デベロッパーガイド

AMB アクセスポリゴン

AMB アクセスポリゴン: デベロッパーガイド

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。Polygon 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。Polygon ブロックチェーンは、proof-of-stakeセンサスメカニズムを使用します。Polygon は、NFTs、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。

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を説明します。

初めて AMB アクセスポリゴン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) Access Polygon
- Amazon Managed Blockchain (AMB) アクセスポリゴンの開始方法
- 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) Access Polygon

> **Note**
>
> このガイドでは、Polygon に不可欠な概念を理解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 は、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を提供します。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必要はありません。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、コンセンサスに参加してブロックチェーンの状態を変更します。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 リモートプロシージャ (JSON-RPC) 呼び出しにアクセスできます。Polygon 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 Polygon 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 Access Polygon を使用して、スマートコントラクトを作成して Polygon ブロックチェーンにデプロイできます。また、Polygon ネットワークへのピアであ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) Access Polygon 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多角形ネットワーク

  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センサスによって保護され、Polygon (POL) トークンが発行されて取引されるパブリック Polygon ブロックチェーン。Mainnet のトランザクションには実際の値があり (つまり、実際のコストが発生します) 、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 月中旬に日没します。この二ュースに従って、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、開発者は使い慣れたツールを使用して Polygon ブロックチェーンを操作し、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すばやく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を次に示します。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 Access Polygon はproof-of-stakeのために Polygon (POL) 検証ノードをサポートしていません。

- Polygon JSON-RPC リクエストの署名バージョン 4

  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 承認された 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、 コールで 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) を指定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 Polygon 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

  アクセサートークンを使用して、署名バージョン 4 (SigV4) 署名プロセスの便利な代替として、Polygon 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 Access Polygon 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 AWS のサービスを含むすべての に 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。

AWS アカウント を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、電話またはテキストメッセージを受け取り、電話キーパッドで検証コードを入力します。

   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。セキュリティベストプラクティスとして、ユーザーに管理アクセス権を割り当て、ルートユーザーアクセスが必要なタスクの実行にはルートユーザーのみを使用するようにしてください。

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 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 承認された 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、 コールで 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) を指定する必要があります。

Accessor トークンを使用して、署名バージョン 4 (SigV4) 署名プロセスの便利な代替として、Polygon 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、、、 SDK を使用して Accessor トークンを作成するアクセス許可を取得するには AWS マネジメントコンソール 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。

のインストールと設定 AWS Command Line Interface

まだインストールしていない場合は、 latest 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) アクセスポリゴンの開始方法

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
- 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 マネジメントコンソール
- awscurl を使用して で AMB アクセスポリゴン JSON-RPC リクエストを作成する AWS CLI
- Node.js で多角形 JSON-RPC リクエストを行う

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 Polygon エンドポイントにアクセスする 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 Polygon 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 マネジメントコンソール、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 マネジメントコンソール

AMB Access Polygon AWS マネジメントコンソール 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、データの取得や多角形ネットワークへのトランザクションの送信など、多角形ネットワークでデータを読み書きできます。

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て、必要な入力を入力します。

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、***###############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、送信 RPC 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。

`awscurl` を使用して で AMB アクセスポリゴン JSON-RPC リクエストを作成する 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。

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、HomeBrew が推奨アプリケーションです。

```
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`、`eth_getBlockByNumber`RPC を呼び出して Polygon Mainnet にリクエストを送信します。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**
>
> また、 を使用して同じリクエストを行い`curl`、`Accessor`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 リクエストを行う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。例については、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 で多角形 JSON-RPC リクエストを行う

HTTPS を使用して署名付きリクエストを送信し、Node.js のネイティブ https モジュールを使用して Polygon Mainnet ネットワークにアクセスすることで、Polygon JSON-RPCs を呼び出すこ

とも、AXIOS などのサードパーティーライブラリを使用することもできます。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 Access Polygon エンドポイントに Polygon 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. マシンには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、こちら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、LTS バージョンの Node 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`と 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完了したら、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");
```

これらのファイルを ディレクトリに保存したら、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

前の例では、トランザクションに署名し、AMB Access Polygon を使用して Polygon Mainnet 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 Polygon Mainnet 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。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。これには、トークンベースのアクセス`billingToken`用の Accessor 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 を受信する受信者のアドレスが含まれます。

> **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、資金供給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 Access Polygon を使用して eth_sendRawTransaction JSON-RPC を呼び出す Polygon Mainnet 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先ほど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、Polygon Mainnet からこのトランザクションに関する追加の詳細を取得します。

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 Polygon 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。このユーティリティは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、受信者アドレスバランスの 16 進数表現 (0x110d9316ec000) は、Ethers の `formatEther()`メソッドを使用して 10 進数に変換されます。これにより、16 進数を 10 進数に変換し、18 進数を 18 (10^18) シフトして POL の真のバランスが得られます。

> **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いくつかの JSON-RPCs 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例を変更したり、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 リクエストを行う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

Accessor トークンを使用して、署名バージョン 4 (SigV4) 署名プロセスの便利な代替として、Polygon 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から AWS アカウント 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 マネジメントコンソール。また、、CreateAccessor、、および の API オペレーションを使用して Accessor GetAccessorトークンを作成ListAccessorsおよび管理することもできます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 AMB Access Polygon JSON-RPC リクエストを請求するために使用されます 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 マネジメントコンソール、、 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 Accessor トークンの作成

Accessor トークンを作成し、これを使用して、 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 マネジメントコンソール

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. 有効な多角形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. Create Accessor を選択して、新しい Accessor 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 Accessor 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。たとえば、Accessor ID または Accessor 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 マネジメントコンソール

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。

が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 Accessor 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 マネジメントコンソール

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. リストから必要な Accessor 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。このページには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。Accessor ID --accessor-idで の値を設定します。

CLI を使用した Accessor AWS 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 Access Polygon がサポートする Polygon 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の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際の固有の考慮事項は、該当する場合に示されています。

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 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 承認された 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、 呼び出しで 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) を指定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) アクセスポリゴン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 (Mainnet) ごとに、リージョンごとに 1 秒あたりのリクエスト数 (RPS) で設定されます。

クォータを引き上げるには、 に連絡する必要があります サポート。に問い合わせるには サポート、 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。カテゴリとして Access:Polygon を選択し、重要度として一般的なガイダンスを選択します。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 で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。`Chain Id` が使用`None`できない場合は を返します。	
	eth_estimateGas	ブロックチェーンにトランザクションを追加せずに、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 の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_getBlockByHash	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_getBlockByNumber	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_getBlockTransactionCountByHash	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_getBlockTransactionCountByNumber	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。	契約アドレスを指定すると、デフォルトで 1K ブロック範囲で任意のブロック範囲で`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_getTransactionByHash	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_getUncleCountByBlockHash	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_getUncleCountByBlockNumber	句番号を使用して指定された句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、優先順位料金または「ヒント」として支払える金額の見積もりであるガスあたりの料金を返します。	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	署名付きトランザクションの新しいメッセージコールトランザクションまたは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。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_traceBlockByHash	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、トレース可能な結果番号を返します (トレースモードが必要)。	
	debug_traceBlockByNumber	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のコンテキスト内で e 番目の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 (トレースモードが必要)。	
	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 opcode 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のコンテキスト内で e 番目の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 (トレースモードが必要) 。	
	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) 。	
Tx 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、キューに入れられているトランザクション (将来の実行のみ予定) の数を提供します。	
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) アクセスポリゴン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、NFTs、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。このトピックでは、Amazon Managed Blockchain (AMB) Access Polygon 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

多角形 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、多角形 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、傾向のある NFTsや、特定のコレクションと最も頻繁にやり取りしているユーザー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、最初の mint、許可リスト、またはセカンダリ市場を使用して 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 Cryptocurrencies を受け入れ、関係するすべての利害関係者をすばやく解決できます。

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。

多角形ウォレットを作成する

AMB Access Polygon を使用すると、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、残高の確認、アセットの転送、アセットの送信、料金の見積もり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ために必要なwallet-as-a-service運用ウォレットを開発できます。

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み、ユーザーのアドレスの 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 のコミュニティ記事です。 AWS re:Post

- 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オフチェーンで取得する
- AMB アクセスを使用して Polygon Mainnet で ERC-20 トークンデータを分析する

Amazon Managed Blockchain (AMB) アクセスポリゴンのセキュリティ

のクラウドセキュリティ AWS が最優先事項です。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を活用できます。

セキュリティは、お客様と AWS お客様の間の責任共有です。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のセキュリティの両方と定義しています。

- 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。 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) Access Polygon に適用されるコンプライアンスプログラムについては、AWS「コンプライアンス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ように AMB Access Polygon を設定する方法について説明します。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 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) アクセスポリゴンでのデータ保護
- Amazon Managed Blockchain (AMB) Access Polygon の Identity and Access Management

Amazon Managed Blockchain (AMB) アクセスポリゴンでのデータ保護

Amazon Managed Blockchain (AMB) Access Polygon でのデータ保護には、 AWS 責任共有モデル責任共有モデルが適用されます。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。ユーザーは、このインフラストラクチャでホストされ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に関する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 アイデンティティセンターまたは 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 が必須で、TLS 1.3 をお勧めします。
- 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。CloudTrail 証跡を使用して AWS アクティビティをキャプチャする方法については、「 AWS CloudTrail ユーザーガイド」のCloudTrail 証跡の使用」を参照してください。
- AWS 暗号化ソリューションと、その中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 などの高度な管理された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 140-3 検証済み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) 140-3」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を、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含め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 SDK を使用して AMB Access Polygon AWS CLI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 タグ、または名前に使用される自由記述のテキストフィールドに入力したデータは、請求または診断ログに使用される場合があります。外部サーバーに URL を提供する場合、そ

のサーバーへのリクエストを検証できるように、認証情報を URL に含めないことを強くお勧めします。

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、転送中のデータと呼ばれる、ネットワークを通過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。

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。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`--no-verify-ssl`。

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 リソースの使用を許可する (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

- オーディエンス
- アイデンティティを使用した認証
- ポリシーを使用したアクセスの管理
- Amazon Managed Blockchain (AMB) Access Polygon と IAM の連携
- 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
- 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

オーディエンス

AWS Identity and Access Management (IAM) の使用方法は、ロールによって異なります。

- サービスユーザー - 機能にアクセスできない場合は、管理者にアクセス許可をリクエストします (「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」を参照)。
- サービス管理者 - ユーザーアクセスを決定し、アクセス許可リクエストを送信します (「Amazon Managed Blockchain (AMB) Access Polygon と IAM の連携」を参照)
- IAM 管理者 - アクセスを管理するためのポリシーを作成します (「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」を参照)

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。、IAM ユーザー AWS アカウントのルートユーザー、または IAM ロールを引き受けることで認証される必要があります。

(AWS IAM アイデンティティセンター IAM Identity Center)、シングルサインオン認証、Google/Facebook 認証情報などの ID ソースからの認証情報を使用して、フェデレーティッド ID としてサインインできます。サインインの詳細については、「AWS サインイン ユーザーガイド」の「AWS アカウントにサインインする方法」を参照してください。

プログラムによるアクセスの場合、 は SDK と CLI AWS を提供してリクエストを暗号化して署名します。詳細については、「IAM ユーザーガイド」の「API リクエストに対するAWS 署名バージョン 4」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、まず、すべての AWS のサービス および リソースへの完全なアクセス権を持つ AWS アカウント root ユーザーと呼ばれる 1 つのサインインアイデンティティから始めます。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。ルートユーザー認証情報を必要とするタスクについては、「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。

フェデレーテッドアイデンティティ

ベストプラクティスとして、人間のユーザーが一時的な認証情報 AWS のサービス を使用して にアクセスするには、ID プロバイダーとのフェデレーションを使用する必要があります。

フェデレーティッド ID は、エンタープライズディレクトリ、ウェブ ID プロバイダー、または ID Directory Service ソースの認証情報 AWS のサービス を使用して にアクセスするユーザーです。フェデレーテッドアイデンティティは、一時的な認証情報を提供するロールを引き受けます。

アクセスを一元管理する場合は、 AWS IAM アイデンティティセンターをお勧めします。詳細については、「AWS IAM アイデンティティセンター ユーザーガイド」の「IAM アイデンティティセンターとは」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、特定の個人やアプリケーションに対する特定のアクセス許可を持つアイデンティティです。長期認証情報を持つ IAM ユーザーの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。詳細については、IAM ユーザーガイドの「ID プロバイダーとのフェデレーションを使用して にアクセスする必要がある AWS」を参照してください。

IAM グループは、IAM ユーザーの集合を指定し、大量のユーザーに対するアクセス許可の管理を容易にします。詳細については、「IAM ユーザーガイド」の「IAM ユーザーに関するユースケース」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を持つアイデンティであり、一時的な認証情報を提供します。ユーザーから IAM ロール (コンソール) に切り替えるか、 または API オペレーションを呼び出すことで、ロールを引き受けることができます。 AWS CLI AWS 詳細については、「IAM ユーザーガイド」の「ロールを引き受けるための各種方法」を参照してください。

IAM ロールは、フェデレーションユーザーアクセス、一時的な IAM ユーザーのアクセス許可、クロスアカウントアクセス、クロスサービスアクセス、および Amazon EC2 で実行するアプリケーションに役立ちます。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。

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、ID AWS 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たときにアクセス許可を定義します。は、プリンシパル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。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、ポリシーを使用して、どのプリンシパルがどのリソースに対して、どのような条件でアクションを実行できるかを定義することで、誰が何にアクセスできるかを指定します。

デフォルトでは、ユーザーやロールにアクセス許可はありません。IAM 管理者は IAM ポリシーを作成してロールに追加し、このロールをユーザーが引き受けられるようにします。IAM ポリシーは、オペレーションの実行方法を問わず、アクセス許可を定義します。

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のポリシーは、アイデンティティ (ユーザー、グループ、またはロール) にアタッチできる JSON アクセス許可ポリシードキュメントです。これらのポリシーは、アイデンティティがどのリソースに対してどのような条件下でどのようなアクションを実行できるかを制御します。アイデンティティベースポリシーの作成方法については、IAM ユーザーガイド の カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する 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、インラインポリシー (単一の ID に直接埋め込む) または管理ポリシー (複数の ID にアタッチされたスタンドアロンポリシー) にすることができます。管理ポリシーと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「管理ポリシーとインラインポリシーのいずれかを選択する」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。例としては、IAM ロール信頼ポリシーや Amazon S3 バケットポリシーなどがあります。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。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。

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。

その他のポリシータイプ

AWS は、より一般的なポリシータイプによって付与されるアクセス許可の最大数を設定できる追加のポリシータイプをサポートしています。

- アクセス許可の境界 – アイデンティティベースのポリシーで IAM エンティティに付与することのできるアクセス許可の数の上限を設定します。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 (SCP) - AWS Organizations内の組織または組織単位の最大のアクセス許可を指定します。詳細については、「AWS Organizations ユーザーガイド」の「サービスコントロールポリシー」を参照してください。
- リソースコントロールポリシー (RCP) – は、アカウント内のリソースで利用できる最大数のアクセス許可を定義します。詳細については、「AWS Organizations ユーザーガイド」の「リソースコントロールポリシー (RCP)」を参照してください。
- セッションポリシー – ロールまたはフェデレーションユーザーの一時セッションを作成する際にパラメータとして渡される高度なポリシーです。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アクセス許可を理解するのがさらに難しくなります。が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 機能を確認してください。

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

IAM 機能	AMB Access Polygon 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

IAM 機能	AMB Access Polygon のサポート
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	いいえ

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については、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: あり

アイデンティティベースポリシーは、IAM ユーザー、ユーザーグループ、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ポリシーの作成方法については、「IAM ユーザーガイド」の「カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する」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。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コントロールできます。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、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、全体のアカウント、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。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のサポート: あり

管理者は JSON AWS 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対してどのような条件下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。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アクセス許可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、「サービス認可リファレンス」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 Access Polygon のポリシーリソース

ポリシーリソースのサポート: なし

管理者は JSON AWS 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対してどのような条件下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。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。リソースレベルのアクセス許可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「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へのサポート: なし

管理者は JSON AWS 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対してどのような条件下でアクションを実行できるかということです。

Condition 要素は、定義された基準に基づいてステートメントが実行される時期を指定します。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して、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。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 Access Polygon 条件キーのリストを確認するには、「サービス認可リファレンス」の「Amazon Managed Blockchain (AMB) Access Polygon 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 のサポート: 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アクセス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 アクセスポリゴンを使用した ABAC

ABAC (ポリシー内のタグ) のサポート: なし

属性ベースのアクセス制御 (ABAC) は、タグと呼ばれる属性に基づいてアクセス許可を定義する認可戦略です。IAM エンティティと AWS リソースにタグをアタッチし、プリンシパルのタグがリソースのタグと一致するときにオペレーションを許可するように ABAC ポリシーを設計できます。

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 認可でアクセス許可を定義する」を参照してください。ABAC をセットアップする手順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「属性ベースの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

一時的な認証情報のサポート: なし

一時的な認証情報は AWS 、リソースへの短期的なアクセスを提供し、フェデレーションまたはスイッチロールの使用時に自動的に作成されます。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 AWS 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「IAM の一時的な認証情報」および「AWS のサービス と IAM との連携」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アクセス許可

転送アクセスセッション (FAS) のサポート: なし

転送アクセスセッション (FAS) は、 を呼び出すプリンシパルのアクセス許可と AWS のサービス、ダウンストリームサービス AWS のサービス へのリクエストをリクエストする を使用します。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

サービスロールのサポート: なし

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 のサービスに許可を委任するロールを作成する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 機能が破損する可能性があります。AMB Access Polygon が指示する場合にのみ、サービスロールを編集します。

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。

サービスにリンクされた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「サービスリンクロール」列に Yes と記載されたサービスを見つけます。サービスにリンクされた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。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。

これらのサンプルの 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 ポリシーを作成する (コンソール)」を参照してください。

各リソースタイプの ARNs「サービス認可リファレンス」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

- ポリシーに関する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
- Polygon ネットワークへのアクセス

ポリシーに関するベストプラクティス

ID ベースのポリシーは、アカウント内で誰かが AMB Access Polygon リソースを作成、アクセス、または削除できるかどうかを決定します。これらのアクションでは、 AWS アカウントに費用が発生する場合があります。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の付与を開始するには、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することで、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 ポリシーでアクセス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。たとえば、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、サービスアクションが などの特定の を通じて使用されている場合に 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 CloudFormation。詳細については、IAM ユーザーガイド の IAM JSON ポリシー要素:条件 を参照してください。
- IAM アクセスアナライザー 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 アクセスアナライザー 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 でポリシーを検証する 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、MFA をオンにしてセキュリティを強化します。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 を使用した安全な API アクセス 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 マネージド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「ユーザーへのアクセス許可の追加」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": "*"
        }
    ]
}
```

Polygon 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 の適切な 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、すべての Polygon 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。

JSON

```
{
    "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 へのアクセスを IAM ユーザーに許可します。

JSON

```
{
    "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ありません
- iam:PassRole を実行する権限がありません
- 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ありません

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、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:PassRole を実行する権限がありません

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、既存のロールをそのサービスに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アクセス許可が必要です。

次の例のエラーは、 という IAM 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marymajorしようとすると発生します。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アクセス許可が必要です。Mary には、ロールをサービスに渡すアクセス許可がありません。

```
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。サインイン資格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を確認するには、「」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、Managed Blockchain のユーザー、ロール、または のサービスによって実行されたアクションを記録する AWS サービスで実行されます。CloudTrail は、マネージドブロックチェーンの AMB Access Polygon エンドポイントをデータプレーンイベントとして呼び出したユーザーをキャプチャします。

必要なデータプレーンイベントを受信するため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 イベントを S3 バケットに継続的に配信できます。CloudTrail によって収集された情報を使用して、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。

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

CloudTrail の AMB アクセスポリゴン情報

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。trail 、CloudTrail はログファイルを 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 を使用して多角形 JSON-RPCs

- 追跡を作成するための概要
- CloudTrail がサポートされているサービスと統合
- 「CloudTrail の Amazon SNS 通知の設定」
- CloudTrail ログファイルを複数のリージョンから受け取る、複数のアカウントから CloudTrail ログファイルを受け取る

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) ユーザー認証情報を使用して行われたかどうか
- 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 エレメント」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、アクションの日時、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**
>
> ログファイルの CloudTrail 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 を使用して多角形 JSON-RPCs

CloudTrail を使用して、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 マネジメントコンソール、、 AWS CLI、および SDK を使用したデータイベントの追加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、証跡の作成後に 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リソース`AWS::ManagedBlockchain::Network`タイプを指定し、データイベントのログ記録を開始して、誰が AMB Access Polygon エンドポイントを呼び出したかを判断します。

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、CloudTrail 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。Polygon 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、実際のログエントリに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 Access Polygon ユーザーガイドのドキュメント履歴

次の表は、AMB Access Polygon のドキュメントリリースを示しています。

変更	説明	日付
JSON-RPC のクォータを更新	サポートされている JSON-RPC ごとに AMB Access Polygon がサポートするクォータが更新されます。	2024 年 4 月 12 日
ムンバイのテストネットネットワークのサポート終了	AMB Access Polygon は、2024 年 4 月 15 日にムンバイの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	AWS re:Post のコミュニティ記事セクションの AMB Access Polygon チュートリアル。	2024 年 4 月 9 日
パブリックプレビュー	Amazon Managed Blockchain (AMB) Access Polygon 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