aws

Guia do Desenvolvedor

# Amazon Managed Blockchain Query

# Amazon Managed Blockchain Query: Guia do Desenvolvedor

# Table of Contents

# O que é a Amazon Managed Blockchain (AMB) Query?

O Amazon Managed Blockchain (AMB) é um serviço totalmente gerenciado projetado para ajudar você a criar aplicativos Web3 resilientes em blockchains públicos e privados. Use o AMB Access para acesso instantâneo e sem servidor a vários blockchains. Crie seus aplicativos prontos para Web3 sem a necessidade de implantar uma infraestrutura de blockchain especializada e mantê-los conectados à rede blockchain. Com o AMB Query, você pode usar operações de API fáceis de usar para desenvolvedores para acessar dados históricos e em tempo real de vários blockchains. Os dados padronizados de blockchain podem ser integrados aos serviços da AWS, sem exigir uma infraestrutura especializada de blockchain ou ETL (extração, transformação e carregamento). Todos os recursos do AMB são dimensionados com segurança para compilações de aplicativos de consumo convencionais e de nível institucional.

O Amazon Managed Blockchain (AMB) Query fornece acesso sem servidor a conjuntos de dados padronizados com várias cadeias de blocos com operações de API fáceis de usar para desenvolvedores. Você pode usar o AMB Query para enviar rapidamente aplicativos que exigem dados de um ou mais blockchains públicos, sem exigir a sobrecarga de analisar dados de blockchain, rastrear contratos e manter uma infraestrutura de indexação especializada. Se você estiver analisando saldos históricos de tokens fungíveis ou tokens não fungíveis (NFTs), visualizando o histórico de transações de um determinado endereço de carteira ou realizando análises de dados sobre a distribuição de criptomoedas nativas, como Ether, o AMB Query fornece acesso aos dados do blockchain.

## Você é um usuário iniciante do AMB Query?

Se você for um usuário iniciante do AMB Query, recomendamos que comece lendo as seguintes seções:

- Conceitos principais: Amazon Managed Blockchain (AMB) Query
- Configurando a consulta Amazon Managed Blockchain (AMB)
- Introdução à Amazon Managed Blockchain (AMB) Query
- Casos de uso com a Amazon Managed Blockchain (AMB) Query

# Conceitos principais: Amazon Managed Blockchain (AMB) Query

> **Note**
>
> Este guia pressupõe que você esteja familiarizado com os conceitos essenciais de blockchain. Esses conceitos incluem descentralização, tokens, contratos, transações, carteiras proof-of-work, chaves públicas e privadas, apostas, mineração, divisões pela metade e outros.

O Amazon Managed Blockchain (AMB) Query fornece acesso conveniente a dados de rede com vários blocos, o que facilita a extração de dados contextuais relacionados à atividade do blockchain. Você pode usar o AMB Query para ler dados de redes públicas de blockchain, como Bitcoin Mainnet e Ethereum Mainnet. Você também pode obter informações, como saldos atuais e históricos de endereços, ou pode obter uma lista de transações de blockchain para um determinado período de tempo. Além disso, você pode obter detalhes de uma determinada transação, como eventos de transação, que podem ser analisados ou usados posteriormente na lógica de negócios de seus aplicativos.

## Considerações e limitações para usar o Amazon Managed Blockchain (AMB) Query

Ao usar o AMB Query, considere o seguinte:

- Regiões disponíveis

  O AMB Query é suportado na `us-east-1` região Leste dos EUA (Norte da Virgínia).

- Service endpoints (Endpoints de serviço)

  O AMB Query pode ser acessado usando o seguinte endpoint:

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

- Redes de blockchain suportadas

O AMB Query suporta as seguintes redes públicas de blockchain:

- Bitcoin Mainnet — A rede pública de blockchain Bitcoin que é protegida por proof-of-work consenso e na qual a criptomoeda Bitcoin (BTC) é emitida e transacionada. As transações na Mainnet têm valor real (ou seja, incorrem em custos reais) e são registradas na blockchain pública.
- Bitcoin Testnet — A rede de teste da Bitcoin Mainnet. O Bitcoin (BTC) nessa rede é separado e distinto do BTC da Mainnet e geralmente não tem nenhum valor.
- Ethereum Mainnet — A proof-of-stake principal rede para o blockchain público do Ethereum. As transações na Mainnet têm valor real (ou seja, incorrem em custos reais) e são registradas no livro distribuído.
- Sepolia Testnet — A rede de teste para a Ethereum Mainnet. O Ether (ETH) nessa rede é separado e distinto do ETH da Mainnet e geralmente não tem nenhum valor.

- Tokens e contratos de blockchain compatíveis

  O AMB Query suporta os seguintes tokens de contrato Ethereum nativos e padrão.

  - Tokens nativos de blockchain público
    - Bitcoin (BTC) — Esse é o token nativo dos blockchains relacionados ao Bitcoin.
    - Ether (ETH) — Esse é o token nativo dos blockchains relacionados ao Ethereum.

  - Padrões contratuais da Ethereum
    - Padrão de token ERC-20 — O ERC-20 é um padrão para tokens fungíveis. Ele tem uma propriedade que faz com que cada token ERC-20 seja exatamente igual (em tipo e valor) a outro token ERC-20 cunhado, o que significa que um token é e sempre será igual a todos os outros tokens. Para obter mais informações, consulte o Padrão de Token ERC-20 em Ethereum.org.
    - Padrão de token não fungível ERC-721 — O ERC-721 é um padrão para tokens não fungíveis (). NFTs Esse tipo de token é único e pode ter um valor diferente de outro token do mesmo contrato, possivelmente devido à sua idade, raridade ou outras propriedades. Para obter mais informações, consulte o Padrão de Token ERC-721 em Ethereum.org.

      Padrão multitoken ERC-1155 — O ERC-1155 é um padrão que cria uma interface de contrato que pode representar e controlar qualquer número de tipos de tokens fungíveis e não fungíveis. Dessa forma, o token ERC-1155 pode funcionar da mesma forma que os tokens ERC-20 e ERC-721, mesmo funcionando como ambos ao mesmo tempo. O token ERC-1155

melhora a funcionalidade dos padrões ERC-20 e ERC-721, tornando-o mais eficiente e corrigindo erros óbvios de implementação. Para obter mais informações, consulte o Padrão de Token ERC-1155 em Ethereum.org.

- Finalidade

  Em blockchains, a finalidade significa que é improvável que transações válidas sejam revertidas. Para o Bitcoin Mainnet, o AMB Query considera uma transação final após 6 blocos. Para o Bitcoin Testnet, ele considera uma transação final após 6 blocos ou 60 minutos, o que ocorrer primeiro. Para redes Ethereum suportadas, o AMB Query considera uma transação final após 64 blocos.

  O saldo de tokens e as operações de API de contratos do AMB Query retornam apenas dados que atingiram a finalidade. No entanto, as operações da API de transação e evento de transação do AMB Query podem retornar dados de transações confirmadas na rede blockchain, mesmo que ainda não tenham atingido a finalidade.

- Endereço NULL não suportado

  O AMB Query não suporta o endereço NULL (`0x0000000000000000000000000000000000000000`).

- Assinatura Versão 4: assinatura de chamadas de API

  Ao fazer chamadas para a AMB Query APIs, você pode fazer isso por meio de uma conexão HTTPS autenticada usando o processo de assinatura Signature Version 4. Isso significa que somente diretores autorizados do IAM na AWS conta podem fazer chamadas à API AMB Query. Para fazer isso, AWS as credenciais (uma ID da chave de acesso e uma chave de acesso secreta) devem ser fornecidas com a chamada.

  > **Important**
  >
  > Não incorpore credenciais do cliente em aplicativos voltados para o usuário.

- O AMB Query suporta identificadores de transações Bitcoin e hashes de transações

  Para redes Bitcoin, as operações da API AMB Query suportam tanto o identificador da transação (`transactionId`) quanto o hash da transação (`transactionHash`). `transactionId`É um hash duplo SHA da transação, sem incluir dados de testemunhas. `transactionHash`É um

hash duplo SHA da transação, incluindo dados da testemunha (também conhecido como ID da transação da testemunha).

Ao invocar as operações `GetTransaction`ou `ListTransactionEvents`API para redes Bitcoin, você pode especificar o `transactionId` ou o. `transactionHash` Além disso, todas as operações de AMB Query em redes Bitcoin que retornam a `transactionId` ou a `transactionHash` incluirão os dois valores como parte da resposta.

# Configurando a consulta Amazon Managed Blockchain (AMB)

Antes de usar o Amazon Managed Blockchain (AMB) Query pela primeira vez, siga as etapas nesta seção para criar uma AWS conta. A seção a seguir discute como começar a usar o AMB Query.

## Pré-requisitos e considerações

Antes de usar a Amazon Web Services pela primeira vez, você deve ter uma AWS conta.

### Inscreva-se para AWS

Quando você se inscreve no Amazon Web Services (AWS), sua AWS conta é automaticamente cadastrada para todos Serviços da AWS, incluindo o Amazon Managed Blockchain (AMB) Query. Você será cobrado apenas pelos serviços que usar.

Se você Conta da AWS já tem um, vá para a próxima etapa. Se você não tem uma Conta da AWS, siga o procedimento abaixo para criar uma.

Para criar uma AWS conta

1. Abra a https://portal.aws.amazon.com/billing/inscrição.
2. Siga as instruções online.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

   Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

## Crie um usuário do IAM com as permissões apropriadas

Para criar e trabalhar com o AMB Query, você deve criar um principal AWS Identity and Access Management (IAM) (usuário ou grupo) com permissões que permitam as ações necessárias do Managed Blockchain.

Somente diretores do IAM podem fazer solicitações da API AMB Query. Ao fazer chamadas para a AMB Query APIs, você pode fazer isso por meio de uma conexão HTTPS autenticada usando o processo de assinatura Signature Version 4. Isso significa que somente diretores autorizados do IAM na AWS conta podem fazer chamadas à API AMB Query. Para fazer isso, AWS as credenciais (uma ID da chave de acesso e uma chave de acesso secreta) devem ser fornecidas com a chamada.

Para obter informações sobre como criar um usuário do IAM, consulte Como criar um usuário do IAM em sua AWS conta. Para obter mais informações sobre como anexar uma política de permissões a um usuário, consulte Alteração de permissões para um usuário do IAM. Para obter um exemplo de uma política de permissões que você pode usar para dar permissão ao usuário para trabalhar com o AMB Query, consulteExemplos de políticas baseadas em identidade para Amazon Managed Blockchain (AMB) Query.

## Instale e configure o AWS Command Line Interface

Se você ainda não tiver feito isso, instale a interface de AWS linha de comando (CLI) mais recente para trabalhar com AWS recursos de um terminal. Para obter mais informações, consulte Instalar ou atualizar a versão mais recente da AWS CLI.

> **Note**
>
> Para acesso à CLI, é necessário ter um ID de chave de acesso e de uma chave de acesso secreta. Use credenciais temporárias em vez de chaves de acesso de longo prazo quando possível. As credenciais temporárias incluem um ID de acesso, uma chave de acesso secreta e um token de segurança que indica quando as credenciais expiram. Para obter mais informações, consulte Uso de credenciais temporárias com AWS recursos no Guia do usuário do IAM.

## Use o AWS Management Console para consultar blockchains usando o Amazon Managed Blockchain (AMB) Query

Você pode acessar o Amazon Managed Blockchain (AMB) Query e fazer consultas em redes de blockchain compatíveis usando o. AWS Management Console As etapas a seguir mostram como fazer isso:

1. Abra o console do Amazon Managed Blockchain em https://console.aws.amazon.com/managedblockchain/.

2. Escolha Editor de consultas na seção Consulta.
3. Escolha uma das redes Blockchain suportadas.
4. Escolha o tipo de consulta que você deseja executar.
5. Insira os parâmetros relevantes para o tipo de consulta que você selecionou e Execute a consulta.

O AMB Query executará sua consulta e você verá os resultados na janela Resultados da consulta.

# Introdução à Amazon Managed Blockchain (AMB) Query

Use os step-by-step tutoriais desta seção para aprender a realizar tarefas usando o Amazon Managed Blockchain (AMB) Query. Esses procedimentos exigem alguns pré-requisitos. Se você não conhece o AMB Query, consulte a seção Configuração deste guia. Para obter mais informações, consulte Configurando a consulta Amazon Managed Blockchain (AMB).

> **Note**
>
> Algumas variáveis nesses exemplos foram deliberadamente ofuscadas. Substitua-os por outros válidos antes de executar esses exemplos.

Tópicos



## Crie uma política do IAM para acessar as operações da API AMB Query

Para fazer solicitações da API AMB Query, você deve usar as credenciais do usuário (AWS_ACCESS_KEY_ID e a AWS_SECRET _ACCESS_KEY) que tenham as permissões apropriadas do IAM para a Amazon Managed Blockchain (AMB) Query. Em um terminal com o AWS CLI instalado, execute o comando a seguir para criar uma política do IAM para acessar as operações da API AMB Query:

```
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
```

Depois de criar a política, anexe essa política à função de um usuário do IAM para que ela entre em vigor. No AWS Management Console, navegue até o serviço do IAM e anexe a política `AmazonManagedBlockchainQueryAccess` à função atribuída ao usuário do IAM que usará o serviço. Para obter mais informações, consulte Como criar uma função e atribuir a um usuário do IAM.

> **Note**
>
> AWS recomenda que você dê acesso a operações específicas da API em vez de usar o curinga*. Para obter mais informações, consulte Acessando ações específicas da API de consulta do Amazon Managed Blockchain (AMB).

# Faça solicitações de API de consulta do Amazon Managed Blockchain (AMB) usando Go

Com o Amazon Managed Blockchain (AMB) Query, você pode criar aplicativos que dependem do acesso instantâneo aos dados do blockchain, uma vez confirmados no blockchain, mesmo que ainda não tenham atingido a finalidade. O AMB Query permite vários casos de uso, como preencher o histórico de transações de uma carteira, fornecer informações contextuais sobre uma transação com base em seu hash de transação ou obter o saldo de tokens nativos, bem como dos tokens ERC-721, ERC-1155 e ERC-20.

Os exemplos a seguir são criados na linguagem Go e usam as operações da API AMB Query. Para obter mais informações sobre Go, consulte a documentação do Go. Para obter mais informações sobre a API de consulta AMB, consulte a documentação de referência da API de consulta do Amazon Managed Blockchain (AMB).

Os exemplos a seguir usam as ações `ListTransactions` e a `GetTransaction` API para primeiro obter uma lista de todas as transações de um determinado endereço de propriedade externa (EOA) na Ethereum Mainnet e, em seguida, o próximo exemplo recupera os detalhes da transação de uma única transação da lista.

Example — Faça a ação **`ListTransactions`** da API usando Go

Copie o código a seguir em um arquivo nomeado `listTransactions.go` no ListTransactionsdiretório.

```
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)
    }
}
```

Depois de salvar o arquivo, execute o código usando o seguinte comando dentro do ListTransactionsdiretório:`go run listTransactions.go`.

A saída a seguir é semelhante à seguinte:

```
{
  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 — Faça a ação **`GetTransaction`** da API usando Go

Este exemplo usa um hash de transação da saída anterior. Copie o código a seguir em um arquivo nomeado `GetTransaction.go` no GetTransactiondiretório.

```
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)
    }
}
```

Depois de salvar o arquivo, execute o código usando o seguinte comando dentro do GetTransactiondiretório:go `run GetTransaction.go`.

A saída a seguir é semelhante à seguinte:

```
{
  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
  }
}
```

A `GetTokenBalance` API fornece uma maneira de obter o saldo de tokens nativos (ETH e BTC), que podem ser usados para obter o saldo atual de uma conta externa (EOA) em um determinado momento.

Example — Use a ação **`GetTokenBalance`** da API para obter o equilíbrio de um token nativo em Go

No exemplo a seguir, você usa a `GetTokenBalance` API para obter um saldo de endereço Ether (ETH) na Ethereum Mainnet. Copie o código a seguir em um arquivo nomeado `GetTokenBalanceEth.go` no GetTokenBalancediretório.

```
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)
    }
}
```

Depois de salvar o arquivo, execute o código usando o seguinte comando dentro do GetTokenBalancediretório:`go run GetTokenBalanceEth.go`.

A saída a seguir é semelhante à seguinte:

```
{
  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"
  }
}
```

# Faça solicitações de API de consulta do Amazon Managed Blockchain (AMB) usando o Node.js

Para executar esses exemplos de Node, os seguintes pré-requisitos se aplicam:

1. Você deve ter o node version manager (nvm) e o Node.js instalados em sua máquina. Você pode encontrar instruções de instalação para seu sistema operacional aqui.
2. Use o `node --version` comando e confirme se você está usando a versão 14 ou superior do Node. Se necessário, você pode usar o `nvm install 14` comando seguido pelo `nvm use 14` comando para instalar a versão 14.
3. As variáveis `AWS_ACCESS_KEY_ID` de ambiente `AWS_SECRET_ACCESS_KEY` devem conter as credenciais associadas à conta.

   Exporte essas variáveis como cadeias de caracteres em seu cliente usando os comandos a seguir. Substitua os valores destacados a seguir pelos valores apropriados da conta de usuário do IAM.

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

> **Note**
>
> - Depois de concluir todos os pré-requisitos, você pode enviar solicitações assinadas via HTTPS para acessar as operações da API de consulta do Amazon Managed Blockchain (AMB) e fazer solicitações usando o módulo https nativo em Node.js, ou você pode usar uma biblioteca de terceiros, como a AXIOS, e recuperar dados do AMB Query.
> - Esses exemplos usam um cliente HTTP de terceiros para Node.js, mas você também pode usar o AWS JavaScript SDK para fazer solicitações ao AMB Query.
> - O exemplo a seguir mostra como fazer solicitações da API AMB Query usando o Axios e os módulos AWS SDK para SigV4.

Copie o `package.json` arquivo a seguir no diretório de trabalho do seu ambiente local:

```
{
```

```
  "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 — Recupere o saldo histórico de tokens de um endereço externo específico (EOA) usando a API AMB Query **`GetTokenBalance`**

Você pode usar a `GetTokenBalance` API para obter o saldo de vários tokens (por exemplo, ERC2 0 ERC721, e ERC1155) e moedas nativas (por exemplo, ETH e BTC), que você pode usar para obter o saldo atual de uma conta externa (EOA) com base em um histórico `timestamp` (timestamp Unix - segundos). Neste exemplo, você usa a `GetTokenBalance`API para obter um saldo de endereço de um token ERC2 0, USDC, na Ethereum Mainnet.

Para testar a `GetTokenBalance` API, copie o código a seguir em um arquivo chamado `token-balance.js` e salve-o no mesmo diretório de trabalho:

```
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);
```

Para executar o código, abra um terminal no mesmo diretório dos seus arquivos e execute o seguinte comando:

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

Esse comando executa o script, passando os argumentos definidos no código para solicitar o saldo de ERC2 0 USDC do EOA listado na Ethereum Mainnet. A resposta é semelhante à seguinte:

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

# Faça solicitações de API de consulta do Amazon Managed Blockchain (AMB) usando Python

Para executar esses exemplos de Python, os seguintes pré-requisitos se aplicam:

1. Você deve ter o Python instalado em sua máquina. Você pode encontrar instruções de instalação para seu sistema operacional aqui.
2. Instale o SDK da AWS para Python (Boto3).
3. Instale a AWS interface de linha de comando e execute o comando `aws configure` para definir as variáveis para seu `Access Key ID`, `Secret Access Key`, `Region` e.

Depois de concluir todos os pré-requisitos, você pode usar o AWS SDK para Python via HTTPS para fazer solicitações à API de consulta do Amazon Managed Blockchain (AMB).

O exemplo de Python a seguir usa módulos do boto3 para enviar solicitações afixadas com os cabeçalhos SigV4 necessários para a operação da API AMB Query. `ListTransactionEvents` Este exemplo recupera uma lista de eventos emitidos por uma determinada transação na Ethereum Mainnet.

Copie o `list-transaction-events.py` arquivo a seguir no diretório de trabalho do seu ambiente local:

```
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')))
```

Para executar o código de amostra em`ListTransactionEvents`, salve o arquivo em seu diretório de trabalho e execute o comando`python3 list-transaction-events.py`. Esse comando executa o script, passando os argumentos definidos no código para solicitar os eventos associados ao hash da transação em questão na rede principal do Ethereum. A resposta é semelhante à seguinte:

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

# Use a consulta Amazon Managed Blockchain (AMB) no AWS Management Console para executar a GetTokenBalance operação

O exemplo a seguir mostra como obter o saldo de um token na rede principal do Ethereum usando a consulta Amazon Managed Blockchain (AMB) no AWS Management Console

Example

1. Abra o console do Amazon Managed Blockchain em https://console.aws.amazon.com/managedblockchain/.
2. Escolha Editor de consultas na seção Consulta.
3. Escolha ETHEREUM_MAINNET como a rede Blockchain.
4. Escolha GetTokenBalancecomo o tipo de consulta.
5. Insira seu endereço Blockchain para o token.
6. Insira o endereço do contrato para o token.
7. Insira o ID do token opcional para o token.
8. Escolha a data de validade para o saldo do token.
9. Insira o opcional No momento para o saldo do token.
10. Selecione Executar consulta.

O AMB Query executará sua consulta e você verá os resultados na janela Resultados da consulta.

# Casos de uso com a Amazon Managed Blockchain (AMB) Query

Este tópico fornece uma lista de casos de uso do AMB Query.

Tópicos



## Consulte saldos de tokens atuais e históricos

A `GetTokenBalance`API obtém o saldo de tokens suportados (ERC20, ERC721, ERC1155) e moedas nativas (ETH, BTC) para obter o saldo atual ou histórico usando um timestamp universal (timestamp Unix, em segundos) de contas externas (). EOAs Por exemplo, você pode usar a operação de `GetTokenBalance` API para obter um saldo de endereço do token ERC2 0, USDC, na Ethereum Mainnet. Você também pode recuperar em lote saldos de tokens e moedas nativas usando a operação da API. `BatchGetTokenBalance`

Para obter mais informações, consulte o Guia de referência de consultas do Amazon Managed Blockchain (AMB).

## Recupere dados históricos de transações

Com o Amazon Managed Blockchain (AMB) Query, você pode recuperar dados históricos de blockchains públicos, como Ethereum e Bitcoin. Esse recurso permite vários casos de uso, como recuperar um histórico de transações em uma carteira blockchain ou fornecer informações contextuais sobre uma transação com base em seu hash de transação. Você pode usar a operação de `ListTransactions`API para obter uma lista de transações para um determinado endereço de propriedade externa (EOA) na Ethereum Mainnet e, em seguida, pode usar a operação de `GetTransaction`API para recuperar os detalhes da transação de uma única transação da lista.

Para obter mais informações, consulte o Guia de referência de consultas do Amazon Managed Blockchain (AMB).

## Obtenha todos os saldos de tokens de um determinado endereço

Você pode usar a operação da `ListTokenBalances`API para obter saldos em carteiras, interfaces de usuário, utilitários web3 e muito mais. Essa operação de API retorna uma lista de todos os saldos de um endereço entre tokens (ERC20, ERC721, ERC1155) e moedas nativas (ETH, BTC) em um determinado blockchain público usando uma única operação de API. Por exemplo, você pode fornecer um endereço de propriedade externa (EOA) e uma rede (a Ethereum Mainnet) e receber uma lista de tokens e saldos de moedas nativas na resposta.

Para obter mais informações, consulte o Guia de referência de consultas do Amazon Managed Blockchain (AMB).

## Listar eventos emitidos para uma transação

Você pode usar a operação da `ListTransactionEvents`API para recuperar uma lista de eventos de contrato emitidos como resultado de uma determinada transação, identificados por seu hash (identificador de transação). Por exemplo, você pode usar `ListTransactionEvents`para recuperar os eventos resultantes de uma transação que chama uma função de um contrato de token ERC2 0 no Ethereum Blockchain, como um evento de transferência ou um evento de retirada do contrato ERC2 0.

Para obter mais informações, consulte o Guia de referência de consultas do Amazon Managed Blockchain (AMB).

## Obtenha todos os tokens cunhados por um contrato

Você pode usar a operação de `ListTokenBalances`API para retornar uma lista de todos os tokens suportados (ERC20, ERC721, ERC1155) emitidos por um contrato quando o endereço do contrato é passado como entrada. Por exemplo, você pode recuperar informações relacionadas a tokens não fungíveis (NFTs) cunhados pelo padrão de ERC721 contrato no blockchain Ethereum usando a operação de API. `ListTokenBalances`

Para obter mais informações, consulte o Guia de referência de consultas do Amazon Managed Blockchain (AMB).

# Liste contratos e obtenha informações sobre contratos

Você pode usar a operação de `ListAssetContracts`API para listar contratos ERC-721, ERC-1155 ou ERC-20 implantados por um determinado endereço. Além disso, se você tiver o endereço do contrato, poderá usar a operação da `GetAssetContract`API para recuperar as propriedades do contrato, como o tipo de contrato, o endereço do implantador e os metadados relevantes do token.

Para obter mais informações, consulte o Guia de referência de consultas do Amazon Managed Blockchain (AMB).

# Referência da API de consulta do Amazon Managed Blockchain (AMB)

O Amazon Managed Blockchain (AMB) Query fornece operações de API para consultar blockchains compatíveis. Isso inclui APIs a consulta de tokens, transações e contratos. Para obter mais informações, consulte a Referência da API AMB Query.

# Segurança na consulta Amazon Managed Blockchain (AMB)

A segurança na nuvem AWS é da mais alta prioridade. Como AWS cliente, você se beneficia de data centers e arquiteturas de rede criados para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O modelo de responsabilidade compartilhada descreve isso como segurança na nuvem e segurança na nuvem:

- Segurança da nuvem — AWS é responsável por proteger a infraestrutura que executa AWS os serviços no Nuvem AWS. AWS também fornece serviços que você pode usar com segurança. Auditores de terceiros testam e verificam regularmente a eficácia da nossa segurança como parte dos Programas de conformidade da AWS. Para saber mais sobre os programas de conformidade que se aplicam à consulta Amazon Managed Blockchain (AMB), consulte AWS Services in Scope by Compliance Program.
- Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade dos dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Para fornecer proteção de dados, autenticação e controle de acesso, o Amazon Managed Blockchain usa AWS recursos e os recursos da estrutura de código aberto executada no Managed Blockchain.

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar o AMB Query. Os tópicos a seguir mostram como configurar o AMB Query para atender aos seus objetivos de segurança e conformidade. Você também pode aprender a usar outros AWS serviços que ajudam a monitorar e proteger seus recursos do AMB Query.

Tópicos



## Criptografia de dados

A criptografia de dados ajuda a impedir que usuários não autorizados leiam dados de uma rede blockchain e dos sistemas de armazenamento de dados associados. Isso inclui dados que podem ser interceptados enquanto viajam pela rede, conhecidos como dados em trânsito.

## Criptografia em trânsito

Por padrão, o Managed Blockchain usa uma conexão HTTPS/TLS para criptografar todos os dados transmitidos do AWS CLI cliente para os endpoints do serviço. AWS

# Gerenciamento de identidade e acesso para Amazon Managed Blockchain (AMB) Query

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) para usar os recursos do AMB Query. O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Tópicos



## Público

A forma como você usa AWS Identity and Access Management (IAM) difere, dependendo do trabalho que você faz no AMB Query.

Usuário do serviço — Se você usar o serviço AMB Query para fazer seu trabalho, seu administrador fornecerá as credenciais e as permissões de que você precisa. À medida que você usa mais recursos do AMB Query para fazer seu trabalho, talvez precise de permissões adicionais. Compreenda como o acesso é gerenciado pode ajudar a solicitar as permissões corretas ao administrador. Se você não conseguir acessar um recurso no AMB Query, consulteSolução de problemas de identidade e acesso à consulta Amazon Managed Blockchain (AMB).

Administrador de serviços — Se você é responsável pelos recursos do AMB Query em sua empresa, provavelmente tem acesso total ao AMB Query. É seu trabalho determinar quais recursos e recursos

do AMB Query seus usuários do serviço devem acessar. Envie as solicitações ao administrador do IAM para alterar as permissões dos usuários de serviço. Revise as informações nesta página para compreender os conceitos básicos do IAM. Para saber mais sobre como sua empresa pode usar o IAM com o AMB Query, consulteComo o Amazon Managed Blockchain (AMB) Query funciona com o IAM.

Administrador do IAM — Se você for administrador do IAM, talvez queira saber detalhes sobre como criar políticas para gerenciar o acesso ao AMB Query. Para ver exemplos de políticas baseadas em identidade do AMB Query que você pode usar no IAM, consulte. Exemplos de políticas baseadas em identidade para Amazon Managed Blockchain (AMB) Query

## Autenticar com identidades

A autenticação é como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado (conectado AWS) como o Usuário raiz da conta da AWS, como usuário do IAM ou assumindo uma função do IAM.

Você pode entrar AWS como uma identidade federada usando credenciais fornecidas por meio de uma fonte de identidade. AWS IAM Identity Center Usuários (IAM Identity Center), a autenticação de login único da sua empresa e suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você faz login como identidade federada, o administrador já configurou anteriormente a federação de identidades usando perfis do IAM. Ao acessar AWS usando a federação, você está assumindo indiretamente uma função.

Dependendo do tipo de usuário que você é, você pode entrar no AWS Management Console ou no portal de AWS acesso. Para obter mais informações sobre como fazer login em AWS, consulte Como fazer login Conta da AWS no Guia do Início de Sessão da AWS usuário.

Se você acessar AWS programaticamente, AWS fornece um kit de desenvolvimento de software (SDK) e uma interface de linha de comando (CLI) para assinar criptograficamente suas solicitações usando suas credenciais. Se você não usa AWS ferramentas, você mesmo deve assinar as solicitações. Para obter mais informações sobre como usar o método recomendado para designar solicitações por conta própria, consulte Versão 4 do AWS Signature para solicitações de API no Guia do usuário do IAM.

Independente do método de autenticação usado, também pode ser necessário fornecer informações adicionais de segurança. Por exemplo, AWS recomenda que você use a autenticação multifator (MFA) para aumentar a segurança da sua conta. Para saber mais, consulte Autenticação multifator

no Guia do usuário do AWS IAM Identity Center e Usar a autenticação multifator da AWS no IAM no Guia do usuário do IAM.

### Conta da AWS usuário root

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos Serviços da AWS os recursos da conta. Essa identidade é chamada de usuário Conta da AWS raiz e é acessada fazendo login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável não usar o usuário-raiz para tarefas diárias. Proteja as credenciais do usuário-raiz e use-as para executar as tarefas que somente ele puder executar. Para obter a lista completa das tarefas que exigem login como usuário-raiz, consulte Tarefas que exigem credenciais de usuário-raiz no Guia do Usuário do IAM.

### Identidade federada

Como prática recomendada, exija que usuários humanos, incluindo usuários que precisam de acesso de administrador, usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório de usuários corporativo, de um provedor de identidade da web AWS Directory Service, do diretório do Identity Center ou de qualquer usuário que acesse usando credenciais fornecidas Serviços da AWS por meio de uma fonte de identidade. Quando as identidades federadas são acessadas Contas da AWS, elas assumem funções, e as funções fornecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, é recomendável usar o AWS IAM Identity Center. Você pode criar usuários e grupos no IAM Identity Center ou pode se conectar e sincronizar com um conjunto de usuários e grupos em sua própria fonte de identidade para uso em todos os seus Contas da AWS aplicativos. Para obter mais informações sobre o Centro de Identidade do IAM, consulte O que é o Centro de Identidade do IAM? no Guia do Usuário do AWS IAM Identity Center .

### Usuários e grupos do IAM

Um usuário do IAM é uma identidade dentro da sua Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, é recomendável contar com credenciais temporárias em vez de criar usuários do IAM com credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com usuários do IAM, é recomendável alternar as chaves de acesso. Para obter mais informações, consulte Alternar as chaves de acesso regularmente para casos de uso que exijam credenciais de longo prazo no Guia do Usuário do IAM.

Um grupo do IAM é uma identidade que especifica uma coleção de usuários do IAM. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdminse conceder a esse grupo permissões para administrar recursos do IAM.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte Casos de uso para usuários do IAM no Guia do usuário do IAM.

## Perfis do IAM

Uma função do IAM é uma identidade dentro da sua Conta da AWS que tem permissões específicas. Ele é semelhante a um usuário do IAM, mas não está associado a uma pessoa específica. Para assumir temporariamente uma função do IAM no AWS Management Console, você pode alternar de um usuário para uma função do IAM (console). Você pode assumir uma função chamando uma operação de AWS API AWS CLI ou usando uma URL personalizada. Para obter mais informações sobre métodos para usar perfis, consulte Métodos para assumir um perfil no Guia do usuário do IAM.

Perfis do IAM com credenciais temporárias são úteis nas seguintes situações:

- Acesso de usuário federado: para atribuir permissões a identidades federadas, é possível criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas por ele. Para ter mais informações sobre perfis para federação, consulte Criar um perfil para um provedor de identidade de terceiros (federação) no Guia do usuário do IAM. Se usar o Centro de Identidade do IAM, configure um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o Centro de Identidade do IAM correlaciona o conjunto de permissões a um perfil no IAM. Para obter informações sobre conjuntos de permissões, consulte Conjuntos de Permissões no Guia do Usuário do AWS IAM Identity Center .
- Permissões temporárias para usuários do IAM: um usuário ou um perfil do IAM pode presumir um perfil do IAM para obter temporariamente permissões diferentes para uma tarefa específica.
- Acesso entre contas: é possível usar um perfil do IAM para permitir que alguém (uma entidade principal confiável) em outra conta acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, com alguns Serviços da AWS, você pode anexar uma política diretamente a um recurso (em vez de usar uma função como proxy). Para conhecer a

diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

- Acesso entre serviços — Alguns Serviços da AWS usam recursos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicativos na Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões da entidade principal da chamada, usando um perfil de serviço ou um perfil vinculado ao serviço.
  - Sessões de acesso direto (FAS) — Quando você usa um usuário ou uma função do IAM para realizar ações AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. As solicitações do FAS são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte Sessões de acesso direto.
  - Perfil de serviço: um perfil de serviço é um perfil do IAM que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do Usuário do IAM.
  - Função vinculada ao serviço — Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode presumir o perfil de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para perfis vinculados ao serviço.
- Aplicativos em execução na Amazon EC2 — Você pode usar uma função do IAM para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma EC2 instância e fazendo solicitações AWS CLI de AWS API. Isso é preferível a armazenar chaves de acesso na EC2 instância. Para atribuir uma AWS função a uma EC2 instância e disponibilizá-la para todos os aplicativos, você cria um perfil de instância anexado à instância. Um perfil de instância contém a função e permite que programas em execução na EC2 instância recebam credenciais temporárias. Para obter mais informações, consulte Usar uma função do IAM para conceder permissões a aplicativos executados em EC2 instâncias da Amazon no Guia do usuário do IAM.

# Gerenciar o acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política é um objeto AWS que, quando associada a uma identidade ou recurso, define suas permissões. AWS avalia essas políticas quando um principal (usuário, usuário raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas é armazenada AWS como documentos JSON. Para obter mais informações sobre a estrutura e o conteúdo de documentos de políticas JSON, consulte Visão geral das políticas JSON no Guia do usuário do IAM.

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

Por padrão, usuários e perfis não têm permissões. Para conceder permissão aos usuários para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM aos perfis e os usuários podem assumir os perfis.

As políticas do IAM definem permissões para uma ação independentemente do método usado para executar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de função da AWS Management Console AWS CLI, da ou da AWS API.

## Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente no Guia do Usuário do IAM.

As políticas baseadas em identidade podem ser categorizadas como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou perfil. As políticas gerenciadas são políticas autônomas que você pode associar a vários usuários, grupos e funções em seu Conta da AWS. As políticas AWS gerenciadas incluem políticas gerenciadas e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do usuário do IAM.

## Políticas baseadas em recursos

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificado pode executar nesse atributo e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

## Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

O Amazon S3 e o AWS WAF Amazon VPC são exemplos de serviços que oferecem suporte. ACLs Para saber mais ACLs, consulte a visão geral da lista de controle de acesso (ACL) no Guia do desenvolvedor do Amazon Simple Storage Service.

## Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- Limites de permissões: um limite de permissões é um recurso avançado no qual você define o máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM (usuário ou perfil do IAM). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade com seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou o perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do usuário do IAM.

- Políticas de controle de serviço (SCPs) — SCPs são políticas JSON que especificam as permissões máximas para uma organização ou unidade organizacional (OU) em AWS Organizations. AWS Organizations é um serviço para agrupar e gerenciar centralmente vários Contas da AWS que sua empresa possui. Se você habilitar todos os recursos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as suas contas. O SCP limita as permissões para entidades nas contas dos membros, incluindo cada uma Usuário raiz da conta da AWS. Para obter mais informações sobre Organizations e SCPs, consulte Políticas de controle de serviços no Guia AWS Organizations do Usuário.
- Políticas de controle de recursos (RCPs) — RCPs são políticas JSON que você pode usar para definir o máximo de permissões disponíveis para recursos em suas contas sem atualizar as políticas do IAM anexadas a cada recurso que você possui. O RCP limita as permissões para recursos nas contas dos membros e pode afetar as permissões efetivas para identidades, incluindo a Usuário raiz da conta da AWS, independentemente de pertencerem à sua organização. Para obter mais informações sobre Organizations e RCPs, incluindo uma lista Serviços da AWS desse suporte RCPs, consulte Políticas de controle de recursos (RCPs) no Guia AWS Organizations do usuário.
- Políticas de sessão: são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em recursos. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações, consulte Políticas de sessão no Guia do usuário do IAM.

### Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de políticas estão envolvidos, consulte Lógica de avaliação de políticas no Guia do usuário do IAM.

## Como o Amazon Managed Blockchain (AMB) Query funciona com o IAM

Antes de usar o IAM para gerenciar o acesso ao AMB Query, saiba quais recursos do IAM estão disponíveis para uso com o AMB Query.

Recursos do IAM que você pode usar com o Amazon Managed Blockchain (AMB) Query

| Atributo do IAM | Suporte ao AMB Query |
|---|---|
| Políticas baseadas em identidade | Sim |
| Políticas baseadas em recurso | Não |
| Ações de políticas | Sim |
| Recursos de políticas | Não |
| Chaves de condição de políticas | Não |
| ACLs | Não |
| ABAC (tags em políticas) | Não |
| Credenciais temporárias | Sim |
| Permissões de entidade principal | Sim |
| Perfis de serviço | Não |
| Funções vinculadas ao serviço | Não |

Para ter uma visão de alto nível de como o AMB Query e outros AWS serviços funcionam com a maioria dos recursos do IAM, consulte AWS os serviços que funcionam com o IAM no Guia do usuário do IAM.

## Políticas baseadas em identidade para AMB Query

Compatível com políticas baseadas em identidade: sim

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário do IAM, grupo de usuários ou perfil. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente no Guia do Usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações e recursos permitidos ou negados, assim como as condições sob as quais as ações são permitidas ou negadas. Você não pode especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou perfil ao qual ela está anexada. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte Referência de elemento de política JSON do IAM no Guia do usuário do IAM.

Exemplos de políticas baseadas em identidade para AMB Query

Para ver exemplos de políticas baseadas em identidade do AMB Query, consulte. Exemplos de políticas baseadas em identidade para Amazon Managed Blockchain (AMB) Query

## Políticas baseadas em recursos no AMB Query

Compatibilidade com políticas baseadas em recursos: não

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificado pode executar nesse atributo e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Para permitir o acesso entre contas, você pode especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em recursos. Adicionar uma entidade principal entre contas à política baseada em recurso é apenas metade da tarefa de estabelecimento da relação de confiança. Quando o principal e o recurso são diferentes Contas da AWS, um administrador do IAM na conta confiável também deve conceder permissão à entidade principal (usuário ou função) para acessar o recurso. Eles concedem permissão ao anexar uma política baseada em identidade para a entidade. No entanto, se uma política baseada em recurso conceder acesso a uma entidade principal na mesma conta, nenhuma política baseada em identidade adicional será necessária. Consulte mais informações em Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

## Ações de política para AMB Query

Compatível com ações de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Action` de uma política JSON descreve as ações que podem ser usadas para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome da operação de AWS API associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Algumas operações também exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista de ações da AMB Query, consulte Actions Defined by Amazon Managed Blockchain (AMB) Query na Referência de Autorização de Serviço.

As ações de política no AMB Query usam o seguinte prefixo antes da ação:

```
managedblockchain-query:
```

Para especificar várias ações em uma única declaração, separe-as com vírgulas.

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

Para ver exemplos de políticas baseadas em identidade do AMB Query, consulte. Exemplos de políticas baseadas em identidade para Amazon Managed Blockchain (AMB) Query

## Recursos de política para AMB Query

Oferece compatibilidade com recursos de políticas: não

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou `NotResource`. Como prática recomendada, especifique um recurso usando seu nome do recurso da Amazon (ARN). Isso pode

ser feito para ações que oferecem compatibilidade com um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem compatibilidade com permissões em nível de recurso, como operações de listagem, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

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

Para ver uma lista dos tipos de recursos do AMB Query e seus ARNs, consulte Resources Defined by Amazon Managed Blockchain (AMB) Query na Referência de Autorização de Serviço. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte Actions Defined by Amazon Managed Blockchain (AMB) Query.

Para ver exemplos de políticas baseadas em identidade do AMB Query, consulte. Exemplos de políticas baseadas em identidade para Amazon Managed Blockchain (AMB) Query

## Chaves de condição de política para AMB Query

Compatível com chaves de condição de política específicas de serviço: não

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Condition` (ou bloco `Condition`) permite que você especifique condições nas quais uma instrução estiver em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usem agentes de condição, como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos de `Condition` em uma declaração ou várias chaves em um único elemento de `Condition`, a AWS os avaliará usando uma operação lógica `AND`. Se você especificar vários valores para uma única chave de condição, AWS avalia a condição usando uma `OR` operação lógica. Todas as condições devem ser atendidas antes que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar condições. Por exemplo, é possível conceder a um usuário do IAM permissão para acessar um recurso somente se ele estiver marcado com seu nome de usuário do IAM. Para obter mais informações, consulte Elementos da política do IAM: variáveis e tags no Guia do usuário do IAM.

AWS suporta chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as chaves de condição AWS globais, consulte as chaves de contexto de condição AWS global no Guia do usuário do IAM.

Para ver uma lista das chaves de condição do AMB Query, consulte Chaves de condição para a consulta do Amazon Managed Blockchain (AMB) na Referência de autorização de serviço. Para saber com quais ações e recursos você pode usar uma chave de condição, consulte Actions Defined by Amazon Managed Blockchain (AMB) Query.

Para ver exemplos de políticas baseadas em identidade do AMB Query, consulte. Exemplos de políticas baseadas em identidade para Amazon Managed Blockchain (AMB) Query

## ACLs em AMB Query

Suportes ACLs: Não

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

## ABAC com AMB Query

Oferece compatibilidade com ABAC (tags em políticas): não

O controle de acesso por atributo (ABAC) é uma estratégia de autorização que define permissões com base em atributos. Em AWS, esses atributos são chamados de tags. Você pode anexar tags a entidades do IAM (usuários ou funções) e a vários AWS recursos. Marcar de entidades e atributos é a primeira etapa do ABAC. Em seguida, você cria políticas de ABAC para permitir operações quando a tag da entidade principal corresponder à tag do recurso que ela estiver tentando acessar.

O ABAC é útil em ambientes que estão crescendo rapidamente e ajuda em situações em que o gerenciamento de políticas se torna um problema.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* ou chaves de condição `aws:TagKeys`.

Se um serviço for compatível com as três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço for compatível com as três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para obter mais informações sobre o ABAC, consulte Definir permissões com autorização do ABAC no Guia do usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte Usar controle de acesso baseado em atributos (ABAC) no Guia do usuário do IAM.

## Usando credenciais temporárias com o AMB Query

Compatível com credenciais temporárias: sim

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, incluindo quais Serviços da AWS funcionam com credenciais temporárias, consulte Serviços da AWS trabalhar com o IAM no Guia do usuário do IAM.

Você está usando credenciais temporárias se fizer login AWS Management Console usando qualquer método, exceto um nome de usuário e senha. Por exemplo, quando você acessa AWS usando o link de login único (SSO) da sua empresa, esse processo cria automaticamente credenciais temporárias. Você também cria automaticamente credenciais temporárias quando faz login no console como usuário e, em seguida, alterna perfis. Para obter mais informações sobre como alternar funções, consulte Alternar para um perfil do IAM (console) no Guia do usuário do IAM.

Você pode criar manualmente credenciais temporárias usando a AWS API AWS CLI ou. Em seguida, você pode usar essas credenciais temporárias para acessar AWS. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para obter mais informações, consulte Credenciais de segurança temporárias no IAM.

## Permissões principais entre serviços para AMB Query

Compatibilidade com o recurso de encaminhamento de sessões de acesso (FAS): sim

Quando você usa um usuário ou uma função do IAM para realizar ações AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. As solicitações do FAS são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte Sessões de acesso direto.

## Funções de serviço para AMB Query

Compatível com perfis de serviço: não

O perfil de serviço é um perfil do IAM que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do Usuário do IAM.

> **Warning**
>
> Alterar as permissões de uma função de serviço pode interromper a funcionalidade do AMB Query. Edite funções de serviço somente quando o AMB Query fornecer orientação para fazer isso.

### Funções vinculadas ao serviço para AMB Query

Compatível com perfis vinculados ao serviço: Não

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode presumir o perfil de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para funções vinculadas ao serviço.

Para obter detalhes sobre como criar ou gerenciar perfis vinculados a serviços, consulte Serviços da AWS que funcionam com o IAM. Encontre um serviço na tabela que inclua um `Yes` na coluna Perfil vinculado ao serviço. Escolha o link Sim para visualizar a documentação do perfil vinculado a serviço desse serviço.

## Exemplos de políticas baseadas em identidade para Amazon Managed Blockchain (AMB) Query

Por padrão, usuários e funções não têm permissão para criar ou modificar recursos do AMB Query. Eles também não podem realizar tarefas usando a AWS API AWS Management Console, AWS Command Line Interface (AWS CLI) ou. Para conceder permissão aos usuários para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM aos perfis e os usuários podem assumir os perfis.

Para aprender a criar uma política baseada em identidade do IAM ao usar esses documentos de política em JSON de exemplo, consulte Criar políticas do IAM (console) no Guia do usuário do IAM.

Para obter detalhes sobre ações e tipos de recursos definidos pelo AMB Query, incluindo o formato de cada um dos ARNs tipos de recursos, consulte Actions, Resources and Condition Keys for Amazon Managed Blockchain (AMB) Query na Referência de Autorização de Serviço.

Tópicos

- Práticas recomendadas de política
- Permitir que os usuários visualizem suas próprias permissões
- Acessando ações específicas da API de consulta do Amazon Managed Blockchain (AMB)

## Práticas recomendadas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir recursos do AMB Query em sua conta. Essas ações podem incorrer em custos para sua Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para obter mais informações, consulte Políticas gerenciadas pela AWS ou Políticas gerenciadas pela AWS para funções de trabalho no Guia do usuário do IAM.
- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em recursos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre como usar o IAM para aplicar permissões, consulte Políticas e permissões no IAM no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: você pode adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, você pode escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como AWS CloudFormation. Para obter mais informações, consulte Elementos da política JSON do IAM: condição no Guia do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas

sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações práticas para ajudar a criar políticas seguras e funcionais. Para obter mais informações, consulte Validação de políticas do IAM Access Analyzer no Guia do Usuário do IAM.

- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para obter mais informações, consulte Configuração de acesso à API protegido por MFA no Guia do Usuário do IAM.

Para obter mais informações sobre as práticas recomendadas do IAM, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.

## Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou 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": "*"
        }
    ]
}
```

## Acessando ações específicas da API de consulta do Amazon Managed Blockchain (AMB)

> **Note**
>
> Para acessar a AMB Query para fazer chamadas de API, você precisará de credenciais de usuário (`AWS_ACCESS_KEY_IDeAWS_SECRET_ACCESS_KEY`) que tenham as permissões apropriadas do IAM para a AMB Query.

### Example Política do IAM para acessar todas as consultas do Amazon Managed Blockchain (AMB) APIs

Este exemplo concede a um usuário do IAM em seu Conta da AWS acesso a todas as consultas APIs AMB.

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

```
}
```

Example Política do IAM para acessar a consulta **`ListTransactions`** do Amazon Managed Blockchain (AMB) e **`GetTransaction`** APIs

Este exemplo concede a um usuário do IAM em seu Conta da AWS acesso à consulta `ListTransaction` AMB e `GetTransaction` APIs

> **Note**
>
> Você pode substituir ou adicionar o APIs no exemplo por outro APIs para dar acesso a outros ou mais APIs. Para obter uma lista de consultas AMB APIs, consulte o Guia de referência da API de consulta do Amazon Managed Blockchain (AMB).

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

## Solução de problemas de identidade e acesso à consulta Amazon Managed Blockchain (AMB)

Use as informações a seguir para ajudá-lo a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com o AMB Query e o IAM.

Tópicos

## Não estou autorizado a realizar uma ação no AMB Query

Se você receber uma mensagem de erro informando que não tem autorização para executar uma ação, suas políticas deverão ser atualizadas para permitir que você realize a ação.

O erro do exemplo a seguir ocorre quando o usuário do IAM `mateojackson` tenta usar o console para visualizar detalhes sobre um atributo *`my-example-widget`* fictício, mas não tem as permissões `managedblockchain-query::`*`GetWidget`* fictícias.

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

Nesse caso, a política do usuário `mateojackson` deve ser atualizada para permitir o acesso ao recurso *`my-example-widget`* usando a ação `managedblockchain-query::`*`GetWidget`*.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

# Métricas de uso da API de consulta do Amazon Managed Blockchain (AMB) na Amazon CloudWatch

## Métricas de uso da API na Amazon CloudWatch

As métricas de uso da API publicadas CloudWatch correspondem às cotas do serviço Amazon Managed Blockchain (AMB) Query. Você pode configurar alarmes para alertá-lo quando seu uso se aproximar de uma cota de serviço. Para obter mais informações sobre a CloudWatch integração com cotas de serviços, consulte as métricas de uso da AWS no Guia do CloudWatch usuário da Amazon.

O AMB Query publica as seguintes métricas de API no `AWS/Usage` namespace, com o nome do serviço. `Amazon Managed Blockchain Query`

| Métrica | Descrição |
| --- | --- |
| `CallCount` | O número total de chamadas feitas para uma API no AMB Query. SUM representa o número total de chamadas para a API durante o período especificado. |

O Amazon Managed Blockchain (AMB) Query publica métricas de uso no `AWS/Usage` namespace com as seguintes dimensões.

| Dimensão | Descrição |
| --- | --- |
| Serviço | O nome do AWS serviço que contém o recurso. `Amazon Managed Blockchain Query`sempre será o valor dessa dimensão. |
| Tipo | O tipo da entidade que está sendo relatada. `API`sempre será o valor dessa dimensão. |
| Recurso | O tipo de recursos que estão sendo relatados. O nome da operação da API AMB Query usada será o valor dessa dimensão. |

| Dimensão | Descrição |
| --- | --- |
| Classe | A classe do recurso que está sendo relatado. Nonesempre será o valor dessa dimensão. |

# Histórico de documentos do Guia do usuário do AMB Query

A tabela a seguir descreve as versões da documentação do AMB Query.

| Alteração | Descrição | Data |
| --- | --- | --- |
| O AMB Query suporta identificadores de transações e hashes de transações de Bitcoin | Para redes Bitcoin, as operações da API AMB Query suportam tanto o identificador da transação (`transactionId`) quanto o hash da transação (`transactionHash`). | 21 de março de 2024 |
| Support para métricas de uso de API na Amazon CloudWatch | O AMB Query adicionou suporte para métricas de uso da API em. CloudWatch Essas métricas de uso correspondem às cotas do serviço AMB Query. | 8 de fevereiro de 2024 |
| Support para transações que não atingiram a finalidade | O AMB Query adicionou suporte para transações que não atingiram a finalidade. Também remove o suporte para a `status` propriedade da resposta da `GetTransaction` operação. Em vez disso, você usará `executionStatus` as propriedades `confirmationStatus` e para determinar o status da transação. | 1.º de fevereiro de 2024 |
| Depreciação da `status` propriedade no tipo de dados de transação | A consulta Amazon Managed Blockchain (AMB) tornou obsoleta a `status` proprieda | 20 de dezembro de 2023 |

| | | |
|---|---|---|
| | de no tipo de dados de transação. Você deve usar os `executionStatus` campos `confirmationStatus` e para determinar se `status` a transação é `FINAL` ou`FAILED`. | |
| Support para Sepolia Testnet | O Amazon Managed Blockchain (AMB) Query agora suporta consultas na Ethereum Sepolia Testnet. | 19 de outubro de 2023 |
| Support para contratos de ativos | Você pode usar a operação de `ListAssetContracts` API para listar os implantados por um determinado endereço. Além disso, se você tiver o endereço do contrato, poderá usar a operação da `GetAssetContract` API para recuperar os detalhes do contrato. | 16 de outubro de 2023 |
| Support para Bitcoin Testnet | O Amazon Managed Blockchain (AMB) Query agora suporta consultas na Bitcoin Testnet. | 16 de outubro de 2023 |
| Lançamento inicial | Lançamento inicial do serviço AMB Query. | 27 de julho de 2023 |

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.