aws

Panduan Developer

Kueri Blockchain yang Dikelola Amazon

Kueri Blockchain yang Dikelola Amazon: Panduan Developer

Table of Contents

Apa itu Kueri Amazon Managed Blockchain (AMB)? 1
Apakah Anda pengguna AMB Query pertama kali? 1
Konsep utama 2
Pertimbangan dan batasan untuk menggunakan Kueri Amazon Managed Blockchain (AMB) 2
Menyiapkan 6
Prasyarat dan pertimbangan 6
Mendaftar untuk AWS 6
Buat pengguna IAM dengan izin yang sesuai 6
Instal dan konfigurasikan AWS Command Line Interface 7
Gunakan Konsol Manajemen AWS untuk menanyakan blockchain menggunakan AMB Query 7
Memulai 9
Buat kebijakan IAM 9
Contoh menggunakan Go 10
Contoh menggunakan Node.js 16
Contoh menggunakan Python 20
Contoh menggunakan Konsol Manajemen AWS 22
Kasus penggunaan Kueri AMB 24
Kueri saldo token saat ini dan historis 24
Mengambil data transaksi historis 24
Dapatkan semua saldo token untuk alamat tertentu 24
Daftar peristiwa yang dipancarkan untuk transaksi 25
Dapatkan semua token yang dicetak oleh kontrak 25
Daftar kontrak dan dapatkan informasi kontrak 25
Referensi API Kueri AMB 27
Keamanan 28
Enkripsi data 29
Enkripsi saat bergerak 29
Manajemen identitas dan akses 29
Audiens 29
Mengautentikasi dengan identitas 30
Mengelola akses menggunakan kebijakan 31
Bagaimana Kueri Amazon Managed Blockchain (AMB) bekerja dengan IAM 33
Contoh kebijakan berbasis identitas 39

Pemecahan masalah 43
Metrik penggunaan API 44
Metrik penggunaan API di Amazon CloudWatch 44
Riwayat dokumen 46
..... xlviii

Apa itu Kueri Amazon Managed Blockchain (AMB)?

Amazon Managed Blockchain (AMB) adalah layanan yang dikelola sepenuhnya yang dirancang untuk membantu Anda membangun aplikasi Web3 yang tangguh di blockchain publik dan pribadi. Gunakan AMB Access untuk akses instan dan tanpa server ke beberapa blockchain. Bangun aplikasi siap Web3 Anda tanpa perlu menggunakan infrastruktur blockchain khusus dan menjaganya tetap terhubung ke jaringan blockchain. Dengan AMB Query, Anda dapat menggunakan operasi API yang ramah pengembang untuk mengakses data real-time dan historis dari beberapa blockchain. Data blockchain standar dapat diintegrasikan dengan layanan AWS, tanpa memerlukan infrastruktur blockchain khusus atau ETL (ekstrak, transformasi, dan muat). Semua fitur AMB diskalakan dengan aman untuk pembuatan aplikasi konsumen tingkat kelembagaan dan arus utama.

Amazon Managed Blockchain (AMB) Query menyediakan akses tanpa server ke kumpulan data multi-blockchain standar dengan operasi API yang ramah pengembang. Anda dapat menggunakan AMB Query untuk mengirimkan aplikasi dengan cepat yang memerlukan data dari satu atau lebih blockchain publik, tanpa memerlukan overhead untuk mengurai data blockchain, melacak kontrak, dan memelihara infrastruktur pengindeksan khusus. Baik Anda menganalisis saldo token historis untuk token yang dapat dipertukarkan atau token yang tidak dapat dipertukarkan (NFTs), melihat riwayat transaksi untuk alamat dompet tertentu, atau melakukan analisis data pada distribusi cryptocurrency asli seperti Ether, AMB Query memberi Anda akses ke data blockchain.

Apakah Anda pengguna AMB Query pertama kali?

Jika Anda adalah pengguna pertama kali AMB Query, kami sarankan Anda mulai dengan membaca bagian berikut:

- Konsep kunci: Kueri Amazon Managed Blockchain (AMB)
- Menyiapkan Kueri Amazon Managed Blockchain (AMB)
- Memulai Kueri Amazon Managed Blockchain (AMB)
- Kasus penggunaan dengan Kueri Amazon Managed Blockchain (AMB)

Konsep kunci: Kueri Amazon Managed Blockchain (AMB)

> **Note**
>
> Panduan ini mengasumsikan bahwa Anda terbiasa dengan konsep blockchain penting. Konsep-konsep ini termasuk desentralisasi, token, kontrak, transaksi, dompet proof-of-work, kunci publik dan pribadi, staking, penambangan, halvings, dan lainnya.

Amazon Managed Blockchain (AMB) Query memberi Anda akses mudah ke data jaringan multi-blockchain, yang memudahkan Anda mengekstrak data kontekstual yang terkait dengan aktivitas blockchain. Anda dapat menggunakan AMB Query untuk membaca data dari jaringan blockchain publik, seperti Bitcoin Mainnet dan Ethereum Mainnet. Anda juga bisa mendapatkan informasi, seperti saldo alamat saat ini dan historis, atau Anda bisa mendapatkan daftar transaksi blockchain untuk jangka waktu tertentu. Selain itu, Anda bisa mendapatkan detail transaksi tertentu, seperti peristiwa transaksi, yang dapat Anda analisis atau gunakan lebih lanjut dalam logika bisnis untuk aplikasi Anda.

Pertimbangan dan batasan untuk menggunakan Kueri Amazon Managed Blockchain (AMB)

Saat Anda menggunakan AMB Query, pertimbangkan hal berikut:

- Wilayah yang Tersedia

  AMB Query didukung di Wilayah AS Timur (Virginia N.)`us-east-1`.

- Titik akhir layanan

  AMB Query dapat diakses dengan menggunakan endpoint berikut:

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

- Jaringan blockchain yang didukung

  AMB Query mendukung jaringan blockchain publik berikut:

- Bitcoin Mainnet — Jaringan blockchain Bitcoin publik yang dijamin dengan proof-of-work konsensus, dan di mana cryptocurrency Bitcoin (BTC) dikeluarkan dan ditransaksikan. Transaksi di Mainnet memiliki nilai aktual (yaitu, mereka mengeluarkan biaya riil) dan dicatat pada blockchain publik.
- Bitcoin Testnet — Testnet untuk Mainnet Bitcoin. Bitcoin (BTC) di jaringan ini terpisah dan berbeda dari Mainnet BTC, dan biasanya tidak memiliki nilai apa pun.
- Ethereum Mainnet — Jaringan proof-of-stake utama untuk blockchain Ethereum publik. Transaksi di Mainnet memiliki nilai aktual (yaitu, mereka mengeluarkan biaya riil) dan dicatat pada buku besar yang didistribusikan.
- Seplia Testnet — Testnet untuk Ethereum Mainnet. Ether (ETH) pada jaringan ini terpisah dan berbeda dari Mainnet ETH, dan biasanya tidak memiliki nilai apa pun.

- Token dan kontrak blockchain yang didukung

  AMB Query mendukung token kontrak Ethereum asli dan standar berikut.

  - Token asli blockchain publik
    - Bitcoin (BTC) — Ini adalah token asli dari blockchain terkait Bitcoin.
    - Ether (ETH) — Ini adalah token asli dari blockchain terkait Ethereum.

  - Standar kontrak Ethereum
    - ERC-20 Token Standard — ERC-20 adalah standar untuk token yang dapat dipertukarkan. Ini memiliki properti yang membuat setiap token ERC-20 persis sama (dalam jenis dan nilai) dengan token ERC-20 lain yang dicetak, yang berarti bahwa satu token adalah dan akan selalu sama dengan semua token lainnya. Untuk informasi selengkapnya, lihat Standar Token ERC-20 di Ethereum.org.
    - ERC-721 Non-fungible Token Standard — ERC-721 adalah standar untuk token yang tidak dapat dipertukarkan (). NFTs Jenis token ini unik dan dapat memiliki nilai yang berbeda dari token lain dari kontrak yang sama, mungkin karena usia, kelangkaan, atau properti lainnya. Untuk informasi selengkapnya, lihat Standar Token ERC-721 di Ethereum.org.

      ERC-1155 Multi-token Standard — ERC-1155 adalah standar yang menciptakan antarmuka kontrak yang dapat mewakili dan mengontrol sejumlah jenis token yang dapat dipertukarkan dan tidak dapat dipertukarkan. Dengan cara ini, token ERC-1155 dapat berfungsi sama dengan token ERC-20 dan ERC-721, bahkan berfungsi sebagai keduanya pada saat yang bersamaan. Token ERC-1155 meningkatkan fungsionalitas standar ERC-20 dan ERC-721,

membuatnya lebih efisien, sambil memperbaiki kesalahan implementasi yang jelas. Untuk informasi selengkapnya, lihat Standar Token ERC-1155 di Ethereum.org.

- Finalitas

  Dalam blockchain, finalitas berarti bahwa transaksi yang valid tidak mungkin dibalik. Untuk Bitcoin Mainnet, AMB Query mempertimbangkan transaksi final setelah 6 blok. Untuk Bitcoin Testnet, ia mempertimbangkan transaksi final setelah 6 blok atau 60 menit, mana yang lebih dulu. Untuk jaringan Ethereum yang didukung, AMB Query mempertimbangkan transaksi final setelah 64 blok.

  Saldo token AMB Query dan operasi API kontrak hanya mengembalikan data yang telah mencapai finalitas. Namun, transaksi AMB Query dan transaksi transaksi operasi API dapat mengembalikan data untuk transaksi yang dikonfirmasi pada jaringan blockchain bahkan jika mereka belum mencapai finalitas.

- Alamat NULL tidak didukung

  AMB Query tidak mendukung alamat NULL (`0x0000000000000000000000000000000000000000`).

- Tanda tangan Versi 4 penandatanganan panggilan API

  Saat melakukan panggilan ke Kueri AMB APIs, Anda dapat melakukannya melalui koneksi HTTPS yang diautentikasi menggunakan proses penandatanganan Versi Tanda Tangan 4. Ini berarti bahwa hanya prinsipal IAM resmi di AWS akun yang dapat melakukan panggilan AMB Query API. Untuk melakukan ini, AWS kredensi (ID kunci akses dan kunci akses rahasia) harus diberikan dengan panggilan.

  > **Important**
  >
  > Jangan menyematkan kredensi klien dalam aplikasi yang menghadap pengguna.

- AMB Query mendukung pengidentifikasi transaksi Bitcoin dan hash transaksi

  Untuk jaringan Bitcoin, operasi AMB Query API mendukung pengenal transaksi (`transactionId`) dan hash transaksi (). `transactionHash` `transactionId`Ini adalah hash Double-SHA dari transaksi yang tidak termasuk data saksi. `transactionHash`Ini adalah hash Double-SHA dari transaksi termasuk data saksi (juga dikenal sebagai id transaksi saksi).

Saat menjalankan operasi `GetTransaction`atau `ListTransactionEvents`API untuk jaringan Bitcoin, Anda dapat menentukan salah satu `transactionId` atau. `transactionHash` Juga, semua operasi AMB Query pada jaringan Bitcoin yang mengembalikan baik a `transactionId` atau a `transactionHash` akan mencakup kedua nilai sebagai bagian dari respons.

Menyiapkan Kueri Amazon Managed Blockchain (AMB)

Sebelum Anda menggunakan Kueri Amazon Managed Blockchain (AMB) untuk pertama kalinya, ikuti langkah-langkah di bagian ini untuk membuat AWS akun. Bagian berikut membahas cara memulai menggunakan AMB Query.

Prasyarat dan pertimbangan

Sebelum Anda menggunakan Amazon Web Services untuk pertama kalinya, Anda harus memiliki AWS akun.

Mendaftar untuk AWS

Saat Anda mendaftar ke Amazon Web Services (AWS), AWS akun Anda secara otomatis mendaftar untuk semua Layanan AWS, termasuk Kueri Amazon Managed Blockchain (AMB). Anda hanya akan dikenakan biaya untuk layanan yang digunakan.

Jika Anda Akun AWS sudah memiliki, lanjutkan ke langkah berikutnya. Jika Anda belum memiliki Akun AWS, gunakan prosedur berikut untuk membuatnya.

Untuk membuat AWS akun

1. Buka https://portal.aws.amazon.com/billing/pendaftaran.
2. Ikuti petunjuk online.

   Bagian dari prosedur pendaftaran melibatkan menerima panggilan telepon atau pesan teks dan memasukkan kode verifikasi pada keypad telepon.

   Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWSdibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

Buat pengguna IAM dengan izin yang sesuai

Untuk membuat dan bekerja dengan AMB Query, Anda harus membuat prinsipal AWS Identity and Access Management (IAM) (pengguna atau grup) dengan izin yang memungkinkan tindakan Blockchain Terkelola yang diperlukan.

Hanya prinsipal IAM yang dapat membuat permintaan AMB Query API. Saat melakukan panggilan ke Kueri AMB APIs, Anda dapat melakukannya melalui koneksi HTTPS yang diautentikasi menggunakan proses penandatanganan Versi Tanda Tangan 4. Ini berarti bahwa hanya prinsipal IAM resmi di AWS akun yang dapat melakukan panggilan AMB Query API. Untuk melakukan ini, AWS kredensi (ID kunci akses dan kunci akses rahasia) harus diberikan dengan panggilan.

Untuk informasi tentang cara membuat pengguna IAM, lihat Membuat pengguna IAM di akun Anda AWS. Untuk informasi selengkapnya tentang cara melampirkan kebijakan izin ke pengguna, lihat Mengubah izin untuk pengguna IAM. Untuk contoh kebijakan izin yang dapat Anda gunakan untuk memberikan izin kepada pengguna agar bekerja dengan Kueri AMB, lihat. Contoh kebijakan berbasis identitas untuk Kueri Amazon Managed Blockchain (AMB)

Instal dan konfigurasikan AWS Command Line Interface

Jika Anda belum melakukannya, instal Antarmuka AWS Baris Perintah (CLI) terbaru untuk bekerja dengan AWS sumber daya dari terminal. Untuk informasi selengkapnya, lihat Menginstal atau memperbarui versi terbaru AWS CLI.

> **Note**
>
> Untuk akses CLI, Anda memerlukan ID kunci akses dan kunci akses rahasia. Gunakan kredensi sementara alih-alih kunci akses jangka panjang jika memungkinkan. Kredensi sementara mencakup ID kunci akses, kunci akses rahasia, dan token keamanan yang menunjukkan kapan kredensialnya kedaluwarsa. Untuk informasi selengkapnya, lihat Menggunakan kredensi sementara dengan AWS sumber daya di Panduan Pengguna IAM.

Gunakan Konsol Manajemen AWS untuk menanyakan blockchain menggunakan Kueri Amazon Managed Blockchain (AMB)

Anda dapat mengakses Kueri Amazon Managed Blockchain (AMB) dan membuat kueri pada jaringan blockchain yang didukung menggunakan. Konsol Manajemen AWS Langkah-langkah berikut menunjukkan cara melakukan ini:

1. Buka konsol Amazon Managed Blockchain di https://console.aws.amazon.com/managedblockchain/.
2. Pilih Editor kueri dari bagian Kueri.

3. Pilih dari salah satu jaringan Blockchain yang didukung.
4. Pilih jenis Kueri yang ingin Anda jalankan.
5. Masukkan parameter yang relevan untuk jenis Kueri yang Anda pilih dan Jalankan kueri.

AMB Query akan menjalankan query Anda dan Anda akan melihat hasilnya di jendela hasil Query.

Memulai Kueri Amazon Managed Blockchain (AMB)

Gunakan step-by-step tutorial di bagian ini untuk mempelajari cara melakukan tugas dengan menggunakan Kueri Amazon Managed Blockchain (AMB). Prosedur ini membutuhkan beberapa prasyarat. Jika Anda baru mengenal AMB Query, Anda dapat meninjau bagian Pengaturan dari panduan ini. Untuk informasi selengkapnya, lihat Menyiapkan Kueri Amazon Managed Blockchain (AMB).

> **Note**
>
> Beberapa variabel dalam contoh-contoh ini sengaja dikaburkan. Ganti dengan yang valid dari Anda sendiri sebelum menjalankan contoh-contoh ini.

Topik

- Membuat kebijakan IAM untuk mengakses operasi AMB Query API
- Buat permintaan API Kueri Amazon Managed Blockchain (AMB) dengan menggunakan Go
- Buat permintaan API Kueri Amazon Managed Blockchain (AMB) dengan menggunakan Node.js
- Buat permintaan API Kueri Amazon Managed Blockchain (AMB) dengan menggunakan Python
- Gunakan Kueri Amazon Managed Blockchain (AMB) Konsol Manajemen AWS untuk menjalankan operasi GetTokenBalance

Membuat kebijakan IAM untuk mengakses operasi AMB Query API

Untuk membuat permintaan AMB Query API, Anda harus menggunakan kredensi pengguna (AWS_ACCESS_KEY_ID dan AWS_SECRET _ACCESS_KEY) yang memiliki izin IAM yang sesuai untuk Kueri Amazon Managed Blockchain (AMB). Di terminal dengan AWS CLI instalasi, jalankan perintah berikut untuk membuat kebijakan IAM untuk mengakses operasi AMB Query API:

```
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
```

Setelah Anda membuat kebijakan, lampirkan kebijakan tersebut ke Peran pengguna IAM agar kebijakan tersebut diterapkan. Di bagian Konsol Manajemen AWS, navigasikan ke layanan IAM, dan lampirkan kebijakan `AmazonManagedBlockchainQueryAccess` ke Peran yang ditetapkan ke pengguna IAM yang akan menggunakan layanan. Untuk informasi selengkapnya, lihat Membuat Peran dan menetapkan ke pengguna IAM.

> **Note**
>
> AWS merekomendasikan agar Anda memberikan akses ke operasi API tertentu daripada menggunakan wild-card*. Untuk informasi selengkapnya, lihat Mengakses tindakan API Kueri Amazon Managed Blockchain (AMB) tertentu.

Buat permintaan API Kueri Amazon Managed Blockchain (AMB) dengan menggunakan Go

Dengan Amazon Managed Blockchain (AMB) Query, Anda dapat membangun aplikasi yang bergantung pada akses instan ke data blockchain setelah dikonfirmasi di blockchain, bahkan jika belum mencapai finalitas. AMB Query memungkinkan beberapa kasus penggunaan seperti mengisi riwayat transaksi dompet, memberikan informasi kontekstual tentang transaksi berdasarkan hash transaksinya, atau mendapatkan saldo token asli serta token ERC-721, ERC-1155, dan ERC-20.

Contoh berikut dibuat dalam bahasa Go dan menggunakan operasi AMB Query API. Untuk informasi selengkapnya tentang Go, lihat Dokumentasi Go. Untuk informasi selengkapnya tentang AMB Query API, lihat Dokumentasi Referensi API Kueri Amazon Managed Blockchain (AMB).

Contoh berikut menggunakan `ListTransactions` dan tindakan `GetTransaction` API untuk terlebih dahulu mendapatkan daftar semua transaksi untuk alamat yang dimiliki eksternal (EOA)

tertentu di Ethereum Mainnet, dan kemudian contoh berikutnya mengambil detail transaksi untuk satu transaksi dari daftar.

Example— Buat tindakan ListTransactions API menggunakan Go

Salin kode berikut ke file bernama `listTransactions.go` dalam ListTransactionsdirektori.

```
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)
    }
}
```

Setelah Anda menyimpan file, jalankan kode dengan menggunakan perintah berikut di dalam ListTransactionsdirektori:`go run listTransactions.go`.

Output yang berikut menyerupai berikut ini:

```
{
  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— Buat tindakan GetTransaction API dengan menggunakan Go

Contoh ini menggunakan hash transaksi dari output sebelumnya. Salin kode berikut ke file bernama `GetTransaction.go` dalam GetTransactiondirektori.

```
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)
    }
}
```

Setelah Anda menyimpan file, jalankan kode dengan menggunakan perintah berikut di dalam GetTransactiondirektori:`go run GetTransaction.go`.

Output yang berikut menyerupai berikut ini:

```
{
  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
  }
}
```

`GetTokenBalance`API menyediakan cara bagi Anda untuk mendapatkan saldo token asli (ETH dan BTC), yang dapat digunakan untuk mendapatkan saldo saat ini dari akun yang dimiliki secara eksternal (EOA) pada suatu titik waktu.

Example— Gunakan tindakan GetTokenBalance API untuk mendapatkan saldo token asli di Go

Dalam contoh berikut, Anda menggunakan `GetTokenBalance` API untuk mendapatkan saldo alamat Ether (ETH) di Ethereum Mainnet. Salin kode berikut ke file bernama `GetTokenBalanceEth.go` dalam GetTokenBalancedirektori.

```
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)
    }
}
```

Setelah Anda menyimpan file, jalankan kode dengan menggunakan perintah berikut di dalam GetTokenBalancedirektori:`go run GetTokenBalanceEth.go`.

Output yang berikut menyerupai berikut ini:

```
{
  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"
  }
}
```

Buat permintaan API Kueri Amazon Managed Blockchain (AMB) dengan menggunakan Node.js

Untuk menjalankan contoh Node ini, prasyarat berikut berlaku:

1. Anda harus memiliki node version manager (nvm) dan Node.js diinstal pada mesin Anda. Anda dapat menemukan instruksi instalasi untuk OS Anda di sini.

2. Gunakan `node --version` perintah dan konfirmasikan bahwa Anda menggunakan Node versi 14 atau lebih tinggi. Jika diperlukan, Anda dapat menggunakan `nvm install 14` perintah, diikuti oleh `nvm use 14` perintah untuk menginstal versi 14.
3. Variabel lingkungan `AWS_ACCESS_KEY_ID` dan `AWS_SECRET_ACCESS_KEY` harus berisi kredensil yang terkait dengan akun.

   Ekspor variabel ini sebagai string pada klien Anda dengan menggunakan perintah berikut. Ganti nilai yang disorot berikut ini dengan nilai yang sesuai dari akun pengguna IAM.

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

> **Note**
>
> - Setelah menyelesaikan semua prasyarat, Anda dapat mengirimkan permintaan yang ditandatangani melalui HTTPS untuk mengakses operasi API Kueri Amazon Managed Blockchain (AMB) dan membuat permintaan dengan menggunakan modul https asli di Node.js, atau Anda dapat menggunakan pustaka pihak ketiga seperti AXIOS dan mengambil data dari AMB Query.
> - Contoh ini menggunakan klien HTTP pihak ketiga untuk Node.js, tetapi Anda juga dapat menggunakan AWS JavaScript SDK untuk membuat permintaan ke AMB Query.
> - Contoh berikut menunjukkan cara membuat permintaan AMB Query API dengan menggunakan Axios dan modul AWS SDK untuk SiGv4.

Salin `package.json` file berikut ke direktori kerja lingkungan lokal Anda:

```
{
  "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— Ambil saldo token historis dari alamat yang dimiliki eksternal (EOA) tertentu dengan menggunakan AMB Query API GetTokenBalance

Anda dapat menggunakan `GetTokenBalance` API untuk mendapatkan saldo berbagai token (misalnya,, ERC20, dan ERC1155) dan koin asli (misalnya ERC721, ETH dan BTC), yang dapat Anda gunakan untuk mendapatkan saldo saat ini dari akun yang dimiliki secara eksternal (EOA) berdasarkan historis `timestamp` (stempel waktu Unix - detik). Dalam contoh ini, Anda menggunakan `GetTokenBalance`API untuk mendapatkan saldo alamat ERC20 token, USDC, di Ethereum Mainnet.

Untuk menguji `GetTokenBalance` API, salin kode berikut ke dalam file bernama`token-balance.js`, dan simpan file ke direktori kerja yang sama:

```
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);
```

Untuk menjalankan kode, buka terminal di direktori yang sama dengan file Anda dan jalankan perintah berikut:

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

Perintah ini menjalankan skrip, meneruskan argumen yang ditentukan dalam kode untuk meminta saldo ERC20 USDC dari EOA yang terdaftar di Ethereum Mainnet. Responsnya terlihat seperti berikut:

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

Buat permintaan API Kueri Amazon Managed Blockchain (AMB) dengan menggunakan Python

Untuk menjalankan contoh Python ini, prasyarat berikut berlaku:

1. Anda harus menginstal Python di mesin Anda. Anda dapat menemukan instruksi instalasi untuk OS Anda di sini.
2. Instal AWS SDK untuk Python (Boto3).
3. Instal AWS Command Line Interface dan jalankan perintah `aws configure` untuk mengatur variabel untuk Anda`Access Key ID`,`Secret Access Key`, dan`Region`.

Setelah menyelesaikan semua prasyarat, Anda dapat menggunakan SDK AWS untuk Python melalui HTTPS untuk membuat permintaan API Kueri Amazon Managed Blockchain (AMB).

Contoh Python berikut menggunakan modul dari boto3 untuk mengirim permintaan yang ditempelkan dengan header SigV4 yang diperlukan ke operasi AMB Query API. `ListTransactionEvents` Contoh ini mengambil daftar peristiwa yang dipancarkan oleh transaksi tertentu di Ethereum Mainnet.

Salin `list-transaction-events.py` file berikut ke direktori kerja lingkungan lokal Anda:

```
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')))
```

Untuk menjalankan kode sampel ke`ListTransactionEvents`, simpan file di direktori kerja Anda dan kemudian jalankan perintah`python3 list-transaction-events.py`. Perintah ini menjalankan skrip, meneruskan argumen yang ditentukan dalam kode untuk meminta peristiwa yang terkait dengan hash transaksi yang diberikan di Ethereum Mainnet. Responsnya terlihat seperti berikut:

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

Gunakan Kueri Amazon Managed Blockchain (AMB) Konsol Manajemen AWS untuk menjalankan operasi GetTokenBalance

Contoh berikut menunjukkan cara mendapatkan saldo token di Ethereum Mainnet menggunakan Amazon Managed Blockchain (AMB) Query di Konsol Manajemen AWS

Example

1. Buka konsol Amazon Managed Blockchain di https://console.aws.amazon.com/managedblockchain/.
2. Pilih Editor kueri dari bagian Kueri.
3. Pilih ETHEREUM_MAINNET sebagai jaringan Blockchain.
4. Pilih GetTokenBalancesebagai tipe Query.
5. Masukkan alamat Blockchain Anda untuk token.
6. Masukkan alamat Kontrak untuk token.
7. Masukkan ID Token opsional untuk token.

8. Pilih Tanggal Pada untuk saldo token.
9. Masukkan opsional Pada waktu untuk saldo token.
10. Pilih Run query (Jalankan kueri).

AMB Query akan menjalankan query Anda dan Anda akan melihat hasilnya di jendela hasil Query.

Kasus penggunaan dengan Kueri Amazon Managed Blockchain (AMB)

Topik ini menyediakan daftar kasus penggunaan Kueri AMB.

Topik

- Kueri saldo token saat ini dan historis
- Mengambil data transaksi historis
- Dapatkan semua saldo token untuk alamat tertentu
- Daftar peristiwa yang dipancarkan untuk transaksi
- Dapatkan semua token yang dicetak oleh kontrak
- Daftar kontrak dan dapatkan informasi kontrak

Kueri saldo token saat ini dan historis

`GetTokenBalance`API mendapatkan saldo token yang didukung (ERC20, ERC721, ERC1155) dan koin asli (ETH, BTC) untuk mendapatkan saldo saat ini atau historis dengan menggunakan stempel waktu universal (stempel waktu Unix, dalam detik) dari akun yang dimiliki secara eksternal (). EOAs Misalnya, Anda dapat menggunakan operasi `GetTokenBalance` API untuk mendapatkan saldo alamat ERC20 token, USDC, di Ethereum Mainnet. Anda juga dapat mengambil saldo token dan koin asli secara batch dengan menggunakan operasi API. `BatchGetTokenBalance`

Untuk informasi selengkapnya, lihat Panduan Referensi Kueri Amazon Managed Blockchain (AMB).

Mengambil data transaksi historis

Dengan Amazon Managed Blockchain (AMB) Query, Anda dapat mengambil data historis dari blockchain publik seperti Ethereum dan Bitcoin. Fitur ini memungkinkan beberapa kasus penggunaan, seperti mengambil riwayat transaksi pada dompet blockchain atau memberikan informasi kontekstual tentang transaksi berdasarkan hash transaksinya. Anda dapat menggunakan operasi `ListTransactions`API untuk mendapatkan daftar transaksi untuk alamat yang dimiliki eksternal (EOA) tertentu di Ethereum Mainnet, dan kemudian Anda dapat menggunakan operasi `GetTransaction`API untuk mengambil detail transaksi untuk satu transaksi dari daftar.

Untuk informasi selengkapnya, lihat Panduan Referensi Kueri Amazon Managed Blockchain (AMB).

Dapatkan semua saldo token untuk alamat tertentu

Anda dapat menggunakan operasi `ListTokenBalances`API untuk mendapatkan saldo pada dompet, antarmuka pengguna, utilitas web3, dan lainnya. Operasi API ini mengembalikan daftar semua saldo untuk alamat di seluruh token (ERC20, ERC721, ERC1155) dan koin asli (ETH, BTC) pada blockchain publik tertentu dengan menggunakan operasi API tunggal. Misalnya, Anda dapat memberikan alamat yang dimiliki secara eksternal (EOA) dan jaringan (Ethereum Mainnet), dan Anda dapat menerima daftar token dan saldo koin asli dalam respons.

Untuk informasi selengkapnya, lihat Panduan Referensi Kueri Amazon Managed Blockchain (AMB).

Daftar peristiwa yang dipancarkan untuk transaksi

Anda dapat menggunakan operasi `ListTransactionEvents`API untuk mengambil daftar peristiwa kontrak yang dipancarkan sebagai hasil dari transaksi tertentu, yang diidentifikasi oleh hash (pengenal transaksi). Misalnya, Anda dapat menggunakan `ListTransactionEvents`untuk mengambil peristiwa yang dihasilkan dari transaksi yang memanggil fungsi kontrak ERC20 token pada Blockchain Ethereum, seperti peristiwa Transfer atau peristiwa Penarikan dari ERC20 kontrak.

Untuk informasi selengkapnya, lihat Panduan Referensi Kueri Amazon Managed Blockchain (AMB).

Dapatkan semua token yang dicetak oleh kontrak

Anda dapat menggunakan operasi `ListTokenBalances`API untuk mengembalikan daftar semua token yang didukung (ERC20, ERC721, ERC1155) yang dicetak oleh kontrak saat melewati alamat kontrak sebagai input. Misalnya, Anda dapat mengambil informasi yang terkait dengan token yang tidak dapat dipertukarkan (NFTs) yang dicetak oleh standar ERC721 kontrak pada blockchain Ethereum dengan menggunakan operasi API. `ListTokenBalances`

Untuk informasi selengkapnya, lihat Panduan Referensi Kueri Amazon Managed Blockchain (AMB).

Daftar kontrak dan dapatkan informasi kontrak

Anda dapat menggunakan operasi `ListAssetContracts`API untuk mencantumkan kontrak ERC-721, ERC-1155, atau ERC-20 yang digunakan oleh alamat tertentu. Selain itu, jika Anda memiliki alamat kontrak, Anda dapat menggunakan operasi `GetAssetContract`API untuk mengambil properti kontrak, seperti alamat deployer tipe kontrak, dan metadata token yang relevan.

Untuk informasi selengkapnya, lihat Panduan Referensi Kueri Amazon Managed Blockchain (AMB).

Referensi API Kueri Amazon Managed Blockchain (AMB)

Kueri Amazon Managed Blockchain (AMB) menyediakan operasi API untuk kueri blockchain yang didukung. Ini termasuk APIs untuk menanyakan token, transaksi, dan kontrak. Untuk informasi selengkapnya, lihat Referensi API Kueri AMB.

Keamanan dalam Kueri Amazon Managed Blockchain (AMB)

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. Model tanggung jawab bersama menggambarkan ini sebagai keamanan cloud dan keamanan di cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi keefektifan keamanan kami sebagai bagian dari program kepatuhan AWS. Untuk mempelajari tentang program kepatuhan yang berlaku untuk Kueri Amazon Managed Blockchain (AMB), lihat AWS Layanan dalam Lingkup berdasarkan Program Kepatuhan.
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor-faktor lain, termasuk sensitivitas data Anda, persyaratan perusahaan Anda, dan hukum dan peraturan yang berlaku.

Untuk memberikan perlindungan data, otentikasi, dan kontrol akses, Amazon Managed Blockchain menggunakan AWS fitur dan fitur kerangka kerja sumber terbuka yang berjalan di Blockchain Terkelola.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan Kueri AMB. Topik berikut menunjukkan cara mengonfigurasi Kueri AMB untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga dapat mempelajari cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan sumber daya Kueri AMB Anda.

Topik

- Enkripsi data
- Manajemen identitas dan akses untuk Kueri Amazon Managed Blockchain (AMB)

Enkripsi data

Enkripsi data membantu mencegah pengguna yang tidak sah membaca data dari jaringan blockchain dan sistem penyimpanan data terkait. Ini termasuk data yang mungkin dicegat saat melakukan perjalanan jaringan, yang dikenal sebagai data dalam perjalanan.

Enkripsi saat bergerak

Secara default, Managed Blockchain menggunakan HTTPS/TLS koneksi untuk mengenkripsi semua data yang dikirimkan dari AWS CLI klien ke titik akhir AWS layanan.

Manajemen identitas dan akses untuk Kueri Amazon Managed Blockchain (AMB)

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya Kueri AMB. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- Audiens
- Mengautentikasi dengan identitas
- Mengelola akses menggunakan kebijakan
- Bagaimana Kueri Amazon Managed Blockchain (AMB) bekerja dengan IAM
- Contoh kebijakan berbasis identitas untuk Kueri Amazon Managed Blockchain (AMB)
- Memecahkan Masalah Amazon Managed Blockchain (AMB) Identitas kueri dan akses

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda berdasarkan peran Anda:

- Pengguna layanan - minta izin dari administrator Anda jika Anda tidak dapat mengakses fitur (lihat Memecahkan Masalah Amazon Managed Blockchain (AMB) Identitas kueri dan akses)
- Administrator layanan - tentukan akses pengguna dan mengirimkan permintaan izin (lihat Bagaimana Kueri Amazon Managed Blockchain (AMB) bekerja dengan IAM)

- Administrator IAM - tulis kebijakan untuk mengelola akses (lihat Contoh kebijakan berbasis identitas untuk Kueri Amazon Managed Blockchain (AMB))

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensi identitas Anda. Anda harus diautentikasi sebagai Pengguna root akun AWS, pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk sebagai identitas federasi menggunakan kredensional dari sumber identitas seperti AWS IAM Identity Center (Pusat Identitas IAM), otentikasi masuk tunggal, atau kredensional. Google/Facebook Untuk informasi selengkapnya tentang cara masuk, lihat Cara masuk ke Akun AWS Anda dalam Panduan Pengguna AWS Sign-In .

Untuk akses terprogram, AWS sediakan SDK dan CLI untuk menandatangani permintaan secara kriptografis. Untuk informasi selengkapnya, lihat AWS Signature Version 4 untuk permintaan API dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang disebut pengguna Akun AWS root yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Untuk tugas yang memerlukan kredensial pengguna root, lihat Tugas yang memerlukan kredensial pengguna root dalam Panduan Pengguna IAM.

Identitas terfederasi

Sebagai praktik terbaik, mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS menggunakan kredensi sementara.

Identitas federasi adalah pengguna dari direktori perusahaan Anda, penyedia identitas web, atau Directory Service yang mengakses Layanan AWS menggunakan kredensi dari sumber identitas. Identitas terfederasi mengambil peran yang memberikan kredensial sementara.

Untuk manajemen akses terpusat, kami menyarankan AWS IAM Identity Center. Untuk informasi selengkapnya, lihat Apa itu Pusat Identitas IAM? dalam Panduan Pengguna AWS IAM Identity Center .

Pengguna dan grup IAM

Pengguna IAM adalah identitas dengan izin khusus untuk satu orang atau aplikasi. Sebaiknya gunakan kredensial sementara alih-alih pengguna IAM dengan kredensial jangka panjang. Untuk informasi selengkapnya, lihat Mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses AWS menggunakan kredensi sementara di Panduan Pengguna IAM.

Grup IAM menentukan kumpulan pengguna IAM dan mempermudah pengelolaan izin untuk pengguna dalam jumlah besar. Untuk mempelajari selengkapnya, lihat Kasus penggunaan untuk pengguna IAM dalam Panduan Pengguna IAM.

Peran IAM

Peran IAM adalah identitas dengan izin khusus yang menyediakan kredensial sementara. Anda dapat mengambil peran dengan beralih dari pengguna ke peran IAM (konsol) atau dengan memanggil operasi AWS CLI atau AWS API. Untuk informasi selengkapnya, lihat Metode untuk mengambil peran dalam Panduan Pengguna IAM.

Peran IAM berguna untuk akses pengguna terfederasi, izin pengguna IAM sementara, akses lintas akun, akses lintas layanan, dan aplikasi yang berjalan di Amazon EC2. Untuk informasi selengkapnya, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan menentukan izin saat dikaitkan dengan identitas atau sumber daya. AWS mengevaluasi kebijakan ini ketika kepala sekolah membuat permintaan. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang dokumen kebijakan JSON, lihat Gambaran umum kebijakan JSON dalam Panduan Pengguna IAM.

Menggunakan kebijakan, administrator menentukan siapa yang memiliki akses ke apa dengan mendefinisikan principal mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Administrator IAM membuat kebijakan IAM dan menambahkannya ke peran, yang kemudian dapat diambil oleh pengguna. Kebijakan IAM mendefinisikan izin terlepas dari metode yang Anda gunakan untuk melakukan operasinya.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang Anda lampirkan ke identitas (pengguna, grup, atau peran). Kebijakan ini mengontrol tindakan apa yang bisa dilakukan oleh identitas tersebut, terhadap sumber daya yang mana, dan dalam kondisi apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan yang dikelola pelanggan dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat berupa kebijakan inline (disematkan langsung ke dalam satu identitas) atau kebijakan terkelola (kebijakan mandiri yang dilampirkan pada banyak identitas). Untuk mempelajari cara memilih antara kebijakan terkelola dan kebijakan inline, lihat Pilih antara kebijakan terkelola dan kebijakan inline dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contohnya termasuk kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Anda harus menentukan principal dalam kebijakan berbasis sumber daya.

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang dapat menetapkan izin maksimum yang diberikan oleh jenis kebijakan yang lebih umum:

- Batasan izin – Menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM. Untuk informasi selengkapnya, lihat Batasan izin untuk entitas IAM dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCPs) — Tentukan izin maksimum untuk organisasi atau unit organisasi di AWS Organizations. Untuk informasi selengkapnya, lihat Kebijakan kontrol layanan dalam Panduan Pengguna AWS Organizations .
- Kebijakan kontrol sumber daya (RCPs) — Tetapkan izin maksimum yang tersedia untuk sumber daya di akun Anda. Untuk informasi selengkapnya, lihat Kebijakan kontrol sumber daya (RCPs) di Panduan AWS Organizations Pengguna.

- Kebijakan sesi – Kebijakan lanjutan yang diteruskan sebagai parameter saat membuat sesi sementara untuk peran atau pengguna terfederasi. Untuk informasi selengkapnya, lihat Kebijakan sesi dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat Logika evaluasi kebijakan di Panduan Pengguna IAM.

Bagaimana Kueri Amazon Managed Blockchain (AMB) bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke Kueri AMB, pelajari fitur IAM apa yang tersedia untuk digunakan dengan AMB Query.

Fitur IAM yang dapat Anda gunakan dengan Kueri Amazon Managed Blockchain (AMB)

Fitur IAM	Dukungan AMB Query
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Tidak
Tindakan kebijakan	Ya
Sumber daya kebijakan	Tidak
Kunci kondisi kebijakan	Tidak
ACLs	Tidak
ABAC (tanda dalam kebijakan)	Tidak
Kredensial sementara	Ya
Izin principal	Ya
Peran layanan	Tidak
Peran terkait layanan	Tidak

Untuk mendapatkan tampilan tingkat tinggi tentang cara AMB Query dan AWS layanan lainnya bekerja dengan sebagian besar fitur IAM, lihat AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM.

Kebijakan berbasis identitas untuk Kueri AMB

Mendukung kebijakan berbasis identitas: Ya

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta kondisi yang menjadi dasar dikabulkan atau ditolaknya tindakan tersebut. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, lihat Referensi elemen kebijakan JSON IAM dalam Panduan Pengguna IAM.

Contoh kebijakan berbasis identitas untuk Kueri AMB

Untuk melihat contoh kebijakan berbasis identitas Kueri AMB, lihat. Contoh kebijakan berbasis identitas untuk Kueri Amazon Managed Blockchain (AMB)

Kebijakan berbasis sumber daya dalam AMB Query

Mendukung kebijakan berbasis sumber daya: Tidak

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh principal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan principal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan secara spesifik seluruh akun atau entitas IAM di akun lain sebagai principal dalam kebijakan berbasis sumber daya. Untuk informasi selengkapnya, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.

Tindakan kebijakan untuk Kueri AMB

Mendukung tindakan kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Untuk melihat daftar tindakan Kueri AMB, lihat Kueri Tindakan yang Ditentukan oleh Amazon Managed Blockchain (AMB) di Referensi Otorisasi Layanan.

Tindakan kebijakan dalam Kueri AMB menggunakan awalan berikut sebelum tindakan:

```
managedblockchain-query:
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan tersebut dengan koma.

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

Untuk melihat contoh kebijakan berbasis identitas Kueri AMB, lihat. Contoh kebijakan berbasis identitas untuk Kueri Amazon Managed Blockchain (AMB)

Sumber daya kebijakan untuk Kueri AMB

Mendukung sumber daya kebijakan: Tidak

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

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

Untuk melihat daftar jenis sumber daya Kueri AMB dan jenisnya ARNs, lihat Kueri Sumber Daya yang Ditentukan oleh Amazon Managed Blockchain (AMB) di Referensi Otorisasi Layanan. Untuk mempelajari tindakan mana yang dapat Anda tentukan ARN dari setiap sumber daya, lihat Kueri Tindakan yang Ditentukan oleh Amazon Managed Blockchain (AMB).

Untuk melihat contoh kebijakan berbasis identitas Kueri AMB, lihat. Contoh kebijakan berbasis identitas untuk Kueri Amazon Managed Blockchain (AMB)

Kunci kondisi kebijakan untuk Kueri AMB

Mendukung kunci kondisi kebijakan khusus layanan: Tidak

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

Elemen `Condition` menentukan ketika pernyataan dieksekusi berdasarkan kriteria yang ditetapkan. Anda dapat membuat ekspresi bersyarat yang menggunakan operator kondisi, misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global di Panduan Pengguna IAM.

Untuk melihat daftar kunci kondisi Kueri AMB, lihat Kunci Kondisi untuk Kueri Amazon Managed Blockchain (AMB) di Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat Kueri Tindakan yang Ditentukan oleh Amazon Managed Blockchain (AMB).

Untuk melihat contoh kebijakan berbasis identitas Kueri AMB, lihat. Contoh kebijakan berbasis identitas untuk Kueri Amazon Managed Blockchain (AMB)

ACLs dalam Kueri AMB

Mendukung ACLs: Tidak

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

ABAC dengan Kueri AMB

Mendukung ABAC (tag dalam kebijakan): Tidak

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut tanda. Anda dapat melampirkan tag ke entitas dan AWS sumber daya IAM, lalu merancang kebijakan ABAC untuk mengizinkan operasi saat tag prinsipal cocok dengan tag pada sumber daya.

Untuk mengendalikan akses berdasarkan tanda, berikan informasi tentang tanda di elemen kondisi dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi untuk hanya beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, lihat Tentukan izin dengan otorisasi ABAC dalam Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, lihat Menggunakan kontrol akses berbasis atribut (ABAC) dalam Panduan Pengguna IAM.

Menggunakan kredensil sementara dengan AMB Query

Mendukung kredensial sementara: Ya

Kredensi sementara menyediakan akses jangka pendek ke AWS sumber daya dan secara otomatis dibuat saat Anda menggunakan federasi atau beralih peran. AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat Kredensial keamanan sementara di IAM dan Layanan AWS yang berfungsi dengan IAM dalam Panduan Pengguna IAM.

Izin utama lintas layanan untuk Kueri AMB

Mendukung sesi akses terusan (FAS): Ya

Sesi akses teruskan (FAS) menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat Sesi akses terusan.

Peran layanan untuk Kueri AMB

Mendukung peran layanan: Tidak

Peran layanan adalah peran IAM yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.

> **Warning**
>
> Mengubah izin untuk peran layanan dapat merusak fungsionalitas Kueri AMB. Edit peran layanan hanya jika AMB Query memberikan panduan untuk melakukannya.

Peran terkait layanan untuk Kueri AMB

Mendukung peran terkait layanan: Tidak

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

Untuk detail tentang pembuatan atau manajemen peran terkait layanan, lihat Layanan AWS yang berfungsi dengan IAM. Cari layanan dalam tabel yang memiliki `Yes` di kolom Peran terkait layanan. Pilih tautan Ya untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Contoh kebijakan berbasis identitas untuk Kueri Amazon Managed Blockchain (AMB)

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya Kueri AMB. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM dengan menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat kebijakan IAM (konsol) di Panduan Pengguna IAM.

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh Kueri AMB, termasuk format ARNs untuk setiap jenis sumber daya, lihat Tindakan, Sumber Daya, dan Kunci Kondisi untuk Kueri Amazon Managed Blockchain (AMB) di Referensi Otorisasi Layanan.

Topik

- Praktik terbaik kebijakan
- Mengizinkan pengguna melihat izin mereka sendiri
- Mengakses tindakan API Kueri Amazon Managed Blockchain (AMB) tertentu

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya Kueri AMB di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Kebijakan yang dikelola AWS atau Kebijakan yang dikelola AWS untuk fungsi tugas dalam Panduan Pengguna IAM.
- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi

selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat Kebijakan dan izin dalam IAM dalam Panduan Pengguna IAM.

- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan JSON IAM: Kondisi dalam Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat Validasi kebijakan dengan IAM Access Analyzer dalam Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Amankan akses API dengan MFA dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat Praktik terbaik keamanan di IAM dalam Panduan Pengguna IAM.

Mengizinkan pengguna melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. 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": "*"
        }
    ]
}
```

Mengakses tindakan API Kueri Amazon Managed Blockchain (AMB) tertentu

> **Note**
>
> Untuk mengakses Kueri AMB untuk melakukan panggilan API, Anda memerlukan kredensi pengguna (`AWS_ACCESS_KEY_ID`dan`AWS_SECRET_ACCESS_KEY`) yang memiliki izin IAM yang sesuai untuk Kueri AMB.

Example Kebijakan IAM untuk mengakses semua Kueri Amazon Managed Blockchain (AMB) APIs

Contoh ini memberikan pengguna IAM dalam Akun AWS akses Anda ke semua Kueri AMB. APIs

JSON

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

Example Kebijakan IAM untuk mengakses Kueri Amazon Managed Blockchain (AMB) dan **`ListTransactions` `GetTransaction`** APIs

Contoh ini memberikan pengguna IAM dalam Akun AWS akses Anda ke Kueri AMB dan `ListTransaction` `GetTransaction` APIs

> **Note**
>
> Anda dapat mengganti atau menambahkan APIs pada contoh dengan yang lain APIs untuk memberikan akses ke yang lain atau lebih APIs. Untuk daftar Kueri AMB APIs, lihat Panduan Referensi API Kueri Amazon Managed Blockchain (AMB).

JSON

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

```
                "managedblockchain-query:GetTransaction"
            ],
            "Resource": "*"
        }
    ]
}
```

Memecahkan Masalah Amazon Managed Blockchain (AMB) Identitas kueri dan akses

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan AMB Query dan IAM.

Topik

- Saya tidak berwenang untuk melakukan tindakan di AMB Query

Saya tidak berwenang untuk melakukan tindakan di AMB Query

Jika Anda menerima pesan kesalahan bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

Contoh kesalahan berikut terjadi ketika pengguna IAM `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang suatu sumber daya *`my-example-widget`* rekaan, tetapi tidak memiliki izin `managedblockchain-query::`*`GetWidget`* rekaan.

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

Dalam hal ini, kebijakan untuk pengguna `mateojackson` harus diperbarui untuk mengizinkan akses ke sumber daya *`my-example-widget`* dengan menggunakan tindakan `managedblockchain-query::`*`GetWidget`*.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Metrik penggunaan API Kueri Amazon Managed Blockchain (AMB) di Amazon CloudWatch

Metrik penggunaan API di Amazon CloudWatch

Metrik penggunaan API yang diterbitkan CloudWatch sesuai dengan kuota layanan Kueri Amazon Managed Blockchain (AMB). Anda dapat mengonfigurasi alarm untuk mengingatkan Anda ketika penggunaan Anda mendekati kuota layanan. Untuk informasi selengkapnya tentang CloudWatch integrasi dengan kuota layanan, lihat metrik penggunaan AWS di CloudWatch Panduan Pengguna Amazon.

AMB Query menerbitkan metrik API berikut di `AWS/Usage` namespace, dengan nama layanan. `Amazon Managed Blockchain Query`

Metrik	Deskripsi
`CallCount`	Jumlah total panggilan yang dilakukan ke API di AMB Query. SUM mewakili jumlah total panggilan ke API selama periode yang ditentukan.

Kueri Amazon Managed Blockchain (AMB) menerbitkan metrik penggunaan ke `AWS/Usage` namespace dengan dimensi berikut.

Dimensi	Deskripsi
Layanan	Nama AWS layanan yang berisi sumber daya. `Amazon Managed Blockchain Query`yakan selalu menjadi nilai untuk dimensi ini.
Tipe	Jenis entitas yang dilaporkan. `API`akan selalu menjadi nilai untuk dimensi ini.

Dimensi	Deskripsi
Sumber daya	Jenis sumber daya yang dilaporkan. Nama operasi AMB Query API yang digunakan akan menjadi nilai untuk dimensi ini.
Kelas	Kelas sumber daya yang dilaporkan. None akan selalu menjadi nilai untuk dimensi ini.

Riwayat dokumen untuk Panduan Pengguna Kueri AMB

Tabel berikut menjelaskan rilis dokumentasi untuk AMB Query.

Perubahan	Deskripsi	Tanggal
AMB Query mendukung pengidentifikasi transaksi Bitcoin dan hash transaksi	Untuk jaringan Bitcoin, operasi AMB Query API mendukung pengenal transaksi (`transactionId`) dan hash transaksi (`transactionHash`).	Maret 21, 2024
Support untuk metrik penggunaan API di Amazon CloudWatch	AMB Query menambahkan dukungan untuk metrik penggunaan API pada. CloudWatch Metrik penggunaan ini sesuai dengan kuota layanan Kueri AMB.	Februari 8, 2024
Support untuk transaksi yang belum mencapai finalitas	AMB Query menambahkan dukungan untuk transaksi yang belum mencapai finalitas. Ini juga menghilangkan dukungan untuk `status` properti dari respons `GetTransaction` operasi. Sebagai gantinya, Anda akan menggunakan `executionStatus` properti `confirmationStatus` dan untuk menentukan status transaksi.	Februari 1, 2024
Penutupan `status` properti dalam tipe data Transaksi	Kueri Amazon Managed Blockchain (AMB) telah menghentikan `status` properti dalam tipe data	20 Desember 2023

	Transaksi. Anda harus menggunakan `executionStatus` bidang `confirmationStatus` dan untuk menentukan `status` apakah transaksi tersebut `FINAL` atau`FAILED`.	
Support untuk Seplia Testnet	Amazon Managed Blockchain (AMB) Query sekarang mendukung kueri di Ethereum Seplia Testnet.	19 Oktober 2023
Support untuk kontrak aset	Anda dapat menggunakan operasi `ListAssetContracts` API untuk daftar yang digunakan oleh alamat yang diberikan. Selain itu, jika Anda memiliki alamat kontrak, Anda dapat menggunakan operasi `GetAssetContract` API untuk mengambil detail kontrak.	16 Oktober 2023
Support untuk Bitcoin Testnet	Amazon Managed Blockchain (AMB) Query sekarang mendukung kueri di Bitcoin Testnet.	16 Oktober 2023
Rilis awal	Rilis awal layanan AMB Query.	Juli 27, 2023

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.