aws

개발자 가이드

AMB 액세스 폴리곤

AMB 액세스 폴리곤: 개발자 가이드

Table of Contents

아마존 관리형 블록체인 (AMB) 액세스 폴리곤은 프리뷰 릴리즈 중이며 변경될 수 있습니다.

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.

아마존 관리형 블록체인 (AMB) 액세스 폴리곤이란 무엇입니까?

Amazon Managed Blockchain (AMB) Access Polygon은 폴리곤 블록체인에서 복원력이 뛰어난 Web3 애플리케이션을 구축하는 데 도움이 되는 완전 관리형 서비스입니다. AMB 액세스 폴리곤은 Polygon 블록체인에 대한 즉각적인 서버리스 액세스를 제공합니다.

Polygon은 이더리움 가상 머신 (EVM) 을 기반으로 사용하는 스케일링 솔루션입니다. Polygon 블록체인은 높은 거래 처리량과 낮은 거래 수수료로 유명합니다. Polygon 블록체인은 proof-of-stake 합의 메커니즘을 사용합니다. Polygon은 NFT, Web3 게임, 토큰화 사용 사례 등과 관련된 분산형 애플리케이션 (dApp) 을 구축하는 데 주로 사용됩니다.

이 가이드에서는 Amazon Managed Blockchain (AMB) Access Polygon을 사용하여 Polygon 블록체인 리소스를 생성하고 관리하는 방법을 다룹니다.

AMB 액세스 폴리곤을 처음 사용하는 사용자를 위한 리소스

AMB 액세스 폴리곤을 처음 사용하는 경우 먼저 다음 섹션을 읽는 것이 좋습니다.

- 주요 개념: 아마존 매니지드 블록체인 (AMB) 액세스 폴리곤
- 아마존 관리형 블록체인 (AMB) 액세스 폴리곤 시작하기
- AMB 액세스 폴리곤을 지원하는 관리형 블록체인 API 및 JSON-RPC

주요 개념: 아마존 매니지드 블록체인 (AMB) 액세스 폴리곤

> **Note**
>
> 이 가이드에서는 사용자가 Polygon에 필수적인 개념을 잘 알고 있다고 가정합니다. 이러한 개념에는 스테이킹, 디앱, 트랜잭션, 지갑, 스마트 컨트랙트, 폴리곤 (POL, 구 MATIC) 등이 포함됩니다. Amazon Managed Blockchain (AMB) 액세스 폴리곤을 사용하기 전에 폴리곤 개발 설명서와 폴리곤 위키를 검토하는 것이 좋습니다.

Amazon Managed Blockchain (AMB) Access Polygon은 노드를 포함한 폴리곤 인프라를 프로비저닝하고 관리할 필요 없이 폴리곤 메인넷 및 폴리곤 메인넷 네트워크에 대한 서버리스 액세스를 제공합니다. 네트워크의 폴리곤 노드는 Polygon 블록체인 상태를 총체적으로 저장하고, 트랜잭션을 확인하고, 합의에 참여하여 블록체인 상태를 변경합니다. 이 관리형 서비스를 사용하면 필요에 따라 신속하게 Polygon 네트워크에 액세스하여 전체 소유 비용을 줄일 수 있습니다.

AMB 액세스 폴리곤을 사용하면 JSON 원격 프로시저 (JSON-RPC) 호출에 액세스할 수 있습니다. 폴리곤 JSON-RPC를 호출하여 관리형 블록체인으로 관리되는 노드를 통해 폴리곤 블록체인과 통신할 수 있습니다. AMB Access Polygon 서비스를 사용하여 Polygon 블록체인과 상호 작용하는 분산형 애플리케이션 (DApp) 을 개발하고 사용할 수 있습니다. DApp의 필수적인 부분은 스마트 계약입니다. AMB Access Polygon을 사용하여 스마트 계약을 생성하고 Polygon 블록체인에 배포할 수 있습니다. Polygon 네트워크에 연결된 모든 노드에서 분산된 방식으로 실행되는 AMB Access Polygon 엔드포인트에 대해 JSON-RPC를 호출하여 지갑의 잔액, 거래 세부 정보, 예상 수수료 등을 확인할 수도 있습니다. Polygon 네트워크의 모든 피어 투 피어는 스마트 컨트랙트를 개발하고 배포할 수 있습니다.

> **Important**
>
> Polygon 주소를 생성, 유지, 사용 및 관리하는 것은 귀하의 책임입니다. 또한 폴리곤 주소의 내용에 대한 책임도 귀하에게 있습니다. AWS Amazon Managed Blockchain에서 Polygon 노드를 사용하여 배포되거나 호출된 트랜잭션에 대해서는 책임을 지지 않습니다.

아마존 관리형 블록체인 (AMB) 액세스 폴리곤 사용에 대한 고려 사항 및 제한

Amazon Managed Blockchain (AMB) 액세스 폴리곤을 사용할 때는 다음 사항을 고려하십시오.

- 지원되는 폴리곤 네트워크

  AMB 액세스 폴리곤은 다음과 같은 공용 네트워크를 지원합니다.

  - 메인넷 — proof-of-stake 합의에 의해 보호되고 폴리곤 (POL) 토큰이 발행되고 거래되는 퍼블릭 폴리곤 블록체인입니다. 메인넷에서의 거래는 실제 가치를 지니며 (즉, 실제 비용이 발생함) 퍼블릭 블록체인에 기록됩니다.

- Polygon은 더 이상 네트워크를 지원하지 않습니다.

  - Polygon Labs의 발표에 따르면 뭄바이 테스트넷 네트워크는 4월 중순에 종료될 예정입니다. 이 소식에 따라 AMB 액세스 폴리곤은 2024년 4월 15일에 뭄바이 테스트넷 지원을 종료했습니다. 테스트 워크로드에는 Amoy 테스트넷을 사용하는 것이 좋습니다.
  - 사설 네트워크는 지원되지 않습니다.
  - 또한 AMB 액세스 폴리곤에는 폴리곤 ZKevM 네트워크에 대한 지원이 포함되어 있지 않습니다.

- 널리 사용되는 타사 프로그래밍 라이브러리와의 호환성

  AMB Access Polygon은 ethers.js 같은 널리 사용되는 프로그래밍 라이브러리와 호환되므로 개발자는 익숙한 도구를 사용하여 Polygon 블록체인과 상호 작용하여 기존 구현과 쉽게 통합하거나 새 애플리케이션을 빠르게 개발할 수 있습니다.

- 지원되는 리전

  이 서비스는 미국 동부 (버지니아 북부) 지역에서만 지원됩니다.

- Service endpoints

  AMB 액세스 폴리곤의 서비스 엔드포인트는 다음과 같습니다. 서비스에 연결하려면 지원되는 지역 중 하나를 포함하는 엔드포인트를 사용해야 합니다.

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

- 스테이킹은 지원되지 않습니다.

  AMB 액세스 폴리곤은 폴리곤 (POL) 유효성 검사기 노드를 지원하지 않습니다. proof-of-stake

- 폴리곤 JSON-RPC 요청의 시그니처 버전 4 서명

  Amazon Managed Blockchain에서 Polygon JSON-RPC를 호출할 때는 서명 버전 4 서명 프로세스를 사용하여 인증된 HTTPS 연결을 통해 호출할 수 있습니다. 즉, 계정의 승인된 IAM 보안 주체만 Polygon JSON-RPC 호출을 할 수 있습니다. AWS 이렇게 하려면 호출과 함께 AWS 자격 증명 (액세스 키 ID 및 보안 액세스 키) 을 제공해야 합니다.

> **Important**
>
> - 사용자 대상 애플리케이션에는 클라이언트 자격 증명을 내장하지 마십시오.
> - IAM 정책을 사용하여 개별 Polygon JSON-RPC에 대한 액세스를 제한할 수는 없습니다.

- 토큰 기반 액세스 지원

  또한 액세스 토큰을 사용하여 서명 버전 4 (SigV4) 서명 프로세스의 편리한 대안으로 Polygon 네트워크 엔드포인트에 JSON-RPC 호출을 할 수 있습니다. 생성한 `BILLING_TOKEN` Accessor 토큰 중 하나의 토큰을 제공하고 호출 시 매개변수로 추가해야 합니다.

  > **Important**
  >
  > - 편리함보다 보안 및 감사 가능성을 우선시한다면 SigV4 서명 프로세스를 대신 사용하세요.
  > - 시그니처 버전 4 (SigV4) 및 토큰 기반 액세스를 사용하여 Polygon JSON-RPC에 액세스할 수 있습니다. 하지만 두 프로토콜을 모두 사용하기로 선택하면 요청이 거부됩니다.
  > - 사용자 대상 애플리케이션에는 Accessor 토큰을 절대 내장해서는 안 됩니다.

- 원시 트랜잭션의 제출만 지원됩니다.

  `eth_sendrawtransaction`JSON-RPC를 사용하여 Polygon 블록체인 상태를 업데이트하는 트랜잭션을 제출하십시오.

아마존 관리형 블록체인 (AMB) 액세스 폴리곤 설정

Amazon Managed Blockchain (AMB) 액세스 폴리곤을 처음 사용하기 전에 이 섹션의 단계에 따라 폴리곤을 생성하십시오. AWS 계정다음 장에서는 AMB 액세스 폴리곤 사용을 시작하는 방법을 설명합니다.

AMB 액세스 폴리곤을 사용하기 위한 사전 요구 사항

AWS 처음 사용하기 전에 반드시 가지고 있어야 합니다. AWS 계정

가입하기 AWS

AWS가입하면 Amazon Managed Blockchain (AMB) 액세스 폴리곤을 포함한 모든 AWS 서비스항목에 자동으로 AWS 계정 가입됩니다. 사용한 서비스에 대해서만 청구됩니다.

AWS 계정 이미 등록했다면 다음 단계로 넘어가세요. AWS 계정이 없는 경우에는 다음 절차에 따라 계정을 만드세요.

생성하려면 AWS 계정

1. https://portal.aws.amazon.com/billing/signup을 엽니다.
2. 온라인 지시 사항을 따릅니다.

   등록 절차 중 전화를 받고 전화 키패드로 확인 코드를 입력하는 과정이 있습니다.

   에 AWS 계정가입하면 AWS 계정 루트 사용자a가 생성됩니다. 루트 사용자에게는 계정의 모든 AWS 서비스 및 리소스에 액세스할 권한이 있습니다. 보안 모범 사례는 사용자에게 관리 액세스 권한을 할당하고, 루트 사용자만 사용하여 루트 사용자 액세스 권한이 필요한 작업을 수행하는 것입니다.

적절한 권한을 가진 IAM 사용자 생성

AMB Access Polygon을 생성하고 사용하려면 필요한 관리형 블록체인 작업을 허용하는 권한을 가진 AWS Identity and Access Management (IAM) 주체 (사용자 또는 그룹) 가 있어야 합니다.

Amazon Managed Blockchain에서 Polygon JSON-RPC를 호출할 때는 서명 버전 4 서명 프로세스를 사용하여 인증된 HTTPS 연결을 통해 호출할 수 있습니다. 즉, 계정의 승인된 IAM 보안 주체만

Polygon JSON-RPC 호출을 할 수 있습니다. AWS 이렇게 하려면 호출과 함께 AWS 자격 증명 (액세스 키 ID 및 보안 액세스 키) 을 제공해야 합니다.

또한 서명 버전 4 (SigV4) 서명 프로세스의 편리한 대안으로 Accessor 토큰을 사용하여 Polygon 네트워크 엔드포인트에 JSON-RPC를 호출할 수 있습니다. 생성한 `BILLING_TOKEN` Accessor 토큰 중 하나의 토큰을 제공하고 호출 시 매개변수로 추가해야 합니다. 하지만,, SDK를 사용하여 Accessor 토큰을 생성할 권한을 얻으려면 여전히 IAM 액세스가 필요합니다. AWS Management Console AWS CLI

IAM 사용자를 생성하는 방법에 대한 자세한 내용은 계정에서 IAM 사용자 생성을 참조하십시오. AWS 권한 정책을 사용자에게 연결하는 방법에 대한 자세한 내용은 IAM 사용자의 권한 변경을 참조하십시오. 사용자에게 AMB Access Polygon을 사용할 수 있는 권한을 부여하는 데 사용할 수 있는 권한 정책의 예는 을 참조하십시오. 아마존 관리형 블록체인 (AMB) 액세스 폴리곤에 대한 ID 기반 정책 예제

AWS Command Line Interface설치 및 구성

아직 설치하지 않았다면, 터미널의 AWS 리소스로 작업하려면 latest AWS Command Line Interface (AWS CLI) 를 설치하십시오. 자세한 내용은 최신 버전의 AWS CLI설치 또는 업데이트를 참조하세요.

> **Note**
>
> CLI 액세스를 위해서는 액세스 키 ID 및 비밀 액세스 키가 필요합니다. 가능하다면 장기 액세스 키 대신 임시 보안 인증 정보를 사용하세요. 임시 보안 인증도 액세스 키 ID와 비밀 액세스 키로 구성되지만 보안 인증이 만료되는 시간을 나타내는 보안 토큰이 포함되어 있습니다. 자세한 내용은 IAM 사용 설명서의 AWS 리소스와 함께 임시 자격 증명 사용을 참조하십시오.

아마존 관리형 블록체인 (AMB) 액세스 폴리곤 시작하기

이 섹션의 정보와 절차를 사용하여 Amazon Managed Blockchain (AMB) 액세스 폴리곤을 시작하십시오.

주제

- Polygon 블록체인 네트워크에 액세스하기 위한 IAM 정책을 생성하십시오.
- 다음을 사용하여 AMB Access RPC 편집기에서 Polygon 원격 프로시저 호출 (RPC) 요청을 생성합니다. AWS Management Console
- 다음을 사용하여 AMB 액세스 폴리곤 JSON-RPC 요청을 전송하십시오. awscurlAWS CLI
- Node.js 에서 폴리곤 JSON-RPC 요청을 생성하세요.

Polygon 블록체인 네트워크에 액세스하기 위한 IAM 정책을 생성하십시오.

Polygon 메인넷의 퍼블릭 엔드포인트에 액세스하여 JSON-RPC 호출을 수행하려면 Amazon Managed Blockchain (AMBAWS_SECRET_ACCESS_KEY) 액세스 폴리곤에 대한 적절한 IAM 권한이 있는 사용자 자격 증명 (AWS_ACCESS_KEY_ID및) 이 있어야 합니다. AWS CLI 설치된 터미널에서 다음 명령을 실행하여 두 Polygon 엔드포인트에 모두 액세스할 수 있는 IAM 정책을 생성합니다.

```
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**
>
> 이전 예제에서는 사용 가능한 모든 Polygon 네트워크에 액세스할 수 있습니다. 특정 엔드포인트에 액세스하려면 다음 `Action` 명령을 사용하십시오.
>
> - `"managedblockchain:InvokeRpcPolygonMainnet"`

정책을 생성한 후 해당 정책을 IAM 사용자의 역할에 연결하면 정책이 적용됩니다. 에서 AWS Management Console IAM 서비스로 `AmazonManagedBlockchainPolygonAccess` 이동하여 정책을 IAM 사용자에게 할당된 역할에 연결합니다.

다음을 사용하여 AMB Access RPC 편집기에서 Polygon 원격 프로시저 호출 (RPC) 요청을 생성합니다. AWS Management Console

AMB 액세스 폴리곤을 AWS Management Console 사용하여 RPC (원격 프로시저 호출) 를 편집, 구성 및 제출할 수 있습니다. 이러한 RPC를 사용하여 Polygon 네트워크에서 데이터를 읽고 트랜잭션을 쓸 수 있습니다. 여기에는 데이터 검색 및 Polygon 네트워크에 트랜잭션 제출이 포함됩니다.

Example

다음 예제는 RPC를 사용하여 최신 블록에 대한 정보를 가져오는 방법을 보여줍니다. `eth_getBlockByNumber` 강조 표시된 변수를 직접 입력으로 변경하거나 나열된 RPC 방법 중 하나를 선택하고 필요한 관련 입력을 입력합니다.

1. https://console.aws.amazon.com/managedblockchain/ 에서 관리형 블록체인 콘솔을 엽니다.
2. RPC 에디터를 선택합니다.
3. 요청 섹션에서 ***#### POLYGON_MAINNET #####*** 선택합니다.
4. RPC ***eth_getBlockByNumber*** 방법으로 선택하세요.
5. ***## ###*** 입력하고 **`latest`** 전체 트랜잭션 ***`False`*** 플래그로 선택합니다.
6. 그런 다음 RPC 제출을 선택합니다.
7. 응답 섹션에서 `latest` 차단 결과를 확인할 수 있습니다. 그런 다음 전체 원시 트랜잭션을 복사하여 추가 분석을 위해 또는 애플리케이션의 비즈니스 로직에 사용할 수 있습니다.

자세한 내용은 AMB 액세스 폴리곤에서 지원하는 RPC를 참조하십시오.

다음을 사용하여 AMB 액세스 폴리곤 JSON-RPC 요청을 전송하십시오. **awscurl**AWS CLI

Example

AMB 액세스 폴리곤 엔드포인트에 폴리곤 JSON-RPC 요청을 보내려면 서명 버전 4 (SigV4) 를 사용하여 IAM 사용자 자격 증명으로 요청에 서명하십시오. awscurl명령줄 도구를 사용하면 SigV4를 사용하여 서비스에 대한 요청에 서명할 수 있습니다. AWS 자세한 내용은 awscurl README.md를 참조하십시오.

운영 awscurl 체제에 적합한 방법을 사용하여 설치합니다. macOS에서는 HomeBrew 다음과 같은 응용 프로그램을 사용하는 것이 좋습니다.

```
brew install awscurl
```

를 이미 설치하고 구성한 경우 IAM 사용자 자격 증명과 기본 AWS 리전 자격 증명이 환경에 설정되어 액세스할 수 있습니다. AWS CLIawscurl 를 사용하여 awscurl RPC를 호출하여 Polygon 메인넷에 요청을 제출하십시오. eth_getBlockByNumber 이 호출은 정보를 검색하려는 블록 번호에 해당하는 문자열 파라미터를 수락합니다.

다음 명령은 params 배열의 블록 번호를 사용하여 헤더를 검색할 특정 블록을 선택하여 Polygon 메인넷에서 블록 데이터를 검색합니다.

```
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**
>
> 토큰을 사용하는 AMB Access 토큰 기반 액세스 기능을 사용하여 curl 이와 동일한 요청을 할 수도 있습니다. Accessor 자세한 정보는 AMB Access Polygon 요청을 위한 토큰 기반 액세스를 위한 액세스 토큰 생성 및 관리을 참조하세요.
>
> ```
> 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'
> ```

두 명령 중 하나의 응답은 최신 블록에 대한 정보를 반환합니다. 설명을 위해 다음 예를 참조하십시오.


```
{"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 에서 폴리곤 JSON-RPC 요청을 생성하세요.

Node.js 네이티브 https 모듈을 사용하여 Polygon 메인넷 네트워크에 액세스하기 위해 HTTPS를 사용하여 서명된 요청을 제출하거나 AXIOS와 같은 타사 라이브러리를 사용하여 Polygon JSON-RPC를 호출할 수 있습니다. 다음 Node.js 예제는 서명 버전 4 (SigV4) 와 토큰 기반 액세스를 모두 사용하여 AMB 액세스 폴리곤 엔드포인트에 Polygon JSON-RPC 요청을 보내는 방법을 보여줍니다. 첫 번째 예시는 한 주소에서 다른 주소로 거래를 보내고 다음 예시는 블록체인에서 거래 세부 정보와 잔액 정보를 요청합니다.

Example

이 예제 Node.js 스크립트를 실행하려면 다음 사전 요구 사항을 적용하십시오.

1. 시스템에 노드 버전 관리자 (nvm) 와 Node.js 가 설치되어 있어야 합니다. 여기에서 해당 OS의 설치 지침을 찾을 수 있습니다.
2. node --version명령을 사용하여 Node 버전 18 이상을 사용하고 있는지 확인합니다. 필요한 경우 nvm install v18.12.0 명령과 명령을 차례로 사용하여 Node의 nvm use v18.12.0 LTS 버전인 버전 18을 설치할 수 있습니다.
3. 환경 변수에는 AWS_ACCESS_KEY_ID 계정과 관련된 자격 증명이 AWS_SECRET_ACCESS_KEY 포함되어야 합니다.

   다음 명령을 사용하여 클라이언트에서 이러한 변수를 문자열로 내보냅니다. 다음 문자열에서 빨간색 값을 IAM 사용자 계정의 적절한 값으로 바꾸십시오.

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

모든 사전 요구 사항을 완료한 후 선호하는 코드 편집기를 사용하여 다음 파일을 로컬 환경의 디렉터리에 복사하십시오.

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**
>
> 다음 코드는 하드코딩된 개인 키를 사용하여 지갑 서명자를 생성하여 Ethers.js 데모용으로만 사용합니다. 실제 자금이 있고 보안상의 위험이 있으므로 프로덕션 환경에서는 이 코드를 사용하지 마십시오.
> 필요한 경우 계정 팀에 문의하여 지갑 및 서명자 모범 사례에 대해 조언하세요.

```
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");
```

이러한 파일이 디렉터리에 저장되면 다음 명령을 사용하여 코드를 실행하는 데 필요한 종속 항목을 설치합니다.

```
npm install
```

Node.js 파일 내 트랜잭션 전송

위 예제에서는 AMB Access Polygon을 사용하여 트랜잭션에 서명하고 이를 Polygon 메인넷으로 브로드캐스트하여 한 주소에서 다른 주소로 네이티브 Polygon 메인넷 토큰 (POL) 을 보냅니다. 이를 위해서는 이더리움 및 Polygon과 같은 이더리움 호환 블록체인과 상호 작용하는 데 널리 사용되는 라이브러리를 사용하는 `sendTx.js` `Ethers.js` 스크립트를 사용하십시오. 빨간색으로 강조 표시된 코드에서 토큰 기반 액세스를 `billingToken` 위한 Accessor 토큰, 트랜잭션에 서명하는 데 사용한 프라이빗 키, POL을 수신하는 수신자 주소 등 세 가지 변수를 바꿔야 합니다.

> **Tip**
>
> 자금 손실 위험을 없애려면 기존 지갑을 재사용하는 대신 이를 위해 새 개인 키 (지갑) 를 만드는 것이 좋습니다. Ethers 라이브러리의 Wallet 클래스 메서드인 createRandom () 을 사용하여 테스트에 사용할 지갑을 생성할 수 있습니다. 또한 Polygon 메인넷에서 POL을 요청해야 하는 경우 공개 POL 파우셋을 사용하여 테스트에 사용할 소량을 요청할 수 있습니다.

자금이 입금된 지갑의 개인 키와 수신자 주소를 코드에 추가했으면 다음 코드를 실행하여 사용자 `billingToken` 주소에서 다른 주소로 전송되는.0001 POL 트랜잭션에 서명하고 AMB Access Polygon을 사용하여 `eth_sendRawTransaction` JSON-RPC를 호출하는 Polygon 메인넷에 브로드캐스트합니다.

```
node sendTx.js
```

다시 수신된 응답은 다음과 비슷합니다.

```
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: []
}
```

응답은 거래 영수증을 구성합니다. 속성 hash 값을 저장합니다. 이것은 방금 블록체인에 제출한 거래의 식별자입니다. 읽기 트랜잭션 예제에서 이 프로퍼티를 사용하여 Polygon 메인넷에서 이 트랜잭션에 대한 추가 세부 정보를 얻을 수 있습니다.

참고로 blockHash null 응답에는 blockNumber 및 가 있습니다. 이는 거래가 아직 Polygon 네트워크의 블록에 기록되지 않았기 때문입니다. 이러한 값은 나중에 정의되며 다음 섹션에서 트랜잭션 세부 정보를 요청할 때 해당 값이 표시될 수 있습니다.

Node.js 내 트랜잭션 읽기

이 섹션에서는 이전에 제출한 거래에 대한 거래 세부 정보를 요청하고 AMB Access Polygon을 사용하는 Polygon 메인넷에 대한 읽기 요청을 사용하여 수신자 주소의 POL 잔액을 검색합니다. readTx.js파일에서 레이블이 지정된 변수를 이전 섹션의 코드 실행 응답에서 저장한 변수로 *your-transaction-id*바꾸십시오. hash

이 코드는 AWS SDK의 필수 서명 버전 4 (SigV4) 모듈을 사용하여 AMB Access Polygon에 대한 HTTPS 요청에 서명하고 널리 사용되는 HTTP 클라이언트인 AXIOS를 사용하여 요청을 보내는 유틸리티를 사용합니다. dispatch-evm-rpc.js

다시 수신된 응답은 다음과 유사합니다.

```
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
```

응답은 거래 세부 정보를 나타냅니다. 참고로`blockHash`, 이제 `blockNumber` AN이 정의되었을 가능성이 높습니다. 이는 거래가 블록에 기록되었음을 나타냅니다. 이 값이 여전히 `null` 유효하다면 몇 분 정도 기다린 다음 코드를 다시 실행하여 거래가 블록에 포함되었는지 확인하세요. 마지막으로, 수신자 주소 잔액의 16진수 표현 (0x110d9316ec000) 은 Ethers의 `formatEther()` 방법을 사용하여 십진수로 변환됩니다. 이 방법은 16진수를 십진수로 변환하고 소수점 자리를 18 (10^18) 씩 이동하여 POL의 실제 잔액을 산출합니다.

> **Tip**
>
> 위의 코드 예제는 Node.js, Ethers 및 Axios를 사용하여 AMB Access Polygon에서 지원되는 몇 가지 JSON-RPC를 활용하는 방법을 보여 주지만, 이 서비스를 사용하여 예제를 수정하고 Polygon에서 애플리케이션을 빌드하는 다른 코드를 작성할 수 있습니다. AMB 액세스 폴리곤에서 지원되는 JSON-RPC의 전체 목록은 을 참조하십시오. AMB 액세스 폴리곤을 지원하는 관리형 블록체인 API 및 JSON-RPC

AMB Access Polygon 요청을 위한 토큰 기반 액세스를 위한 액세스 토큰 생성 및 관리

또한 서명 버전 4 (SigV4) 서명 프로세스의 편리한 대안으로 Accessor 토큰을 사용하여 Polygon 네트워크 엔드포인트에 JSON-RPC를 호출할 수 있습니다. 생성한 `BILLING_TOKEN` Accessor 토큰 중 하나의 토큰을 제공하고 호출 시 매개변수로 추가해야 합니다.

> **Important**
>
> - 편리함보다 보안 및 감사 가능성을 우선시한다면 SigV4 서명 프로세스를 대신 사용하세요.
> - 시그니처 버전 4 (SigV4) 및 토큰 기반 액세스를 사용하여 Polygon JSON-RPC에 액세스할 수 있습니다. 하지만 두 프로토콜을 모두 사용하기로 선택하면 요청이 거부됩니다.
> - 사용자 대상 애플리케이션에는 Accessor 토큰을 절대 내장해서는 안 됩니다.

콘솔의 토큰 접근자 페이지에는 클라이언트의 발신 코드에서 AMB Access Polygon JSON-RPC를 호출하는 데 사용할 수 있는 모든 접근자 토큰 목록이 표시됩니다. AWS 계정

AMB 액세스 폴리곤 JSON-RPC 요청에 대한 자세한 내용은 을 참조하십시오. AMB 액세스 폴리곤을 지원하는 관리형 블록체인 API 및 JSON-RPC

를 사용하여 접근자 토큰을 생성하고 관리할 수 있습니다. AWS Management Console`CreateAccessor`, `GetAccessorListAccessors`, 및 같은 API 작업을 사용하여 접근자 토큰을 만들고 관리할 수도 있습니다. `DeleteAccessor` A는 `BILLING_TOKEN` 접근자의 속성입니다. 이 `BILLING_TOKEN` 속성은 접근자를 추적하고 사용자의 AMB Access Polygon JSON-RPC 요청에 대한 비용을 청구하는 데 사용됩니다. AWS 계정

접근자 토큰 생성 및 관리와 관련된 모든 API 작업은,, SDK를 통해서도 사용할 수 있습니다. AWS Management Console AWS CLI

토큰 기반 액세스를 위한 접근자 토큰 생성

접근자 토큰을 생성하고 이를 사용하여 내 모든 AMB 액세스 폴리곤 노드에서 AMB 액세스 폴리곤 API를 호출할 수 있습니다. AWS 계정

다음을 사용하여 AMB 액세스 폴리곤 JSON-RPC 요청을 할 수 있는 접근자 토큰을 생성하십시오. AWS Management Console

1. https://console.aws.amazon.com/managedblockchain/ 에서 관리형 블록체인 콘솔을 엽니다.
2. 토큰 접근자를 선택합니다.
3. 접근자 생성을 선택합니다.
4. 유효한 Polygon 블록체인 네트워크를 선택하세요.
5. 선택사항, 접근자에 대한 태그를 추가하세요.
6. 접근자 생성을 선택하여 새 접근자 토큰을 생성합니다.

다음을 사용하여 AMB 액세스 폴리곤 JSON-RPC 요청을 만들려면 접근자 토큰을 생성하십시오. AWS CLI

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

이전 명령은 다음 예와 같이 와 `AccessorId` 함께 `BillingToken` 를 반환합니다.

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

응답의 핵심 요소는 다음과 같습니다`BillingToken`. 이 속성을 사용하여 AMB 액세스 폴리곤 JSON-RPC 호출을 수행할 수 있습니다. 예제의 일부 값은 보안상의 이유로 난독화되었지만 실제 응답에는 완전히 표시됩니다.

> **Note**
>
> 작업이 실행되면 관리형 블록체인이 사용자를 대신하여 토큰을 프로비저닝하고 구성합니다. 이 프로세스의 길이는 여러 변수에 따라 달라집니다.

접근자 토큰 세부 정보 보기

소유하고 있는 각 접근자 토큰의 속성을 볼 수 있습니다. AWS 계정 예를 들어, 접근자의 접근자 ID 또는 Amazon 리소스 이름 (ARN) 을 볼 수 있습니다. 상태, 유형, 생성 날짜 및 기간도 볼 수 있습니다. `BillingToken`

를 사용하여 접근자 토큰의 정보를 보려면 AWS Management Console

1. https://console.aws.amazon.com/managedblockchain/ 에서 관리형 블록체인 콘솔을 엽니다.
2. 탐색 창에서 토큰 접근자를 선택합니다.
3. 목록에서 토큰의 접근자 ID를 선택합니다.

그러면 토큰 세부 정보 페이지가 나타납니다. 이 페이지에서 토큰의 속성을 볼 수 있습니다.

를 사용하여 접근자 토큰의 정보를 보려면 AWS CLI

다음 명령을 실행하여 접근자 토큰의 세부 정보를 확인합니다. 의 값을 접근자 `--accessor-id` ID로 바꾸십시오.

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

`BillingToken`및 기타 키 속성은 다음 예와 같이 반환됩니다. 예제의 일부 값은 보안상의 이유로 난독화되었지만 실제 응답에는 완전히 나타납니다.

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

접근자 토큰 삭제

접근자 토큰을 삭제하면 토큰이 에서 AVAILABLE PENDING_DELETION 상태로 바뀝니다. 상태에는 접근자 토큰을 PENDING_DELETION 사용할 수 없습니다.

를 사용하여 접근자 토큰을 삭제하려면 AWS Management Console

1. https://console.aws.amazon.com/managedblockchain/ 에서 관리형 블록체인 콘솔을 엽니다.
2. 탐색 창에서 토큰 접근자를 선택합니다.
3. 목록에서 원하는 접근자 토큰을 선택합니다.
4. 삭제를 선택합니다.
5. 선택 내용을 확인합니다.

삭제된 접근자 토큰과 함께 토큰 접근자 페이지로 돌아갑니다. 페이지에 상태가 표시됩니다. PENDING_DELETION

를 사용하여 접근자 토큰을 삭제하려면 AWS CLI

다음 예제는 토큰을 삭제하는 방법을 보여줍니다. delete-accessor명령을 사용하여 토큰을 삭제합니다. 접근자 --accessor-id ID로 값을 설정합니다.

CLI를 사용하여 접근자 AWS 토큰 삭제

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

이 명령이 성공적으로 실행되면 메시지가 반환되지 않습니다.

AMB 액세스 폴리곤을 지원하는 관리형 블록체인 API 및 JSON-RPC

Amazon Managed Blockchain은 AMB 액세스 폴리곤에 대한 토큰 접근자를 생성하고 관리하기 위한 API 작업을 제공합니다. 자세한 내용은 관리형 블록체인 API 참조 안내서를 참조하십시오.

다음 항목은 AMB 액세스 폴리곤이 지원하는 폴리곤 JSON-RPC의 목록과 참조를 제공합니다. 지원되는 각 JSON-RPC에는 용도에 대한 간략한 설명이 있습니다. Polygon JSON-RPC를 사용하여 스마트 컨트랙트 데이터를 쿼리 및 가져오고, 트랜잭션 세부 정보를 가져오고, 트랜잭션을 제출하고, 트랜잭션 추적 실행, 수수료 추정 등의 기타 유틸리티를 사용할 수 있습니다.

AMB 액세스 폴리곤은 다음과 같은 JSON-RPC 메서드를 지원합니다. 지원되는 각 JSON-RPC에는 해당 유틸리티와 기본 요청 할당량에 대한 범주와 간략한 설명이 있습니다. Amazon Managed Blockchain과 함께 JSON-RPC 방법을 사용할 때 고려해야 할 고유한 고려 사항이 해당하는 경우 명시되어 있습니다.

Note

- 목록에 없는 메서드는 지원되지 않습니다.
- Amazon Managed Blockchain에서 Polygon JSON-RPC를 호출할 때는 서명 버전 4 서명 프로세스를 사용하여 인증된 HTTPS 연결을 통해 호출할 수 있습니다. 즉, 계정의 승인된 IAM 보안 주체만 Polygon JSON-RPC 호출을 할 수 있습니다. AWS 이렇게 하려면 호출과 함께 AWS 자격 증명 (액세스 키 ID 및 보안 액세스 키) 을 제공해야 합니다.
- 서명 버전 4 (SigV4) 서명 프로세스의 편리한 대안으로 토큰 기반 액세스를 사용할 수도 있습니다. 편리함보다 보안 및 감사 가능성을 우선시한다면 SigV4 서명 프로세스를 대신 사용하십시오. 하지만 SigV4와 토큰 기반 액세스를 모두 사용하는 경우 요청이 작동하지 않습니다.
- 이 프리뷰에서는 아마존 관리형 블록체인 (AMB) 액세스 폴리곤에서 JSON-RPC 배치 요청이 지원되지 않습니다.
- 다음 표의 할당량 열에는 각 JSON-RPC의 할당량이 나열되어 있습니다. 할당량은 각 JSON-RPC의 지역별 초당 요청 수 (RPS) /폴리곤 네트워크 (메인넷) 로 설정됩니다.

  할당량을 늘리려면 문의해야 합니다. AWS Support AWS Support문의하려면 로그인하십시오 AWS Support Center Console. 사례 생성을 선택합니다. 테크니컬을 선택합니다. 관리형 블록체인을 서비스로 선택하세요. 카테고리로 Access:Polygon을 선택하고 심각도로는 일반 지침을 선택하세요. 제목으로 RPC 할당량을 입력하고 설명 텍스트 상자에 JSON-RPC와 필

요에 맞는 할당량 한도를 지역별 Polygon 네트워크당 RPS로 나열하십시오. 사례를 제출하세요.

범주	JSON-RPC	설명	고려 사항
이더리움	ETH_블록 번호	가장 최근 블록의 수를 반환합니다.	
	eth_call	블록체인에서 트랜잭션을 생성하지 않고 새 메시지 호출을 즉시 실행합니다.	`eth_call`가스를 전혀 소비하지 않지만, 가스를 필요로 하는 메시지에는 가스 파라미터가 있습니다.
	ETH_ChainID	EIP-155 내에 도입된 현재 구성된 값에 대한 정수 `Chain Id` 값을 반환합니다. 사용할 수 없는 `None` `Chain Id` 경우 반환합니다.	
	ETH_ESTIMATEGAS	트랜잭션을 블록체인에 추가하지 않고도 거래에 필요한 가스를 추정하고 반환합니다.	
	ETH_FEE/내역	과거 가스 정보 모음을 반환합니다.	

범주	JSON-RPC	설명	고려 사항
	ETH_GasPrice	가스당 현재 가격을 Wei 단위로 반환합니다.	
	ETH_GetBalance	지정된 계정 주소 및 블록 식별자에 대한 계정 잔액을 반환합니다.	
	eth_get 해시 BlockBy	블록 해시를 사용하여 지정된 블록에 대한 정보를 반환합니다.	
	eth_get 번호 BlockBy	블록 번호를 사용하여 지정된 블록에 대한 정보를 반환합니다.	
	eth_get BlockReceipts	블록 번호를 사용하여 지정된 블록에 대한 영수증을 반환합니다.	
	eth_get 해시 BlockTransaction CountBy	블록 해시를 사용하여 지정된 블록의 트랜잭션 수를 반환합니다.	
	eth_get 번호 BlockTransaction CountBy	블록 번호를 사용하여 지정된 블록의 트랜잭션 수를 반환합니다.	

범주	JSON-RPC	설명	고려 사항
	ETH_GetCode	지정된 계정 주소 및 블록 식별자의 코드를 반환합니다.	
	ETH_GetLogs	지정된 필터 개체에 대한 모든 로그의 배열을 반환합니다.	계약 주소가 제공되면 기본적으로 1K 블록 범위를 갖는 모든 블록 범위에서 `eth_getloqs` 요청할 수 있습니다. 활동이 많은 계약은 더 작은 블록 범위로 제한될 수 있습니다. 계약 주소가 제공되지 않는 경우 블록 범위는 8이 됩니다.
	eth_get RawTransaction ByHash	에서 지정한 원시 형태의 트랜잭션을 반환합니다. `transacti on_hash`	
	eth_get StorageAt	지정된 계정 주소 및 블록 식별자에 대해 지정된 저장 위치의 값을 반환합니다.	

범주	JSON-RPC	설명	고려 사항
	eth_get TransactionBy BlockHash AndIndex	지정된 블록 해시와 트랜잭션 인덱스 위치를 사용하여 트랜잭션에 대한 정보를 반환합니다.	
	eth_get TransactionBy BlockNumber AndIndex	지정된 블록 번호와 트랜잭션 인덱스 위치를 사용하여 트랜잭션에 대한 정보를 반환합니다.	
	eth_get 해시 TransactionBy	지정된 트랜잭션 해시가 있는 트랜잭션에 대한 정보를 반환합니다.	
	eth_get TransactionCount	지정된 주소 및 블록 식별자로부터 전송된 트랜잭션 수를 반환합니다.	
	eth_get TransactionReceipt	지정된 트랜잭션 해시를 사용하여 트랜잭션 수신을 반환합니다.	
	eth_get UncleBy BlockHash AndIndex	블록 해시와 엉클 인덱스 위치를 사용하여 지정된 엉클 블록에 대한 정보를 반환합니다.	

범주	JSON-RPC	설명	고려 사항
	eth_get UncleBy BlockNumber AndIndex	블록 번호와 엉클 인덱스 위치를 사용하여 지정된 엉클 블록에 대한 정보를 반환합니다.	
	eth_get 해시 UncleCount ByBlock	엉클 해시를 사용하여 지정된 아저씨의 카운트 수를 반환합니다.	
	eth_get 넘버 UncleCount ByBlock	삼촌 번호를 사용하여 지정된 삼촌의 카운트 수를 반환합니다.	
	eth_max PriorityFee PerGas	현재 블록에 거래를 포함시키기 위해 우선 수수료 또는 “팁”으로 지불할 수 있는 금액의 추정치인 가스당 수수료를 반환합니다.	일반적으로 이 메서드에서 반환된 값을 사용하여 제출하는 후속 `maxFeePerGas` 거래에서 값을 설정합니다.
	ETH_프로토콜/버전	현재 이더리움 프로토콜 버전을 반환합니다.	

범주	JSON-RPC	설명	고려 사항
	eth_send RawTransaction	서명된 트랜잭션에 대한 새 메시지 호출 트랜잭션 또는 컨트랙트 생성을 생성합니다.	관리형 블록체인은 원시 트랜잭션만 지원합니다. 트랜잭션을 전송하기 전에 트랜잭션을 생성하고 서명해야 합니다.
디버그	디버그_트레이스 해시 BlockBy	블록 해시로 지정된 블록의 모든 트랜잭션을 추적기로 실행하여 가능한 추적 결과 수를 반환합니다 (Trace Mode 필요).	
	디버그_트레이스 번호 BlockBy	숫자로 지정된 블록의 모든 트랜잭션을 추적기로 실행하여 추적 결과를 반환합니다 (추적 모드 필요).	
	디버그_트레이스콜	지정된 블록 실행 컨텍스트 내에서 eth 호출을 실행하여 가능한 추적 결과 수를 반환합니다 (추적 모드 필요).	

범주	JSON-RPC	설명	고려 사항
	Debug_TraceTransaction은	지정된 트랜잭션의 모든 트레이스를 반환합니다 (트레이스 모드 필요).	
Net	넷_버전	현재 네트워크 ID를 반환합니다.	
추적	트레이스_블록	블록에 포함된 모든 트랜잭션의 호출된 모든 opcode에 대한 전체 스택 트레이스를 반환합니다.	
	trace_call	지정된 블록 실행 컨텍스트 내에서 eth 호출을 실행하여 가능한 추적 결과 수를 반환합니다 (추적 모드 필요).	
	trace_transaction	지정된 트랜잭션의 모든 트레이스를 반환합니다 (트레이스 모드 필요).	
Tx 풀	txpool_content	보류 중인 모든 트랜잭션과 대기 중인 트랜잭션을 반환합니다.	

범주	JSON-RPC	설명	고려 사항
	txpool_status	현재 다음 블록에 포함되도록 보류 중인 모든 트랜잭션과 대기 중인 트랜잭션(향후 실행에만 스케줄링됨) 의 수를 제공합니다.	
웹	Web3_ClientVersion	현재 클라이언트 버전을 반환합니다.	

아마존 관리형 블록체인 (AMB) 액세스 폴리곤을 사용한 폴리곤 사용 사례

Polygon 블록체인은 NFT, Web3 게임, 토큰화 사용 사례 등과 관련된 분산형 애플리케이션 (dApp) 을 구축하는 데 주로 사용됩니다. 이 주제에서는 Amazon Managed Blockchain (AMB) 액세스 폴리곤을 사용하여 구현할 수 있는 몇 가지 사용 사례 목록을 제공합니다.

주제

폴리곤 NFT 데이터 분석

전송 이벤트 및 지정된 기간 동안의 NFT 메타데이터와 같은 정보를 포함하여 Polygon NFT에 대한 데이터를 수집할 수 있습니다. 그런 다음 이 데이터를 분석하여 어떤 NFT가 유행하고 있는지 또는 어떤 사용자가 특정 컬렉션과 가장 자주 상호작용하는지와 같은 통찰력을 얻을 수 있습니다.

자세한 정보는 AMB 액세스 폴리곤을 지원하는 관리형 블록체인 API 및 JSON-RPC을 참조하세요.

NFT 구매 지원

AMB Access Polygon을 사용하여 이니셜 민트, 허용 목록 또는 2차 시장에서 NFT 구매 거래를 제출할 수 있습니다. 다른 AWS 서비스를 함께 사용하면 신용 카드를 사용한 구매, 법정화폐 또는 암호화폐를 이용한 구매를 허용하고 관련된 모든 이해 관계자가 신속하게 결제할 수 있습니다.

자세한 정보는 AMB 액세스 폴리곤을 지원하는 관리형 블록체인 API 및 JSON-RPC을 참조하세요.

폴리곤 지갑 만들기

AMB Access Polygon을 사용하면 블록체인의 스마트 계약에서 사용자 토큰 잔고를 읽거나 서명된 거래를 블록체인으로 브로드캐스팅하는 등 디지털 자산 지갑의 중요한 기능을 수행할 수 있습니다.

자세한 정보는 AMB 액세스 폴리곤을 지원하는 관리형 블록체인 API 및 JSON-RPC을 참조하세요.

서비스형 지갑

AMB Access Polygon을 사용하면 지원되는 Polygon JSON-RPC를 사용하여 잔액 확인, 자산 이전, 자산 전송, 수수료 추정 등 일반적인 지갑 거래를 지원하는 wallet-as-a-service 데 필요한 작업을 개발할 수 있습니다.

자세한 정보는 AMB 액세스 폴리곤을 지원하는 관리형 블록체인 API 및 JSON-RPC을 참조하세요.

토큰 게이트 경험

AMB Access Polygon을 사용하여 사용자를 위한 토큰 게이트 경험을 구축할 수 있습니다. 예를 들어 특정 NFT의 소유자에게만 콘텐츠에 대한 액세스를 조건부로 제공할 수 있습니다. 이를 위해서는 블록체인을 읽고 사용자 주소의 NFT 소유권을 확인해야 합니다.

자세한 내용은 AMB 액세스 폴리곤을 지원하는 관리형 블록체인 API 및 JSON-RPC을(를) 참조하세요.

아마존 관리형 블록체인 (AMB) 액세스 폴리곤 튜토리얼

이 섹션에서 강조 표시된 다음 자습서는 AMB Access Polygon을 사용하여 Polygon 블록체인에서 AWS re:Post 몇 가지 일반적인 작업을 수행하는 방법을 배우는 데 도움이 되는 안내를 제공하는 커뮤니티 기사입니다.

- AMB 액세스 폴리곤 및 web3.js 를 사용하여 트랜잭션 전송
- AMB 액세스 폴리곤과 하드햇 이그니션을 사용하여 스마트 컨트랙트를 배포하세요.
- 스마트 컨트랙트와의 상호작용
- AMB Access Polygon 및 Chainlink 데이터 피드를 사용하여 오프체인에서 현재 가격 데이터를 검색합니다.
- AMB 액세스를 사용하여 폴리곤 메인넷의 ERC-20 토큰 데이터를 분석하십시오.

아마존 관리형 블록체인 (AMB) 액세스 폴리곤의 보안

클라우드 AWS 보안은 최우선 과제입니다. AWS 고객은 가장 보안에 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 혜택을 누릴 수 있습니다.

보안은 기업과 귀사 간의 공동 책임입니다. AWS 공동 책임 모델은 이를 클라우드의 보안과 클라우드에서의 보안 모두로 설명합니다.

- 클라우드 보안 - AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호하는 역할을 합니다 AWS 클라우드. AWS 또한 안전하게 사용할 수 있는 서비스를 제공합니다. 서드 파티 감사자는 정기적으로 AWS 규정 준수 프로그램의 일환으로 보안 효과를 테스트하고 검증합니다. Amazon Managed Blockchain (AMB) Access Polygon에 적용되는 규정 준수 프로그램에 대해 알아보려면 규정 준수 프로그램별 범위 내AWS 서비스를 참조하십시오.
- 클라우드에서의 보안 — 귀하의 책임은 사용하는 AWS 서비스에 따라 결정됩니다. 또한 사용자는 데이터의 민감도, 회사 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

Amazon Managed Blockchain은 데이터 보호, 인증 및 액세스 제어를 제공하기 위해 관리형 블록체인에서 실행되는 오픈 소스 프레임워크의 특징과 AWS 특징을 사용합니다.

이 설명서는 AMB Access Polygon을 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 항목에서는 보안 및 규정 준수 목표를 충족하도록 AMB Access Polygon을 구성하는 방법을 보여줍니다. 또한 AMB Access Polygon 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법도 알아봅니다.

주제

- 아마존 관리형 블록체인 (AMB) 액세스 폴리곤의 데이터 보호
- 아마존 매니지드 블록체인 (AMB) 액세스 폴리곤에 대한 ID 및 액세스 관리

아마존 관리형 블록체인 (AMB) 액세스 폴리곤의 데이터 보호

AWS 공동 책임 모델 공유 책임 모델 이 모델에 설명된 대로 은 (AWS 는) 모두를 실행하는 글로벌 인프라를 보호할 책임이 있습니다. AWS 클라우드사용자는 인프라에서 호스팅되는 콘텐츠를 관리해야 합니다. 사용하는 AWS 서비스 의 보안 구성과 관리 작업에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 데이터 프라이버시 FAQ를 참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 AWS 공동 책임 모델 및 GDPR 블로그 게시물을 참조하십시오.

데이터 보호를 위해 AWS 계정 자격 증명을 보호하고 AWS IAM Identity Center OR AWS Identity and Access Management (IAM) 을 사용하여 개별 사용자를 설정하는 것이 좋습니다. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 멀티 팩터 인증 설정(MFA)을 사용하세요.
- SSL/TLS를 사용하여 리소스와 통신하세요. AWS TLS 1.2는 필수이며 TLS 1.3를 권장합니다.
- 를 사용하여 API 및 사용자 활동 로깅을 설정합니다. AWS CloudTrail
- 포함된 모든 기본 보안 제어와 함께 AWS 암호화 솔루션을 사용하십시오 AWS 서비스.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용하세요.
- 명령줄 인터페이스 또는 API를 AWS 통해 액세스할 때 FIPS 140-2로 검증된 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 사용하십시오. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 Federal Information Processing Standard(FIPS) 140-2를 참조하십시오.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 양식 필드에 입력하지 않는 것이 좋습니다. 여기에는 콘솔, API 또는 SDK를 AWS 서비스 사용하여 AMB Access Polygon 또는 기타 작업을 수행하는 경우가 포함됩니다. AWS CLI AWS 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 보안 인증 정보를 URL에 포함해서는 안 됩니다.

데이터 암호화

데이터 암호화는 권한이 없는 사용자가 블록체인 네트워크 및 관련 데이터 스토리지 시스템에서 데이터를 읽는 것을 방지하는 데 도움이 됩니다. 여기에는 네트워크를 이동할 때 가로챌 수 있는 데이터 (전송 중인 데이터) 가 포함됩니다.

전송 중 암호화

기본적으로 Managed Blockchain은 HTTPS/TLS 연결을 사용하여 서비스 엔드포인트를 실행하는 클라이언트 컴퓨터에서 전송되는 모든 데이터를 암호화합니다. AWS CLI AWS

HTTPS/TLS 사용을 활성화하기 위해 어떤 조치도 필요하지 않습니다. 명령을 사용하여 개별 명령에 대해 명시적으로 비활성화하지 않는 한 항상 활성화됩니다. AWS CLI `--no-verify-ssl`

아마존 매니지드 블록체인 (AMB) 액세스 폴리곤에 대한 ID 및 액세스 관리

AWS Identity and Access Management (IAM) 은 관리자가 리소스에 대한 액세스를 안전하게 제어할 수 AWS 서비스 있도록 AWS 도와줍니다. IAM 관리자는 AMB Access Polygon 리소스를 사용할 수 있는 인증 (로그인) 및 권한 부여 (권한 보유) 를 받을 수 있는 사용자를 제어합니다. IAM은 추가 비용 AWS 서비스 없이 사용할 수 있습니다.

주제

고객

AMB 액세스 폴리곤에서 수행하는 작업에 따라 사용 방법 AWS Identity and Access Management (IAM) 이 다릅니다.

서비스 사용자 — AMB Access Polygon 서비스를 사용하여 작업을 수행하는 경우 관리자가 필요한 자격 증명과 권한을 제공합니다. 더 많은 AMB Access Polygon 기능을 사용하여 작업을 수행함에 따라 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방식을 이해하면 적절한 권한을 관리자에게 요청할 수 있습니다. AMB 액세스 폴리곤의 기능에 액세스할 수 없는 경우 을 참조하십시오. 아마존 관리형 블록체인 (AMB) 액세스 폴리곤 ID 및 액세스 문제 해결

서비스 관리자 — 회사에서 AMB 액세스 폴리곤 리소스를 담당하는 경우 AMB 액세스 폴리곤에 대한 전체 액세스 권한이 있을 것입니다. 서비스 사용자가 액세스해야 하는 AMB Access Polygon 기능 및 리소스를 결정하는 것은 여러분의 몫입니다. 그런 다음, IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경해야 합니다. 이 페이지의 정보를 검토하여 IAM의 기본 개념을 이해하십시오. 회사에서 AMB 액세스 폴리곤과 함께 IAM을 사용하는 방법에 대한 자세한 내용은 을 참조하십시오. 아마존 매니지드 블록체인 (AMB) 액세스 폴리곤이 IAM과 연동되는 방식

IAM 관리자 — IAM 관리자라면 AMB Access Polygon에 대한 액세스를 관리하기 위한 정책을 작성하는 방법에 대해 자세히 알고 싶을 것입니다. IAM에서 사용할 수 있는 AMB 액세스 폴리곤 ID 기반 정책

의 예를 보려면 을 참조하십시오. 아마존 관리형 블록체인 (AMB) 액세스 폴리곤에 대한 ID 기반 정책 예제

ID를 통한 인증

인증은 ID 자격 증명을 사용하여 로그인하는 AWS 방법입니다. IAM 사용자로 인증 (로그인 AWS) 하거나 IAM 역할을 맡아 인증 (로그인) 해야 합니다. AWS 계정 루트 사용자

ID 소스를 통해 제공된 자격 증명을 사용하여 페더레이션 ID로 로그인할 수 있습니다. AWS AWS IAM Identity Center (IAM ID 센터) 사용자, 회사의 싱글 사인온 인증, Google 또는 Facebook 자격 증명이 페더레이션 ID의 예입니다. 페더레이션 ID로 로그인할 때 관리자가 이전에 IAM 역할을 사용하여 ID 페더레이션을 설정했습니다. 페더레이션을 사용하여 액세스하는 경우 AWS 간접적으로 역할을 맡게 됩니다.

사용자 유형에 따라 AWS Management Console 또는 AWS 액세스 포털에 로그인할 수 있습니다. 로그인에 대한 자세한 내용은 AWS 로그인 사용 설명서의 내 로그인 방법을 참조하십시오. AWS AWS 계정

AWS 프로그래밍 방식으로 액세스하는 경우 자격 증명을 사용하여 요청에 암호화 방식으로 서명할 수 있는 소프트웨어 개발 키트 (SDK) 와 명령줄 인터페이스 (CLI) 를 AWS 제공합니다. AWS 도구를 사용하지 않는 경우 요청에 직접 서명해야 합니다. 권장 방법을 사용하여 직접 요청에 서명하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 AWS API 요청 서명을 참조하십시오.

사용하는 인증 방법에 상관없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어, AWS 계정의 보안을 강화하기 위해 다단계 인증 (MFA) 을 사용할 것을 권장합니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서의 다중 인증 및 IAM 사용 설명서의 AWS에서 다중 인증(MFA) 사용을 참조하십시오.

AWS 계정 루트 사용자

계정을 AWS 계정만들 때는 먼저 계정의 모든 AWS 서비스 리소스에 대한 완전한 액세스 권한을 가진 하나의 로그인 ID로 시작합니다. 이 ID를 AWS 계정 루트 사용자라고 하며, 계정을 만들 때 사용한 이메일 주소와 비밀번호로 로그인하여 액세스할 수 있습니다. 일상적인 태스크에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 보안 인증 정보를 보호하고 루트 사용자만 수행할 수 있는 태스크를 수행하는 데 사용하세요. 루트 사용자로 로그인해야 하는 전체 작업 목록은 IAM 사용 설명서의 루트 사용자 보안 인증이 필요한 작업을 참조하십시오.

페더레이션 자격 증명

가장 좋은 방법은 관리자 액세스가 필요한 사용자를 비롯한 수동 AWS 서비스 사용자가 ID 공급자와의 페더레이션을 사용하여 임시 자격 증명을 사용하여 액세스하도록 하는 것입니다.

페더레이션 ID는 기업 사용자 디렉토리, 웹 ID 공급자, Identity Center 디렉터리의 사용자 또는 ID 소스를 통해 제공된 자격 증명을 사용하여 액세스하는 AWS 서비스 모든 사용자를 말합니다. AWS Directory Service페더레이션 ID에 AWS 계정액세스하면 이들이 역할을 맡고 역할은 임시 자격 증명을 제공합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center(을)를 사용하는 것이 좋습니다. IAM Identity Center에서 사용자 및 그룹을 생성하거나 자체 ID 소스의 사용자 및 그룹 집합에 연결하고 동기화하여 모든 사용자 및 애플리케이션에서 사용할 수 있습니다. AWS 계정 IAM Identity Center에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서에서 IAM Identity Center란 무엇입니까?를 참조하십시오.

IAM 사용자 및 그룹

IAM 사용자는 단일 사용자 또는 애플리케이션에 대한 특정 권한을 AWS 계정 가진 사용자 내 자격 증명입니다. 가능하면 암호 및 액세스 키와 같은 장기 보안 인증이 있는 IAM 사용자를 생성하는 대신 임시 보안 인증을 사용하는 것이 좋습니다. 하지만 IAM 사용자의 장기 보안 인증이 필요한 특정 사용 사례가 있는 경우, 액세스 키를 교체하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 장기 보안 인증이 필요한 사용 사례의 경우 정기적으로 액세스 키 교체를 참조하십시오.

IAM 그룹은 IAM 사용자 컬렉션을 지정하는 자격 증명입니다. 사용자는 그룹으로 로그인할 수 없습니다. 그룹을 사용하여 여러 사용자의 권한을 한 번에 지정할 수 있습니다. 그룹을 사용하면 대규모 사용자 집합의 권한을 더 쉽게 관리할 수 있습니다. 예를 들어, IAMAdmins라는 그룹이 있고 이 그룹에 IAM 리소스를 관리할 권한을 부여할 수 있습니다.

사용자는 역할과 다릅니다. 사용자는 한 사람 또는 애플리케이션과 고유하게 연결되지만, 역할은 해당 역할이 필요한 사람이라면 누구나 수임할 수 있습니다. 사용자는 영구적인 장기 보안 인증 정보를 가지고 있지만, 역할은 임시 보안 인증만 제공합니다. 자세한 내용은 IAM 사용 설명서의 IAM 사용자를 만들어야 하는 경우(역할이 아님)를 참조하십시오.

IAM 역할

IAM 역할은 특정 권한을 가진 사용자 AWS 계정 내의 자격 증명입니다. IAM 사용자와 유사하지만, 특정 개인과 연결되지 않습니다. 역할을 AWS Management Console 전환하여 에서 일시적으로 IAM 역할을 맡을 수 있습니다. AWS CLI 또는 AWS API 작업을 호출하거나 사용자 지정 URL을 사용하여 역할을 수임할 수 있습니다. 역할 사용 방법에 대한 자세한 내용은 IAM 사용 설명서의 IAM 역할 사용을 참조하십시오.

임시 보안 인증이 있는 IAM 역할은 다음과 같은 상황에서 유용합니다.

- 페더레이션 사용자 액세스 - 페더레이션 ID에 권한을 부여하려면 역할을 생성하고 해당 역할의 권한을 정의합니다. 페더레이션 ID가 인증되면 역할이 연결되고 역할에 정의된 권한이 부여됩니다. 페더레이션 역할에 대한 자세한 내용은 IAM 사용 설명서의 서드 파티 ID 공급자의 역할 생성 단원을 참조하십시오. IAM Identity Center를 사용하는 경우, 권한 집합을 구성합니다. 인증 후 ID가 액세스할 수 있는 항목을 제어하기 위해 IAM Identity Center는 권한 세트를 IAM의 역할과 연관짓습니다. 권한 세트에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 권한 세트를 참조하십시오.
- 임시 IAM 사용자 권한 - IAM 사용자 또는 역할은 IAM 역할을 수임하여 특정 태스크에 대한 다양한 권한을 임시로 받을 수 있습니다.
- 크로스 계정 액세스 - IAM 역할을 사용하여 다른 계정의 사용자(신뢰할 수 있는 보안 주체)가 내 계정의 리소스에 액세스하도록 허용할 수 있습니다. 역할은 계정 간 액세스를 부여하는 기본적인 방법입니다. 그러나 일부 AWS 서비스경우에는 역할을 프록시로 사용하는 대신 정책을 리소스에 직접 연결할 수 있습니다. 계정 간 액세스에 대한 역할과 리소스 기반 정책의 차이점을 알아보려면 IAM 사용 설명서의 IAM의 교차 계정 리소스 액세스를 참조하십시오.
- 서비스 간 액세스 — 일부는 다른 기능을 사용합니다. AWS 서비스 AWS 서비스예를 들어 서비스에서 직접적 호출을 수행하면 일반적으로 해당 서비스는 Amazon EC2에서 애플리케이션을 실행하거나 Amazon S3에 객체를 저장합니다. 서비스는 직접적으로 호출하는 보안 주체의 권한을 사용하거나, 서비스 역할을 사용하거나, 또는 서비스 연결 역할을 사용하여 이 태스크를 수행할 수 있습니다.
  - 순방향 액세스 세션 (FAS) — IAM 사용자 또는 역할을 사용하여 작업을 수행하는 경우 보안 AWS 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는 전화를 거는 주체의 권한을 다운스트림 AWS 서비스서비스에 AWS 서비스 요청하기 위한 요청과 결합하여 사용합니다. FAS 요청은 다른 서비스 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 서비스가 수신한 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하세요.
  - 서비스 역할 - 서비스 역할은 서비스가 사용자를 대신하여 태스크를 수행하기 위해 맡는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 AWS 서비스에 대한 권한을 위임할 역할 생성을 참조하십시오.
  - 서비스 연결 역할 — 서비스 연결 역할은 에 연결된 서비스 역할의 한 유형입니다. AWS 서비스서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 사용자에게 AWS 계정 표시되며 해당 서비스가 소유합니다. IAM 관리자는 서비스 링크 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.
- Amazon EC2에서 실행되는 애플리케이션 — IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 API 요청을 AWS CLI 하는 애플리케이션의 임시 자격 증명을 관리할 수 있습니다. AWS 이는 EC2 인스턴스 내에 액세스 키를 저장할 때 권장되는 방법입니다. EC2 인스턴스에 AWS 역할을 할당하고 모든 애플리케이션에서 사용할 수 있게 하려면 인스턴스에 연결된 인스턴스 프로필을 생성합니다. 인

스턴스 프로파일에는 역할이 포함되어 있으며 EC2 인스턴스에서 실행되는 프로그램이 임시 보안 인증을 얻을 수 있습니다. 자세한 내용은 IAM 사용 설명서의 IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여를 참조하십시오.

IAM 역할을 사용할지 또는 IAM 사용자를 사용할지를 알아보려면 IAM 사용 설명서의 IAM 역할(사용자 대신)을 생성하는 경우를 참조하십시오.

정책을 사용한 액세스 관리

정책을 생성하고 이를 AWS ID 또는 리소스에 AWS 연결하여 액세스를 제어할 수 있습니다. 정책은 ID 또는 리소스와 연결될 때 AWS 해당 권한을 정의하는 객체입니다. AWS 주도자 (사용자, 루트 사용자 또는 역할 세션) 가 요청할 때 이러한 정책을 평가합니다. 정책에서 권한은 요청이 허용되거나 거부되는 지를 결정합니다. 대부분의 정책은 JSON 문서로 AWS 저장됩니다. JSON 정책 문서의 구조와 콘텐츠에 대한 자세한 내용은 IAM 사용 설명서의 JSON 정책 개요를 참조하십시오.

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

기본적으로, 사용자와 역할에는 어떠한 권한도 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 수임할 수 있습니다.

IAM 정책은 작업을 수행하기 위해 사용하는 방법과 상관없이 작업에 대한 권한을 정의합니다. 예를 들어, `iam:GetRole` 작업을 허용하는 정책이 있다고 가정합니다. 해당 정책을 사용하는 사용자는 AWS Management Console, AWS CLI, 또는 AWS API에서 역할 정보를 가져올 수 있습니다.

보안 인증 기반 정책

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 태스크를 수행할 수 있는지를 제어합니다. ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성을 참조하십시오.

보안 인증 기반 정책은 인라인 정책 또는 관리형 정책으로 한층 더 분류할 수 있습니다. 인라인 정책은 단일 사용자, 그룹 또는 역할에 직접 포함됩니다. 관리형 정책은 내 여러 사용자, 그룹 및 역할에 연결할 수 있는 독립형 정책입니다. AWS 계정관리형 정책에는 AWS 관리형 정책과 고객 관리형 정책이 포함됩니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 관리형 정책과 인라인 정책의 선택을 참조하십시오.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우, 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 태스크를 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 연동 사용자 등이 포함될 수 있습니다. AWS 서비스

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. IAM의 AWS 관리형 정책은 리소스 기반 정책에 사용할 수 없습니다.

액세스 제어 목록(ACL)

액세스 제어 목록(ACL)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACLs는 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

ACL을 지원하는 서비스의 예로는 아마존 S3와 아마존 VPC가 있습니다. AWS WAF ACL에 대해 자세히 알아보려면 Amazon Simple Storage Service 개발자 가이드의 ACL(액세스 제어 목록) 개요를 참조하십시오.

기타 정책 타입

AWS 일반적이지 않은 추가 정책 유형을 지원합니다. 이러한 정책 타입은 더 일반적인 정책 타입에 따라 사용자에게 부여되는 최대 권한을 설정할 수 있습니다.

- 권한 경계 – 권한 경계는 자격 증명 기반 정책에 따라 IAM 엔터티(IAM 사용자 또는 역할)에 부여할 수 있는 최대 권한을 설정하는 고급 기능입니다. 개체에 대한 권한 경계를 설정할 수 있습니다. 그 결과로 얻는 권한은 개체의 보안 인증 기반 정책과 그 권한 경계의 교집합입니다. `Principal` 필드에서 사용자나 역할을 지정하는 리소스 기반 정책은 권한 경계를 통해 제한되지 않습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 권한 경계에 대한 자세한 내용은 IAM 사용 설명서의 IAM 엔터티에 대한 권한 경계를 참조하십시오.
- 서비스 제어 정책 (SCP) - SCP는 조직 또는 조직 단위 (OU) 에 대한 최대 권한을 지정하는 JSON 정책입니다. AWS Organizations AWS Organizations 사업체가 소유한 여러 AWS 계정 개를 그룹화하고 중앙에서 관리하는 서비스입니다. 조직에서 모든 기능을 활성화할 경우, 서비스 제어 정책(SCP)을 임의의 또는 모든 계정에 적용할 수 있습니다. SCP는 구성원 계정의 엔티티 (각 엔티티 포함) 에 대한 권한을 제한합니다. AWS 계정 루트 사용자조직 및 SCP에 대한 자세한 내용은 AWS Organizations 사용 설명서의 SCP 작동 방식을 참조하십시오.

- 세션 정책 - 세션 정책은 역할 또는 페더레이션 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 결과적으로 얻는 세션의 권한은 사용자 또는 역할의 보안 인증 기반 정책의 교차와 세션 정책입니다. 또한 권한을 리소스 기반 정책에서 가져올 수도 있습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 자세한 내용은 IAM 사용 설명서의 세션 정책을 참조하십시오.

여러 정책 타입

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 여러 정책 유형이 관련된 경우 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 참조하십시오.

아마존 매니지드 블록체인 (AMB) 액세스 폴리곤이 IAM과 연동되는 방식

IAM을 사용하여 AMB 액세스 폴리곤에 대한 액세스를 관리하기 전에 AMB 액세스 폴리곤에서 사용할 수 있는 IAM 기능에 대해 알아보십시오.

아마존 관리형 블록체인 (AMB) 액세스 폴리곤과 함께 사용할 수 있는 IAM 기능

IAM 특성	AMB 액세스 폴리곤 지원
ID 기반 정책	예
리소스 기반 정책	아니요
정책 작업	예
정책 리소스	아니요
정책 조건 키	아니요
ACL	아니요
ABAC(정책 내 태그)	아니요
임시 보안 인증	아니요
보안 주체 권한	아니요

IAM 특성	AMB 액세스 폴리곤 지원
서비스 역할	아니요
서비스 연결 역할	아니요

AMB Access Polygon 및 기타 제품이 대부분의 IAM 기능과 어떻게 AWS 서비스 작동하는지 자세히 알아보려면 IAM 사용 설명서의 IAM과 함께 작동하는AWS 서비스를 참조하십시오.

AMB 액세스 폴리곤에 대한 ID 기반 정책

보안 인증 기반 정책 지원	예

자격 증명 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 자격 증명에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 태스크를 수행할 수 있는지를 제어합니다. ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성을 참조하십시오.

IAM ID 기반 정책을 사용하면 허용되거나 거부되는 작업과 리소스뿐 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. 보안 인증 기반 정책에서는 보안 주체가 연결된 사용자 또는 역할에 적용되므로 보안 주체를 지정할 수 없습니다. JSON 정책에서 사용하는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 IAM JSON 정책 요소 참조를 참조하십시오.

AMB 액세스 폴리곤의 ID 기반 정책 예제

AMB 액세스 폴리곤 ID 기반 정책의 예를 보려면 을 참조하십시오. 아마존 관리형 블록체인 (AMB) 액세스 폴리곤에 대한 ID 기반 정책 예제

AMB 액세스 폴리곤 내의 리소스 기반 정책

리소스 기반 정책 지원	아니요

리소스 기반 정책은 리소스에 연결하는 JSON 정책 문서입니다. 리소스 기반 정책의 예는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경

우, 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 태스크를 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 등이 포함될 수 있습니다. AWS 서비스

교차 계정 액세스를 활성화하려는 경우, 전체 계정이나 다른 계정의 IAM 개체를 리소스 기반 정책의 보안 주체로 지정할 수 있습니다. 리소스 기반 정책에 크로스 계정 보안 주체를 추가하는 것은 트러스트 관계 설정의 절반밖에 되지 않는다는 것을 유념하십시오. 보안 주체와 리소스가 다른 AWS 계정경우 신뢰할 수 있는 계정의 IAM 관리자는 보안 주체 개체 (사용자 또는 역할) 에게 리소스에 액세스할 수 있는 권한도 부여해야 합니다. 엔터티에 ID 기반 정책을 연결하여 권한을 부여합니다. 하지만 리소스 기반 정책이 동일 계정의 보안 주체에 액세스를 부여하는 경우, 추가 자격 증명 기반 정책이 필요하지 않습니다. 자세한 내용은 IAM 사용 설명서의 IAM의 교차 계정 리소스 액세스를 참조하십시오.

AMB 액세스 폴리곤에 대한 정책 조치

정책 작업 지원	예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

JSON 정책의 `Action`요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 태스크를 설명합니다. 정책 작업은 일반적으로 관련 AWS API 작업과 이름이 같습니다. 일치하는 API 작업이 없는 권한 전용 작업 같은 몇 가지 예외도 있습니다. 정책에서 여러 작업이 필요한 몇 가지 작업도 있습니다. 이러한 추가 작업을 일컬어 종속 작업이라고 합니다.

연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하십시오.

AMB 액세스 폴리곤 작업 목록을 보려면 서비스 권한 부여 참조의 Amazon Managed Blockchain (AMB) 액세스 폴리곤에서 정의한 작업을 참조하십시오.

AMB 액세스 폴리곤의 정책 조치는 작업 앞에 다음 접두사를 사용합니다.

```
managedblockchain:
```

단일 문에서 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

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

```
      "managedblockchain::action2"
         ]
```

와일드카드(*)를 사용하여 여러 작업을 지정할 수 있습니다. 예를 들어, InvokeRpcPolygon라는 단어로 시작하는 모든 태스크를 지정하려면 다음 태스크를 포함합니다.

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

AMB 액세스 폴리곤 ID 기반 정책의 예를 보려면 을 참조하십시오. 아마존 관리형 블록체인 (AMB) 액세스 폴리곤에 대한 ID 기반 정책 예제

AMB 액세스 폴리곤의 정책 리소스

정책 리소스 지원	아니요

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

Resource JSON 정책 요소는 작업이 적용되는 하나 이상의 개체를 지정합니다. 문장에는 Resource또는 NotResource요소가 반드시 추가되어야 합니다. 모범 사례에 따라 Amazon 리소스 이름(ARN)을 사용하여 리소스를 지정합니다. 리소스 수준 권한이라고 하는 특정 리소스 유형을 지원하는 작업에 대해 이 태스크를 수행할 수 있습니다.

작업 나열과 같이 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

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

AMB 액세스 폴리곤 리소스 유형 및 해당 ARN 목록을 보려면 서비스 권한 부여 참조의 Amazon Managed Blockchain (AMB) 액세스 폴리곤에서 정의한 리소스를 참조하십시오. 각 리소스의 ARN을 지정할 수 있는 작업에 대해 알아보려면 Amazon Managed Blockchain (AMB) 액세스 폴리곤에서 정의한 작업을 참조하십시오.

AMB 액세스 폴리곤 ID 기반 정책의 예를 보려면 을 참조하십시오. 아마존 관리형 블록체인 (AMB) 액세스 폴리곤에 대한 ID 기반 정책 예제

AMB 액세스 폴리곤의 정책 조건 키

서비스별 정책 조건 키 지원	아니요

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Condition` 요소(또는 `Condition` 블록)를 사용하면 정책이 발효되는 조건을 지정할 수 있습니다. `Condition` 요소는 옵션입니다. 같거나 작음과 같은 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다.

한 문에서 여러 `Condition` 요소를 지정하거나 단일 `Condition` 요소에서 여러 키를 지정하는 경우, AWS 는 논리적 `AND` 태스크를 사용하여 평가합니다. 단일 조건 키에 여러 값을 지정하는 경우 는 논리적 `OR` 연산을 사용하여 조건을 AWS 평가합니다. 명문의 권한을 부여하기 전에 모든 조건을 충족해야 합니다.

조건을 지정할 때 자리 표시자 변수를 사용할 수도 있습니다. 예컨대, IAM 사용자에게 IAM 사용자 이름으로 태그가 지정된 경우에만 리소스에 액세스할 수 있는 권한을 부여할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 IAM 정책 요소: 변수 및 태그를 참조하십시오.

AWS 글로벌 조건 키 및 서비스별 조건 키를 지원합니다. 모든 AWS 글로벌 조건 키를 보려면 IAM 사용 AWS 설명서의 글로벌 조건 컨텍스트 키를 참조하십시오.

AMB 액세스 폴리곤 조건 키 목록을 보려면 서비스 권한 부여 참조의 Amazon Managed Blockchain (AMB) 액세스 폴리곤의 조건 키를 참조하십시오. 조건 키를 사용할 수 있는 작업 및 리소스를 알아보려면 Amazon Managed Blockchain (AMB) 액세스 폴리곤에서 정의한 작업을 참조하십시오.

AMB 액세스 폴리곤 ID 기반 정책의 예를 보려면 을 참조하십시오. 아마존 관리형 블록체인 (AMB) 액세스 폴리곤에 대한 ID 기반 정책 예제

AMB 액세스 폴리곤의 ACL

ACL 지원	아니요

ACL(액세스 통제 목록)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACLs는 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

AMB 액세스 폴리곤이 있는 ABAC

ABAC 지원(정책의 태그)	아니요

ABAC(속성 기반 액세스 통제)는 속성에 근거하여 권한을 정의하는 권한 부여 전략입니다. AWS에서는 이러한 속성을 태그라고 합니다. IAM 엔티티 (사용자 또는 역할) 및 여러 AWS 리소스에 태그를 첨부할 수 있습니다. ABAC의 첫 번째 단계로 개체 및 리소스에 태그를 지정합니다. 그런 다음 보안 주체의 태그가 액세스하려는 리소스의 태그와 일치할 때 작업을 허용하도록 ABAC 정책을 설계합니다.

ABAC는 빠르게 성장하는 환경에서 유용하며 정책 관리가 번거로운 상황에 도움이 됩니다.

태그에 근거하여 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` 또는 `aws:TagKeys` 조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다.

서비스가 모든 리소스 유형에 대해 세 가지 조건 키를 모두 지원하는 경우, 값은 서비스에 대해 예입니다. 서비스가 일부 리소스 유형에 대해서만 세 가지 조건 키를 모두 지원하는 경우, 값은 부분적입니다.

ABAC에 대한 자세한 정보는 IAM 사용 설명서의 ABAC란 무엇입니까?를 참조하십시오. ABAC 설정 단계가 포함된 자습서를 보려면 IAM 사용 설명서의 속성 기반 액세스 제어(ABAC) 사용을 참조하십시오.

AMB 액세스 폴리곤에서 임시 자격 증명 사용

임시 보안 인증 정보 지원	아니요

임시 자격 증명을 사용하여 로그인하면 일부 자격 증명이 AWS 서비스 작동하지 않습니다. 임시 자격 증명을 사용하는 방법을 AWS 서비스 비롯한 추가 정보는 IAM 사용 설명서의 IAM과AWS 서비스 연동되는 내용을 참조하십시오.

사용자 이름과 암호를 제외한 다른 방법을 AWS Management Console 사용하여 로그인하면 임시 자격 증명을 사용하는 것입니다. 예를 들어 회사의 SSO (Single Sign-On) 링크를 AWS 사용하여 액세스하는 경우 이 프로세스에서 자동으로 임시 자격 증명을 생성합니다. 또한 콘솔에 사용자로 로그인한 다음 역할을 전환할 때 임시 보안 인증을 자동으로 생성합니다. 역할 전환에 대한 자세한 내용은 IAM 사용 설명서의 역할로 전환(콘솔)을 참조하십시오.

또는 API를 사용하여 임시 자격 증명을 수동으로 생성할 수 있습니다 AWS CLI . AWS 그런 다음 해당 임시 자격 증명을 사용하여 액세스할 수 AWS있습니다. AWS 장기 액세스 키를 사용하는 대신 임시 자

격 증명을 동적으로 생성할 것을 권장합니다. 자세한 정보는 IAM의 임시 보안 자격 증명 섹션을 참조하십시오.

AMB 액세스 폴리곤에 대한 서비스 간 보안 주체 권한

전달 액세스 세션(FAS) 지원	아니요

IAM 사용자 또는 역할을 사용하여 에서 작업을 수행하는 AWS경우 사용자는 보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는 전화를 거는 주체의 권한을 다운스트림 서비스에 AWS 서비스 요청하라는 요청과 결합하여 사용합니다. AWS 서비스 FAS 요청은 다른 서비스 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 서비스가 수신한 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하세요.

AMB 액세스 폴리곤의 서비스 역할

서비스 역할 지원	아니요

서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하기 위해 수임하는 IAM role(IAM 역할)입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 AWS 서비스에 대한 권한을 위임할 역할 생성을 참조하십시오.

> **Warning**
>
> 서비스 역할의 권한을 변경하면 AMB 액세스 폴리곤 기능이 손상될 수 있습니다. AMB Access Polygon에서 이에 대한 지침을 제공하는 경우에만 서비스 역할을 편집하십시오.

AMB 액세스 폴리곤의 서비스 연결 역할

서비스 연결 역할 지원	아니요

서비스 연결 역할은 에 연결된 서비스 역할 유형입니다. AWS 서비스서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 사용자에게 AWS 계정 표시되며 해

당 서비스가 소유합니다. IAM 관리자는 서비스 링크 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

서비스 연결 역할 생성 또는 관리에 대한 자세한 내용은 IAM으로 작업하는AWS 서비스를 참조하십시오. 서비스 연결 역할 열에서 Yes(이)가 포함된 서비스를 테이블에서 찾습니다. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 Yes(네) 링크를 선택합니다.

아마존 관리형 블록체인 (AMB) 액세스 폴리곤에 대한 ID 기반 정책 예제

기본적으로 사용자와 역할은 AMB Access Polygon 리소스를 생성하거나 수정할 권한이 없습니다. 또한 AWS Management Console, AWS Command Line Interface (AWS CLI) 또는 AWS API를 사용하여 작업을 수행할 수 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 맡을 수 있습니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성을 참조하십시오.

각 리소스 유형의 ARN 형식을 비롯하여 AMB Access Polygon에서 정의한 작업 및 리소스 유형에 대한 자세한 내용은 서비스 권한 부여 참조의 Amazon Managed Blockchain (AMB) Access Polygon용 작업, 리소스 및 조건 키를 참조하십시오.

주제

- 정책 모범 사례
- AMB 액세스 폴리곤 콘솔 사용
- 사용자가 자신의 고유한 권한을 볼 수 있도록 허용
- 폴리곤 네트워크 액세스

정책 모범 사례

ID 기반 정책에 따라 계정에서 AMB Access Polygon 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부가 결정됩니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따릅니다.

- AWS 관리형 정책을 시작하고 최소 권한 권한으로 이동 — 사용자와 워크로드에 권한을 부여하려면 여러 일반적인 사용 사례에 권한을 부여하는 AWS 관리형 정책을 사용하십시오. 해당 내용은 에서 사용할 수 있습니다. AWS 계정사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 더 줄이

는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 AWS 관리형 정책 또는 직무에 대한AWS 관리형 정책을 참조하십시오.

- 최소 권한 적용 – IAM 정책을 사용하여 권한을 설정하는 경우, 태스크를 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. IAM을 사용하여 권한을 적용하는 방법에 대한 자세한 정보는 IAM 사용 설명서의 IAM의 정책 및 권한을 참조하십시오.
- IAM 정책의 조건을 사용하여 액세스 추가 제한 – 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어 SSL을 사용하여 모든 요청을 전송해야 한다고 지정하는 정책 조건을 작성할 수 있습니다. 예를 AWS 서비스들어 특정 작업을 통해 서비스 작업을 사용하는 경우 조건을 사용하여 서비스 작업에 대한 액세스 권한을 부여할 수도 AWS CloudFormation있습니다. 자세한 내용은 IAM 사용 설명서의 IAM JSON 정책 요소: 조건을 참조하십시오.
- IAM Access Analyzer를 통해 IAM 정책을 확인하여 안전하고 기능적인 권한 보장 - IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 모범 사례가 정책에서 준수되도록 신규 및 기존 정책을 확인합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 추천을 제공하여 안전하고 기능적인 정책을 작성하도록 돕습니다. 자세한 내용은 IAM 사용 설명서의 IAM Access Analyzer 정책 검증을 참조하십시오.
- 멀티 팩터 인증 (MFA) 필요 - IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 AWS 계정 MFA를 활성화하십시오. API 작업을 직접적으로 호출할 때 MFA가 필요하면 정책에 MFA 조건을 추가합니다. 자세한 내용은 IAM 사용 설명서의 MFA 보호 API 액세스 구성을 참조하십시오.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 IAM의 보안 모범 사례를 참조하십시오.

AMB 액세스 폴리곤 콘솔 사용

Amazon Managed Blockchain (AMB) 액세스 폴리곤 콘솔에 액세스하려면 최소 권한 집합이 있어야 합니다. 이러한 권한을 통해 내 AMB Access Polygon 리소스에 대한 세부 정보를 나열하고 볼 수 있어야 합니다. AWS 계정최소 필수 권한보다 더 제한적인 자격 증명 기반 정책을 만들면 콘솔이 해당 정책에 연결된 엔터티(사용자 또는 역할)에 대해 의도대로 작동하지 않습니다.

AWS CLI 또는 API만 호출하는 사용자에게 최소 콘솔 권한을 허용할 필요는 없습니다. AWS 그 대신, 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 합니다.

사용자와 역할이 AMB 액세스 폴리곤 콘솔을 계속 사용할 수 있도록 하려면 AMB 액세스 폴리곤 *ConsoleAccess* 또는 *ReadOnly* AWS 관리형 정책도 엔티티에 연결하십시오. 자세한 내용은 IAM 사용 설명서의 사용자에게 권한 추가를 참조하십시오.

사용자가 자신의 고유한 권한을 볼 수 있도록 허용

이 예제는 IAM 사용자가 자신의 사용자 ID에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여줍니다. 이 정책에는 콘솔에서 또는 API를 사용하여 프로그래밍 방식으로 이 작업을 완료할 수 있는 권한이 포함됩니다. AWS CLI AWS

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

폴리곤 네트워크 액세스

> **Note**
>
> Polygon의 퍼블릭 엔드포인트에 `mainnet` 액세스하고 `mainnet` JSON-RPC를 호출하려면 AMB Access Polygon에 대한 적절한 IAM 권한이 있는 사용자 자격 증명(`AWS_ACCESS_KEY_ID`및`AWS_SECRET_ACCESS_KEY`) 이 필요합니다.

Example 모든 폴리곤 네트워크에 액세스하기 위한 IAM 정책

이 예시에서는 IAM 사용자에게 모든 Polygon 네트워크에 AWS 계정 대한 액세스 권한을 부여합니다.

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

Example Polygon 메인넷 네트워크에 액세스하기 위한 IAM 정책

이 예시에서는 IAM 사용자에게 Polygon 메인넷 네트워크 AWS 계정 액세스 권한을 부여합니다.

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

```
        }
    ]
}
```

아마존 관리형 블록체인 (AMB) 액세스 폴리곤 ID 및 액세스 문제 해결

다음 정보를 사용하면 AMB Access Polygon 및 IAM으로 작업할 때 발생할 수 있는 일반적인 문제를 진단하고 해결하는 데 도움이 됩니다.

주제

- AMB 액세스 폴리곤에서 작업을 수행할 권한이 없습니다.
- 저는 IAM을 수행할 권한이 없습니다. PassRole
- 외부 사용자가 내 AMB Access Polygon AWS 계정 리소스에 액세스할 수 있도록 허용하고 싶습니다.

AMB 액세스 폴리곤에서 작업을 수행할 권한이 없습니다.

작업을 수행할 권한이 없다는 오류가 수신되면, 작업을 수행할 수 있도록 정책을 업데이트해야 합니다.

다음 예제 오류는 mateojacksonIAM 사용자가 콘솔을 사용하여 가상 *my-example-widget*리소스에 대한 세부 정보를 보려고 하지만 가상 managedblockchain::*GetWidget*권한이 없을 때 발생합니다.

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

이 경우 managedblockchain::*GetWidget*작업을 사용하여 *my-example-widget*리소스에 액세스할 수 있도록 mateojackson사용자 정책을 업데이트해야 합니다.

도움이 필요하면 관리자에게 문의하십시오. AWS 관리자는 로그인 자격 증명을 제공한 사람입니다.

저는 IAM을 수행할 권한이 없습니다. PassRole

작업을 수행할 권한이 없다는 오류가 발생하는 경우 AMB Access Polygon에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다. iam:PassRole

일부 AWS 서비스 서비스에서는 새 서비스 역할 또는 서비스 연결 역할을 생성하는 대신 기존 역할을 해당 서비스에 전달할 수 있습니다. 이렇게 하려면 사용자가 서비스에 역할을 전달할 수 있는 권한을 가지고 있어야 합니다.

다음 예제 오류는 이라는 IAM 사용자가 AMB Access Polygon에서 콘솔을 사용하여 작업을 `marymajor` 수행하려고 할 때 발생합니다. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 수 있는 권한을 가지고 있지 않습니다.

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

이 경우, Mary가 `iam:PassRole` 작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요하면 관리자에게 문의하세요. AWS 관리자는 로그인 자격 증명을 제공한 사람입니다.

외부 사용자가 내 AMB Access Polygon AWS 계정 리소스에 액세스할 수 있도록 허용하고 싶습니다.

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스할 때 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 액세스 제어 목록(ACL)을 지원하는 서비스의 경우 이러한 정책을 사용하여 다른 사람에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세히 알아보려면 다음을 참조하십시오.

- AMB Access Polygon이 이러한 기능을 지원하는지 여부를 알아보려면 을 참조하십시오. 아마존 매니지드 블록체인 (AMB) 액세스 폴리곤이 IAM과 연동되는 방식
- 소유한 리소스에 대한 액세스를 제공하는 방법을 알아보려면 IAM 사용 설명서의 다른 AWS 계정 IAM 사용자에게 액세스 권한 제공을 참조하십시오. AWS 계정
- 제3자에게 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 타사 AWS 계정AWS 계정 소유에 대한 액세스 제공을 참조하십시오.
- ID 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 외부에서 인증된 사용자에게 액세스 권한 제공(자격 증명 페더레이션)을 참조하십시오.
- 교차 계정 액세스에 대한 역할 사용과 리소스 기반 정책의 차이점을 알아보려면 IAM 사용 설명서의 IAM의 교차 계정 리소스 액세스를 참조하십시오.

를 사용하여 아마존 관리형 블록체인 (AMB) 액세스 폴리곤 이벤트 로깅 AWS CloudTrail

> **Note**
>
> 아마존 관리형 블록체인 (AMB) 액세스 폴리곤은 관리 이벤트를 지원하지 않습니다.

Amazon Managed Blockchain은 관리형 블록체인에서 AWS CloudTrail사용자, 역할 또는 서비스가 수행한 작업의 기록을 제공하는 AWS 서비스인 에서 실행됩니다. CloudTrail 관리형 블록체인의 AMB Access Polygon 엔드포인트를 데이터 플레인 이벤트로 호출한 사람을 캡처합니다.

원하는 데이터 플레인 이벤트를 수신하도록 구독되는 적절하게 구성된 트레일을 생성하면 AMB Access Polygon 관련 이벤트를 S3 버킷으로 지속적으로 전송할 수 있습니다. CloudTrail 에서 수집한 정보를 사용하여 AMB Access Polygon 엔드포인트 중 하나에 요청이 이루어졌는지 CloudTrail, 요청을 보낸 IP 주소, 요청한 사람, 요청 시기 및 기타 추가 세부 정보를 확인할 수 있습니다.

자세한 내용은 사용 CloudTrail 설명서를 참조하십시오.AWS CloudTrail

AMB 액세스 폴리곤 정보는 CloudTrail

CloudTrail 정보를 생성할 AWS 계정 때 활성화됩니다. 하지만 AMB Access Polygon 엔드포인트를 호출한 사람을 보려면 데이터 플레인 이벤트를 구성해야 합니다.

AMB Access Polygon에 대한 이벤트를 포함하여 내 이벤트를 지속적으로 기록하려면 트레일을 생성하십시오 AWS 계정. 트레일을 사용하면 S3 CloudTrail 버킷에 로그 파일을 전송할 수 있습니다. 콘솔에서 추적을 생성하면 기본적으로 모든 AWS 리전에 추적이 적용됩니다. 트레일은 AWS 파티션에 있는 모든 지원 지역의 이벤트를 기록하고 지정한 S3 버킷으로 로그 파일을 전달합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 더 자세히 분석하고 이에 따라 조치를 AWS 서비스 취하도록 기타를 구성할 수 있습니다. 자세한 내용은 다음 자료를 참조하십시오.

- 폴리곤 CloudTrail JSON-RPC를 추적하는 데 사용
- 추적 생성 개요
- CloudTrail 지원되는 서비스 및 통합
- 에 대한 Amazon SNS 알림 구성 CloudTrail
- 여러 지역에서 CloudTrail 로그 파일 수신 및 여러 계정으로부터 CloudTrail 로그 파일 수신

CloudTrail 데이터 이벤트를 분석하여 AMB Access Polygon 엔드포인트를 호출한 사람을 모니터링할 수 있습니다.

모든 이벤트 및 로그 항목에는 요청을 생성한 사용자에 대한 정보가 들어 있습니다. 보안 인증 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청이 루트 또는 AWS Identity and Access Management (IAM) 사용자 자격 증명으로 이루어졌는지 여부
- 역할 또는 연합된 사용자에 대한 임시 보안 자격 증명을 사용하여 요청이 생성되었는지 여부
- 다른 사람이 요청했는지 여부 AWS 서비스

자세한 내용은 CloudTrail UserIdentity 요소를 참조하십시오.

AMB 액세스 폴리곤 로그 파일 항목의 이해

데이터 플레인 이벤트의 경우 트레일은 이벤트를 지정된 S3 버킷에 로그 파일로 전송할 수 있는 구성입니다. 각 CloudTrail 로그 파일에는 모든 소스의 단일 요청을 나타내는 하나 이상의 로그 항목이 포함되어 있습니다. 이러한 항목은 작업 날짜 및 시간, 관련 요청 매개 변수를 포함하여 요청된 작업에 대한 세부 정보를 제공합니다.

> **Note**
>
> CloudTrail 로그 파일의 데이터 이벤트는 AMB Access Polygon API 호출의 정렬된 스택 추적이 아니므로 특정 순서로 표시되지 않습니다.

폴리곤 CloudTrail JSON-RPC를 추적하는 데 사용

CloudTrail 이를 사용하여 계정에서 AMB Access Polygon 엔드포인트를 호출한 사람과 데이터 이벤트로 호출된 JSON-RPC를 추적할 수 있습니다. 기본적으로 트레일을 생성할 때 데이터 이벤트는 로깅되지 않습니다. AMB Access Polygon 엔드포인트를 CloudTrail 데이터 이벤트로 호출한 사용자를 기록하려면 활동을 수집하려는 지원되는 리소스 또는 리소스 유형을 트레일에 명시적으로 추가해야 합니다. AMB Access Polygon은,, SDK를 사용하여 데이터 이벤트를 추가할 수 있도록 지원합니다. AWS Management Console AWS CLI자세한 내용은 사용 설명서의 고급 선택기를 사용한 이벤트 로깅을 참조하십시오.AWS CloudTrail

트레일에 데이터 이벤트를 로깅하려면 트레일을 만든 후 put-event-selectors 작업을 사용하십시오. AMB Access Polygon 엔드포인트를 호출한 사람을 확인하기 위한 데이터 이벤트 로깅을 시작하려면

이 --advanced-event-selectors 옵션을 사용하여 AWS::ManagedBlockchain::Network 리소스 유형을 지정합니다.

Example 모든 계정의 AMB Access Polygon 엔드포인트 요청의 데이터 이벤트 로그 항목

다음 예시는 put-event-selectors 작업을 사용하여 해당 지역의 트레일에 대한 모든 계정의 AMB Access Polygon 엔드포인트 요청을 기록하는 방법을 보여줍니다. my-polygon-trail us-east-1

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

구독한 후에는 이전 예제에서 지정한 트레일에 연결된 S3 버킷의 사용량을 추적할 수 있습니다.

다음 결과는 에서 수집한 정보의 CloudTrail 데이터 이벤트 로그 항목을 보여줍니다 CloudTrail. Polygon JSON-RPC 요청이 AMB Access Polygon 엔드포인트 중 하나에 이루어졌는지, 요청을 보낸 IP 주소, 요청한 사람, 요청 시기 및 기타 추가 세부 정보를 확인할 수 있습니다. 다음 예제의 일부 값은 보안상의 이유로 난독화되었지만 실제 로그 항목에는 완전히 나타납니다.

```
{
        "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 액세스 폴리곤 사용 설명서의 문서 기록

다음 표에는 AMB 액세스 폴리곤의 설명서 릴리스가 설명되어 있습니다.

변경 사항	설명	날짜
JSON-RPC의 할당량이 업데이트되었습니다.	지원되는 각 JSON-RPC에 대해 AMB 액세스 폴리곤이 지원하는 할당량이 업데이트되었습니다.	2024년 4월 12일
뭄바이 테스트넷 네트워크 지원 종료	AMB 액세스 폴리곤은 2024년 4월 15일에 뭄바이 테스트넷 지원을 종료했습니다.	2024년 4월 10일
튜토리얼 주제 추가	AMB 액세스 폴리곤 자습서는 AWS re:Post의 커뮤니티 기사 섹션에서 제공됩니다.	2024년 4월 9일
공개 미리 보기	아마존 관리형 블록체인 (AMB) 액세스 폴리곤 서비스의 공개 프리뷰 릴리즈.	2023년 11월 24일