aws

개발자 가이드

Amazon Managed Blockchain 쿼리

Amazon Managed Blockchain 쿼리: 개발자 가이드

Table of Contents

아마존 관리형 블록체인 (AMB) 쿼리란 무엇입니까?

Amazon Managed Blockchain (AMB) 은 퍼블릭 및 프라이빗 블록체인 모두에서 복원력이 뛰어난 Web3 애플리케이션을 구축할 수 있도록 설계된 완전 관리형 서비스입니다. AMB Access를 사용하면 여러 블록체인에 서버리스 방식으로 즉시 액세스할 수 있습니다. 특수 블록체인 인프라를 배포하고 블록체인 네트워크에 연결된 상태를 유지할 필요 없이 Web3 지원 애플리케이션을 구축하십시오. AMB Query를 사용하면 개발자 친화적인 API 작업을 사용하여 여러 블록체인의 실시간 및 과거 데이터에 액세스할 수 있습니다. 표준화된 블록체인 데이터는 특수 블록체인 인프라 또는 ETL (추출, 변환, 로드) 없이 AWS 서비스와 통합될 수 있습니다. 모든 AMB 기능은 기관 등급 및 주류 소비자 애플리케이션 빌드에 맞게 안전하게 확장됩니다.

Amazon Managed Blockchain (AMB) 쿼리는 개발자 친화적인 API 작업을 통해 표준화된 다중 블록체인 데이터 세트에 대한 서버리스 액세스를 제공합니다. AMB Query를 사용하면 하나 이상의 퍼블릭 블록체인의 데이터가 필요한 애플리케이션을 신속하게 출시할 수 있습니다. 이때 블록체인 데이터를 파싱하고, 계약을 추적하고, 특수 인덱싱 인프라를 유지 관리하는 데 드는 오버헤드가 필요하지 않습니다. 대체 가능한 토큰 또는 대체 불가능한 토큰 (NFT) 의 과거 토큰 잔고를 분석하든, 특정 지갑 주소의 거래 내역을 확인하든, 이더와 같은 고유 암호화폐의 배포에 대한 데이터 분석을 수행하든 AMB Query를 사용하면 블록체인 데이터에 액세스할 수 있습니다.

AMB Query를 처음 사용하시나요?

AMB Query를 처음 사용하는 경우 먼저 다음 섹션을 읽는 것이 좋습니다.

- 주요 개념: 아마존 관리형 블록체인 (AMB) 쿼리
- 아마존 관리형 블록체인 (AMB) 쿼리 설정
- 아마존 관리형 블록체인 (AMB) 쿼리 시작하기
- 아마존 관리형 블록체인 (AMB) 쿼리를 사용한 사용 사례

주요 개념: 아마존 관리형 블록체인 (AMB) 쿼리

> **Note**
>
> 이 가이드에서는 사용자가 필수 블록체인 개념을 잘 알고 있다고 가정합니다. 이러한 개념에는 탈중앙화, 토큰, 계약, 거래, 지갑 proof-of-work, 공개 및 개인 키, 스테이킹, 채굴, 반감기 등이 포함됩니다.

Amazon Managed Blockchain (AMB) 쿼리를 사용하면 다중 블록체인 네트워크 데이터에 편리하게 액세스할 수 있으므로 블록체인 활동과 관련된 컨텍스트 데이터를 더 쉽게 추출할 수 있습니다. AMB 쿼리를 사용하여 비트코인 메인넷 및 이더리움 메인넷과 같은 퍼블릭 블록체인 네트워크에서 데이터를 읽을 수 있습니다. 또한 주소의 현재 및 과거 잔액과 같은 정보를 얻거나 특정 기간 동안의 블록체인 거래 목록을 얻을 수 있습니다. 또한 트랜잭션 이벤트와 같은 특정 트랜잭션의 세부 정보를 얻을 수 있으며, 이를 추가로 분석하거나 애플리케이션의 비즈니스 로직에 사용할 수 있습니다.

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

AMB 쿼리를 사용할 때는 다음 사항을 고려하십시오.

- 사용 가능한 지역

  AMB 쿼리는 미국 동부 (버지니아 북부) *us-east-1* 지역에서 지원됩니다.

- Service endpoints

  AMB 쿼리는 다음 엔드포인트를 사용하여 액세스할 수 있습니다.

  `https://managedblockchain-query.us-east-1.amazonaws.com`.

- 지원되는 블록체인 네트워크

  AMB Query는 다음과 같은 퍼블릭 블록체인 네트워크를 지원합니다.

  - 비트코인 메인넷 — proof-of-work 합의에 의해 보호되고 비트코인 (BTC) 암호화폐가 발행되고 거래되는 퍼블릭 비트코인 블록체인 네트워크입니다. 메인넷에서의 거래는 실제 가치를 가지며 (즉, 실제 비용이 발생함) 퍼블릭 블록체인에 기록됩니다.
  - 비트코인 테스트넷 — 비트코인 메인넷을 위한 테스트넷. 이 네트워크의 비트코인 (BTC) 은 메인넷 BTC와 별개이며, 보통 가치가 없습니다.

  - 이더리움 메인넷 — 퍼블릭 이더리움 블록체인의 proof-of-stake 메인 네트워크입니다. 메인넷에서의 거래는 실제 가치를 가지며 (즉, 실제 비용이 발생함) 분산 원장에 기록됩니다.
  - 세폴리아 테스트넷 — 이더리움 메인넷을 위한 테스트넷. 이 네트워크의 이더 (ETH) 는 메인넷 ETH와 별개이며, 보통 가치가 없습니다.

- 지원되는 블록체인 토큰 및 계약

  AMB Query는 다음과 같은 네이티브 및 표준 이더리움 컨트랙트 토큰을 지원합니다.

  - 퍼블릭 블록체인 네이티브 토큰
    - 비트코인 (BTC) — 비트코인 관련 블록체인의 기본 토큰입니다.
    - 이더 (ETH) — 이더리움 관련 블록체인의 기본 토큰입니다.

  - 이더리움 계약 표준
    - ERC-20 토큰 표준 — ERC-20 토큰은 대체 가능한 토큰의 표준입니다. 각 ERC-20 토큰을 다른 ERC-20 토큰이 발행된 다른 토큰과 정확히 같게 만드는 속성을 가지고 있습니다. 즉, 한 토큰은 다른 모든 토큰과 동일하며 앞으로도 항상 같을 것입니다. 자세한 내용은 Ethereum.org의 ERC-20 토큰 표준을 참조하십시오.
    - ERC-721 대체 불가 토큰 표준 — ERC-721 표준은 대체 불가능한 토큰 (NFT) 의 표준입니다. 이 유형의 토큰은 고유하며 연식, 희귀성 또는 기타 특성 때문에 동일한 계약의 다른 토큰과 가치가 다를 수 있습니다. 자세한 내용은 Ethereum.org의 ERC-721 토큰 표준을 참조하십시오.

      ERC-1155 다중 토큰 표준 — ERC-1155 표준은 대체 가능한 토큰 유형과 대체 불가능한 토큰 유형을 원하는 수만큼 표현하고 제어할 수 있는 계약 인터페이스를 생성하는 표준입니다. 이러한 방식으로 ERC-1155 토큰은 ERC-20 및 ERC-721 토큰과 동일하게 작동할 수 있으며, 동시에 두 가지 기능을 모두 수행할 수도 있습니다. ERC-1155 토큰은 ERC-20 및 ERC-721 표준의 기능을 개선하여 효율성을 높이는 동시에 명백한 구현 오류를 수정합니다. 자세한 내용은 이더리움.org의 ERC-1155 토큰 표준을 참조하십시오.

- 최종성

  블록체인에서 최종성이란 유효한 거래가 취소될 가능성이 낮다는 것을 의미합니다. 비트코인 메인넷의 경우 AMB Query는 6블록 이후 거래를 최종 트랜잭션으로 간주합니다. 비트코인 테스트넷의 경우 6블록 또는 60분 중 먼저 도래하는 시점을 기준으로 거래가 완료된 것으로 간주합니다. 지원되는 이더리움 네트워크의 경우 AMB Query는 64블록 이후 거래를 최종 트랜잭션으로 간주합니다.

AMB Query의 토큰 잔고 및 계약 API 작업은 유효성에 도달한 데이터만 반환합니다. 하지만 AMB Query의 트랜잭션 및 트랜잭션 이벤트 API 작업은 아직 완료되지 않았더라도 블록체인 네트워크에서 확인된 트랜잭션에 대한 데이터를 반환할 수 있습니다.

- NULL 주소는 지원되지 않습니다.

  AMB 쿼리는 NULL (`0x0000000000000000000000000000000000000000`) 주소를 지원하지 않습니다.

- API 호출의 서명 버전 4 서명

  AMB 쿼리 API를 호출할 때는 서명 버전 4 서명 프로세스를 사용하여 인증된 HTTPS 연결을 통해 호출할 수 있습니다. 즉, AWS 계정의 승인된 IAM 보안 주체만 AMB 쿼리 API 호출을 할 수 있습니다. 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.

- AMB Query는 비트코인 트랜잭션 식별자와 트랜잭션 해시를 지원합니다.

  비트코인 네트워크의 경우 AMB 쿼리 API 작업은 트랜잭션 식별자 (`transactionId`) 와 트랜잭션 해시 () 를 모두 지원합니다. `transactionHash` 증인 데이터를 제외한 거래의 이중 SHA `transactionId` 해시입니다. 증인 데이터 (증인 거래 ID라고도 함) 를 포함한 거래의 이중 SHA `transactionHash` 해시입니다.

  비트코인 네트워크에 대한 `GetTransaction`또는 `ListTransactionEvents`API 작업을 호출할 때 또는 를 지정할 수 있습니다. `transactionId` `transactionHash` 또한 a `transactionId` `transactionHash` 또는 a를 반환하는 비트코인 네트워크의 모든 AMB 쿼리 작업에는 두 값이 모두 응답의 일부로 포함됩니다.

아마존 관리형 블록체인 (AMB) 쿼리 설정

Amazon Managed Blockchain (AMB) 쿼리를 처음 사용하기 전에 이 섹션의 단계에 따라 AWS 계정을 생성하십시오. 다음 섹션에서는 AMB 쿼리를 사용하여 시작하는 방법을 설명합니다.

필수 조건 및 고려 사항

Amazon Web Services를 처음 사용하기 전에 AWS 계정이 있어야 합니다.

가입하기: AWS

Amazon Web Services (AWS) 에 가입하면 Amazon Managed Blockchain (AMB) 쿼리를 포함한 모든 AWS 서비스쿼리에 AWS 계정이 자동으로 등록됩니다. 사용한 서비스에 대해서만 청구됩니다.

AWS 계정 이미 등록한 경우 다음 단계로 이동하십시오. AWS 계정이 없는 경우에는 다음 절차에 따라 계정을 만드세요.

AWS 계정을 만들려면

1. https://portal.aws.amazon.com/billing/signup을 여세요.
2. 온라인 지시 사항을 따르세요.

   등록 절차 중에는 전화를 받고 키패드로 인증 코드를 입력하는 과정이 있습니다.

   에 AWS 계정가입하면 AWS 계정 루트 사용자a가 생성됩니다. 루트 사용자에게는 계정의 모든 AWS 서비스 및 리소스 액세스 권한이 있습니다. 보안 모범 사례로, 사용자에게 관리자 액세스 권한을 할당하고 루트 사용자 액세스가 필요한 작업을 수행할 때는 루트 사용자만 사용하십시오.

적절한 권한을 가진 IAM 사용자 생성

AMB 쿼리를 생성하고 사용하려면 필요한 관리형 블록체인 작업을 허용하는 권한을 가진 AWS Identity and Access Management (IAM) 보안 주체 (사용자 또는 그룹) 를 생성해야 합니다.

IAM 보안 주체만 AMB 쿼리 API 요청을 할 수 있습니다. AMB 쿼리 API를 호출할 때는 서명 버전 4 서명 프로세스를 사용하여 인증된 HTTPS 연결을 통해 호출할 수 있습니다. 즉, AWS 계정의 승인된 IAM 보안 주체만 AMB 쿼리 API 호출을 할 수 있습니다. 이렇게 하려면 호출과 함께 AWS 자격 증명 (액세스 키 ID 및 보안 액세스 키) 을 제공해야 합니다.

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

설치 및 구성 AWS Command Line Interface

아직 설치하지 않았다면 최신 AWS 명령줄 인터페이스 (CLI) 를 설치하여 터미널의 AWS 리소스로 작업하십시오. 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 인증을 사용합니다. 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.

Amazon Managed Blockchain (AMB) 쿼리를 사용하여 블록체인을 쿼리하는 AWS Management Console 데 사용합니다.

Amazon Managed Blockchain (AMB) 쿼리에 액세스하고 를 사용하여 지원되는 블록체인 네트워크에서 쿼리를 작성할 수 있습니다. AWS Management Console다음 단계는 이를 수행하는 방법을 보여줍니다.

1. https://console.aws.amazon.com/managedblockchain/ 에서 아마존 매니지드 블록체인 콘솔을 엽니다.
2. 쿼리 섹션에서 쿼리 편집기를 선택합니다.
3. 지원되는 블록체인 네트워크 중 하나를 선택하세요.
4. 실행하려는 쿼리 유형을 선택합니다.
5. 선택한 쿼리 유형에 대한 관련 매개변수를 입력하고 쿼리를 실행합니다.

AMB Query가 쿼리를 실행하고 쿼리 결과 창에 결과가 표시됩니다.

아마존 관리형 블록체인 (AMB) 쿼리 시작하기

이 섹션의 step-by-step 자습서를 통해 Amazon Managed Blockchain (AMB) 쿼리를 사용하여 작업을 수행하는 방법을 알아보십시오. 이러한 절차에는 몇 가지 사전 요구 사항이 필요합니다. AMB Query를 처음 사용하는 경우 이 가이드의 설정 섹션을 검토할 수 있습니다. 자세한 정보는 아마존 관리형 블록체인 (AMB) 쿼리 설정을 참조하세요.

> **Note**
>
> 이 예제의 일부 변수는 의도적으로 난독화되었습니다. 이 예제를 실행하기 전에 먼저 유효한 것으로 바꾸십시오.

주제

- AMB 쿼리 API 작업에 액세스하기 위한 IAM 정책을 생성하십시오.
- Go를 사용하여 아마존 관리형 블록체인 (AMB) 쿼리 API 요청 생성
- Node.js 를 사용하여 아마존 관리형 블록체인 (AMB) 쿼리 API 요청 생성
- Python을 사용하여 아마존 관리형 블록체인 (AMB) 쿼리 API 요청 생성
- 에서 Amazon Managed Blockchain (AMB) 쿼리를 AWS Management Console 사용하여 작업을 실행합니다. GetTokenBalance

AMB 쿼리 API 작업에 액세스하기 위한 IAM 정책을 생성하십시오.

AMB 쿼리 API 요청을 하려면 Amazon Managed Blockchain (AMB) 쿼리에 대한 적절한 IAM 권한이 있는 사용자 자격 증명 (AWS_ACCESS_KEY_ID 및 AWS_SECRET_ACCESS_KEY) 을 사용해야 합니다. 설치된 터미널에서 다음 AWS CLI 명령을 실행하여 AMB 쿼리 API 작업에 액세스하는 IAM 정책을 생성합니다.

```
cat <<EOT > ~/amb-query-access-policy.json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid" : "AMBQueryAccessPolicy",
            "Effect": "Allow",
            "Action": [
                "managedblockchain-query:*"
```

```
            ],
            "Resource": "*"
        }
    ]
}
EOT
aws iam create-policy --policy-name AmazonManagedBlockchainQueryAccess --policy-
document file://$HOME/amb-query-access-policy.json
```

정책을 생성한 후 정책을 IAM 사용자의 역할에 연결하여 정책을 적용하십시오. 에서 IAM 서비스로 이동하여 해당 서비스를 사용할 IAM 사용자에게 할당된 역할에 정책을 AmazonManagedBlockchainQueryAccess 연결합니다. AWS Management Console자세한 내용은 역할 생성 및 IAM 사용자에게 할당을 참조하십시오.

> **Note**
>
> AWS 와일드카드를 사용하는 대신 특정 API 작업에 대한 액세스 권한을 부여하는 것이 좋습니다. * 자세한 정보는 특정 아마존 관리형 블록체인 (AMB) 쿼리 API 작업에 액세스을 참조하세요.

Go를 사용하여 아마존 관리형 블록체인 (AMB) 쿼리 API 요청 생성

Amazon Managed Blockchain (AMB) Query를 사용하면 블록체인 데이터가 아직 최종 단계에 도달하지 않았더라도 일단 블록체인에서 확인되면 블록체인 데이터에 즉시 액세스할 수 있는 애플리케이션을 구축할 수 있습니다. AMB 쿼리는 지갑의 거래 내역을 채우거나, 거래 해시를 기반으로 거래에 대한 컨텍스트 정보를 제공하거나, 네이티브 토큰과 ERC-721, ERC-1155, ERC-20 토큰의 잔액을 구하는 등 여러 사용 사례를 지원합니다.

다음 예제는 Go 언어로 작성되었으며 AMB 쿼리 API 작업을 사용합니다. Go에 대한 자세한 내용은 Go 설명서를 참조하십시오. AMB 쿼리 API에 대한 자세한 내용은 아마존 관리형 블록체인 (AMB) 쿼리 API 참조 설명서를 참조하십시오.

다음 예제는 ListTransactions 및 GetTransaction API 작업을 사용하여 먼저 이더리움 메인넷의 특정 외부 소유 주소 (EOA) 에 대한 모든 트랜잭션 목록을 가져온 다음 목록에서 단일 트랜잭션의 세부 정보를 검색합니다.

Example — Go를 사용하여 API 작업 수행 **ListTransactions**

다음 코드를 listTransactions.go ListTransactions디렉터리에 이름이 지정된 파일에 복사합니다.

```
package main

import (
    "fmt"
    "github.com/aws/aws-sdk-go/aws"
    "github.com/aws/aws-sdk-go/aws/session"
    "github.com/aws/aws-sdk-go/service/managedblockchainquery"
    "time"
)

func main() {

    // Set up a session
    ambQuerySession := session.Must(session.NewSessionWithOptions(session.Options{
        Config: aws.Config{
            Region: aws.String("us-east-1"),
        },
    }))
    client := managedblockchainquery.New(ambQuerySession)

    // Inputs for ListTransactions API
    ownerAddress := "0x00000bf26964af9d7eed9e03e53415d********"
    network := managedblockchainquery.QueryNetworkEthereumMainnet
    sortOrder := managedblockchainquery.SortOrderAscending
    fromTime := time.Date(1971, 1, 1, 1, 1, 1, 1, time.UTC)
    toTime := time.Now()
    nonFinal := "NONFINAL"
    // Call ListTransactions API. Transactions that have reached finality are always
 returned
    listTransactionRequest, listTransactionResponse :=
 client.ListTransactionsRequest(&managedblockchainquery.ListTransactionsInput{
        Address: &ownerAddress,
        Network: &network,
        Sort: &managedblockchainquery.ListTransactionsSort{
            SortOrder: &sortOrder,
        },
        FromBlockchainInstant: &managedblockchainquery.BlockchainInstant{
            Time: &fromTime,
        },
        ToBlockchainInstant: &managedblockchainquery.BlockchainInstant{
            Time: &toTime,
        },
```

```
        ConfirmationStatusFilter: &managedblockchainquery.ConfirmationStatusFilter{
            Include: []*string{&nonFinal},
          },
    })
    errors := listTransactionRequest.Send()

    if errors == nil {
        // handle API response
        fmt.Println(listTransactionResponse)
    } else {
        // handle API errors
        fmt.Println(errors)
    }
}
```

파일을 저장한 후 ListTransactions디렉터리 내에서 다음 명령을 사용하여 코드를 실행합니다`go run listTransactions.go`.

다음 출력은 다음과 비슷합니다.

```
{
  Transactions: [
    {
      ConfirmationStatus: "FINAL",
      Network: "ETHEREUM_MAINNET",
      TransactionHash:
 "0x12345ea404b45323c0cf458ac755ecc45985fbf2b18e2996af3c8e8693354321",
      TransactionTimestamp: 2020-06-01 01:59:11 +0000 UTC
    },
    {
      ConfirmationStatus: "FINAL",
      Network: "ETHEREUM_MAINNET",
      TransactionHash:
 "0x1234547c65675d867ebd2935bb7ebe0996e9ec8e432a579a4516c7113bf54321",
      TransactionTimestamp: 2021-09-01 20:06:59 +0000 UTC
    },
     {
      ConfirmationStatus: "NONFINAL",
      Network: "ETHEREUM_MAINNET",
      TransactionHash:
 "0x123459df7c1cd42336cd1c444cae0eb660ccf13ef3a159f05061232a24954321",
      TransactionTimestamp: 2024-01-23 17:10:11 +0000 UTC
    }
```

```
  ]
}
```

Example — Go를 사용하여 **GetTransaction** API 작업 수행

이 예제에서는 이전 출력의 트랜잭션 해시를 사용합니다. 다음 코드를 GetTransaction.go GetTransaction디렉터리에 이름이 지정된 파일에 복사합니다.

```
package main

import (
    "fmt"
    "github.com/aws/aws-sdk-go/aws"
    "github.com/aws/aws-sdk-go/aws/session"
    "github.com/aws/aws-sdk-go/service/managedblockchainquery"
)

func main() {

    // Set up a session
    ambQuerySession := session.Must(session.NewSessionWithOptions(session.Options{
        Config: aws.Config{
            Region: aws.String("us-east-1"),
        },
    }))
    client := managedblockchainquery.New(ambQuerySession)

    // inputs for GetTransaction API
    transactionHash :=
 "0x123452695a82868950d9db8f64dfb2f6f0ad79284a6c461d115ede8930754321"
    network := managedblockchainquery.QueryNetworkEthereumMainnet

    // Call GetTransaction API. This operation will return transaction details for all
    // transactions that are con#rmed on the blockchain, even if they have not
    // reached #nality.
    getTransactionRequest, getTransactionResponse :=
 client.GetTransactionRequest(&managedblockchainquery.GetTransactionInput{
        Network:         &network,
        TransactionHash: &transactionHash,
    })

    errors := getTransactionRequest.Send()
    if errors == nil {
```

```
        // handle API response
        fmt.Println(getTransactionResponse)
    } else {
        // handle API errors
        fmt.Println(errors)
    }
}
```

파일을 저장한 후 GetTransaction디렉터리 내에서 다음 명령을 사용하여 코드를 실행합니다go run GetTransaction.go.

다음 출력은 다음과 비슷합니다.

```
{
  Transaction: {
    BlockHash: "0x000005c6a71d1afbc005a652b6ceca71cd516d97b0fc514c2a1d0f2ca3912345",
    BlockNumber: "11111111",
    CumulativeGasUsed: "5555555",
    EffectiveGasPrice: "44444444444",
    From: "0x9157f4de39ab4c657ad22b9f19997536********",
    GasUsed: "22222",
    Network: "ETHEREUM_MAINNET",
    NumberOfTransactions: 111,
    SignatureR: "0x99999894fd2df2d039b3555dab80df66753f84be475069dfaf6c6103********",
    SignatureS: "0x77777a101e7f37dd2dd0bf878b39080d5ecf3bf082c9bd4f40de783e********",
    SignatureV: 0,
    ConfirmationStatus: "FINAL",
    ExecutionStatus: "SUCCEEDED",
    To: "0x5555564f282bf135d62168c1e513280d********",
    TransactionHash:
 "0x123452695a82868950d9db8f64dfb2f6f0ad79284a6c461d115ede8930754321",
    TransactionIndex: 11,
    TransactionTimestamp: 2022-02-02 01:01:59 +0000 UTC
  }
}
```

GetTokenBalanceAPI는 네이티브 토큰 (ETH 및 BTC) 의 잔액을 확인할 수 있는 방법을 제공하며, 이를 통해 특정 시점의 외부 소유 계정 (EOA) 의 현재 잔고를 가져올 수 있습니다.

Example — **`GetTokenBalance`** API 작업을 사용하여 Go에서 네이티브 토큰의 잔액을 확인하세요.

다음 예시에서는 `GetTokenBalance` API를 사용하여 이더리움 메인넷의 주소 이더 (ETH) 잔고를 가져옵니다. 다음 코드를 디렉터리에 이름이 지정된 `GetTokenBalanceEth.go` 파일에 복사합니다.

GetTokenBalance

```
package main

import (
    "fmt"
    "github.com/aws/aws-sdk-go/aws"
    "github.com/aws/aws-sdk-go/aws/session"
    "github.com/aws/aws-sdk-go/service/managedblockchainquery"
)

func main() {
    // Set up a session
    ambQuerySession := session.Must(session.NewSessionWithOptions(session.Options{
        Config: aws.Config{
            Region: aws.String("us-east-1"),
        },
    }))
    client := managedblockchainquery.New(ambQuerySession)

    // inputs for GetTokenBalance API
    ownerAddress := "0xBeE510AF9804F3B459C0419826b6f225********"
    network := managedblockchainquery.QueryNetworkEthereumMainnet
    nativeTokenId  := "eth" //Ether on Ethereum mainnet

    // call GetTokenBalance API
    getTokenBalanceRequest, getTokenBalanceResponse :=
 client.GetTokenBalanceRequest(&managedblockchainquery.GetTokenBalanceInput{
        TokenIdentifier: &managedblockchainquery.TokenIdentifier{
            Network:         &network,
            TokenId: &nativeTokenId,
        },
        OwnerIdentifier: &managedblockchainquery.OwnerIdentifier{
            Address: &ownerAddress,
        },
    })
    errors := getTokenBalanceRequest.Send()

    if errors == nil {
```

```
        // process API response
        fmt.Println(getTokenBalanceResponse)
    } else {
        // process API errors
        fmt.Println(errors)
    }
}
```

파일을 저장한 후 GetTokenBalance디렉터리 내에서 다음 명령을 사용하여 코드를 실행합니다go run GetTokenBalanceEth.go.

다음 출력은 다음과 비슷합니다.

```
{
  AtBlockchainInstant: {
    Time: 2020-12-05 11:51:01 +0000 UTC
  },
  Balance: "4343260710",
  LastTransactionHash:
 "0x00000ce94398e56641888f94a7d586d51664eb9271bf2b3c48297a50a0711111",
  LastTransactionTime: 2023-03-14 18:33:59 +0000 UTC,
  OwnerIdentifier: {
    Address: "0x12345d31750D727E6A3a7B534255BADd********"
  },
  TokenIdentifier: {
    Network: "ETHEREUM_MAINNET",
    TokenId: "eth"
  }
}
```

Node.js 를 사용하여 아마존 관리형 블록체인 (AMB) 쿼리 API 요청 생성

이러한 Node 예제를 실행하려면 다음 사전 요구 사항이 적용됩니다.

1. 컴퓨터에 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 14 이상을 사용하고 있는지 확인합니다. 필요한 경우 명령을 사용한 다음 nvm install 14 명령을 사용하여 버전 14를 설치할 수 있습니다. nvm use 14

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. 다음에서 강조 표시된 값을 IAM 사용자 계정의 적절한 값으로 바꿉니다.

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

> **Note**
>
> - 사전 요구 사항을 모두 완료한 후에는 HTTPS를 통해 서명된 요청을 제출하여 Amazon Managed Blockchain (AMB) 쿼리 API 작업에 액세스하고 Node.js 내 네이티브 https 모듈을 사용하여 요청하거나 AXIOS와 같은 타사 라이브러리를 사용하여 AMB 쿼리에서 데이터를 검색할 수 있습니다.
> - 이 예시에서는 Node.js 타사 HTTP 클라이언트를 사용하지만 AWS JavaScript SDK를 사용하여 AMB 쿼리에 요청할 수도 있습니다.
> - 다음 예제는 Axios와 SigV4용 AWS SDK 모듈을 사용하여 AMB 쿼리 API 요청을 만드는 방법을 보여줍니다.

다음 package.json 파일을 로컬 환경의 작업 디렉터리에 복사합니다.

```
{
  "name": "amb-query-examples",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "@aws-crypto/sha256-js": "^4.0.0",
    "@aws-sdk/credential-provider-node": "^3.360.0",
    "@aws-sdk/protocol-http": "^3.357.0",
    "@aws-sdk/signature-v4": "^3.357.0",
```

```
    "axios": "^1.4.0"
  }
}
```

Example — AMB Query API를 사용하여 특정 외부 소유 주소 (EOA) 에서 과거 토큰 잔고 검색 **`GetTokenBalance`**

`GetTokenBalance`API를 사용하여 다양한 토큰 (예: ERC20, ERC721, ERC1155) 과 네이티브 코인 (예: ETH 및 BTC) 의 잔액을 가져올 수 있으며, 이를 사용하여 과거 `timestamp` (Unix 타임스탬프 - 초) 를 기반으로 외부 소유 계정 (EOA) 의 현재 잔액을 가져올 수 있습니다. 이 예시에서는 `GetTokenBalance`API를 사용하여 이더리움 메인넷에 있는 ERC20 토큰 USDC의 주소 잔고를 가져옵니다.

`GetTokenBalance`API를 테스트하려면 다음 코드를 라는 `token-balance.js` 파일에 복사하고 파일을 동일한 작업 디렉터리에 저장합니다.

```
const axios = require('axios').default;
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-query',
  region: 'us-east-1',
  sha256: SHA256,
});

const queryRequest = async (path, data) => {
  //query endpoint
  let queryEndpoint = `https://managedblockchain-query.us-east-1.amazonaws.com/
${path}`;

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

  // create an HTTP Request object
  const req = new HttpRequest({
    hostname: url.hostname.toString(),
    path: url.pathname.toString(),
```

```
    body: JSON.stringify(data),
    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: queryEndpoint, data: data})

    console.log(response.data)
  } catch (error) {
    console.error('Something went wrong: ', error)
    throw error
  }


}


let methodArg = 'get-token-balance';

let dataArg = {
  " atBlockchainInstant": {
    "time": 1688071493
  },
  "ownerIdentifier": {
      "address": "0xf3B0073E3a7F747C7A38B36B805247B2********" // externally owned
 address
  },
  "tokenIdentifier": {
      "contractAddress":"0xA0b86991c6218b36c1d19D4a2e9Eb0cE********", //USDC contract
 address
      "network":"ETHEREUM_MAINNET"
  }
}
```

```
//Run the query request.
queryRequest(methodArg, dataArg);
```

코드를 실행하려면 파일과 같은 디렉터리에서 터미널을 열고 다음 명령을 실행합니다.

```
npm i
node token-balance.js
```

이 명령은 스크립트를 실행하고 코드에 정의된 인수를 전달하여 이더리움 메인넷에 나열된 EOA의 ERC20 USDC 잔액을 요청합니다. 응답은 다음과 유사합니다.

```
{
 atBlockchainInstant: { time: 1688076218 },
 balance: '140386693440144',
 lastUpdatedTime: { time: 1688074727 },
 ownerIdentifier: { address: '0xf3b0073e3a7f747c7a38b36b805247b2********' },
 tokenIdentifier: {
   contractAddress: '0xa0b86991c6218b36c1d19d4a2e9eb0ce********',
   network: 'ETHEREUM_MAINNET'
 }
```

Python을 사용하여 아마존 관리형 블록체인 (AMB) 쿼리 API 요청 생성

이러한 Python 예제를 실행하려면 다음 전제 조건이 적용됩니다.

1. 컴퓨터에 Python이 설치되어 있어야 합니다. 여기에서 해당 OS의 설치 지침을 찾을 수 있습니다.
2. 파이썬용 AWS SDK (Boto3) 를 설치합니다.
3. AWS 명령줄 인터페이스를 설치하고 명령을 `aws configure` 실행하여`Access Key ID`, `Secret Access Key` 및 변수를 설정합니다. `Region`

모든 사전 요구 사항을 완료한 후에는 HTTPS를 통한 AWS Python용 SDK를 사용하여 Amazon Managed Blockchain (AMB) 쿼리 API 요청을 생성할 수 있습니다.

다음 Python 예제는 boto3의 모듈을 사용하여 필수 SigV4 헤더가 부착된 요청을 AMB 쿼리 API 작업에 보냅니다. `ListTransactionEvents` 이 예제는 이더리움 메인넷에서 지정된 트랜잭션이 내보낸 이벤트 목록을 검색합니다.

다음 `list-transaction-events.py` 파일을 로컬 환경의 작업 디렉터리에 복사하세요.

```
import json
from botocore.auth import SigV4Auth
from botocore.awsrequest import AWSRequest
from botocore.session import Session
from botocore.httpsession import URLLib3Session

def signed_request(url, method,  params, service, region):

    session = Session()
    sigv4 = SigV4Auth(session.get_credentials(), service, region)
    data = json.dumps(params)
    request = AWSRequest(method, url, data=data)
    sigv4.add_auth(request)
    http_session = URLLib3Session()
    response = http_session.send(request.prepare())

    return(response)

url = 'https://managedblockchain-query.us-east-1.amazonaws.com/list-transaction-events'
method = 'POST'
params = {
'network': 'ETHEREUM_MAINNET',
'transactionHash': '0x125714bb4db48757007fff2671b37637bbfd6d47b3a4757ebbd0c5222984f905'
}
service = 'managedblockchain-query'
region = 'us-east-1'

# Call the listTransactionEvents operation. This operation will return transaction
 details for
# all transactions that are con#rmed on the blockchain, even if they have not reached
# #nality.
listTransactionEvents = signed_request(url, method, params, service, region)

print(json.loads(listTransactionEvents.content.decode('utf-8')))
```

샘플 코드를 ListTransactionEvents 실행하려면 작업 디렉터리에 파일을 저장한 다음 명령을 실행합니다python3 list-transaction-events.py. 이 명령은 스크립트를 실행하고 코드에 정의된 인수를 전달하여 이더리움 메인넷에서 지정된 트랜잭션 해시와 관련된 이벤트를 요청합니다. 응답은 다음과 유사합니다.

```
{
 'events':
```

```
 [
  {
      'contractAddress': '0x95ad61b0a150d79219dcf64e1e6cc01f********',
      'eventType': 'ERC20_TRANSFER',
      'from': '0xab5801a7d398351b8be11c439e05c5b3********',
      'network': 'ETHEREUM_MAINNET',
      'to': '0xdead000000000000000420694206942********',
      'transactionHash':
 '0x125714bb4db48757007fff2671b37637bbfd6d47b3a4757ebbd0c522********',
      'value': '4102419967718718947718261747554 64'
  }
 ]
}
```

에서 Amazon Managed Blockchain (AMB) 쿼리를 AWS Management Console 사용하여 작업을 실행합니다. GetTokenBalance

다음 예는 Amazon Managed Blockchain (AMB) 쿼리를 사용하여 이더리움 메인넷에서 토큰 잔고를 확인하는 방법을 보여줍니다. AWS Management Console

Example

1. https://console.aws.amazon.com/managedblockchain/ 에서 아마존 매니지드 블록체인 콘솔을 엽니다.
2. 쿼리 섹션에서 쿼리 편집기를 선택합니다.
3. 블록체인 네트워크로 이더리움_메인넷을 선택합니다.
4. 쿼리 GetTokenBalance유형으로 선택하세요.
5. 토큰의 블록체인 주소를 입력합니다.
6. 토큰의 계약 주소를 입력합니다.
7. 토큰의 선택적 토큰 ID를 입력합니다.
8. 토큰 잔액의 At 날짜를 선택합니다.
9. 토큰 잔액에 At time (선택 사항) 을 입력합니다.
10. 쿼리 실행을 선택합니다.

AMB Query가 쿼리를 실행하고 쿼리 결과 창에 결과가 표시됩니다.

아마존 관리형 블록체인 (AMB) 쿼리를 사용한 사용 사례

이 항목에서는 AMB 쿼리 사용 사례 목록을 제공합니다.

주제

현재 및 과거 토큰 잔고를 쿼리합니다.

`GetTokenBalance`API는 외부 소유 계정 (EOA) 의 범용 타임스탬프 (Unix 타임스탬프, 초 단위) 를 사용하여 지원되는 토큰 (ERC20, ERC721, ERC1155) 과 네이티브 코인 (ETH, BTC) 의 잔액을 가져와 현재 또는 과거 잔액을 가져옵니다. 예를 들어, `GetTokenBalance` API 연산을 사용하여 이더리움 메인넷에 있는 ERC20 토큰 USDC의 주소 잔고를 가져올 수 있습니다. API 작업을 사용하여 토큰과 네이티브 코인의 잔액을 일괄 검색할 수도 있습니다. `BatchGetTokenBalance`

자세한 내용은 Amazon Managed Blockchain (AMB) 쿼리 참조 안내서를 참조하십시오.

과거 거래 데이터 검색

Amazon Managed Blockchain (AMB) 쿼리를 사용하면 이더리움 및 비트코인과 같은 퍼블릭 블록체인에서 과거 데이터를 검색할 수 있습니다. 이 기능을 사용하면 블록체인 지갑에서 거래 내역을 검색하거나 거래 해시를 기반으로 거래에 대한 컨텍스트 정보를 제공하는 등 여러 사용 사례가 가능합니다. `ListTransactions`API 작업을 사용하여 이더리움 메인넷의 특정 외부 소유 주소 (EOA) 에 대한 트랜잭션 목록을 가져온 다음 `GetTransaction`API 작업을 사용하여 목록에서 단일 트랜잭션에 대한 트랜잭션 세부 정보를 검색할 수 있습니다.

자세한 내용은 Amazon Managed Blockchain (AMB) 쿼리 참조 안내서를 참조하십시오.

해당 주소의 모든 토큰 잔고 가져오기

ListTokenBalancesAPI 작업을 사용하여 지갑, 사용자 인터페이스, web3 유틸리티 등의 잔액을 확인할 수 있습니다. 이 API 연산은 단일 API 작업을 사용하여 주어진 퍼블릭 블록체인의 토큰 (ERC20, ERC721, ERC1155) 및 네이티브 코인 (ETH, BTC) 간 주소의 모든 잔고 목록을 반환합니다. 예를 들어 외부 소유 주소 (EOA) 와 네트워크 (이더리움 메인넷) 를 제공하면 응답으로 토큰 및 네이티브 코인 잔고 목록을 받을 수 있습니다.

자세한 내용은 Amazon Managed Blockchain (AMB) 쿼리 참조 안내서를 참조하십시오.

트랜잭션에 대해 발생한 이벤트를 나열합니다.

ListTransactionEventsAPI 작업을 사용하여 해시 (거래 식별자) 로 식별되는 특정 거래의 결과로 발생한 계약 이벤트 목록을 검색할 수 있습니다. 예를 들어, 이더리움 블록체인에서 ERC20 토큰 컨트랙트의 함수를 호출하는 트랜잭션의 결과 이벤트 (예: ERC20 컨트랙트의 전송 이벤트 또는 인출 이벤트) 를 검색하는 데 사용할 ListTransactionEvents수 있습니다.

자세한 내용은 Amazon Managed Blockchain (AMB) 쿼리 참조 안내서를 참조하십시오.

계약을 통해 발행된 모든 토큰 가져오기

ListTokenBalancesAPI 작업을 사용하여 계약 주소를 입력으로 전달하면 컨트랙트에서 발행된 모든 지원 토큰 (ERC20, ERC721, ERC1155) 의 목록을 반환할 수 있습니다. 예를 들어, API 작업을 사용하여 이더리움 블록체인의 ERC721 계약 표준에 따라 발행된 대체 불가능한 토큰 (NFT) 과 관련된 정보를 검색할 수 있습니다. ListTokenBalances

자세한 내용은 Amazon Managed Blockchain (AMB) 쿼리 참조 안내서를 참조하십시오.

계약을 나열하고 계약 정보를 얻으십시오.

ListAssetContractsAPI 작업을 사용하여 지정된 주소로 배포된 ERC-721, ERC-1155 또는 ERC-20 계약을 나열할 수 있습니다. 또한 계약 주소가 있는 경우 GetAssetContractAPI 작업을 사용하여 계약 유형 배포자 주소 및 관련 토큰 메타데이터와 같은 계약 속성을 검색할 수 있습니다.

자세한 내용은 Amazon Managed Blockchain (AMB) 쿼리 참조 안내서를 참조하십시오.

아마존 매니지드 블록체인 (AMB) 쿼리 API 레퍼런스

Amazon Managed Blockchain (AMB) 쿼리는 지원되는 블록체인을 쿼리하기 위한 API 작업을 제공합니다. 여기에는 토큰, 트랜잭션 및 계약을 쿼리하기 위한 API가 포함됩니다. 자세한 내용은 AMB 쿼리 API 참조를 참조하십시오.

아마존 관리형 블록체인 (AMB) 쿼리의 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 기업 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) 쿼리에 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 Query를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 항목에서는 보안 및 규정 준수 목표를 충족하도록 AMB Query를 구성하는 방법을 보여줍니다. AMB 쿼리 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법도 배울 수 있습니다.

주제

데이터 암호화

데이터 암호화는 승인되지 않은 사용자가 블록체인 네트워크 및 관련 데이터 저장 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 관리형 블록체인은 HTTPS/TLS 연결을 사용하여 클라이언트에서 서비스 엔드포인트로 전송되는 모든 데이터를 암호화합니다. AWS CLI AWS

아마존 관리형 블록체인 (AMB) 쿼리의 자격 증명 및 액세스 관리

AWS Identity and Access Management (IAM) 은 관리자가 리소스에 대한 액세스를 안전하게 제어할 수 AWS 서비스 있도록 도와줍니다. AWS IAM 관리자는 AMB 쿼리 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 Query에서 수행하는 작업에 따라 사용 방법 AWS Identity and Access Management (IAM) 이 다릅니다.

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

서비스 관리자 — 회사에서 AMB Query 리소스를 담당하는 경우 AMB Query에 대한 전체 액세스 권한이 있을 것입니다. 서비스 사용자가 액세스해야 하는 AMB Query 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 IAM을 AMB Query와 함

께 사용하는 방법에 대한 자세한 내용은 을 참조하십시오. 아마존 관리형 블록체인 (AMB) 쿼리가 IAM과 작동하는 방식

IAM 관리자 - IAM 관리자라면 정책을 작성하여 AMB Query에 대한 액세스를 관리하는 방법에 대해 자세히 알고 싶을 것입니다. IAM에서 사용할 수 있는 AMB 쿼리 ID 기반 정책의 예를 보려면 을 참조하십시오. 아마존 관리형 블록체인 (AMB) 쿼리에 대한 ID 기반 정책 예제

ID를 통한 인증

인증은 ID 자격 증명을 AWS 사용하여 로그인하는 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 사람이 ID 공급자와의 페더레이션을 사용하여 임시 자격 증명을 사용하여 AWS 서비스 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 역

할을 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. 크로스 계정 액세스에 대한 역할과 리소스 기반 정책의 차이점을 알아보려면 IAM 사용 설명서의 IAM의 크로스 계정 리소스 액세스를 참조하세요.
- 서비스 간 액세스 — 일부는 다른 AWS 서비스서비스의 기능을 AWS 서비스 사용합니다. 예를 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. 인스턴스 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 관련되어 있을 때 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 참조하십시오.

아마존 관리형 블록체인 (AMB) 쿼리가 IAM과 작동하는 방식

IAM을 사용하여 AMB 쿼리에 대한 액세스를 관리하기 전에 AMB 쿼리에 사용할 수 있는 IAM 기능에 대해 알아보십시오.

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

IAM 특성	AMB 쿼리 지원
ID 기반 정책	예
리소스 기반 정책	아니요
정책 작업	예
정책 리소스	아니요
정책 조건 키	아니요
ACL	아니요

IAM 특성	AMB 쿼리 지원
ABAC(정책 내 태그)	아니요
임시 보안 인증	예
보안 주체 권한	예
서비스 역할	아니요
서비스 연결 역할	아니요

AMB Query 및 기타 AWS 서비스가 대부분의 IAM 기능과 어떻게 작동하는지 자세히 알아보려면 IAM 사용 설명서의 IAM과 함께 작동하는AWS 서비스를 참조하십시오.

AMB 쿼리에 대한 자격 증명 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 쿼리의 ID 기반 정책 예제

AMB Query ID 기반 정책의 예를 보려면 을 참조하십시오. 아마존 관리형 블록체인 (AMB) 쿼리에 대한 ID 기반 정책 예제

AMB 쿼리 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 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, 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 쿼리에 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 쿼리 작업 목록을 보려면 서비스 권한 부여 참조의 Amazon Managed Blockchain (AMB) 쿼리로 정의된 작업을 참조하십시오.

AMB 쿼리의 정책 작업은 작업 앞에 다음 접두사를 사용합니다.

```
managedblockchain-query:
```

단일 문에서 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": [
      "managedblockchain-query::ListTransaction",
      "managedblockchain-query::GetTransaction"
         ]
```

AMB Query ID 기반 정책의 예를 보려면 을 참조하십시오. 아마존 관리형 블록체인 (AMB) 쿼리에 대한 ID 기반 정책 예제

AMB 쿼리의 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 `Resource`또는 `NotResource`요소가 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 쿼리 리소스 유형 및 해당 ARN 목록을 보려면 서비스 권한 부여 참조의 Amazon Managed Blockchain (AMB) 쿼리로 정의된 리소스를 참조하십시오. 각 리소스의 ARN을 지정할 수 있는 작업에 대해 알아보려면 Amazon Managed Blockchain (AMB) 쿼리에서 정의된 작업을 참조하십시오.

AMB 쿼리 ID 기반 정책의 예를 보려면 을 참조하십시오. 아마존 관리형 블록체인 (AMB) 쿼리에 대한 ID 기반 정책 예제

AMB 쿼리의 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 쿼리 조건 키 목록을 보려면 서비스 권한 부여 참조의 Amazon Managed Blockchain (AMB) 쿼리의 조건 키를 참조하십시오. 조건 키를 사용할 수 있는 작업 및 리소스를 알아보려면 Amazon Managed Blockchain (AMB) 쿼리로 정의된 작업을 참조하십시오.

AMB 쿼리 ID 기반 정책의 예를 보려면 을 참조하십시오. 아마존 관리형 블록체인 (AMB) 쿼리에 대한 ID 기반 정책 예제

AMB 쿼리의 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.

ABAC (AMB 쿼리 포함)

ABAC 지원(정책의 태그)	아니요

ABAC(속성 기반 액세스 통제)는 속성에 근거하여 권한을 정의하는 권한 부여 전략입니다. AWS에서는 이러한 속성을 태그라고 합니다. IAM 개체 (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 쿼리에 임시 자격 증명 사용

임시 보안 인증 지원	예

임시 자격 증명을 사용하여 로그인하면 작동하지 AWS 서비스 않는 것도 있습니다. 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 자격 증명을 동적으로 생성할 것을 권장합니다. 자세한 정보는 IAM의 임시 보안 자격 증명 섹션을 참조하십시오.

AMB 쿼리에 대한 서비스 간 보안 주체 권한

전달 액세스 세션(FAS) 지원	예

IAM 사용자 또는 역할을 사용하여 작업을 수행하는 AWS경우 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 요청하기 위한 요청과 함께 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 쿼리의 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 Query 기능이 작동하지 않을 수 있습니다. AMB Query에서 이에 대한 지침을 제공하는 경우에만 서비스 역할을 편집하십시오.

AMB 쿼리의 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 해당 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) 쿼리에 대한 ID 기반 정책 예제

기본적으로 사용자와 역할에는 AMB 쿼리 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 쿼리에 정의된 작업 및 리소스 유형에 대한 자세한 내용은 서비스 인증 참조의 Amazon Managed Blockchain (AMB) 쿼리의 작업, 리소스 및 조건 키를 참조하십시오.

주제

- 정책 모범 사례
- 사용자가 자신의 고유한 권한을 볼 수 있도록 허용
- 특정 아마존 관리형 블록체인 (AMB) 쿼리 API 작업에 액세스

정책 모범 사례

ID 기반 정책에 따라 계정에서 AMB 쿼리 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 줄이는 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.

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": "*"
        }
    ]
}
```

특정 아마존 관리형 블록체인 (AMB) 쿼리 API 작업에 액세스

> **Note**
>
> AMB 쿼리에 액세스하여 API를 호출하려면 AMB 쿼리에 대한 적절한 IAM 권한이 있는 사용자 자격 증명 (AWS_ACCESS_KEY_ID및AWS_SECRET_ACCESS_KEY) 이 필요합니다.

Example 모든 아마존 관리형 블록체인 (AMB) 쿼리 API에 액세스하기 위한 IAM 정책

이 예시는 모든 AMB 쿼리 API에 AWS 계정 대한 액세스 권한을 IAM 사용자에게 부여합니다.

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

Example 아마존 관리형 블록체인 (AMB) 쿼리 **`ListTransactions`** 및 API에 액세스하기 위한 IAM 정책 **`GetTransaction`**

이 예시에서는 IAM 사용자에게 AMB 쿼리 및 API에 AWS 계정 대한 액세스 권한을 부여합니다. `ListTransaction` `GetTransaction`

> **Note**
>
> 예제의 API를 다른 API로 대체하거나 추가하여 다른 또는 더 많은 API에 대한 액세스 권한을 부여할 수 있습니다. AMB 쿼리 API 목록은 아마존 관리형 블록체인 (AMB) 쿼리 API 참조 가이드를 참조하십시오.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AccessAMBQueryAPIs",
            "Effect": "Allow",
            "Action": [
                "managedblockchain-query:ListTransactions",
                "managedblockchain-query:GetTransaction"
            ],
            "Resource": "*"
        }
    ]
}
```

아마존 관리형 블록체인 (AMB) 쿼리 ID 및 액세스 문제 해결

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

주제

- AMB Query에서 작업을 수행할 권한이 없습니다.

AMB Query에서 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-query::*GetWidget*권한이 없을 때 발생합니다.

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

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

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

아마존 매니지드 블록체인 (AMB) 쿼리 API 사용 지표: 아마존 CloudWatch

아마존의 API 사용 지표 CloudWatch

게시된 API 사용 지표는 Amazon Managed Blockchain (AMB) 쿼리 서비스 할당량에 CloudWatch 해당합니다. 사용량이 서비스 할당량에 가까워지면 알리도록 경보를 구성할 수 있습니다. 서비스 CloudWatch 할당량과의 통합에 대한 자세한 내용은 Amazon 사용 CloudWatch 설명서의 AWS 사용 지표를 참조하십시오.

AMB Query는 `AWS/Usage` 네임스페이스에 다음 API 메트릭을 서비스 이름과 함께 게시합니다. `Amazon Managed Blockchain Query`

지표	설명
`CallCount`	AMB 쿼리에서 API에 대한 총 호출 수입니다. SUM은 지정된 기간 동안 API에 대한 총 직접 호출 수를 나타냅니다.

Amazon Managed Blockchain (AMB) 쿼리는 사용 지표를 `AWS/Usage` 네임스페이스에 다음과 같은 차원으로 게시합니다.

측정기준	설명
Service	리소스가 포함된 AWS 서비스의 이름. `Amazon Managed Blockchain Query` 은 항상 이 차원의 값입니다.
유형	보고되는 개체의 유형. `API`은 항상 이 차원의 값입니다.
Resource	보고되는 리소스 유형. 사용된 AMB Query API 작업의 이름이 이 차원의 값이 됩니다.

측정기준	설명
Class	보고되는 리소스의 클래스. None은 항상 이 차원의 값입니다.

AMB 쿼리 사용 설명서의 문서 기록

다음 표에는 AMB Query의 설명서 릴리스가 설명되어 있습니다.

변경 사항	설명	날짜
AMB 쿼리는 비트코인 트랜잭션 식별자와 트랜잭션 해시를 지원합니다.	비트코인 네트워크의 경우 AMB 쿼리 API 작업은 트랜잭션 식별자 (`transactionId`) 와 트랜잭션 해시 () 를 모두 지원합니다. `transactionHash`	2024년 3월 21일
Amazon의 API 사용 지표 지원 CloudWatch	AMB Query는 API 사용 지표에 대한 지원을 추가했습니다. CloudWatch 이러한 사용량 지표는 AMB 쿼리 서비스 할당량에 해당합니다.	2024년 2월 8일
완료되지 않은 거래에 대한 지원	AMB Query는 완료되지 않은 트랜잭션에 대한 지원을 추가했습니다. 또한 작업 응답에서 `status` 속성에 대한 지원을 제거합니다. `GetTransaction` 대신 `confirmationStatus` 및 `executionStatus` 속성을 사용하여 트랜잭션 상태를 확인합니다.	2024년 2월 1일
거래 데이터 유형의 `status` 속성 지원 중단	Amazon Managed Blockchain (AMB) 쿼리는 트랜잭션 데이터 유형에서 해당 `status` 속성을 더 이상 사용하지 않습니다. `confirmationStatus` 및 `executionStatus` 필드를 사용하여 트랜잭션이 또는인	2023년 12월 20일

	지 확인해야 합니다. `status` `FINAL FAILED`	
세폴리아 테스트넷 지원	Amazon Managed Blockchain (AMB) 쿼리는 이제 이더리움 세폴리아 테스트넷에서 쿼리를 지원합니다.	2023년 10월 19일
자산 계약 지원	`ListAssetContracts` API 작업을 사용하여 지정된 주소로 배포된 목록을 표시할 수 있습니다. 또한 계약 주소가 있는 경우 `GetAssetContract` API 작업을 사용하여 계약 세부 정보를 검색할 수 있습니다.	2023년 10월 16일
비트코인 테스트넷 지원	Amazon Managed Blockchain (AMB) 쿼리는 이제 비트코인 테스트넷에서 쿼리를 지원합니다.	2023년 10월 16일
최초 릴리스	AMB 쿼리 서비스의 최초 출시.	2023년 7월 27일

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