い契約は、より小さなブロック範囲に制限される場合があります。契約住所が指定されていない場合、ブロック範囲は 8 になります。
	eth_getRawTransactionByHash	で指定されたトランザクションの raw 形式を返しますtransaction_hash 。	
	eth_getStorageAt	指定されたアカウントアドレスとブロック識別子の指定されたストレージ位置の値を返します。	

カテゴリ	JSON-RPC	説明	考慮事項
	eth_getTransactionByBlockHashAndIndex	指定されたブロックハッシュとトランザクションインデックスの位置を使用して、トランザクションに関する情報を返します。	
	eth_getTransactionByBlockNumberAndIndex	指定されたブロック番号とトランザクションインデックスの位置を使用して、トランザクションに関する情報を返します。	
	eth_get TransactionByハッシュ	指定されたトランザクションハッシュを持つトランザクションに関する情報を返します。	
	eth_getTransactionCount	指定されたアドレスとブロック識別子から送信されたトランザクションの数を返します。	

カテゴリ	JSON-RPC	説明	考慮事項
	eth_getTransactionReceipt	指定されたトランザクションハッシュを使用してトランザクションの受信を返します。	
	eth_getUncleByBlockHashAndIndex	ブロックハッシュと Uncle インデックスの位置を使用して指定された Uncle ブロックに関する情報を返します。	
	eth_getUncleByBlockNumberAndIndex	ブロック番号と Uncle インデックスの位置を使用して指定された Uncle ブロックに関する情報を返します。	
	eth_get UncleCountByBlockハッシュ	Uncle ハッシュを使用して指定された Uncle のカウント数を返します。	
	eth_get UncleCountByBlock番号	Uncle 番号を使用して指定された Uncle のカウント数を返します。	

カテゴリ	JSON-RPC	説明	考慮事項
	eth_maxPriorityFeePerGas	現在のブロックに含まれるトランザクションを取得するために、優先順位料金として支払うことができる金額の見積もりであるガスあたりの料金、または「ヒント」を返します。	通常、このメソッドから返される値を使用して、送信する後続のトランザクションmaxFeePerGas で を設定します。
	eth_protocolVersion	現在の Ethereum プロトコルバージョンを返します。	
	eth_sendRawTransaction	新しいメッセージコールトランザクションまたは署名付きトランザクションの契約作成を作成します。	Managed Blockchain は raw トランザクションのみをサポートします。トランザクションを送信する前に、トランザクションを作成して署名する必要があります。

カテゴリ	JSON-RPC	説明	考慮事項
デバッグ	debug_trace BlockByハッシュ	トレーサーを使用してブロックハッシュで指定されたブロック内のすべてのトランザクションを実行することで、可能なトレース結果番号を返します (トレースモードが必要)。	
	debug_trace BlockBy番号	トレーサーで数値で指定されたブロック内のすべてのトランザクションを実行して、トレース結果を返します (トレースモードが必要)。	
	debug_traceCall	特定のブロック実行 (トレースモードが必要) のコンテキスト内で eth 呼び出しを実行して、可能なトレース結果の数を返します。	
	debug_traceTransaction	特定のトランザクションのすべてのトレースを返します (トレースモードが必要)。	

カテゴリ	JSON-RPC	説明	考慮事項
正味	net_version	現在のネットワーク ID を返します。	
トレース	trace_block	ブロックに含まれていたすべてのトランザクションの呼び出されたすべてのオプコードの完全なスタックトレースを返します。	
	trace_call	特定のブロック実行 (トレースモードが必要) のコンテキスト内で eth 呼び出しを実行して、可能なトレース結果の数を返します。	
	trace_transaction	特定のトランザクションのすべてのトレースを返します (トレースモードが必要)。	
送信プール	txpool_content	保留中およびキューに入っているすべてのトランザクションを返します。	

カテゴリ	JSON-RPC	説明	考慮事項
	txpool_status	次のブロックに現在含まれているすべてのトランザクションと、キューに入っているトランザクションの数 (将来の実行のみにスケジュールされます) を提供します。	
Web	web3_clientVersion	現在のクライアントバージョンを返します。	

Amazon Managed Blockchain (AMB) Access Polygon を使用したポリゴンのユースケース

Polygon ブロックチェーンは、NFT 、Web3 ゲーム、トークン化のユースケースなどに関連する分散アプリケーション (dApps) の構築によく使用されます。 NFTs このトピックでは、Amazon Managed Blockchain (AMB) アクセスポリゴンを使用して実装できるいくつかのユースケースのリストを示します。

トピック

ポリゴン NFT データを分析する

指定した期間の転送イベントや NFTsメタデータなどの情報を含む、Polygon NFT に関するデータを収集できます。その後、このデータを分析して、どの NFTs がトレンドであるか、どのユーザーが特定のコレクションと最も頻繁にやり取りしているかなどのインサイトを引き出すことができます。

詳細については、「AMB Access Polygon でサポートされている Managed Blockchain API と JSON-RPCs 」を参照してください。

NFT 購入のサポート

AMB Access Polygon を使用して、初期ミント、許可リスト、またはセカンダリマーケットで NFT 購入のトランザクションを送信できます。他の AWS サービスを組み合わせて使用することで、クレジットカードを使用した購入を許可し、Fiat または暗号通貨を受け入れ、関係するすべての利害関係者に迅速に解決できます。

詳細については、「AMB Access Polygon でサポートされている Managed Blockchain API と JSON-RPCs 」を参照してください。

ポリゴンウォレットを作成する

AMB Access Polygon を使用して、ブロックチェーン上のスマートコントラクトからのユーザートークンバランスの読み取りや、署名付きトランザクションのブロックチェーンへのブロードキャストなど、デジタルアセットウォレットの重要な機能を提供できます。

詳細については、「AMB Access Polygon でサポートされている Managed Blockchain API と JSON-RPCs 」を参照してください。

サービスとしてのウォレット

AMB Access Polygon を使用すると、サポートされている Polygon JSON-RPCs を使用して、残高、アセット転送、アセット送信、料金の見積もりの確認などの一般的なウォレットトランザクションをサポートする wallet-as-a-service ために必要な運用を開発できます。

詳細については、「AMB Access Polygon でサポートされている Managed Blockchain API と JSON-RPCs 」を参照してください。

トークンゲートエクスペリエンス

AMB Access Polygon を使用して、ユーザーのトークンゲートエクスペリエンスを構築できます。例えば、特定の NFT の所有者にのみ、条件付きでコンテンツへのアクセスを提供できます。これを実現するには、ブロックチェーンを読んで、ユーザーの住所の NFT 所有権を決定する必要があります。

詳細については、「AMB Access Polygon でサポートされている Managed Blockchain API と JSON-RPCs 」を参照してください。

Amazon Managed Blockchain (AMB) アクセスポリゴンのチュートリアル

このセクションで強調表示されている以下のチュートリアルは、AMB Access Polygon を使用して Polygon ブロックチェーンで一般的なタスクを実行する方法を学ぶのに役立つチュートリアルを提供する の AWS re:Post コミュニティ記事です。

- AMB Access Polygon と web3.js を使用したトランザクションの送信
- AMB Access Polygon と Hardhat Ignition を使用してスマートコントラクトをデプロイする
- スマートコントラクトの操作
- AMB Access Polygon と Chainlink データフィードを使用して現在の価格データをチェーン外から取得する
- AMB アクセスを使用してポリゴンメインネットで ERC-20 トークンデータを分析

Amazon Managed Blockchain (AMB) Access Polygon のセキュリティ

のクラウドセキュリティが最優先事項 AWS です。お客様は AWS 、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャからメリットを得られます。

セキュリティは、 AWS とユーザーの間で共有される責任です。責任共有モデルでは、これをクラウドのセキュリティとクラウド内のセキュリティの両方として説明しています。

- クラウドのセキュリティ — AWS は、 で AWS サービスを実行するインフラストラクチャを保護する責任を担います AWS クラウド。また、 は、お客様が安全に使用できるサービス AWS も提供します。サードパーティーの監査人は、AWS コンプライアンスプログラム の一環として、セキュリティの有効性を定期的にテストおよび検証します。Amazon Managed Blockchain (AMB) アクセスポリゴンに適用されるコンプライアンスプログラムの詳細については、「コンプライアンスAWS プログラムによる対象範囲内のサービス」を参照してください。
- クラウド内のセキュリティ — お客様の責任は、使用する AWS サービスによって決まります。また、お客様は、 お客様のデータの機密性、企業の要件、および適用可能な法律や規制といった他の要因 についても責任を担います。

データ保護、認証、アクセス制御を提供するために、Amazon Managed Blockchain は AWS Managed Blockchain で実行されているオープンソースフレームワークの機能を使用します。

このドキュメントは、AMB Access Polygon を使用する際の責任共有モデルの適用方法を理解するのに役立ちます。以下のトピックでは、セキュリティおよびコンプライアンスの目的を達成するために AMB Access Polygon を設定する方法を示します。また、AMB Access Polygon リソースのモニタリングや保護に役立つ他の AWS のサービスの使用方法についても説明します。

トピック

- Amazon Managed Blockchain (AMB) Access Polygon でのデータ保護
- Amazon Managed Blockchain (AMB) Access Polygon の Identity and Access Management

Amazon Managed Blockchain (AMB) Access Polygon でのデータ保護

責任 AWS 共有モデル、Amazon Managed Blockchain (AMB) Access Polygon でのデータ保護に適用されます。このモデルで説明されているように、 AWS はすべての を実行するグローバルインフラストラクチャを保護する責任があります AWS クラウド。お客様は、このインフラストラクチャでホストされているコンテンツに対する管理を維持する責任があります。また、使用する AWS のサービス のセキュリティ設定と管理タスクもユーザーの責任となります。データプライバシーの詳細については、「データプライバシーのよくある質問」を参照してください。欧州でのデータ保護の詳細については、AWS セキュリティブログに投稿された「AWS 責任共有モデルおよび GDPR」のブログ記事を参照してください。

データ保護の目的で、認証情報を保護し AWS アカウント 、 AWS IAM Identity Center または AWS Identity and Access Management (IAM) を使用して個々のユーザーを設定することをお勧めします。この方法により、それぞれのジョブを遂行するために必要な権限のみが各ユーザーに付与されます。また、次の方法でデータを保護することもお勧めします:

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して AWS リソースと通信します。TLS 1.2 は必須であり TLS 1.3 がお勧めです。
- で API とユーザーアクティビティのログ記録を設定します AWS CloudTrail。
- AWS 暗号化ソリューションと、 内のすべてのデフォルトのセキュリティコントロールを使用します AWS のサービス。
- Amazon Macie などの高度なマネージドセキュリティサービスを使用します。これらは、Amazon S3 に保存されている機密データの検出と保護を支援します。
- コマンドラインインターフェイスまたは API AWS を介して にアクセスするときに FIPS 140-2 検証済みの暗号化モジュールが必要な場合は、FIPS エンドポイントを使用します。利用可能な FIPS エンドポイントの詳細については、「連邦情報処理規格 (FIPS) 140-2」を参照してください。

お客様の E メールアドレスなどの極秘または機密情報は、タグ、または名前フィールドなどの自由形式のテキストフィールドに配置しないことを強くお勧めします。これは、コンソール、API、または AWS CLI SDK を使用して AMB Access Polygon または他の AWS のサービス を使用する場合も同様です。 AWS SDKs 名前に使用する自由記述のテキストフィールドやタグに入力したデータは、課金や診断ログに使用される場合があります。外部サーバーへの URL を提供する場合は、そのサーバーへのリクエストを検証するための認証情報を URL に含めないように強くお勧めします。

データ暗号化

データ暗号化は、権限のないユーザーがブロックチェーンネットワークおよび関連するデータストレージシステムからデータを読み取るのを防ぐのに役立ちます。これには、ネットワークを移動するときに傍受される可能性のあるデータが含まれます。これは、転送中のデータと呼ばれます。

転送中の暗号化

デフォルトでは、 Managed Blockchain は HTTPS/TLS 接続を使用して、 を実行するクライアントコンピュータから AWS サービスエンドポイントに送信されるすべてのデータを暗号化します AWS CLI 。

HTTPS/TLS の使用を有効にするために必要な操作はありません。`--no-verify-ssl` コマンドを使用して個々の AWS CLI コマンドに対して明示的に無効にしない限り、常に有効になります。

Amazon Managed Blockchain (AMB) Access Polygon の Identity and Access Management

AWS Identity and Access Management (IAM) は、管理者が AWS リソースへのアクセスを安全に制御 AWS のサービス するのに役立つ です。IAM 管理者は、誰を認証 (サインイン) し、誰に AMB Access Polygon リソースの使用を承認する (アクセス許可を付与する) かを制御します。IAM は、追加料金なしで AWS のサービス 使用できる です。

トピック

対象者

AWS Identity and Access Management (IAM) の使用方法は、AMB Access Polygon で行う作業によって異なります。

サービスユーザー – ジョブを実行するために AMB Access Polygon サービスを使用する場合、管理者から必要な認証情報とアクセス許可が与えられます。さらに多くの AMB Access Polygon 機能を使用して作業を行う場合は、追加のアクセス許可が必要になることがあります。アクセスの管理方法を理解しておくと、管理者に適切な許可をリクエストするうえで役立ちます。AMB アクセスポリゴンの機能にアクセスできない場合は、「」を参照してくださいAmazon Managed Blockchain (AMB) アクセスポリゴンアイデンティティとアクセスのトラブルシューティング。

サービス管理者 – 社内の AMB Access Polygon リソースを担当している場合は、通常、AMB Access Polygon へのフルアクセスがあります。サービスユーザーがどの AMB Access Polygon 機能やリソースにアクセスするかを決めるのは管理者の仕事です。その後、IAM 管理者にリクエストを送信して、サービスユーザーの権限を変更する必要があります。このページの情報を点検して、IAM の基本概念を理解してください。会社で AMB Access Polygon で IAM を使用する方法の詳細については、「」を参照してくださいAmazon Managed Blockchain (AMB) Access Polygon と IAM の連携。

IAM 管理者 – IAM 管理者は、AMB Access Polygon へのアクセスを管理するポリシーの作成方法の詳細について確認する場合があります。IAM で使用できる AMB Access Polygon アイデンティティベースのポリシーの例を表示するには、「」を参照してくださいAmazon Managed Blockchain (AMB) Access Polygon のアイデンティティベースのポリシーの例。

アイデンティティを使用した認証

認証とは、ID 認証情報 AWS を使用して にサインインする方法です。として、IAM ユーザーとして AWS アカウントのルートユーザー、または IAM ロールを引き受けて認証（ にサインイン AWS) される必要があります。

ID ソースを介して提供された認証情報を使用して、フェデレーティッド ID AWS として にサインインできます。 AWS IAM Identity Center（ IAM Identity Center) ユーザー、会社のシングルサインオン認証、Google または Facebook の認証情報は、フェデレーティッド ID の例です。フェデレーティッド ID としてサインインする場合、IAM ロールを使用して、前もって管理者により ID フェデレーションが設定されています。フェデレーション AWS を使用して にアクセスすると、間接的にロールを引き受けることになります。

ユーザーのタイプに応じて、 AWS Management Console または AWS アクセスポータルにサインインできます。へのサインインの詳細については AWS、「 ユーザーガイド」の「 へのサインイン AWS アカウント方法AWS サインイン 」を参照してください。

AWS プログラムで にアクセスする場合、 は Software Development Kit (SDK) とコマンドラインインターフェイス (CLI) AWS を提供し、認証情報を使用してリクエストに暗号で署名します。 AWS ツールを使用しない場合は、リクエストに自分で署名する必要があります。推奨される方法を使用

してリクエストを自分で署名する方法の詳細については、IAM ユーザーガイドの API AWS リクエストの署名を参照してください。

使用する認証方法を問わず、追加セキュリティ情報の提供をリクエストされる場合もあります。例えば、 AWS では、多要素認証 (MFA) を使用してアカウントのセキュリティを向上させることをお勧めします。詳細については、「AWS IAM Identity Center ユーザーガイド」の「Multi-factor authentication」(多要素認証) および「IAM ユーザーガイド」の「AWSでの多要素認証 (MFA) の使用」を参照してください。

AWS アカウント ルートユーザー

を作成するときは AWS アカウント、アカウント内のすべての およびリソースへの AWS のサービス 完全なアクセス権を持つ 1 つのサインインアイデンティティから始めます。この ID は AWS アカウント ルートユーザーと呼ばれ、アカウントの作成に使用した E メールアドレスとパスワードでサインインすることでアクセスできます。日常的なタスクには、ルートユーザーを使用しないことを強くお勧めします。ルートユーザーの認証情報は保護し、ルートユーザーでしか実行できないタスクを実行するときに使用します。ルートユーザーとしてサインインする必要があるタスクの完全なリストについては、IAM ユーザーガイドのルートユーザー認証情報が必要なタスクを参照してください。

フェデレーティッドアイデンティティ

ベストプラクティスとして、管理者アクセスを必要とするユーザーを含む人間のユーザーに、一時的な認証情報を使用して にアクセスするための ID プロバイダーとのフェデレーションの使用を要求 AWS のサービス します。

フェデレーティッド ID は、エンタープライズユーザーディレクトリ、ウェブ ID プロバイダー、AWS Directory Service、アイデンティティセンターディレクトリのユーザー、または ID ソースを通じて提供された認証情報 AWS のサービス を使用して にアクセスするユーザーです。フェデレーティッド ID が にアクセスすると AWS アカウント、ロールを引き受け、ロールは一時的な認証情報を提供します。

アクセスを一元管理する場合は、 AWS IAM Identity Centerを使用することをお勧めします。IAM Identity Center でユーザーとグループを作成することも、独自の ID ソース内のユーザーとグループのセットに接続して同期して、すべての AWS アカウント とアプリケーションで使用できるようにすることもできます。IAM Identity Center の詳細については、「AWS IAM Identity Center ユーザーガイド」の「What is IAM Identity Center?」(IAM Identity Center とは) を参照してください。

IAM ユーザーとグループ

IAM ユーザーは、単一のユーザーまたはアプリケーションに対して特定のアクセス許可 AWS アカウント を持つ 内のアイデンティティです。可能であれば、パスワードやアクセスキーなどの長期的な認証情報を保有する IAM ユーザーを作成する代わりに、一時的な認証情報を使用することをお勧めします。ただし、IAM ユーザーでの長期的な認証情報が必要な特定のユースケースがある場合は、アクセスキーをローテーションすることをお勧めします。詳細については、IAM ユーザーガイドの長期的な認証情報を必要とするユースケースのためにアクセスキーを定期的にローテーションするを参照してください。

IAM グループは、IAM ユーザーの集団を指定するアイデンティティです。グループとしてサインインすることはできません。グループを使用して、複数のユーザーに対して一度に権限を指定できます。多数のユーザーグループがある場合、グループを使用することで権限の管理が容易になります。例えば、IAMAdminsという名前のグループを設定して、そのグループにIAM リソースを管理する許可を与えることができます。

ユーザーは、ロールとは異なります。ユーザーは 1 人の人または 1 つのアプリケーションに一意に関連付けられますが、ロールはそれを必要とする任意の人が引き受けるようになっています。ユーザーには永続的な長期の認証情報がありますが、ロールでは一時的な認証情報が提供されます。詳細については、「IAM ユーザーガイド」の「IAM ユーザー (ロールではなく) の作成が適している場合」を参照してください。

IAM ロール

IAM ロールは、特定のアクセス許可 AWS アカウント を持つ 内のアイデンティティです。これは IAM ユーザーに似ていますが、特定のユーザーには関連付けられていません。ロール を切り替える AWS Management Console ことで、 で IAM ロールを一時的に引き受けることができます。ロールを引き受けるには、 または AWS API AWS CLI オペレーションを呼び出すか、カスタム URL を使用します。ロールを使用する方法の詳細については、「IAM ユーザーガイド」の「IAM ロールの使用」を参照してください。

IAM ロールと一時的な認証情報は、次の状況で役立ちます:

- フェデレーションユーザーアクセス – フェデレーティッド ID に許可を割り当てるには、ロールを作成してそのロールの許可を定義します。フェデレーティッド ID が認証されると、その ID はロールに関連付けられ、ロールで定義されている許可が付与されます。フェデレーションの詳細については、「IAM ユーザーガイド」の「Creating a role for a third-party Identity Provider」(サードパーティーアイデンティティプロバイダー向けロールの作成) を参照してください。IAM Identity Center を使用する場合は、許可セットを設定します。アイデンティティが認証後にアクセスでき

るものを制御するため、IAM Identity Center は、権限セットを IAM のロールに関連付けます。アクセス許可セットの詳細については、「AWS IAM Identity Center ユーザーガイド」の「アクセス許可セット」を参照してください。

- 一時的な IAM ユーザー権限 - IAM ユーザーまたはロールは、特定のタスクに対して複数の異なる権限を一時的に IAM ロールで引き受けることができます。
- クロスアカウントアクセス - IAM ロールを使用して、自分のアカウントのリソースにアクセスすることを、別のアカウントの人物 (信頼済みプリンシパル) に許可できます。クロスアカウントアクセス権を付与する主な方法は、ロールを使用することです。ただし、一部の では AWS のサービス、(ロールをプロキシとして使用する代わりに) ポリシーをリソースに直接アタッチできます。クロスアカウントアクセスにおけるロールとリソースベースのポリシーの違いについては、「IAM ユーザーガイド」の「IAM でのクロスアカウントのリソースへのアクセス」を参照してください。
- クロスサービスアクセス — 一部の は、他の の機能 AWS のサービス を使用します AWS のサービス。例えば、あるサービスで呼び出しを行うと、通常そのサービスによって Amazon EC2 でアプリケーションが実行されたり、Amazon S3 にオブジェクトが保存されたりします。サービスでは、呼び出し元プリンシパルの許可、サービスロール、またはサービスリンクロールを使用してこれを行う場合があります。
  - 転送アクセスセッション (FAS) – IAM ユーザーまたはロールを使用して でアクションを実行する場合 AWS、ユーザーはプリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、 を呼び出すプリンシパルのアクセス許可を AWS のサービス、ダウンストリーム サービス AWS のサービス へのリクエストのリクエストと組み合わせて使用します。FAS リクエストは、サービスが他の AWS のサービス またはリソースとのやり取りを完了する必要があるリクエストを受け取った場合にのみ行われます。この場合、両方のアクションを実行するためのアクセス許可が必要です。FAS リクエストを行う際のポリシーの詳細については、「転送アクセスセッション」を参照してください。
  - サービスロール - サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、「IAM ユーザーガイド」の「AWS のサービスにアクセス許可を委任するロールの作成」を参照してください。
  - サービスにリンクされたロール – サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント 、サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールのアクセス許可を表示できますが、編集することはできません。

- Amazon EC2 で実行されているアプリケーション – IAM ロールを使用して、EC2 インスタンスで実行され、 AWS CLI または AWS API リクエストを行うアプリケーションの一時的な認証情報を管理できます。これは、EC2 インスタンス内でのアクセスキーの保存に推奨されます。 AWS ロールを EC2 インスタンスに割り当て、そのすべてのアプリケーションで使用できるようにするには、インスタンスにアタッチされたインスタンスプロファイルを作成します。インスタンスプロファイルにはロールが含まれ、EC2 インスタンスで実行されるプログラムは一時的な認証情報を取得できます。詳細については、IAM ユーザーガイドのAmazon EC2 インスタンスで実行されるアプリケーションに IAM ロールを使用して許可を付与するを参照してください。

IAM ロールと IAM ユーザーのどちらを使用するかについては、IAM ユーザーガイドの(IAM ユーザーではなく) IAM ロールをいつ作成したら良いのか?を参照してください。

ポリシーを使用したアクセスの管理

でアクセスを制御する AWS には、ポリシーを作成し、 AWS ID またはリソースにアタッチします。ポリシーは、アイデンティティまたはリソースに関連付けられているときにアクセス許可を定義する のオブジェクトです。 は、プリンシパル (ユーザー、ルートユーザー、またはロールセッション) AWS がリクエストを行うときに、これらのポリシー AWS を評価します。ポリシーでの権限により、リクエストが許可されるか拒否されるかが決まります。ほとんどのポリシーは JSON ドキュメント AWS として に保存されます。JSON ポリシードキュメントの構造と内容の詳細については、IAM ユーザーガイドのJSON ポリシー概要を参照してください。

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどんなリソースにどんな条件でアクションを実行できるかということです。

デフォルトでは、ユーザーやロールに権限はありません。IAM 管理者は、リソースで必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き継ぐことができます。

IAM ポリシーは、オペレーションの実行方法を問わず、アクションの許可を定義します。例えば、`iam:GetRole`アクションを許可するポリシーがあるとします。そのポリシーを持つユーザーは、 AWS Management Console、、 AWS CLIまたは AWS API からロール情報を取得できます。

アイデンティティベースのポリシー

アイデンティティベースポリシーは、IAM ユーザーグループ、ユーザーのグループ、ロールなど、アイデンティティにアタッチできる JSON 許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデン

ティティベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「IAM ポリシーの作成」を参照してください。

アイデンティティベースのポリシーは、さらにインラインポリシーまたはマネージドポリシーに分類できます。インラインポリシーは、単一のユーザー、グループ、またはロールに直接埋め込まれています。管理ポリシーは、 内の複数のユーザー、グループ、ロールにアタッチできるスタンドアロンポリシーです AWS アカウント。管理ポリシーには、 AWS 管理ポリシーとカスタマー管理ポリシーが含まれます。マネージドポリシーまたはインラインポリシーのいずれかを選択する方法については、IAM ユーザーガイドのマネージドポリシーとインラインポリシーの比較を参照してください。

リソースベースのポリシー

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシー や Amazon S3 バケットポリシー があげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、プリンシパルを指定する必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、または を含めることができます AWS のサービス。

リソースベースのポリシーは、そのサービス内にあるインラインポリシーです。リソースベースのポリシーでは、IAM の AWS マネージドポリシーを使用できません。

アクセスコントロールリスト (ACL)

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための許可を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

Amazon S3、 AWS WAF、および Amazon VPC は、ACLs。ACL の詳細については、Amazon Simple Storage Service デベロッパーガイドのアクセスコントロールリスト (ACL) の概要を参照してください。

その他のポリシータイプ

AWS は、一般的ではない追加のポリシータイプをサポートします。これらのポリシータイプでは、より一般的なポリシータイプで付与された最大の権限を設定できます。

- アクセス許可の境界 - アクセス許可の境界は、アイデンティティベースのポリシーによって IAM エンティティ (IAM ユーザーまたはロール) に付与できる権限の上限を設定する高度な機能です。

エンティティにアクセス許可の境界を設定できます。結果として得られる権限は、エンティティのアイデンティティベースポリシーとそのアクセス許可の境界の共通部分になります。`Principal` フィールドでユーザーまたはロールを指定するリソースベースのポリシーでは、アクセス許可の境界は制限されません。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。アクセス許可の境界の詳細については、IAM ユーザーガイドのIAM エンティティのアクセス許可の境界を参照してください。

- サービスコントロールポリシー (SCPs) – SCPs は、 の組織または組織単位 (OU) に対する最大アクセス許可を指定する JSON ポリシーです AWS Organizations。 AWS Organizations は、 AWS アカウント ビジネスが所有する複数の をグループ化して一元管理するサービスです。組織内のすべての機能を有効にすると、サービスコントロールポリシー (SCP) を一部またはすべてのアカウントに適用できます。SCP は、各 を含むメンバーアカウントのエンティティのアクセス許可を制限します AWS アカウントのルートユーザー。Organizations と SCP の詳細については、AWS Organizations ユーザーガイドの「SCP の仕組み」を参照してください。
- セッションポリシー - セッションポリシーは、ロールまたはフェデレーションユーザーの一時的なセッションをプログラムで作成する際にパラメータとして渡す高度なポリシーです。結果としてセッションの権限は、ユーザーまたはロールのアイデンティティベースポリシーとセッションポリシーの共通部分になります。また、リソースベースのポリシーから権限が派生する場合もあります。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。詳細については、IAM ユーザーガイドのセッションポリシーを参照してください。

複数のポリシータイプ

1 つのリクエストに複数のタイプのポリシーが適用されると、結果として作成される権限を理解するのがさらに難しくなります。複数のポリシータイプが関与する場合にリクエストを許可するかどうか AWS を決定する方法については、IAM ユーザーガイドの「ポリシー評価ロジック」を参照してください。

Amazon Managed Blockchain (AMB) Access Polygon と IAM の連携

IAM を使用して AMB Access Polygon へのアクセスを管理する前に、AMB Access Polygon で使用できる IAM 機能について学びます。

Amazon Managed Blockchain (AMB) Access Polygon で使用できる IAM の機能

IAM 機能	AMB アクセスポリゴンのサポート
アイデンティティベースのポリシー	あり
リソースベースのポリシー	なし
ポリシーアクション	あり
ポリシーリソース	いいえ
ポリシー条件キー	いいえ
ACL	なし
ABAC (ポリシー内のタグ)	いいえ
一時的な認証情報	いいえ
プリンシパル権限	いいえ
サービスロール	いいえ
サービスリンクロール	なし

AMB Access Polygon およびその他の がほとんどの IAM 機能と AWS のサービス 連携する方法の概要を把握するには、「IAM ユーザーガイド」のAWS 「IAM と連携する のサービス」を参照してください。

AMB Access Polygon のアイデンティティベースのポリシー

アイデンティティベースのポリシーのサポート: はい

アイデンティティベースポリシーは、IAM ユーザーグループ、ユーザーのグループ、ロールなど、アイデンティティにアタッチできる JSON 許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティベースのポリシーを作成する方法については、IAM ユーザーガイドのIAM ポリシーの作成を参照してください。

IAM アイデンティティベースのポリシーでは、許可または拒否するアクションとリソース、およびアクションを許可または拒否する条件を指定できます。プリンシパルは、それが添付されているユーザーまたはロールに適用されるため、アイデンティティベースのポリシーでは指定できません。JSON ポリシーで使用できるすべての要素について学ぶには、IAM ユーザーガイドのIAM JSON ポリシーの要素のリファレンスを参照してください。

AMB Access Polygon のアイデンティティベースのポリシーの例

AMB Access Polygon アイデンティティベースのポリシーの例を表示するには、「」を参照してくださいAmazon Managed Blockchain (AMB) Access Polygon のアイデンティティベースのポリシーの例。

AMB Access Polygon 内のリソースベースのポリシー

リソースベースのポリシーのサポート: なし

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシー や Amazon S3 バケットポリシー があげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、プリンシパルを指定する必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、または を含めることができます AWS のサービス。

クロスアカウントアクセスを有効にするには、アカウント全体、または別のアカウントの IAM エンティティをリソースベースのポリシーのプリンシパルとして指定します。リソースベースのポリシーにクロスアカウントのプリンシパルを追加しても、信頼関係は半分しか確立されない点に注意してください。プリンシパルとリソースが異なる にある場合 AWS アカウント、信頼されたアカウントの IAM 管理者は、プリンシパルエンティティ (ユーザーまたはロール) にリソースへのアクセス許可も付与する必要があります。IAM 管理者は、アイデンティティベースのポリシーをエンティティにアタッチすることで権限を付与します。ただし、リソースベースのポリシーで、同じアカウントのプリンシパルへのアクセス権が付与されている場合は、アイデンティティベースのポリシーをさらに付与する必要はありません。詳細については、「IAM ユーザーガイド」の「IAM でのクロスアカウントリソースアクセス」を参照してください。

AMB アクセスポリゴンのポリシーアクション

ポリシーアクションに対するサポート: はい

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどんなリソースにどんな条件でアクションを実行できるかということです。

JSON ポリシーのAction要素には、ポリシー内のアクセスを許可または拒否するために使用できるアクションが記述されます。ポリシーアクションの名前は通常、関連付けられた AWS API オペレーションと同じです。一致する API オペレーションのない許可のみのアクションなど、いくつかの例外があります。また、ポリシーに複数のアクションが必要なオペレーションもあります。これらの追加アクションは、依存アクションと呼ばれます。

このアクションは、関連付けられたオペレーションを実行するための権限を付与するポリシーで使用されます。

AMB Access Polygon アクションのリストを確認するには、「サービス認証リファレンス」の「Amazon Managed Blockchain (AMB) Access Polygon で定義されるアクション」を参照してください。

AMB Access Polygon のポリシーアクションは、アクションの前に次のプレフィックスを使用します。

```
managedblockchain:
```

単一のステートメントで複数のアクションを指定するには、アクションをカンマで区切ります。

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

ワイルドカード (*) を使用して複数アクションを指定できます。例えば、InvokeRpcPolygon という単語で始まるすべてのアクションを指定するには、次のアクションを含めます。

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

AMB Access Polygon アイデンティティベースのポリシーの例を表示するには、「」を参照してくださいAmazon Managed Blockchain (AMB) Access Polygon のアイデンティティベースのポリシーの例。

AMB アクセスポリゴンのポリシーリソース

ポリシーリソースのサポート： いいえ

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースにどのような条件でアクションを実行できるかということです。

`Resource` JSON ポリシー要素は、アクションが適用されるオブジェクトを指定します。ステートメントには、`Resource` または `NotResource` 要素を含める必要があります。ベストプラクティスとして、Amazon リソースネーム (ARN) を使用してリソースを指定します。これは、リソースレベルの許可と呼ばれる特定のリソースタイプをサポートするアクションに対して実行できます。

オペレーションのリスト化など、リソースレベルの権限をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (*) を使用します。

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

AMB Access Polygon リソースタイプとその ARNs」の「Amazon Managed Blockchain (AMB) Access Polygon で定義されるリソース」を参照してください。 各リソースの ARN を指定できるアクションについては、「Amazon Managed Blockchain (AMB) Access Polygon で定義されるアクション」を参照してください。

AMB Access Polygon アイデンティティベースのポリシーの例を表示するには、「」を参照してくださいAmazon Managed Blockchain (AMB) Access Polygon のアイデンティティベースのポリシーの例。

AMB Access Polygon のポリシー条件キー

サービス固有のポリシー条件キーをサポート： いいえ

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどんなリソースにどんな条件でアクションを実行できるかということです。

`Condition` 要素 (または `Condition` ブロック) を使用すると、ステートメントが有効な条件を指定できます。`Condition` 要素はオプションです。イコールや未満などの 条件演算子 を使用して条件式を作成することで、ポリシーの条件とリクエスト内の値を一致させることができます。

1 つのステートメントに複数の Condition 要素を指定する場合、または 1 つの Condition 要素に複数のキーを指定する場合、 AWS では AND 論理演算子を使用してそれらを評価します。1 つの条件キーに複数の値を指定すると、 は論理ORオペレーションを使用して条件 AWS を評価します。ステートメントの権限が付与される前にすべての条件が満たされる必要があります。

条件を指定する際にプレースホルダー変数も使用できます。例えば IAM ユーザーに、IAM ユーザー名がタグ付けされている場合のみリソースにアクセスできる権限を付与することができます。詳細については、IAM ユーザーガイドのIAM ポリシーの要素: 変数およびタグを参照してください。

AWS は、グローバル条件キーとサービス固有の条件キーをサポートします。すべての AWS グローバル条件キーを確認するには、「IAM ユーザーガイド」のAWS 「 グローバル条件コンテキストキー」を参照してください。

AMB アクセスポリゴンの条件キーのリストを確認するには、「サービス認証リファレンス」の「Amazon Managed Blockchain (AMB) アクセスポリゴンの条件キー」を参照してください。条件キーを使用できるアクションとリソースについては、「Amazon Managed Blockchain (AMB) Access Polygon で定義されるアクション」を参照してください。

AMB Access Polygon アイデンティティベースのポリシーの例を表示するには、「」を参照してくださいAmazon Managed Blockchain (AMB) Access Polygon のアイデンティティベースのポリシーの例。

AMB アクセスポリゴンACLs

ACL をサポート: なし

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための許可を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

AMB アクセスポリゴンでの ABAC

ABAC をサポート (ポリシー内のタグ): いいえ

属性ベースのアクセス制御 (ABAC) は、属性に基づいてアクセス許可を定義する認可戦略です。では AWS、これらの属性はタグ と呼ばれます。タグは、IAM エンティティ (ユーザーまたはロール) および多くの AWS リソースにアタッチできます。エンティティとリソースのタグ付けは、ABAC の最初の手順です。その後、プリンシパルのタグがアクセスしようとしているリソースのタグと一致した場合にオペレーションを許可するように ABAC ポリシーをします。

ABAC は、急成長する環境やポリシー管理が煩雑になる状況で役立ちます。

タグに基づいてアクセスを管理するには、aws:ResourceTag/*key-name*、aws:RequestTag/*key-name*、または aws:TagKeys の条件キーを使用して、ポリシーの条件要素でタグ情報を提供します。

サービスがすべてのリソースタイプに対して 3 つの条件キーすべてをサポートする場合、そのサービスの値はありです。サービスが一部のリソースタイプに対してのみ 3 つの条件キーのすべてをサポートする場合、値は「部分的」になります。

ABAC の詳細については、IAM ユーザーガイドのABAC とは?を参照してください。ABAC をセットアップするステップを説明するチュートリアルについては、IAM ユーザーガイドの属性に基づくアクセスコントロール (ABAC) を使用するを参照してください。

AMB Access Polygon での一時的な認証情報の使用

一時的な認証情報をサポート： いいえ

一部の は、一時的な認証情報を使用してサインインすると機能 AWS のサービス しません。一時的な認証情報 AWS のサービス を使用する などの詳細については、IAM ユーザーガイドのAWS のサービス「IAM と連携する 」を参照してください。

ユーザー名とパスワード以外の AWS Management Console 方法で にサインインする場合、一時的な認証情報を使用します。例えば、会社の Single Sign-On (SSO) リンク AWS を使用して にアクセスすると、そのプロセスによって一時的な認証情報が自動的に作成されます。また、ユーザーとしてコンソールにサインインしてからロールを切り替える場合も、一時的な認証情報が自動的に作成されます。ロールの切り替えに関する詳細については、IAM ユーザーガイドのロールへの切り替え (コンソール)を参照してください。

一時的な認証情報は、 AWS CLI または AWS API を使用して手動で作成できます。その後、これらの一時的な認証情報を使用して . AWS recommends にアクセスできます AWS。これは、長期的なアクセスキーを使用する代わりに、一時的な認証情報を動的に生成することを推奨しています。詳細については、IAM の一時的セキュリティ認証情報を参照してください。

AMB Access Polygon のクロスサービスプリンシパル許可

転送アクセスセッション (FAS) のサポート: なし

IAM ユーザーまたはロールを使用して でアクションを実行すると AWS、プリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、 を呼び出すプリンシパルのアクセス許可を AWS のサービス、ダウンストリームサービス AWS のサービス へのリクエストリクエストリクエストと組み合わせて使用します。FAS リクエストは、サービスが他の AWS のサービス またはリソースとの

やり取りを完了する必要があるリクエストを受け取った場合にのみ行われます。この場合、両方のアクションを実行するためのアクセス許可が必要です。FAS リクエストを行う際のポリシーの詳細については、「転送アクセスセッション」を参照してください。

AMB Access Polygon のサービスロール

サービスロールをサポート： いいえ

サービスロールとは、サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、「IAM ユーザーガイド」の「AWS のサービスにアクセス許可を委任するロールの作成」を参照してください。

> **Warning**
>
> サービスロールのアクセス許可を変更すると、AMB Access Polygon の機能が破損する可能性があります。AMB Access Polygon が指示する場合以外は、サービスロールを編集しないでください。

AMB Access Polygon のサービスにリンクされたロール

サービスにリンクされたロールのサポート: なし

サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント 、サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールのアクセス許可を表示できますが、編集することはできません。

サービスリンクロールの作成または管理の詳細については、IAM と提携するAWS のサービスを参照してください。表の中から、[Service-linked role] (サービスにリンクされたロール) 列に Yes と記載されたサービスを見つけます。サービスリンクロールに関するドキュメントをサービスで表示するには、はい リンクを選択します。

Amazon Managed Blockchain (AMB) Access Polygon のアイデンティティベースのポリシーの例

デフォルトでは、ユーザーとロールには AMB Access Polygon リソースを作成または変更するアクセス許可はありません。また、、 AWS Command Line Interface （ AWS CLI ） AWS Management

Console、または AWS API を使用してタスクを実行することはできません。IAM 管理者は、リソースで必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き受けることができます。

これらサンプルの JSON ポリシードキュメントを使用して、IAM アイデンティティベースのポリシーを作成する方法については、IAM ユーザーガイドのIAM ポリシーの作成を参照してください。

各リソースタイプの ARNs「サービス認証リファレンス」の「Amazon Managed Blockchain (AMB) Access Polygon のアクション、リソース、および条件キー」を参照してください。

トピック

- ポリシーのベストプラクティス
- AMB Access Polygon コンソールの使用
- 自分の権限の表示をユーザーに許可する
- ポリゴンネットワークへのアクセス

ポリシーのベストプラクティス

ID ベースのポリシーは、ユーザーのアカウントで誰かが AMB Access Polygon リソースを作成、アクセス、または削除できるどうかを決定します。これらのアクションを実行すると、 AWS アカウントに料金が発生する可能性があります。アイデンティティベースポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください:

- AWS 管理ポリシーを開始し、最小特権のアクセス許可に移行する – ユーザーとワークロードにアクセス許可を付与するには、多くの一般的なユースケースにアクセス許可を付与する AWS 管理ポリシーを使用します。これらは で使用できます AWS アカウント。ユースケースに固有の AWS カスタマー管理ポリシーを定義して、アクセス許可をさらに減らすことをお勧めします。詳細については、「IAM ユーザーガイド」の「AWS マネージドポリシー」または「ジョブ機能のAWS マネージドポリシー」を参照してください。
- 最小特権を適用する – IAM ポリシーで許可を設定する場合は、タスクの実行に必要な許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権アクセス許可とも呼ばれています。IAM を使用して許可を適用する方法の詳細については、IAM ユーザーガイドのIAM でのポリシーとアクセス許可を参照してください。
- IAM ポリシーで条件を使用してアクセスをさらに制限する - ポリシーに条件を追加して、アクションやリソースへのアクセスを制限できます。例えば、ポリシー条件を記述して、すべてのリクエ

ストを SSL を使用して送信するように指定できます。条件を使用して、 などの特定の を介してサービスアクションが使用される場合に AWS のサービス、サービスアクションへのアクセスを許可することもできます AWS CloudFormation。詳細については、「IAM ユーザーガイド」の [IAM JSON policy elements: Condition] (IAM JSON ポリシー要素:条件) を参照してください。

- IAM Access Analyzer を使用して IAM ポリシーを検証し、安全で機能的な権限を確保する - IAM Access Analyzer は、新規および既存のポリシーを検証して、ポリシーが IAM ポリシー言語 (JSON) および IAM のベストプラクティスに準拠するようにします。IAM アクセスアナライザーは 100 を超えるポリシーチェックと実用的な推奨事項を提供し、安全で機能的なポリシーの作成をサポートします。詳細については、IAM ユーザーガイドのIAM Access Analyzer ポリシーの検証を参照してください。
- 多要素認証 (MFA) を要求する – で IAM ユーザーまたはルートユーザーを必要とするシナリオがある場合は AWS アカウント、セキュリティを強化するために MFA を有効にします。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、IAM ユーザーガイドのMFA 保護 API アクセスの設定を参照してください。

IAM でのベストプラクティスの詳細については、IAM ユーザーガイドのIAM でのセキュリティのベストプラクティスを参照してください。

AMB Access Polygon コンソールの使用

Amazon Managed Blockchain (AMB) Access Polygon コンソールにアクセスするには、最小限のアクセス許可のセットが必要です。これらのアクセス許可により、 の AMB Access Polygon リソースの詳細を一覧表示および表示できます AWS アカウント。最小限必要な許可よりも制限が厳しいアイデンティティベースのポリシーを作成すると、そのポリシーを持つエンティティ (ユーザーまたはロール) に対してコンソールが意図したとおりに機能しません。

AWS CLI または AWS API のみを呼び出すユーザーには、最小限のコンソールアクセス許可を付与する必要はありません。代わりに、実行しようとしている API オペレーションに一致するアクションのみへのアクセスが許可されます。

ユーザーとロールが引き続き AMB Access Polygon コンソールを使用できるようにするには、エンティティに AMB Access Polygon *ConsoleAccess*または *ReadOnly* AWS 管理ポリシーもアタッチします。詳細については、IAM ユーザーガイドのユーザーへの許可の追加を参照してください。

自分の権限の表示をユーザーに許可する

この例では、ユーザーアイデンティティにアタッチされたインラインおよびマネージドポリシーの表示を IAM ユーザーに許可するポリシーの作成方法を示します。このポリシーには、コンソールで、

または AWS CLI または AWS API を使用してプログラムでこのアクションを実行するアクセス許可が含まれています。

```
{
    "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": "*"
        }
    ]
}
```

ポリゴンネットワークへのアクセス

> **Note**
>
> Polygon のパブリックエンドポイントにアクセスしmainnet、JSON-RPC 呼び出しmainnetを行うには、AMB Access Polygon に適切な IAM アクセス許可を持つユーザー認証情報 (AWS_ACCESS_KEY_ID および AWS_SECRET_ACCESS_KEY) が必要です。

Example すべてのポリゴンネットワークにアクセスするための IAM ポリシー

この例では、 の IAM ユーザーにすべてのポリゴンネットワーク AWS アカウント へのアクセスを許可します。

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

Example Polygon Mainnet ネットワークにアクセスするための IAM ポリシー

この例では、Polygon Mainnet ネットワーク AWS アカウント へのアクセス権を IAM ユーザーに付与します。

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

```
            ],
            "Resource": "*"
        }
    ]
}
```

Amazon Managed Blockchain (AMB) アクセスポリゴンアイデンティティとアクセスのトラブルシューティング

以下の情報は、AMB Access Polygon と IAM の使用時に発生する可能性がある一般的な問題の診断と修正に役立ちます。

トピック

- AMB Access Polygon でアクションを実行する権限がない
- iam を実行する権限がありません。PassRole
- 自分の 以外のユーザーに AMB Access Polygon リソース AWS アカウント へのアクセスを許可したい

AMB Access Polygon でアクションを実行する権限がない

「I am not authorized to perform an action in Amazon Bedrock」というエラーが表示された場合、そのアクションを実行できるようにポリシーを更新する必要があります。

次のエラー例は、mateojackson IAM ユーザーがコンソールを使用して、ある *my-example-widget* リソースに関する詳細情報を表示しようとしたことを想定して、その際に必要なmanagedblockchain::*GetWidget* アクセス許可を持っていない場合に発生するものです。

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

この場合、managedblockchain::*GetWidget* アクションを使用して *my-example-widget*リソースへのアクセスを許可するように、mateojackson ユーザーのポリシーを更新する必要があります。

サポートが必要な場合は、 AWS 管理者にお問い合わせください。サインイン認証情報を提供した担当者が管理者です。

iam を実行する権限がありません。PassRole

`iam:PassRole` アクションを実行する権限がないというエラーが表示された場合は、ポリシーを更新して AMB Access Polygon にロールを渡すことができるようにする必要があります。

一部の AWS のサービス では、新しいサービスロールまたはサービスにリンクされたロールを作成する代わりに、そのサービスに既存のロールを渡すことができます。そのためには、サービスにロールを渡す権限が必要です。

次の例のエラーは、 という IAM `marymajor` ユーザーがコンソールを使用して AMB Access Polygon でアクションを実行しようする場合に発生します。ただし、このアクションをサービスが実行するには、サービスロールから付与された権限が必要です。メアリーには、ロールをサービスに渡す許可がありません。

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

この場合、Mary のポリシーを更新してメアリーに `iam:PassRole` アクションの実行を許可する必要があります。

サポートが必要な場合は、 AWS 管理者にお問い合わせください。サインイン認証情報を提供した担当者が管理者です。

自分の 以外のユーザーに AMB Access Polygon リソース AWS アカウント へのアクセスを許可したい

他のアカウントのユーザーや組織外の人が、リソースにアクセスするために使用できるロールを作成できます。ロールの引き受けを委託するユーザーを指定できます。リソースベースのポリシーまたはアクセスコントロールリスト (ACL) をサポートするサービスの場合、それらのポリシーを使用して、リソースへのアクセスを付与できます。

詳細については、以下を参照してください:

- AMB Access Polygon がこれらの機能をサポートしているかどうかについては、「」を参照してくださいAmazon Managed Blockchain (AMB) Access Polygon と IAM の連携。
- 所有 AWS アカウント している のリソースへのアクセスを提供する方法については、IAM ユーザーガイドの「所有 AWS アカウント している別の の IAM ユーザーへのアクセスを提供する」を参照してください。

- リソースへのアクセスをサードパーティー に提供する方法については AWS アカウント、IAM ユーザーガイドの「サードパーティー AWS アカウント が所有する へのアクセスを提供する」を参照してください。
- ID フェデレーションを介してアクセスを提供する方法については、IAM ユーザーガイドの外部で認証されたユーザー (ID フェデレーション) へのアクセスの許可を参照してください。
- クロスアカウントアクセスにおけるロールとリソースベースのポリシーの使用方法の違いについては、「IAM ユーザーガイド」の「IAM でのクロスアカウントのリソースへのアクセス」を参照してください。

を使用した Amazon Managed Blockchain (AMB) アクセスポリゴンイベントのログ記録 AWS CloudTrail

> **Note**
>
> Amazon Managed Blockchain (AMB) Access Polygon は管理イベントをサポートしていません。

Amazon Managed Blockchain は AWS CloudTrail、データプレーンイベントとして Managed Blockchain の AMB Access Polygon エンドポイントを呼び出した Managed Blockchain. CloudTrail captures のユーザー、ロール、または AWS サービスによって実行されたアクションを記録するサービスである で実行されます。

必要なデータプレーンイベントを受信するようにサブスクライブされている適切に設定された証跡を作成すると、AMB Access Polygon 関連の CloudTrail イベントが S3 バケットに継続的に配信されます。によって収集された情報を使用して CloudTrail、AMB Access Polygon エンドポイントの 1 つ、リクエスト元の IP アドレス、リクエスト者、リクエスト日時などの詳細に対してリクエストが行われたかどうかを判断できます。

の詳細については CloudTrail、「AWS CloudTrail ユーザーガイド」を参照してください。

での AMB アクセスポリゴン情報 CloudTrail

CloudTrail は、作成 AWS アカウント 時に で有効になります。ただし、AMB Access Polygon エンドポイントを呼び出したユーザーを表示するようにデータプレーンイベントを設定する必要があります。

AMB Access Polygon のイベントなど AWS アカウント、 のイベントの継続的な記録については、証跡を作成します。証跡により、 はログファイル CloudTrail を S3 バケットに配信できます。デフォルトでは、コンソールで証跡を作成するときに、証跡がすべての AWS リージョンに適用されます。証跡は、 AWS パーティションでサポートされているすべてのリージョンからのイベントを口グに記録し、指定した S3 バケットにログファイルを配信します。さらに、他の を設定 AWS のサービス してさらに分析し、 CloudTrail ログで収集されたイベントデータに対応できます。詳細については、次を参照してください:

- CloudTrail を使用してポリゴン JSON-RPCs

- 追跡を作成するための概要
- CloudTrail がサポートするサービスと統合
- の Amazon SNS 通知の設定 CloudTrail
- 複数のリージョンからの CloudTrail ログファイルの受信と複数のアカウントからの CloudTrail ログファイルの受信

CloudTrail データイベントを分析することで、AMB Access Polygon エンドポイントを呼び出したユーザーをモニタリングできます。

各イベントまたはログエントリには、誰がリクエストを生成したかという情報が含まれます。アイデンティティ情報は、以下を判別するのに役立ちます:

- リクエストがルートまたは AWS Identity and Access Management (IAM) ユーザー認証情報のどちらを使用して行われたか
- リクエストが、ロールとフェデレーティッドユーザーのどちらの一時的なセキュリティ認証情報を使用して送信されたか
- リクエストが別の によって行われたかどうか AWS のサービス

詳細については、CloudTrail userIdentity 要素」を参照してください。

AMB Access Polygon ログファイルエントリについて

データプレーンイベントの場合、証跡は、指定された S3 バケットにイベントをログファイルとして配信できるようにする設定です。各 CloudTrail ログファイルには、任意のソースからの 1 つのリクエストを表す 1 つ以上のログエントリが含まれます。これらのエントリは、アクションの日付と時刻、および関連するリクエストパラメータなど、リクエストされたアクションに関する詳細を提供します。

> **Note**
>
> CloudTrail ログファイル内の データイベントは、AMB Access Polygon API コールの順序付けられたスタックトレースではないため、特定の順序では表示されません。

CloudTrail を使用してポリゴン JSON-RPCs

を使用して CloudTrail 、アカウント内の誰が AMB Access Polygon エンドポイントを呼び出し、どの JSON-RPC がデータイベント として呼び出されたかを追跡できます。デフォルトでは、証跡を作成すると、データイベントはログに記録されません。AMB Access Polygon エンドポイントを呼び出したユーザーを CloudTrail データイベントとして記録するには、アクティビティを収集するサポートされているリソースまたはリソースタイプを証跡に明示的に追加する必要があります。AMB Access Polygon は AWS Management Console、、 AWS CLI、 SDK を使用してデータイベントを追加することをサポートしています。詳細については、「 AWS CloudTrail ユーザーガイド」の「高度なセレクタを使用してイベントをログに記録する」を参照してください。

証跡のデータイベントをログに記録するには、証跡を作成した後、put-event-selectors オペレーションを使用します。`--advanced-event-selectors` オプションを使用して`AWS::ManagedBlockchain::Network`リソースタイプを指定し、データイベントのログ記録を開始して、AMB Access Polygon エンドポイントを呼び出したユーザーを決定します。

Example アカウントのすべての AMB Access Polygon エンドポイントリクエストのデータイベントログエントリ

次の例は、 `put-event-selectors`オペレーションを使用して、 `us-east-1`リージョン`my-polygon-trail`の証跡に対するアカウントの AMB Access Polygon エンドポイントリクエストをすべてログに記録する方法を示しています。

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

サブスクライブすると、前の例で指定した証跡に接続されている S3 バケットの使用状況を追跡できます。

次の結果は、 によって収集された情報の CloudTrail データイベントログエントリを示しています CloudTrail。ポリゴン JSON-RPC リクエストが AMB Access Polygon エンドポイントの 1 つ、リクエスト元の IP アドレス、リクエスト者、リクエスト日時などの詳細に対して行われたかどうかを判

断できます。次の例の一部の値は、セキュリティ上の理由から難読化されていますが、実際の口グエントリには完全に表示されます。


```
{
        "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 アクセスポリゴンユーザーガイドのドキュメント履歴

次の表は、AMB アクセスポリゴンのドキュメントリリースをまとめたものです。

変更	説明	日付
JSON-RPC のクォータが更新されました。	AMB アクセスポリゴンがサポートする各 JSON-RPC でサポートされるクォータが更新されました。	2024 年 4 月 12 日
ムンバイテストネットネットワークのサポート終了	AMB アクセスポリゴンは 2024 年 4 月 15 日にムンバイテストネットのサポートを終了しました。	2024 年 4 月 10 日
「チュートリアル」トピックの追加	AMB は AWS re: Post のコミュニティ記事セクションからポリゴンチュートリアルにアクセスできます。	2024 年 4 月 9 日
パブリックプレビュー	Amazon Managed Blockchain (AMB) アクセスポリゴンサービスのパブリックプレビューリリース。	2023 年 11 月 24 日