aws

Guía para desarrolladores

Amazon Managed Blockchain Query

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Managed Blockchain Query: Guía para desarrolladores

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas registradas y la imagen comercial de Amazon no se pueden utilizar en ningún producto o servicio que no sea de Amazon de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

¿Qué es Amazon Managed Blockchain (AMB) Query?

Amazon Managed Blockchain (AMB) es un servicio totalmente gestionado diseñado para ayudarle a crear aplicaciones Web3 resilientes en cadenas de bloques públicas y privadas. Utilice AMB Access para acceder de forma instantánea y sin servidor a múltiples cadenas de bloques. Cree sus aplicaciones preparadas para Web3 sin necesidad de implementar una infraestructura de cadena de bloques especializada ni de mantenerlas conectadas a la red de cadenas de bloques. Con AMB Query, puede utilizar operaciones de API fáciles de usar para los desarrolladores para acceder a datos históricos y en tiempo real de múltiples cadenas de bloques. Los datos estandarizados de la cadena de bloques se pueden integrar con los servicios de AWS, sin necesidad de una infraestructura de cadena de bloques especializada o ETL (extracción, transformación y carga). Todas las funciones de AMB se escalan de forma segura para crear aplicaciones de consumo estándar y de nivel institucional.

Amazon Managed Blockchain (AMB) Query proporciona acceso sin servidor a conjuntos de datos estandarizados de múltiples cadenas de bloques con operaciones de API fáciles de usar para los desarrolladores. Puede usar AMB Query para enviar rápidamente aplicaciones que requieran datos de una o más cadenas de bloques públicas, sin la sobrecarga de analizar los datos de la cadena de bloques, rastrear los contratos y mantener una infraestructura de indexación especializada. Ya sea que estés analizando los saldos históricos de fichas en busca de fichas fungibles o no fungibles (NFTs), consultando el historial de transacciones de una dirección de cartera determinada o realizando análisis de datos sobre la distribución de criptomonedas nativas, como Ether, AMB Query te permite acceder a los datos de la cadena de bloques.

¿Es la primera vez que utiliza AMB Query?

Si es la primera vez que utiliza AMB Query, le recomendamos que comience leyendo las siguientes secciones:

- Conceptos clave: consulta de Amazon Managed Blockchain (AMB)
- Configuración de una consulta de Amazon Managed Blockchain (AMB)
- Introducción a Amazon Managed Blockchain (AMB) Query
- Casos de uso con Amazon Managed Blockchain (AMB) Query

Conceptos clave: consulta de Amazon Managed Blockchain (AMB)

> **Note**
>
> En esta guía se asume que está familiarizado con los conceptos esenciales de la cadena de bloques. Estos conceptos incluyen la descentralización, los tokens, los contratos, las transacciones, las carteras proof-of-work, las claves públicas y privadas, las apuestas, la minería, las partidas a medias y otros.

Amazon Managed Blockchain (AMB) Query le proporciona un acceso cómodo a los datos de red de varias cadenas de bloques, lo que le facilita la extracción de datos contextuales relacionados con la actividad de la cadena de bloques. Puede utilizar AMB Query para leer datos de redes públicas de cadenas de bloques, como Bitcoin Mainnet y Ethereum Mainnet. También puede obtener información, como los saldos actuales e históricos de las direcciones, o puede obtener una lista de las transacciones de la cadena de bloques durante un período de tiempo determinado. Además, puede obtener detalles de una transacción determinada, como los eventos de la transacción, que puede analizar más a fondo o utilizar en la lógica empresarial para sus aplicaciones.

Consideraciones y limitaciones para usar Amazon Managed Blockchain (AMB) Query

Cuando utilice AMB Query, tenga en cuenta lo siguiente:

- Regiones disponibles

  La consulta AMB es compatible con la `us-east-1` región EE.UU. Este (Norte de Virginia).

- Service endpoints

  Se puede acceder a AMB Query mediante el siguiente punto final:

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

- Redes de cadenas de bloques compatibles

AMB Query es compatible con las siguientes redes públicas de cadenas de bloques:

- Bitcoin Mainnet: la red pública de cadenas de bloques de Bitcoin que está protegida por proof-of-work consenso y en la que se emite y realiza transacciones con la criptomoneda Bitcoin (BTC). Las transacciones en Mainnet tienen un valor real (es decir, incurren en costes reales) y se registran en la cadena de bloques pública.
- Bitcoin Testnet: la red de pruebas de la red principal de Bitcoin. El Bitcoin (BTC) de esta red es independiente y distinto del BTC de la red principal y, por lo general, no tiene ningún valor.
- Ethereum Mainnet: la red proof-of-stake principal de la cadena de bloques pública de Ethereum. Las transacciones en Mainnet tienen un valor real (es decir, incurren en costos reales) y se registran en el libro mayor distribuido.
- Sepolia Testnet: la red de pruebas para la red principal de Ethereum. El éter (ETH) de esta red es independiente y distinto del ETH de la red principal y, por lo general, no tiene ningún valor.

- Tokens y contratos de cadena de bloques compatibles

  AMB Query admite los siguientes tokens de contrato nativos y estándar de Ethereum.

  - Tokens nativos de cadenas de bloques públicas
    - Bitcoin (BTC): este es el token nativo de las cadenas de bloques relacionadas con Bitcoin.
    - Ether (ETH): este es el token nativo de las cadenas de bloques relacionadas con Ethereum.

  - Estándares contractuales de Ethereum
    - Estándar de fichas ERC-20: el ERC-20 es un estándar para fichas fungibles. Tiene una propiedad que hace que cada token ERC-20 sea exactamente igual (en tipo y valor) a otro token ERC-20 acuñado, lo que significa que un token es y será siempre igual a todos los demás tokens. Para obtener más información, consulta el estándar de fichas ERC-20 en Ethereum.org.
    - Estándar de fichas no fungibles ERC-721: el ERC-721 es un estándar para fichas no fungibles (). NFTs Este tipo de token es único y puede tener un valor diferente al de otro token del mismo contrato, posiblemente debido a su antigüedad, rareza u otras propiedades. Para obtener más información, consulta el estándar de fichas ERC-721 en Ethereum.org.

      Estándar ERC-1155 para múltiples fichas: el ERC-1155 es un estándar que crea una interfaz contractual que puede representar y controlar cualquier número de tipos de fichas fungibles y no fungibles. De esta forma, el token ERC-1155 puede funcionar de la misma manera que los tokens ERC-20 y ERC-721, e incluso funcionar como ambos al mismo tiempo. El token

ERC-1155 mejora la funcionalidad de los estándares ERC-20 y ERC-721, lo que lo hace más eficiente y corrige errores de implementación obvios. Para obtener más información, consulte el estándar de token ERC-1155 en Ethereum.org.

- Finalidad

  En las cadenas de bloques, la finalidad significa que es poco probable que las transacciones válidas se anulen. Para la red principal de Bitcoin, AMB Query considera que una transacción es definitiva después de 6 bloques. En el caso de la red de pruebas de Bitcoin, se considera que una transacción es definitiva tras 6 bloques o 60 minutos, lo que ocurra primero. En el caso de las redes Ethereum compatibles, AMB Query considera que una transacción es definitiva después de 64 bloques.

  Las operaciones de API de saldos simbólicos y contratos de AMB Query solo devuelven datos definitivos. Sin embargo, las operaciones de la API de transacciones y eventos de transacción de AMB Query pueden devolver datos de transacciones confirmadas en la red de cadenas de bloques, incluso si aún no han alcanzado la finalidad definitiva.

- No se admiten direcciones NULAS

  AMB Query no admite la dirección `NULL` (`0x0000000000000000000000000000000000000000`).

- Firma (versión 4): firma de llamadas a la API

  Al realizar llamadas a la AMB Query APIs, puede hacerlo a través de una conexión HTTPS autenticada mediante el proceso de firma de la versión 4 de Signature. Esto significa que solo los directores de IAM autorizados de la AWS cuenta pueden realizar llamadas a la API de AMB Query. Para ello, se deben proporcionar AWS las credenciales (un identificador de clave de acceso y una clave de acceso secreta) junto con la llamada.

  > **Important**
  >
  > No inserte las credenciales del cliente en las aplicaciones orientadas al usuario.

- AMB Query admite identificadores de transacciones y hashes de transacciones de Bitcoin

En las redes de Bitcoin, las operaciones de la API de AMB Query admiten tanto el identificador de transacción (`transactionId`) como el hash de transacción (). `transactionHash` `transactionId`Se trata de un hash SHA doble de la transacción que no incluye los datos de los testigos. `transactionHash`Se trata de un hash SHA doble de la transacción que incluye los datos del testigo (también conocido como identificador de la transacción del testigo).

Al invocar las operaciones de la `ListTransactionEvents`API `GetTransaction`o para las redes de Bitcoin, puede especificar la `transactionId` o la. `transactionHash` Además, todas las operaciones de consulta AMB en las redes de Bitcoin que devuelvan a `transactionId` o `transactionHash` a incluirán ambos valores como parte de la respuesta.

Configuración de una consulta de Amazon Managed Blockchain (AMB)

Antes de utilizar Amazon Managed Blockchain (AMB) Query por primera vez, siga los pasos de esta sección para crear una AWS cuenta. En la siguiente sección, se explica cómo empezar a utilizar AMB Query.

Requisitos y consideraciones previos

Antes de utilizar Amazon Web Services por primera vez, debe tener una AWS cuenta.

Inscríbase en AWS

Cuando te registras en Amazon Web Services (AWS), tu AWS cuenta se registra automáticamente para todos Servicios de AWS, incluida Amazon Managed Blockchain (AMB) Query. Solo se le cobrará por los servicios que utilice.

Si Cuenta de AWS ya tienes una, continúa con el siguiente paso. Si no dispone de una Cuenta de AWS, utilice el siguiente procedimiento para crear una.

Para crear una AWS cuenta

1. Abre el https://portal.aws.amazon.com/billing/registro.
2. Siga las instrucciones que se le indiquen.

   Parte del procedimiento de registro consiste en recibir una llamada telefónica o mensaje de texto e indicar un código de verificación en el teclado del teléfono.

   Cuando te registras en un Cuenta de AWS, Usuario raíz de la cuenta de AWSse crea un. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar tareas que requieren acceso de usuario raíz.

Cree un usuario de IAM con los permisos adecuados

Para crear AMB Query y trabajar con ella, debes crear un director AWS Identity and Access Management (IAM) (usuario o grupo) con permisos que permitan realizar las acciones necesarias de la cadena de bloques gestionada.

Solo los directores de IAM pueden realizar solicitudes a la API de AMB Query. Al realizar llamadas a la AMB Query APIs, puede hacerlo a través de una conexión HTTPS autenticada mediante el proceso de firma de la versión 4 de Signature. Esto significa que solo los directores de IAM autorizados de la AWS cuenta pueden realizar llamadas a la API de AMB Query. Para ello, se deben proporcionar AWS las credenciales (un identificador de clave de acceso y una clave de acceso secreta) junto con la llamada.

Para obtener información sobre cómo crear un usuario de IAM, consulte Crear un usuario de IAM en su AWS cuenta. Para obtener más información sobre cómo adjuntar una política de permisos a un usuario, consulte Cambiar los permisos de un usuario de IAM. Para ver un ejemplo de una política de permisos que puede utilizar para conceder permiso a un usuario para que trabaje con AMB Query, consulte. Ejemplos de políticas basadas en identidad para Amazon Managed Blockchain (AMB) Query

Instale y configure el AWS Command Line Interface

Si aún no lo ha hecho, instale la interfaz de AWS línea de comandos (CLI) más reciente para trabajar con AWS los recursos de un terminal. Para obtener más información, consulte Instalación o actualización de la versión de AWS CLI más reciente.

> **Note**
>
> Para acceder a la CLI, necesita un ID de clave de acceso y una clave de acceso secreta. Cuando sea posible, utilice credenciales temporales en lugar de claves de acceso. Las credenciales temporales incluyen un ID de clave de acceso y una clave de acceso secreta, pero, además, incluyen un token de seguridad que indica cuándo caducan las credenciales. Para obtener más información, consulte Uso de credenciales temporales con AWS recursos en la Guía del usuario de IAM.

Utilice la consulta AWS Management Console para consultar cadenas de bloques mediante Amazon Managed Blockchain (AMB) Query

Puede acceder a Amazon Managed Blockchain (AMB) Query y realizar consultas en las redes de cadenas de bloques compatibles mediante. AWS Management Console Los siguientes pasos muestran cómo hacerlo:

1. Abra la consola Amazon Managed Blockchain en https://console.aws.amazon.com/managedblockchain/.
2. Elija el editor de consultas en la sección de consultas.
3. Elija una de las redes Blockchain compatibles.
4. Elija el tipo de consulta que desea ejecutar.
5. Introduzca los parámetros relevantes para el tipo de consulta que ha seleccionado y ejecute la consulta.

AMB Query ejecutará su consulta y verá los resultados en la ventana de resultados de la consulta.

Introducción a Amazon Managed Blockchain (AMB) Query

Utilice los step-by-step tutoriales de esta sección para aprender a realizar tareas con Amazon Managed Blockchain (AMB) Query. Estos procedimientos requieren algunos requisitos previos. Si es la primera vez que utiliza AMB Query, puede consultar la sección de configuración de esta guía. Para obtener más información, consulte Configuración de una consulta de Amazon Managed Blockchain (AMB).

> **Note**
>
> Algunas variables de estos ejemplos se han ocultado deliberadamente. Sustitúyalas por otras válidas antes de ejecutar estos ejemplos.

Temas

Cree una política de IAM para acceder a las operaciones de la API de consultas de AMB

Para realizar solicitudes a la API AMB Query, debe utilizar las credenciales de usuario (AWS_ACCESS_KEY_ID y AWS_SECRET _ACCESS_KEY) que tengan los permisos de IAM adecuados para Amazon Managed Blockchain (AMB) Query. En una terminal con los AWS CLI instalados, ejecute el siguiente comando para crear una política de IAM que permita acceder a las operaciones de la API de consultas de AMB:

```
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
```

Después de crear la política, asocie esa política al rol de un usuario de IAM para que surta efecto. En el AWS Management Console, navegue hasta el servicio de IAM y asocie la política `AmazonManagedBlockchainQueryAccess` al rol asignado al usuario de IAM que utilizará el servicio. Para obtener más información, consulte Crear un rol y asignarlo a un usuario de IAM.

> **Note**
>
> AWS recomienda dar acceso a operaciones de API específicas en lugar de utilizar el comodín. * Para obtener más información, consulte Acceso a acciones específicas de la API Query de Amazon Managed Blockchain (AMB).

Realice solicitudes de API de consultas de Amazon Managed Blockchain (AMB) mediante Go

Con Amazon Managed Blockchain (AMB) Query, puede crear aplicaciones que dependan del acceso instantáneo a los datos de la cadena de bloques una vez confirmados en la cadena de bloques, incluso si aún no han alcanzado la finalidad definitiva. AMB Query permite varios casos de uso, como rellenar el historial de transacciones de una cartera, proporcionar información contextual sobre una transacción en función de su hash de transacción u obtener el saldo de un token nativo, así como de los tokens ERC-721, ERC-1155 y ERC-20.

Los siguientes ejemplos se crearon en el lenguaje Go y utilizan las operaciones de la API AMB Query. Para obtener más información sobre Go, consulte la documentación de Go. Para obtener más

información sobre la API de consultas de AMB, consulte la documentación de referencia de la API de consultas de Amazon Managed Blockchain (AMB).

Los ejemplos siguientes utilizan las acciones `ListTransactions` y las de la `GetTransaction` API para obtener primero una lista de todas las transacciones de una dirección de propiedad externa (EOA) determinada en la red principal de Ethereum y, a continuación, el siguiente ejemplo recupera los detalles de la transacción de una sola transacción de la lista.

Example — Realiza la acción de la **`ListTransactions`** API con Go

Copia el siguiente código en un archivo nombrado `listTransactions.go` en el ListTransactionsdirectorio.

```
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)
    }
}
```

Después de guardar el archivo, ejecute el código mediante el siguiente comando dentro del ListTransactionsdirectorio:`go run listTransactions.go`.

El resultado que se muestra a continuación es similar al siguiente:

```
{
  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 — Realice la acción **`GetTransaction`** de la API mediante Go

En este ejemplo, se usa un hash de transacción del resultado anterior. Copie el siguiente código en un archivo con un nombre `GetTransaction.go` en el GetTransactiondirectorio.

```
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)
    }
}
```

Después de guardar el archivo, ejecute el código mediante el siguiente comando dentro del GetTransactiondirectorio:`go run GetTransaction.go`.

El resultado que se muestra a continuación es similar al siguiente:

```
{
  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
  }
}
```

La `GetTokenBalance` API te permite obtener el saldo de los tokens nativos (ETH y BTC), que se puede utilizar para obtener el saldo actual de una cuenta de propiedad externa (EOA) en un momento dado.

Example — Usa la acción de la **`GetTokenBalance`** API para obtener el saldo de un token nativo en Go

En el siguiente ejemplo, utilizas la `GetTokenBalance` API para obtener el saldo de una dirección en Ether (ETH) en la red principal de Ethereum. Copia el siguiente código en un archivo con un nombre `GetTokenBalanceEth.go` en el GetTokenBalancedirectorio.

```
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)
    }
}
```

Después de guardar el archivo, ejecute el código mediante el siguiente comando dentro del GetTokenBalancedirectorio:`go run GetTokenBalanceEth.go`.

El resultado que se muestra a continuación es similar al siguiente:

```
{
  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"
  }
}
```

Realice solicitudes de API de consultas de Amazon Managed Blockchain (AMB) mediante Node.js

Para ejecutar estos ejemplos de nodos, se deben cumplir los siguientes requisitos previos:

1. Debe tener el administrador de versiones de nodos (nvm) y Node.js instalados en su máquina. Puede encontrar las instrucciones de instalación de su sistema operativo aquí.
2. Utilice el `node --version` comando y confirme que está utilizando la versión 14 o superior de Node. Si es necesario, puede usar el `nvm install 14` comando, seguido del `nvm use 14` comando para instalar la versión 14.
3. Las variables `AWS_ACCESS_KEY_ID` de entorno `AWS_SECRET_ACCESS_KEY` deben contener las credenciales asociadas a la cuenta.

   Exporte estas variables como cadenas en su cliente mediante los siguientes comandos. Sustituya los valores resaltados a continuación por los valores correspondientes de la cuenta de usuario de IAM.

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

> **Note**
>
> - Una vez cumplidos todos los requisitos previos, puede enviar solicitudes firmadas a través de HTTPS para acceder a las operaciones de la API de consultas de Amazon Managed Blockchain (AMB) y realizar solicitudes mediante el módulo https nativo de Node.js, o puede utilizar una biblioteca de terceros, como AXIOS, y recuperar datos de AMB Query.
> - En estos ejemplos se utiliza un cliente HTTP de terceros para Node.js, pero también se puede utilizar el AWS JavaScript SDK para realizar solicitudes a AMB Query.
> - En el siguiente ejemplo, se muestra cómo realizar solicitudes a la API de AMB Query mediante Axios y los módulos del AWS SDK para SiGv4.

Copie el siguiente `package.json` archivo en el directorio de trabajo de su entorno 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 el saldo histórico de fichas de una dirección de propiedad externa (EOA) específica mediante la API de consulta AMB **`GetTokenBalance`**

Puedes usar la `GetTokenBalance` API para obtener el saldo de varios tokens (por ejemplo, ERC2 0 ERC721, y ERC1155) y monedas nativas (por ejemplo, ETH y BTC), que puedes usar para obtener el saldo actual de una cuenta de propiedad externa (EOA) en función de un historial `timestamp` (marca de tiempo de Unix: segundos). En este ejemplo, utilizas la `GetTokenBalance`API para obtener el saldo de direcciones de un token de ERC2 0, USDC, en la red principal de Ethereum.

Para probar la `GetTokenBalance` API, copia el siguiente código en un archivo denominado `token-balance.js` y guárdalo en el mismo directorio de trabajo:

```
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 ejecutar el código, abre una terminal en el mismo directorio que tus archivos y ejecuta el siguiente comando:

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

Este comando ejecuta el script y pasa los argumentos definidos en el código para solicitar el saldo de ERC2 0 USDC de la EOA que cotiza en la red principal de Ethereum. La respuesta será similar a la siguiente:

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

Realice solicitudes de API de consultas de Amazon Managed Blockchain (AMB) mediante Python

Para ejecutar estos ejemplos de Python, se deben cumplir los siguientes requisitos previos:

1. Debe tener Python instalado en su máquina. Puede encontrar las instrucciones de instalación para su sistema operativo aquí.
2. Instale el AWS SDK para Python (Boto3).
3. Instale la interfaz de línea de AWS comandos y ejecute el comando `aws configure` para establecer las variables para su `Access Key ID`, `Secret Access Key`, y. `Region`

Una vez cumplidos todos los requisitos previos, puede utilizar el AWS SDK para Python a través de HTTPS para realizar solicitudes a la API de consultas de Amazon Managed Blockchain (AMB).

El siguiente ejemplo de Python usa módulos de boto3 para enviar solicitudes adjuntas con los encabezados SigV4 necesarios a la operación AMB Query API. `ListTransactionEvents` Este ejemplo recupera una lista de eventos emitidos por una transacción determinada en la red principal de Ethereum.

Copie el siguiente `list-transaction-events.py` archivo en el directorio de trabajo de su entorno 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 ejecutar el código de ejemploListTransactionEvents, guarde el archivo en su directorio de trabajo y, a continuación, ejecute el comandopython3 list-transaction-events.py. Este comando ejecuta el script y pasa los argumentos definidos en el código para solicitar los eventos asociados al hash de transacción dado en la red principal de Ethereum. La respuesta será similar a la siguiente:

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

Utilice Amazon Managed Blockchain (AMB) Query AWS Management Console para ejecutar la operación GetTokenBalance

El siguiente ejemplo muestra cómo obtener el saldo de un token en la red principal de Ethereum mediante una consulta de Amazon Managed Blockchain (AMB) en el AWS Management Console

Example

1. Abra la consola Amazon Managed Blockchain en https://console.aws.amazon.com/managedblockchain/.
2. Elija el editor de consultas en la sección de consultas.
3. Elija ETHEREUM_MAINNET como red de cadena de bloques.
4. Elija GetTokenBalancecomo tipo de consulta.
5. Introduce tu dirección de cadena de bloques para el token.
6. Introduce la dirección del contrato para el token.
7. Introduzca el ID de token opcional para el token.
8. Selecciona la fecha A para el saldo del token.
9. Introduce la hora A opcional para el saldo simbólico.
10. Elija Ejecutar consulta.

AMB Query ejecutará su consulta y verá los resultados en la ventana de resultados de la consulta.

Casos de uso con Amazon Managed Blockchain (AMB) Query

En este tema se proporciona una lista de los casos de uso de AMB Query.

Temas

Consulta los saldos actuales e históricos de los tokens

La `GetTokenBalance`API obtiene el saldo de los tokens admitidos (ERC20, ERC721, ERC1155) y las monedas nativas (ETH, BTC) para obtener el saldo actual o histórico mediante una marca de tiempo universal (marca de tiempo de Unix, en segundos) de cuentas de propiedad externa (). EOAs Por ejemplo, puedes usar la operación de la `GetTokenBalance` API para obtener el saldo de direcciones del token ERC2 0, USDC, en la red principal de Ethereum. También puedes recuperar por lotes los saldos de los tokens y las monedas nativas mediante la operación de API. `BatchGetTokenBalance`

Para obtener más información, consulte la Guía de referencia de consultas de Amazon Managed Blockchain (AMB).

Recupera datos históricos de transacciones

Con Amazon Managed Blockchain (AMB) Query, puede recuperar datos históricos de cadenas de bloques públicas como Ethereum y Bitcoin. Esta función permite varios casos de uso, como recuperar el historial de transacciones en una cartera de cadena de bloques o proporcionar información contextual sobre una transacción en función de su hash de transacción. Puedes usar la operación de `ListTransactions`API para obtener una lista de transacciones de una dirección de propiedad externa (EOA) determinada en la red principal de Ethereum y, luego, puedes usar

la operación de `GetTransaction`API para recuperar los detalles de la transacción de una sola transacción de la lista.

Para obtener más información, consulte la Guía de referencia de consultas de Amazon Managed Blockchain (AMB).

Obtenga todos los saldos simbólicos de una dirección determinada

Puede usar la operación de la `ListTokenBalances`API para obtener saldos en carteras, interfaces de usuario, utilidades web3 y más. Esta operación de API devuelve una lista de todos los saldos de una dirección entre los tokens (ERC20, ERC721, ERC1155) y las monedas nativas (ETH, BTC) de una cadena de bloques pública determinada mediante una sola operación de API. Por ejemplo, puedes proporcionar una dirección de propiedad externa (EOA) y una red (la red principal de Ethereum) y, en la respuesta, puedes recibir una lista de los saldos de tokens y monedas nativas.

Para obtener más información, consulte la Guía de referencia de consultas de Amazon Managed Blockchain (AMB).

Enumere los eventos emitidos para una transacción

Puede utilizar la operación de `ListTransactionEvents`API para recuperar una lista de los eventos contractuales que se emiten como resultado de una transacción determinada, identificados por su hash (identificador de transacción). Por ejemplo, puedes utilizarla `ListTransactionEvents`para recuperar los eventos resultantes de una transacción que invoque una función de un contrato de token ERC2 0 en la cadena de bloques de Ethereum, como un evento de transferencia o un evento de retirada del contrato ERC2 0.

Para obtener más información, consulte la Guía de referencia de consultas de Amazon Managed Blockchain (AMB).

Obtenga todos los tokens acuñados mediante un contrato

Puede utilizar la operación de `ListTokenBalances`API para obtener una lista de todos los tokens admitidos (ERC20, ERC721, ERC1155) acuñados por un contrato cuando se introduce la dirección del contrato. Por ejemplo, puedes recuperar información relacionada con los tokens no fungibles (NFTs) acuñados según el estándar ERC721 contractual en la cadena de bloques de Ethereum mediante la operación de API. `ListTokenBalances`

Para obtener más información, consulte la Guía de referencia de consultas de Amazon Managed Blockchain (AMB).

Enumere los contratos y obtenga información sobre los contratos

Puede usar la operación de `ListAssetContracts`API para enumerar los contratos ERC-721, ERC-1155 o ERC-20 implementados por una dirección determinada. Además, si tienes la dirección del contrato, puedes usar la operación de `GetAssetContract`API para recuperar las propiedades del contrato, como el tipo de contrato, la dirección del implementador y los metadatos de los tokens relevantes.

Para obtener más información, consulte la Guía de referencia de consultas de Amazon Managed Blockchain (AMB).

Referencia de la API de consultas de Amazon Managed Blockchain (AMB)

Amazon Managed Blockchain (AMB) Query proporciona operaciones de API para consultar cadenas de bloques compatibles. Esto incluye APIs la consulta de tokens, transacciones y contratos. Para obtener más información, consulta la referencia de la API de consultas de AMB.

Consulta de seguridad en Amazon Managed Blockchain (AMB)

La seguridad en la nube AWS es de máxima prioridad. Como AWS cliente, usted se beneficia de los centros de datos y las arquitecturas de red diseñados para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre AWS usted y usted. El modelo de responsabilidad compartida describe esto como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la Nube de AWS. AWS también le proporciona servicios que puede utilizar de forma segura. Auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad en el marco de los programas de conformidad de AWS. Para obtener más información sobre los programas de conformidad que se aplican a Amazon Managed Blockchain (AMB) Query, consulte AWS Services in Scope by Compliance Program.
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos aplicables.

Para proporcionar protección de datos, autenticación y control de acceso, Amazon Managed Blockchain utiliza AWS características y características del marco de código abierto que se ejecuta en Managed Blockchain.

Esta documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida al utilizar AMB Query. Los siguientes temas muestran cómo configurar AMB Query para cumplir sus objetivos de seguridad y conformidad. También puede aprender a utilizar otros AWS servicios que le ayudan a supervisar y proteger sus recursos de AMB Query.

Temas

Cifrado de datos

El cifrado de datos ayuda a evitar que usuarios no autorizados lean datos de una red blockchain y los sistemas de almacenamiento de datos asociados. Esto incluye los datos que podrían interceptarse a medida que viajan por la red, lo que se conoce como datos en tránsito.

Cifrado en tránsito

De forma predeterminada, Managed Blockchain utiliza una conexión HTTPS/TLS para cifrar todos los datos que se transmiten desde el AWS CLI cliente a los puntos finales del servicio. AWS

Gestión de identidad y acceso para Amazon Managed Blockchain (AMB) Query

AWS Identity and Access Management (IAM) es una herramienta Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a los recursos. AWS Los administradores de IAM controlan quién puede autenticarse (iniciar sesión) y quién puede autorizarse (tener permisos) para usar los recursos de AMB Query. Puede utilizar Servicio de AWS IAM sin coste adicional.

Temas

Público

La forma de usar AWS Identity and Access Management (IAM) varía según el trabajo que realice en AMB Query.

Usuario del servicio: si utiliza el servicio AMB Query para realizar su trabajo, el administrador le proporcionará las credenciales y los permisos que necesita. A medida que vaya utilizando más funciones de AMB Query para realizar su trabajo, es posible que necesite permisos adicionales.

Entender cómo se administra el acceso puede ayudarle a solicitar los permisos correctos al administrador. Si no puede acceder a una función de AMB Query, consulte. Solución de problemas de acceso e identidad de consultas de Amazon Managed Blockchain (AMB)

Administrador de servicios: si está a cargo de los recursos de AMB Query en su empresa, probablemente tenga acceso completo a AMB Query. Su trabajo consiste en determinar a qué funciones y recursos de AMB Query deben acceder los usuarios del servicio. Luego, debe enviar solicitudes a su gestionador de IAM para cambiar los permisos de los usuarios de su servicio. Revise la información de esta página para conocer los conceptos básicos de IAM. Para obtener más información sobre cómo su empresa puede utilizar IAM con AMB Query, consulte. Cómo funciona Amazon Managed Blockchain (AMB) Query con IAM

Administrador de IAM: si es administrador de IAM, puede que desee obtener más información sobre cómo redactar políticas para administrar el acceso a AMB Query. Para ver ejemplos de políticas basadas en la identidad de AMB Query que puede usar en IAM, consulte. Ejemplos de políticas basadas en identidad para Amazon Managed Blockchain (AMB) Query

Autenticación con identidades

La autenticación es la forma de iniciar sesión con sus AWS credenciales de identidad. Debe estar autenticado (con quien haya iniciado sesión AWS) como usuario de IAM o asumiendo una función de IAM. Usuario raíz de la cuenta de AWS

Puede iniciar sesión AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (Centro de identidades de IAM), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son ejemplos de identidades federadas. Al iniciar sesión como una identidad federada, su gestionador habrá configurado previamente la federación de identidades mediante roles de IAM. Cuando accedes AWS mediante la federación, estás asumiendo un rol de forma indirecta.

Según el tipo de usuario que sea, puede iniciar sesión en el portal AWS Management Console o en el de AWS acceso. Para obtener más información sobre cómo iniciar sesión AWS, consulte Cómo iniciar sesión Cuenta de AWS en su Guía del AWS Sign-In usuario.

Si accede AWS mediante programación, AWS proporciona un kit de desarrollo de software (SDK) y una interfaz de línea de comandos (CLI) para firmar criptográficamente sus solicitudes con sus credenciales. Si no utilizas AWS herramientas, debes firmar las solicitudes tú mismo. Para obtener más información sobre la firma de solicitudes, consulte AWS Signature Versión 4 para solicitudes API en la Guía del usuario de IAM.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo, le AWS recomienda que utilice la autenticación multifactor (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte Autenticación multifactor en la Guía del usuario de AWS IAM Identity Center y Autenticación multifactor AWS en IAM en la Guía del usuario de IAM.

Cuenta de AWS usuario root

Al crear una Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todos Servicios de AWS los recursos de la cuenta. Esta identidad se denomina usuario Cuenta de AWS raíz y se accede a ella iniciando sesión con la dirección de correo electrónico y la contraseña que utilizaste para crear la cuenta. Recomendamos encarecidamente que no utiliza el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las tareas que solo el usuario raíz pueda realizar. Para ver la lista completa de las tareas que requieren que inicie sesión como usuario raíz, consulta Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

Identidad federada

Como práctica recomendada, exija a los usuarios humanos, incluidos los que requieren acceso de administrador, que utilicen la federación con un proveedor de identidades para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio de usuarios de su empresa, un proveedor de identidades web AWS Directory Service, el directorio del Centro de Identidad o cualquier usuario al que acceda Servicios de AWS mediante las credenciales proporcionadas a través de una fuente de identidad. Cuando las identidades federadas acceden Cuentas de AWS, asumen funciones y las funciones proporcionan credenciales temporales.

Para una administración de acceso centralizada, le recomendamos que utiliza AWS IAM Identity Center. Puede crear usuarios y grupos en el Centro de identidades de IAM, o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus Cuentas de AWS aplicaciones. Para obtener más información, consulta ¿Qué es el Centro de identidades de IAM? en la Guía del usuario de AWS IAM Identity Center .

Usuarios y grupos de IAM

Un usuario de IAM es una identidad propia Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos emplear credenciales temporales, en lugar de crear usuarios de IAM que tengan credenciales de larga duración como

contraseñas y claves de acceso. No obstante, si tiene casos de uso específicos que requieran credenciales de larga duración con usuarios de IAM, recomendamos rotar las claves de acceso. Para más información, consulta Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración en la Guía del usuario de IAM.

Un grupo de IAM es una identidad que especifica un conjunto de usuarios de IAM. No puedes iniciar sesión como grupo. Puedes usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos para grandes conjuntos de usuarios. Por ejemplo, puede asignar un nombre a un grupo IAMAdminsy concederle permisos para administrar los recursos de IAM.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales de larga duración permanentes; no obstante, los roles proporcionan credenciales temporales. Para obtener más información, consulte Casos de uso para usuarios de IAM en la Guía del usuario de IAM.

Roles de IAM

Un rol de IAM es una identidad dentro de usted Cuenta de AWS que tiene permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una persona determinada. Para asumir temporalmente un rol de IAM en el AWS Management Console, puede cambiar de un rol de usuario a uno de IAM (consola). Puedes asumir un rol llamando a una operación de AWS API AWS CLI o usando una URL personalizada. Para más información sobre los métodos para el uso de roles, consulta Métodos para asumir un rol en la Guía del usuario de IAM.

Los roles de IAM con credenciales temporales son útiles en las siguientes situaciones:

- Acceso de usuario federado: para asignar permisos a una identidad federada, puedes crear un rol y definir sus permisos. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información acerca de roles de federación, consulte Crear un rol para un proveedor de identidad de terceros (federación) en la Guía de usuario de IAM. Si utiliza el IAM Identity Center, debe configurar un conjunto de permisos. IAM Identity Center correlaciona el conjunto de permisos con un rol en IAM para controlar a qué puedes acceder las identidades después de autenticarse. Para obtener información acerca de los conjuntos de permisos, consulta Conjuntos de permisos en la Guía del usuario de AWS IAM Identity Center .
- Permisos de usuario de IAM temporales: un usuario de IAM puedes asumir un rol de IAM para recibir temporalmente permisos distintos que le permitan realizar una tarea concreta.

- Acceso entre cuentas: puedes utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta acceda a los recursos de la cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunas Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). Para obtener información acerca de la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulta Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.
- Acceso entre servicios: algunos Servicios de AWS utilizan funciones en otros Servicios de AWS. Por ejemplo, cuando realizas una llamada en un servicio, es habitual que ese servicio ejecute aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado al servicio.
  - Sesiones de acceso directo (FAS): cuando utilizas un usuario o un rol de IAM para realizar acciones en AWS ellas, se te considera principal. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. El FAS utiliza los permisos del principal que llama Servicio de AWS y los solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulta Reenviar sesiones de acceso.
  - Rol de servicio: un rol de servicio es un rol de IAM que adopta un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.
  - Función vinculada al servicio: una función vinculada a un servicio es un tipo de función de servicio que está vinculada a un. Servicio de AWS El servicio puedes asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puedes ver, pero no editar, los permisos de los roles vinculados a servicios.
- Aplicaciones que se ejecutan en Amazon EC2: puedes usar un rol de IAM para administrar las credenciales temporales de las aplicaciones que se ejecutan en una EC2 instancia y realizan AWS CLI solicitudes a la AWS API. Esto es preferible a almacenar las claves de acceso en la EC2 instancia. Para asignar un AWS rol a una EC2 instancia y ponerlo a disposición de todas sus aplicaciones, debe crear un perfil de instancia adjunto a la instancia. Un perfil de instancia contiene

el rol y permite que los programas que se ejecutan en la EC2 instancia obtengan credenciales temporales. Para obtener más información, consulte Usar un rol de IAM para conceder permisos a las aplicaciones que se ejecutan en EC2 instancias de Amazon en la Guía del usuario de IAM.

Administración de acceso mediante políticas

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando un director (usuario, usuario raíz o sesión de rol) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre la estructura y el contenido de los documentos de política JSON, consulta Información general de políticas JSON en la Guía del usuario de IAM.

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM puedes crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puedes añadir las políticas de IAM a roles y los usuarios puedes asumirlos.

Las políticas de IAM definen permisos para una acción independientemente del método que se utiliza para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con esa política puede obtener información sobre el rol de la API AWS Management Console AWS CLI, la o la AWS API.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puedes asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Las políticas basadas en identidades puedes clasificarse además como políticas insertadas o políticas administradas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y roles de su Cuenta de AWS empresa. Las políticas administradas incluyen políticas AWS

administradas y políticas administradas por el cliente. Para obtener más información sobre cómo elegir una política administrada o una política insertada, consulte Elegir entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Los ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios puedes utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puedes realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Listas de control de acceso () ACLs

Las listas de control de acceso (ACLs) controlan qué responsables (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLs son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios compatibles. AWS WAF ACLs Para obtener más información ACLs, consulte la descripción general de la lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite tipos de políticas adicionales y menos comunes. Estos tipos de políticas puedes establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- Límites de permisos: un límite de permisos es una característica avanzada que le permite establecer los permisos máximos que una política basada en identidad puedes conceder a una entidad de IAM (usuario o rol de IAM). Puedes establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites de permisos. Las políticas basadas en recursos que especifiquen el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites

de los permisos, consulta Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.

- Políticas de control de servicios (SCPs): SCPs son políticas de JSON que especifican los permisos máximos para una organización o unidad organizativa (OU). AWS Organizations AWS Organizations es un servicio para agrupar y administrar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilitas todas las funciones de una organización, puedes aplicar políticas de control de servicios (SCPs) a una o a todas tus cuentas. El SCP limita los permisos de las entidades en las cuentas de los miembros, incluidas las de cada una Usuario raíz de la cuenta de AWS. Para obtener más información sobre Organizations SCPs, consulte las políticas de control de servicios en la Guía del AWS Organizations usuario.
- Políticas de control de recursos (RCPs): RCPs son políticas de JSON que puedes usar para establecer los permisos máximos disponibles para los recursos de tus cuentas sin actualizar las políticas de IAM asociadas a cada recurso que poseas. El RCP limita los permisos de los recursos en las cuentas de los miembros y puede afectar a los permisos efectivos de las identidades, incluidos los permisos Usuario raíz de la cuenta de AWS, independientemente de si pertenecen a su organización. Para obtener más información sobre Organizations e RCPs incluir una lista de Servicios de AWS ese apoyo RCPs, consulte Políticas de control de recursos (RCPs) en la Guía del AWS Organizations usuario.
- Políticas de sesión: las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también puedes proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para más información, consulta Políticas de sesión en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo se AWS determina si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la lógica de evaluación de políticas en la Guía del usuario de IAM.

Cómo funciona Amazon Managed Blockchain (AMB) Query con IAM

Antes de usar IAM para administrar el acceso a AMB Query, infórmese sobre las funciones de IAM disponibles para su uso con AMB Query.

Funciones de IAM que puede utilizar con Amazon Managed Blockchain (AMB) Query

Característica de IAM	Soporte para consultas AMB
Políticas basadas en identidades	Sí
Políticas basadas en recursos	No
Acciones de políticas	Sí
Recursos de políticas	No
Claves de condición de política	No
ACLs	No
ABAC (etiquetas en políticas)	No
Credenciales temporales	Sí
Permisos de entidades principales	Sí
Roles de servicio	No
Roles vinculados al servicio	No

Para obtener una visión general de cómo funcionan AMB Query y otros AWS servicios con la mayoría de las funciones de IAM, consulte los AWS servicios que funcionan con IAM en la Guía del usuario de IAM.

Políticas basadas en la identidad para AMB Query

Compatibilidad con las políticas basadas en identidad: sí

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte Creación de políticas de IAM en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está asociada. Para obtener más información sobre los elementos que puede utilizar en una política de JSON, consulte Referencia de los elementos de las políticas de JSON de IAM en la Guía del usuario de IAM.

Ejemplos de políticas basadas en la identidad para AMB Query

Para ver ejemplos de políticas basadas en la identidad de AMB Query, consulte. Ejemplos de políticas basadas en identidad para Amazon Managed Blockchain (AMB) Query

Políticas basadas en recursos dentro de AMB Query

Admite políticas basadas en recursos: no

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Los ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios puedes utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puedes realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Para habilitar el acceso entre cuentas, puede especificar toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de recursos. Añadir a una política en función de recursos una entidad principal entre cuentas es solo una parte del establecimiento de una relación de confianza. Cuando el principal y el recurso son diferentes Cuentas de AWS, el administrador de IAM de la cuenta de confianza también debe conceder a la entidad principal (usuario o rol) permiso para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política basada en recursos concede acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para obtener más información, consulte Cross account resource access in IAM en la Guía del usuario de IAM.

Acciones políticas para AMB Query

Compatibilidad con las acciones de políticas: sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puedes realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puedes utilizar para conceder o denegar el acceso en una política. Las acciones políticas suelen tener el mismo nombre que la operación de AWS API asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de las acciones de consulta de AMB, consulte Acciones definidas por la consulta de Amazon Managed Blockchain (AMB) en la Referencia de autorización de servicios.

Las acciones políticas de AMB Query utilizan el siguiente prefijo antes de la acción:

```
managedblockchain-query:
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

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

Para ver ejemplos de políticas basadas en la identidad de AMB Query, consulte. Ejemplos de políticas basadas en identidad para Amazon Managed Blockchain (AMB) Query

Recursos de políticas para AMB Query

Compatibilidad con recursos de políticas: no

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puedes realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica

recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). Puedes hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utiliza un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

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

Para ver una lista de los tipos de recursos de AMB Query y sus correspondientes ARNs, consulte Resources Defined by Amazon Managed Blockchain (AMB) Query en la Referencia de autorización de servicios. Para saber con qué acciones puede especificar el ARN de cada recurso, consulte Actions Defined by Amazon Managed Blockchain (AMB) Query.

Para ver ejemplos de políticas basadas en la identidad de AMB Query, consulte. Ejemplos de políticas basadas en identidad para Amazon Managed Blockchain (AMB) Query

Claves de condición de política para AMB Query

Compatibilidad con claves de condición de políticas específicas del servicio: no

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puedes realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puedes crear expresiones condicionales que utilizan operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación `AND` lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición mediante una `OR` operación lógica. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puedes utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puedes conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado

con su nombre de usuario de IAM. Para más información, consulta Elementos de la política de IAM: variables y etiquetas en la Guía del usuario de IAM.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición AWS globales, consulte las claves de contexto de condición AWS globales en la Guía del usuario de IAM.

Para ver una lista de claves de condición de AMB Query, consulte Claves de condición para la consulta Amazon Managed Blockchain (AMB) en la Referencia de autorización de servicio. Para saber con qué acciones y recursos puede utilizar una clave de condición, consulte Actions Defined by Amazon Managed Blockchain (AMB) Query.

Para ver ejemplos de políticas basadas en la identidad de AMB Query, consulte. Ejemplos de políticas basadas en identidad para Amazon Managed Blockchain (AMB) Query

ACLs en AMB Query

Soporta ACLs: No

Las listas de control de acceso (ACLs) controlan qué directores (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLs son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

ABAC con AMB Query

Compatibilidad con ABAC (etiquetas en las políticas): no

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a las entidades de IAM (usuarios o roles) y a muchos AWS recursos. El etiquetado de entidades y recursos es el primer paso de ABAC. A continuación, designa las políticas de ABAC para permitir operaciones cuando la etiqueta de la entidad principal coincida con la etiqueta del recurso al que se intenta acceder.

ABAC es útil en entornos que crecen con rapidez y ayuda en situaciones en las que la administración de las políticas resulta engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información sobre ABAC, consulte Definición de permisos con la autorización de ABAC en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulta Uso del control de acceso basado en atributos (ABAC) en la Guía del usuario de IAM.

Uso de credenciales temporales con AMB Query

Compatibilidad con credenciales temporales: sí

Algunas Servicios de AWS no funcionan cuando inicias sesión con credenciales temporales. Para obtener información adicional, incluidas las que Servicios de AWS funcionan con credenciales temporales, consulta Cómo Servicios de AWS funcionan con IAM en la Guía del usuario de IAM.

Utiliza credenciales temporales si inicia sesión en ellas AWS Management Console mediante cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accedes AWS mediante el enlace de inicio de sesión único (SSO) de tu empresa, ese proceso crea automáticamente credenciales temporales. También crea credenciales temporales de forma automática cuando inicia sesión en la consola como usuario y luego cambia de rol. Para obtener más información sobre el cambio de roles, consulte Cambio de un usuario a un rol de IAM (consola) en la Guía del usuario de IAM.

Puedes crear credenciales temporales manualmente mediante la AWS CLI API o. AWS A continuación, puede utilizar esas credenciales temporales para acceder AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para obtener más información, consulte Credenciales de seguridad temporales en IAM.

Permisos principales entre servicios para AMB Query

Admite sesiones de acceso directo (FAS): sí

Cuando utilizas un usuario o un rol de IAM para realizar acciones en él AWS, se te considera principal. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos del principal que llama y los que solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Servicio de AWS Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para

realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulta Reenviar sesiones de acceso.

Funciones de servicio para AMB Query

Compatible con roles de servicio: No

Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Creación de un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.

> **Warning**
>
> Cambiar los permisos de un rol de servicio podría interrumpir la funcionalidad de AMB Query. Edite las funciones de servicio solo cuando AMB Query proporcione instrucciones para hacerlo.

Funciones vinculadas al servicio para AMB Query

Compatibilidad con roles vinculados al servicio: no

Un rol vinculado a un servicio es un tipo de rol de servicio que está vinculado a un. Servicio de AWS El servicio puedes asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puedes ver, pero no editar, los permisos de los roles vinculados a servicios.

Para más información sobre cómo crear o administrar roles vinculados a servicios, consulta Servicios de AWS que funcionan con IAM. Busque un servicio en la tabla que incluya `Yes` en la columna Rol vinculado a un servicio. Seleccione el vínculo Sí para ver la documentación acerca del rol vinculado a servicios para ese servicio.

Ejemplos de políticas basadas en identidad para Amazon Managed Blockchain (AMB) Query

De forma predeterminada, los usuarios y los roles no tienen permiso para crear o modificar los recursos de AMB Query. Tampoco pueden realizar tareas mediante la AWS Management Console, AWS Command Line Interface (AWS CLI) o la AWS API. Un administrador de IAM puedes crear

políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puedes añadir las políticas de IAM a roles y los usuarios puedes asumirlos.

Para obtener información acerca de cómo crear una política basada en identidades de IAM mediante el uso de estos documentos de políticas JSON de ejemplo, consulte Creación de políticas de IAM (consola) en la Guía del usuario de IAM.

Para obtener más información sobre las acciones y los tipos de recursos definidos por AMB Query, incluido el formato de cada uno de los tipos de recursos, consulte Actions, Resources and Condition Keys for Amazon Managed Blockchain (AMB) Query en la Referencia de autorización de servicios. ARNs

Temas

- Prácticas recomendadas sobre las políticas
- Cómo permitir a los usuarios consultar sus propios permisos
- Acceso a acciones específicas de la API Query de Amazon Managed Blockchain (AMB)

Prácticas recomendadas sobre las políticas

Las políticas basadas en la identidad determinan si alguien puede crear, eliminar o acceder a los recursos de AMB Query de su cuenta. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están disponibles en su. Cuenta de AWS Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulta las políticas administradas por AWS o las políticas administradas por AWS para funciones de tarea en la Guía de usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se puedes llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulta Políticas y permisos en IAM en la Guía del usuario de IAM.

- Utiliza condiciones en las políticas de IAM para restringir aún más el acceso: puedes agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puedes escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo AWS CloudFormation. Para obtener más información, consulta Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.
- Utiliza el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Validación de políticas con el Analizador de acceso de IAM en la Guía del usuario de IAM.
- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para exigir la MFA cuando se invoquen las operaciones de la API, añada condiciones de MFA a sus políticas. Para más información, consulte Acceso seguro a la API con MFA en la Guía del usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte Prácticas recomendadas de seguridad en IAM en la Guía del usuario de IAM.

Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas gestionadas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API o. 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": "*"
        }
    ]
}
```

Acceso a acciones específicas de la API Query de Amazon Managed Blockchain (AMB)

> **Note**
>
> Para acceder a AMB Query y realizar llamadas a la API, necesitará credenciales de usuario (`AWS_ACCESS_KEY_IDyAWS_SECRET_ACCESS_KEY`) disponer de los permisos de IAM adecuados para AMB Query.

Example Política de IAM para acceder a todas las consultas de Amazon Managed Blockchain (AMB) APIs

En este ejemplo, se concede a un usuario de IAM el Cuenta de AWS acceso a todas las consultas de AMB. APIs

```
{
```

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

Example Política de IAM para acceder a Amazon Managed Blockchain (AMB) Query y **`ListTransactionsGetTransaction`** APIs

En este ejemplo, se concede a un usuario de IAM el Cuenta de AWS acceso a la consulta de AMB y `ListTransaction` `GetTransaction` APIs

> **Note**
>
> Puede reemplazar o agregar el del APIs ejemplo por otro APIs para dar acceso a otro o más. APIs Para obtener una lista de AMB Query APIs, consulte la Guía de referencia de la API de consultas de Amazon Managed Blockchain (AMB).

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

Solución de problemas de acceso e identidad de consultas de Amazon Managed Blockchain (AMB)

Utilice la siguiente información como ayuda para diagnosticar y solucionar los problemas habituales que pueden surgir al trabajar con AMB Query e IAM.

Temas

No estoy autorizado a realizar ninguna acción en AMB Query

Si recibe un error que indica que no tiene autorización para realizar una acción, las políticas se deben actualizar para permitirle realizar la acción.

En el siguiente ejemplo, el error se produce cuando el usuario de IAM `mateojackson` intenta utilizar la consola para consultar los detalles acerca de un recurso ficticio *`my-example-widget`*, pero no tiene los permisos ficticios `managedblockchain-query::`*`GetWidget`*.

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

En este caso, la política del usuario `mateojackson` debe actualizarse para permitir el acceso al recurso *`my-example-widget`* mediante la acción `managedblockchain-query::`*`GetWidget`*.

Si necesita ayuda, póngase en contacto con su AWS administrador. El gestionador es la persona que le proporcionó las credenciales de inicio de sesión.

Métricas de uso de la API de consulta de Amazon Managed Blockchain (AMB) en Amazon CloudWatch

Métricas de uso de API en Amazon CloudWatch

Las métricas de uso de la API publicadas CloudWatch corresponden a las cuotas del servicio de consultas de Amazon Managed Blockchain (AMB). Puede configurar alarmas para que le avisen cuando su uso se acerque a una cuota de servicio. Para obtener más información sobre CloudWatch la integración con las cuotas de servicio, consulte las métricas de uso de AWS en la Guía del CloudWatch usuario de Amazon.

AMB Query publica las siguientes métricas de API en el espacio de `AWS/Usage` nombres, con el nombre del `Amazon Managed Blockchain Query` servicio.

Métrica	Descripción
`CallCount`	El número total de llamadas realizadas a una API en AMB Query. SUM representa el número total de llamadas a la API durante el período especificado.

Amazon Managed Blockchain (AMB) Query publica las métricas de uso en el espacio de `AWS/Usage` nombres con las siguientes dimensiones.

Dimensión	Descripción
Servicio	El nombre del AWS servicio que contiene el recurso. `Amazon Managed Blockchain Query`siempre será el valor de esta dimensión.
Tipo	El tipo de entidad sobre la que se informa. `API`siempre será el valor de esta dimensión.
Recurso	El tipo de recursos sobre los que se informa. El nombre de la operación de la API de

Dimensión	Descripción
	consulta de AMB utilizada será el valor de esta dimensión.
Clase	La clase del recurso del que se informa. Nonesiempre será el valor de esta dimensión.

Historial de documentos de la Guía del usuario de AMB Query

En la siguiente tabla se describen las versiones de la documentación de AMB Query.

Cambio	Descripción	Fecha
AMB Query admite identificadores de transacciones y hashes de transacciones de Bitcoin	En las redes de Bitcoin, las operaciones de la API de AMB Query admiten tanto el identificador de transacción (`transactionId`) como el hash de transacción (). `transactionHash`	21 de marzo de 2024
Support para métricas de uso de API en Amazon CloudWatch	AMB Query agregó soporte para las métricas de uso de la API en CloudWatch. Estas métricas de uso corresponden a las cuotas del servicio de AMB Query.	8 de febrero de 2024
Support para transacciones que no han alcanzado la finalidad	AMB Query agregó soporte para las transacciones que no han alcanzado la finalidad. También elimina la compatibilidad con la `status` propiedad de la respuesta a la `GetTransaction` operación. En su lugar, utilizará las `executionStatus` propiedades `confirmationStatus` y para determinar el estado de la transacción.	1 de febrero de 2024

Obsolación de la `status` propiedad en el tipo de datos de transacción	Amazon Managed Blockchain (AMB) Query ha dejado de usar la `status` propiedad en el tipo de datos Transaction. Debe usar los `executionStatus` campos `confirmationStatus` y para determinar si `status` la transacción es `FINAL` o. `FAILED`	20 de diciembre de 2023
Support para Sepolia Testnet	Amazon Managed Blockchain (AMB) Query ahora admite consultas en la red de pruebas Sepolia de Ethereum.	19 de octubre de 2023
Support for assets contracts	Puede utilizar la operación de la `ListAssetContracts` API para enumerar las unidades implementadas por una dirección determinada. Además, si tienes la dirección del contrato, puedes usar la operación de la `GetAssetContract` API para recuperar los detalles del contrato.	16 de octubre de 2023
Support para Bitcoin Testnet	Amazon Managed Blockchain (AMB) Query ahora admite consultas en la red de pruebas de Bitcoin.	16 de octubre de 2023
Versión inicial	Versión inicial del servicio AMB Query.	27 de julio de 2023

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.