aws

Entwicklerhandbuch

Amazon Managed Blockchain Query

Amazon Managed Blockchain Query: Entwicklerhandbuch

Table of Contents

Was ist Amazon Managed Blockchain (AMB) Query?

Amazon Managed Blockchain (AMB) ist ein vollständig verwalteter Service, mit dem Sie robuste Web3-Anwendungen auf öffentlichen und privaten Blockchains erstellen können. Verwenden Sie AMB Access für den sofortigen und serverlosen Zugriff auf mehrere Blockchains. Erstellen Sie Ihre Web3-fähigen Anwendungen, ohne eine spezielle Blockchain-Infrastruktur bereitstellen und diese mit dem Blockchain-Netzwerk verbinden zu müssen. Mit AMB Query können Sie entwicklerfreundliche API-Operationen verwenden, um auf Echtzeit- und historische Daten aus mehreren Blockchains zuzugreifen. Die standardisierten Blockchain-Daten können in AWS-Services integriert werden, ohne dass eine spezielle Blockchain-Infrastruktur oder ETL (Extrahieren, Transformieren und Laden) erforderlich ist. Alle AMB-Funktionen lassen sich sicher skalieren und eignen sich sowohl für Anwendungen auf institutioneller Ebene als auch für Standardanwender.

Amazon Managed Blockchain (AMB) Query bietet serverlosen Zugriff auf standardisierte Multi-Blockketten-Datensätze mit entwicklerfreundlichen API-Operationen. Sie können AMB Query verwenden, um schnell Anwendungen bereitzustellen, die Daten aus einer oder mehreren öffentlichen Blockchains benötigen, ohne dass Sie den Aufwand für das Parsen von Blockchain-Daten, die Rückverfolgung von Verträgen und die Wartung einer speziellen Indexierungsinfrastruktur aufwenden müssen. Egal, ob Sie historische Token-Salden für fungible Token oder nicht fungible Token (NFTs) analysieren, den Transaktionsverlauf für eine bestimmte Wallet-Adresse anzeigen oder Datenanalysen zur Verteilung nativer Kryptowährungen wie Ether durchführen, AMB Query bietet Ihnen Zugriff auf die Blockchain-Daten.

Verwenden Sie AMB Query zum ersten Mal?

Wenn Sie AMB Query zum ersten Mal verwenden, empfehlen wir Ihnen, zunächst die folgenden Abschnitte zu lesen:

- Schlüsselkonzepte: Amazon Managed Blockchain (AMB) Query
- Amazon Managed Blockchain (AMB) -Abfrage einrichten
- Erste Schritte mit Amazon Managed Blockchain (AMB) Query
- Anwendungsfälle mit Amazon Managed Blockchain (AMB) Query

Schlüsselkonzepte: Amazon Managed Blockchain (AMB) Query

> **Note**
>
> In diesem Leitfaden wird davon ausgegangen, dass Sie mit den wichtigsten Blockchain-Konzepten vertraut sind. Zu diesen Konzepten gehören Dezentralisierung, Tokens, Verträge, Transaktionen proof-of-work, Wallets, öffentliche und private Schlüssel, Staking, Mining, Halbierungen und andere.

Amazon Managed Blockchain (AMB) Query bietet Ihnen bequemen Zugriff auf Netzwerkdaten mit mehreren Blockchains, sodass Sie Kontextdaten im Zusammenhang mit Blockchain-Aktivitäten leichter extrahieren können. Sie können AMB Query verwenden, um Daten aus öffentlichen Blockchain-Netzwerken wie Bitcoin Mainnet und Ethereum Mainnet zu lesen. Sie können auch Informationen wie aktuelle und historische Adresssalden abrufen oder eine Liste von Blockchain-Transaktionen für einen bestimmten Zeitraum abrufen. Darüber hinaus können Sie Details zu einer bestimmten Transaktion abrufen, z. B. Transaktionsereignisse, die Sie weiter analysieren oder in der Geschäftslogik für Ihre Anwendungen verwenden können.

Überlegungen und Einschränkungen bei der Verwendung von Amazon Managed Blockchain (AMB) Query

Beachten Sie bei der Verwendung von AMB Query Folgendes:

- Verfügbare Regionen

  AMB Query wird in der `us-east-1` Region USA Ost (Nord-Virginia) unterstützt.

- Service-Endpunkte

  Auf AMB Query kann über den folgenden Endpunkt zugegriffen werden:

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

- Unterstützte Blockchain-Netzwerke

AMB Query unterstützt die folgenden öffentlichen Blockchain-Netzwerke:

- Bitcoin Mainnet — Das öffentliche Bitcoin-Blockchainnetzwerk, das durch proof-of-work Konsens gesichert ist und über das die Bitcoin (BTC) -Kryptowährung ausgegeben und abgewickelt wird. Transaktionen im Mainnet haben einen tatsächlichen Wert (das heißt, sie verursachen echte Kosten) und werden in der öffentlichen Blockchain aufgezeichnet.
- Bitcoin Testnet — Das Testnetz für das Bitcoin Mainnet. Bitcoin (BTC) in diesem Netzwerk ist getrennt und unterscheidet sich von Mainnet BTC und hat normalerweise keinen Wert.
- Ethereum Mainnet — Das proof-of-stake Hauptnetzwerk für die öffentliche Ethereum-Blockchain. Transaktionen im Mainnet haben einen tatsächlichen Wert (das heißt, sie verursachen echte Kosten) und werden im Distributed-Ledger aufgezeichnet.
- Sepolia Testnet — Das Testnetz für das Ethereum-Mainnet. Ether (ETH) in diesem Netzwerk ist getrennt und unterscheidet sich von Mainnet ETH und hat normalerweise keinen Wert.

- Unterstützte Blockchain-Token und Verträge

  AMB Query unterstützt die folgenden systemeigenen und standardmäßigen Ethereum-Vertragstoken.

  - Native Tokens für öffentliche Blockchains
    - Bitcoin (BTC) — Dies ist das native Token von Bitcoin-bezogenen Blockchains.
    - Ether (ETH) — Dies ist das native Token von Ethereum-bezogenen Blockchains.

  - Vertragsstandards von Ethereum
    - ERC-20-Token-Standard — Der ERC-20 ist ein Standard für fungible Token. Er hat eine Eigenschaft, die dafür sorgt, dass jedes ERC-20-Token (in Typ und Wert) genau dem anderen geprägten ERC-20-Token entspricht, was bedeutet, dass ein Token allen anderen Token entspricht und immer sein wird. Weitere Informationen finden Sie im ERC-20-Token-Standard auf Ethereum.org.
    - ERC-721-Standard für nicht fungible Token — Der ERC-721 ist ein Standard für nicht fungible Token (). NFTs Diese Art von Token ist einzigartig und kann einen anderen Wert haben als ein anderes Token aus demselben Vertrag, möglicherweise aufgrund seines Alters, seiner Seltenheit oder anderer Eigenschaften. Weitere Informationen finden Sie im ERC-721 Token Standard auf Ethereum.org.

      ERC-1155 Multi-Token-Standard — Der ERC-1155 ist ein Standard, der eine Vertragsschnittstelle schafft, die eine beliebige Anzahl von fungiblen und nicht fungiblen

Tokentypen darstellen und steuern kann. Auf diese Weise kann das ERC-1155-Token genauso funktionieren wie die ERC-20- und ERC-721-Token und sogar als beide gleichzeitig funktionieren. Das ERC-1155-Token verbessert die Funktionalität der Standards ERC-20 und ERC-721, macht sie effizienter und korrigiert gleichzeitig offensichtliche Implementierungsfehler. Weitere Informationen finden Sie im ERC-1155-Token-Standard auf Ethereum.org.

- Endgültigkeit

  In Blockchains bedeutet Finalität, dass es unwahrscheinlich ist, dass gültige Transaktionen rückgängig gemacht werden. Für das Bitcoin-Mainnet betrachtet AMB Query eine Transaktion nach 6 Blöcken als endgültig. Für das Bitcoin-Testnet wird davon ausgegangen, dass eine Transaktion entweder nach 6 Blöcken oder nach 60 Minuten abgeschlossen ist, je nachdem, was zuerst eintritt. Bei unterstützten Ethereum-Netzwerken betrachtet AMB Query eine Transaktion nach 64 Blöcken als abgeschlossen.

  Die Token-Balance- und Vertrags-API-Operationen von AMB Query geben nur Daten zurück, die ihre Endgültigkeit erreicht haben. Die Transaktions- und Transaktionsereignis-API-Operationen von AMB Query können jedoch Daten für Transaktionen zurückgeben, die im Blockchain-Netzwerk bestätigt wurden, auch wenn sie noch nicht abgeschlossen sind.

- NULL-Adresse wird nicht unterstützt

  AMB Query unterstützt die Adresse `NULL` (`0x0000000000000000000000000000000000000000`) nicht.

- Signatur, Version 4, Signierung von API-Aufrufen

  Wenn Sie die AMB-Abfrage aufrufen APIs, können Sie dies über eine HTTPS-Verbindung tun, die mithilfe des Signaturprozesses von Signature Version 4 authentifiziert wurde. Das bedeutet, dass nur autorisierte IAM-Prinzipale im AWS Konto AMB Query-API-Aufrufe tätigen können. Zu diesem Zweck müssen beim AWS Aufruf Anmeldeinformationen (eine Zugriffsschlüssel-ID und ein geheimer Zugriffsschlüssel) bereitgestellt werden.

  > **Important**
  >
  > Betten Sie keine Kundenanmeldedaten in benutzerseitige Anwendungen ein.

- AMB Query unterstützt Bitcoin-Transaktions-Identifikatoren und Transaktions-Hashes

  Für Bitcoin-Netzwerke unterstützen AMB Query API-Operationen sowohl die Transaktions-ID (`transactionId`) als auch den Transaktions-Hash (). `transactionHash` Das `transactionId` ist ein Double-SHA-Hash der Transaktion ohne Zeugendaten. Das `transactionHash` ist ein Double-SHA-Hash der Transaktion, einschließlich Zeugendaten (auch bekannt als Zeugentransaktions-ID).

  Beim Aufrufen der `ListTransactionEvents`API-Operationen `GetTransaction`oder für Bitcoin-Netzwerke können Sie entweder die `transactionId` oder die angeben. `transactionHash` Außerdem enthalten alle AMB-Query-Operationen in Bitcoin-Netzwerken, die entweder a `transactionId` oder a `transactionHash` zurückgeben, beide Werte als Teil der Antwort.

Amazon Managed Blockchain (AMB) -Abfrage einrichten

Bevor Sie Amazon Managed Blockchain (AMB) Query zum ersten Mal verwenden, folgen Sie den Schritten in diesem Abschnitt, um ein AWS Konto zu erstellen. Im folgenden Abschnitt werden die ersten Schritte mit AMB Query beschrieben.

Voraussetzungen und Überlegungen

Bevor Sie Amazon Web Services zum ersten Mal nutzen können, müssen Sie über ein AWS Konto verfügen.

Melden Sie sich an für AWS

Wenn Sie sich für Amazon Web Services (AWS) registrieren, wird Ihr AWS Konto automatisch für alle registriert AWS-Services, einschließlich Amazon Managed Blockchain (AMB) Query. Berechnet werden Ihnen aber nur die Services, die Sie nutzen.

Wenn Sie AWS-Konto bereits eine haben, fahren Sie mit dem nächsten Schritt fort. Wenn Sie kein AWS-Konto haben, führen Sie die folgenden Schritte zum Erstellen eines Kontos aus.

Um ein AWS Konto zu erstellen

1. Öffnen Sie https://portal.aws.amazon.com/billing/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

   Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Benutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

Erstellen Sie einen IAM-Benutzer mit den entsprechenden Berechtigungen

Um AMB Query zu erstellen und damit zu arbeiten, müssen Sie einen AWS Identity and Access Management (IAM-) Prinzipal (Benutzer oder Gruppe) mit Berechtigungen erstellen, die die erforderlichen Managed Blockchain-Aktionen ermöglichen.

Nur IAM-Prinzipale können AMB Query API-Anfragen stellen. Wenn Sie die AMB-Abfrage aufrufen, können Sie dies über eine HTTPS-Verbindung tun APIs, die mithilfe des Signature Version 4-Signaturprozesses authentifiziert wurde. Das bedeutet, dass nur autorisierte IAM-Prinzipale im AWS Konto AMB Query-API-Aufrufe tätigen können. Zu diesem Zweck müssen beim AWS Aufruf Anmeldeinformationen (eine Zugriffsschlüssel-ID und ein geheimer Zugriffsschlüssel) bereitgestellt werden.

Informationen zum Erstellen eines IAM-Benutzers finden Sie unter Einen IAM-Benutzer in Ihrem AWS Konto erstellen. Weitere Informationen dazu, wie Sie einem Benutzer eine Berechtigungsrichtlinie zuordnen, finden Sie unter Berechtigungen für einen IAM-Benutzer ändern. Ein Beispiel für eine Berechtigungsrichtlinie, mit der Sie einem Benutzer die Erlaubnis erteilen können, mit AMB Query zu arbeiten, finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Amazon Managed Blockchain (AMB) Query

Installieren und konfigurieren Sie den AWS Command Line Interface

Falls Sie dies noch nicht getan haben, installieren Sie die neueste AWS Befehlszeilenschnittstelle (CLI), um mit AWS Ressourcen von einem Terminal aus zu arbeiten. Weitere Informationen finden Sie unter Installieren oder Aktualisierung auf die neueste Version von AWS CLI.

> **Note**
>
> Für CLI-Zugriff benötigen Sie eine Zugriffsschlüssel-ID und einen geheimen Zugriffsschlüssel. Verwenden Sie möglichst temporäre Anmeldeinformationen anstelle langfristiger Zugriffsschlüssel. Temporäre Anmeldeinformationen bestehen aus einer Zugriffsschlüssel-ID, einem geheimen Zugriffsschlüssel und einem Sicherheits-Token, das angibt, wann die Anmeldeinformationen ablaufen. Weitere Informationen finden Sie unter Verwenden temporärer Anmeldeinformationen mit AWS Ressourcen im IAM-Benutzerhandbuch.

Verwenden Sie die AWS-Managementkonsole , um Blockchains mithilfe von Amazon Managed Blockchain (AMB) Query abzufragen

Mit dem können Sie auf Amazon Managed Blockchain (AMB) Query zugreifen und Abfragen zu unterstützten Blockchain-Netzwerken stellen. AWS-Managementkonsole Die folgenden Schritte zeigen, wie das geht:

1. Öffnen Sie die Amazon Managed Blockchain Blockchain-Konsole unter https://console.aws.amazon.com/managedblockchain/.
2. Wählen Sie im Abfragebereich den Abfrage-Editor aus.
3. Wählen Sie aus einem der unterstützten Blockchain-Netzwerke.
4. Wählen Sie den Abfragetyp aus, den Sie ausführen möchten.
5. Geben Sie die relevanten Parameter für den ausgewählten Abfragetyp ein und klicken Sie auf Abfrage ausführen.

AMB Query führt Ihre Abfrage aus und Sie sehen die Ergebnisse im Fenster mit den Abfrageergebnissen.

Erste Schritte mit Amazon Managed Blockchain (AMB) Query

In den step-by-step Tutorials in diesem Abschnitt erfahren Sie, wie Sie Aufgaben mithilfe von Amazon Managed Blockchain (AMB) Query ausführen. Für diese Verfahren sind einige Voraussetzungen erforderlich. Wenn Sie mit AMB Query noch nicht vertraut sind, können Sie den Abschnitt Einrichtung dieses Handbuchs lesen. Weitere Informationen finden Sie unter Amazon Managed Blockchain (AMB) -Abfrage einrichten.

> **Note**
>
> Einige Variablen in diesen Beispielen wurden bewusst verschleiert. Ersetzen Sie sie durch eigene gültige, bevor Sie diese Beispiele ausführen.

Themen

Erstellen Sie eine IAM-Richtlinie für den Zugriff auf AMB Query API-Operationen

Um AMB Query API-Anfragen zu stellen, müssen Sie die Benutzeranmeldedaten (AWS_ACCESS_KEY_ID und AWS_SECRET _ACCESS_KEY) verwenden, die über die entsprechenden IAM-Berechtigungen für Amazon Managed Blockchain (AMB) Query verfügen. Führen Sie in einem Terminal, auf dem das AWS CLI installiert ist, den folgenden Befehl aus, um eine IAM-Richtlinie für den Zugriff auf AMB Query API-Operationen zu erstellen:

```
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
```

Nachdem Sie die Richtlinie erstellt haben, fügen Sie diese Richtlinie der Rolle eines IAM-Benutzers hinzu, damit sie wirksam wird. Navigieren Sie in der AWS-Managementkonsole zum IAM-Dienst und fügen Sie die Richtlinie der Rolle `AmazonManagedBlockchainQueryAccess` hinzu, die dem IAM-Benutzer zugewiesen ist, der den Dienst verwenden wird. Weitere Informationen finden Sie unter Rolle erstellen und sie einem IAM-Benutzer zuweisen.

> **Note**
>
> AWS empfiehlt, dass Sie Zugriff auf bestimmte API-Operationen gewähren, anstatt den Platzhalter zu verwenden. * Weitere Informationen finden Sie unter Zugreifen auf bestimmte Amazon Managed Blockchain (AMB) Query API-Aktionen.

Stellen Sie mithilfe von Go API-Anfragen für Amazon Managed Blockchain (AMB) -Abfragen

Mit Amazon Managed Blockchain (AMB) Query können Sie Anwendungen erstellen, die auf sofortigen Zugriff auf Blockchain-Daten angewiesen sind, sobald sie in der Blockchain bestätigt wurden, auch wenn sie noch nicht endgültig sind. AMB Query ermöglicht verschiedene Anwendungsfälle, z. B. das Auffüllen des Transaktionsverlaufs einer Wallet, die Bereitstellung von Kontextinformationen zu einer Transaktion auf der Grundlage ihres Transaktions-Hash oder das Abrufen des Saldos von systemeigenen Token sowie von ERC-721-, ERC-1155- und ERC-20-Token.

Die folgenden Beispiele wurden in der Sprache Go erstellt und verwenden die AMB Query API-Operationen. Weitere Informationen zu Go finden Sie in der Go-Dokumentation. Weitere Informationen zur AMB Query API finden Sie in der Referenzdokumentation zur Amazon Managed Blockchain (AMB) Query API.

In den folgenden Beispielen werden die API-Aktionen `ListTransactions` und die `GetTransaction` API-Aktionen verwendet, um zunächst eine Liste aller Transaktionen für eine bestimmte externe Adresse (EOA) im Ethereum-Mainnet abzurufen. Im nächsten Beispiel werden dann die Transaktionsdetails für eine einzelne Transaktion aus der Liste abgerufen.

Example— Führen Sie die ListTransactions API-Aktion mit Go durch

Kopieren Sie den folgenden Code in eine Datei mit dem Namen `listTransactions.go` im ListTransactionsVerzeichnis.

```
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)
    }
}
```

Nachdem Sie die Datei gespeichert haben, führen Sie den Code mit dem folgenden Befehl im ListTransactionsVerzeichnis aus:`go run listTransactions.go`.

Die folgende Ausgabe ähnelt der folgenden:

```
{
  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— Führen Sie die GetTransaction API-Aktion mithilfe von Go durch

In diesem Beispiel wird ein Transaktions-Hash aus der vorherigen Ausgabe verwendet. Kopieren Sie den folgenden Code in eine Datei mit dem Namen `GetTransaction.go` im GetTransactionVerzeichnis.

```
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)
    }
}
```

Nachdem Sie die Datei gespeichert haben, führen Sie den Code mit dem folgenden Befehl im GetTransactionVerzeichnis aus:`go run GetTransaction.go`.

Die folgende Ausgabe ähnelt der folgenden:

```
{
  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
  }
}
```

Die `GetTokenBalance` API bietet Ihnen die Möglichkeit, den Saldo der systemeigenen Tokens (ETH und BTC) abzurufen. Dieser kann verwendet werden, um den aktuellen Saldo eines externen Kontos (EOA) zu einem bestimmten Zeitpunkt abzurufen.

Example— Verwenden Sie die GetTokenBalance API-Aktion, um den Saldo eines nativen Tokens in Go abzurufen

Im folgenden Beispiel verwenden Sie die `GetTokenBalance` API, um einen Adress-Ether-Saldo (ETH) im Ethereum-Mainnet abzurufen. Kopieren Sie den folgenden Code in eine Datei mit dem Namen `GetTokenBalanceEth.go` im GetTokenBalanceVerzeichnis.

```
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)
    }
}
```

Nachdem Sie die Datei gespeichert haben, führen Sie den Code mit dem folgenden Befehl im GetTokenBalanceVerzeichnis aus:`go run GetTokenBalanceEth.go`.

Die folgende Ausgabe ähnelt der folgenden:

```
{
  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"
  }
```

```
}
```

Stellen Sie mithilfe von Node.js Anfragen an die Amazon Managed Blockchain (AMB) Query API

Für die Ausführung dieser Node-Beispiele gelten die folgenden Voraussetzungen:

1. Sie müssen den Node Version Manager (nvm) und Node.js auf Ihrem Computer installiert haben. Eine Installationsanleitung für Ihr Betriebssystem finden Sie hier.
2. Verwenden Sie den `node --version` Befehl und bestätigen Sie, dass Sie Node Version 14 oder höher verwenden. Bei Bedarf können Sie den `nvm install 14` Befehl verwenden, gefolgt vom `nvm use 14` Befehl, um Version 14 zu installieren.
3. Die Umgebungsvariablen `AWS_ACCESS_KEY_ID` und `AWS_SECRET_ACCESS_KEY` müssen die Anmeldeinformationen enthalten, die dem Konto zugeordnet sind.

   Exportieren Sie diese Variablen mithilfe der folgenden Befehle als Zeichenfolgen auf Ihrem Client. Ersetzen Sie die im Folgenden hervorgehobenen Werte durch entsprechende Werte aus dem IAM-Benutzerkonto.

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

> **Note**
>
> - Nachdem Sie alle Voraussetzungen erfüllt haben, können Sie signierte Anfragen über HTTPS einreichen, um auf Amazon Managed Blockchain (AMB) Query API-Operationen zuzugreifen und Anfragen mithilfe des nativen https-Moduls in Node.js zu stellen, oder Sie können eine Drittanbieterbibliothek wie AXIOS verwenden und Daten aus AMB Query abrufen.
> - In diesen Beispielen wird ein HTTP-Client eines Drittanbieters für Node.js verwendet, Sie können jedoch auch das AWS JavaScript SDK verwenden, um Anfragen an AMB Query zu stellen.
> - Das folgende Beispiel zeigt Ihnen, wie Sie mithilfe von Axios und den AWS SDK-Modulen für SigV4 AMB Query-API-Anfragen stellen.

Kopieren Sie die folgende `package.json` Datei in das Arbeitsverzeichnis Ihrer lokalen Umgebung:

```
{
  "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— Rufen Sie mithilfe der AMB Query API den historischen Token-Saldo von einer bestimmten externen Adresse (EOA) ab GetTokenBalance

Sie können die `GetTokenBalance` API verwenden, um den Saldo verschiedener Tokens (z. B., ERC20 ERC721, und ERC1155) und systemeigener Münzen (z. B. ETH und BTC) abzurufen, sodass Sie den aktuellen Saldo eines externen Kontos (EOA) auf der Grundlage eines historischen `timestamp` (Unix-Zeitstempel — Sekunden) ermitteln können. In diesem Beispiel verwenden Sie die `GetTokenBalance`API, um den Adresssaldo eines ERC20 Tokens, USDC, im Ethereum-Mainnet abzurufen.

Um die `GetTokenBalance` API zu testen, kopieren Sie den folgenden Code in eine Datei mit dem Namen `token-balance.js` und speichern Sie die Datei im selben Arbeitsverzeichnis:

```
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);
```

Um den Code auszuführen, öffnen Sie ein Terminal im selben Verzeichnis wie Ihre Dateien und führen Sie den folgenden Befehl aus:

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

Dieser Befehl führt das Skript aus und übergibt die im Code definierten Argumente, um den ERC20 USDC-Kontostand der im Ethereum-Mainnet gelisteten EOA anzufordern. Die Antwort ähnelt dem folgenden Beispiel:

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

Stellen Sie mithilfe von Python API-Anfragen für Amazon Managed Blockchain (AMB) -Abfragen

Um diese Python-Beispiele auszuführen, gelten die folgenden Voraussetzungen:

1. Sie müssen Python auf Ihrem Computer installiert haben. Eine Installationsanleitung für Ihr Betriebssystem finden Sie hier.
2. Installieren Sie das AWS-SDK SDK for Python (Boto3).
3. Installieren Sie die AWS Befehlszeilenschnittstelle und führen Sie den Befehl aus`aws configure`, um die Variablen für Ihr `Access Key ID``Secret Access Key`, und festzulegen. `Region`

Nachdem Sie alle Voraussetzungen erfüllt haben, können Sie das AWS SDK für Python über HTTPS verwenden, um API-Anfragen für Amazon Managed Blockchain (AMB) Query zu stellen.

Das folgende Python-Beispiel verwendet Module von boto3, um Anfragen mit den erforderlichen SigV4-Headern an den AMB Query API-Vorgang zu senden. `ListTransactionEvents` In diesem Beispiel wird eine Liste von Ereignissen abgerufen, die von einer bestimmten Transaktion im Ethereum-Mainnet ausgelöst wurden.

Kopieren Sie die folgende `list-transaction-events.py` Datei in das Arbeitsverzeichnis Ihrer lokalen Umgebung:

```
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')))
```

Um den Beispielcode auszuführen`ListTransactionEvents`, speichern Sie die Datei in Ihrem Arbeitsverzeichnis und führen Sie dann den Befehl aus`python3 list-transaction-events.py`. Dieser Befehl führt das Skript aus und übergibt die im Code definierten Argumente, um die Ereignisse anzufordern, die mit dem angegebenen Transaktions-Hash im Ethereum-Mainnet verknüpft sind. Die Antwort ähnelt dem folgenden Beispiel:

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

Verwenden Sie Amazon Managed Blockchain (AMB) Query auf dem AWS-Managementkonsole , um den GetTokenBalance Vorgang auszuführen

Das folgende Beispiel zeigt, wie Sie den Saldo eines Tokens im Ethereum-Mainnet mithilfe von Amazon Managed Blockchain (AMB) Query auf dem AWS-Managementkonsole

Example

1. Öffnen Sie die Amazon Managed Blockchain Blockchain-Konsole unter https://console.aws.amazon.com/managedblockchain/.
2. Wählen Sie im Abfragebereich den Abfrage-Editor aus.
3. Wählen Sie ETHEREUM_MAINNET als Blockchain-Netzwerk.
4. Wählen Sie GetTokenBalanceals Abfragetyp.
5. Geben Sie Ihre Blockchain-Adresse für das Token ein.
6. Geben Sie die Vertragsadresse für das Token ein.
7. Geben Sie die optionale Token-ID für das Token ein.
8. Wählen Sie das Datum „Datum“ für das Tokenguthaben aus.
9. Geben Sie optional die Uhrzeit für das Token-Guthaben ein.
10. Wählen Sie Abfrage ausführen.

AMB Query führt Ihre Abfrage aus und Sie sehen die Ergebnisse im Fenster mit den Abfrageergebnissen.

Anwendungsfälle mit Amazon Managed Blockchain (AMB) Query

Dieses Thema enthält eine Liste der Anwendungsfälle von AMB Query.

Themen

Fragen Sie aktuelle und historische Token-Salden ab

Die `GetTokenBalance`API ruft den Saldo der unterstützten Token (ERC20, ERC721, ERC1155) und nativen Münzen (ETH, BTC) ab, um den aktuellen oder historischen Saldo mithilfe eines universellen Zeitstempels (Unix-Zeitstempel, in Sekunden) externer Konten () zu ermitteln. EOAs Sie können beispielsweise den `GetTokenBalance` API-Vorgang verwenden, um einen Adresssaldo des ERC20 Tokens (USDC) im Ethereum-Mainnet abzurufen. Mithilfe der API-Operation können Sie auch Salden von Tokens und nativen Münzen stapelweise abrufen. `BatchGetTokenBalance`

Weitere Informationen finden Sie im Amazon Managed Blockchain (AMB) Query Reference Guide.

Rufen Sie historische Transaktionsdaten ab

Mit Amazon Managed Blockchain (AMB) Query können Sie historische Daten aus öffentlichen Blockchains wie Ethereum und Bitcoin abrufen. Diese Funktion ermöglicht verschiedene Anwendungsfälle, z. B. das Abrufen eines Transaktionsverlaufs in einer Blockchain-Brieftasche oder die Bereitstellung von Kontextinformationen zu einer Transaktion auf der Grundlage ihres Transaktions-Hashs. Sie können die `ListTransactions`API-Operation verwenden, um eine Liste von Transaktionen für eine bestimmte externe Adresse (EOA) im Ethereum-Mainnet abzurufen, und dann können Sie die `GetTransaction`API-Operation verwenden, um die Transaktionsdetails für eine einzelne Transaktion aus der Liste abzurufen.

Weitere Informationen finden Sie im Amazon Managed Blockchain (AMB) Query Reference Guide.

Ruft alle Token-Guthaben für eine bestimmte Adresse ab

Sie können den `ListTokenBalances`API-Vorgang verwenden, um Guthaben auf Wallets, Benutzeroberflächen, Web3-Dienstprogrammen und mehr abzurufen. Diese API-Operation gibt mithilfe einer einzigen API-Operation eine Liste aller Salden für eine Adresse in Bezug auf Tokens (ERC20 ERC721,, ERC1155) und native Coins (ETH, BTC) in einer bestimmten öffentlichen Blockchain zurück. Sie können beispielsweise eine externe Adresse (EOA) und ein Netzwerk (das Ethereum-Mainnet) angeben und in der Antwort eine Liste mit Tokens und systemeigenen Münzguthaben erhalten.

Weitere Informationen finden Sie im Amazon Managed Blockchain (AMB) Query Reference Guide.

Listet die für eine Transaktion ausgelösten Ereignisse auf

Sie können den `ListTransactionEvents`API-Vorgang verwenden, um eine Liste von Vertragsereignissen abzurufen, die als Ergebnis einer bestimmten Transaktion ausgelöst werden, identifiziert durch ihren Hash (Transaktions-ID). Sie können dies beispielsweise verwenden, `ListTransactionEvents`um die resultierenden Ereignisse einer Transaktion abzurufen, die eine Funktion eines ERC20 Token-Vertrags in der Ethereum-Blockchain aufruft, z. B. ein Übertragungsereignis oder ein Auszahlungsereignis aus dem ERC20 Vertrag.

Weitere Informationen finden Sie im Amazon Managed Blockchain (AMB) Query Reference Guide.

Holen Sie sich alle Tokens, die durch einen Vertrag geprägt wurden

Sie können die `ListTokenBalances`API-Operation verwenden, um eine Liste aller unterstützten Token (ERC20,, ERC1155) zurückzugeben ERC721, die durch einen Vertrag geprägt wurden, wenn die Vertragsadresse als Eingabe übergeben wird. Beispielsweise können Sie mithilfe der API-Operation Informationen zu nicht fungiblen Token (NFTs) abrufen, die nach dem ERC721 Vertragsstandard in der Ethereum-Blockchain geprägt wurden. `ListTokenBalances`

Weitere Informationen finden Sie im Amazon Managed Blockchain (AMB) Query Reference Guide.

Verträge auflisten und Vertragsinformationen abrufen

Sie können den `ListAssetContracts`API-Vorgang verwenden, um ERC-721-, ERC-1155- oder ERC-20-Verträge aufzulisten, die von einer bestimmten Adresse bereitgestellt werden. Wenn Sie über die Vertragsadresse verfügen, können Sie den `GetAssetContract`API-Vorgang außerdem verwenden, um die Eigenschaften des Vertrags abzurufen, z. B. die Adresse des Vertragstyps, der Bereitsteller, und die relevanten Token-Metadaten.

Weitere Informationen finden Sie im Amazon Managed Blockchain (AMB) Query Reference Guide.

API-Referenz für Amazon Managed Blockchain (AMB) - Abfragen

Amazon Managed Blockchain (AMB) Query bietet API-Operationen für die Abfrage unterstützter Blockchains. Dies beinhaltet APIs die Abfrage von Token, Transaktionen und Verträgen. Weitere Informationen finden Sie in der AMB Query API-Referenz.

Sicherheit in der Amazon Managed Blockchain (AMB) - Abfrage

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die so konzipiert sind, dass sie die Anforderungen der sicherheitssensibelsten Unternehmen erfüllen.

Sicherheit ist eine gemeinsame AWS Verantwortung von Ihnen und Ihnen. Das Modell der gemeinsamen Verantwortung beschreibt dies sowohl als Sicherheit in der Cloud als auch als Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, auf der AWS Dienste in der ausgeführt AWS Cloud werden. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der AWS -Compliance-Programme regelmäßig. Weitere Informationen zu den Compliance-Programmen, die für Amazon Managed Blockchain (AMB) Query gelten, finden Sie unter AWS Services in Scope by Compliance Program.
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Service, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, die Anforderungen Ihres Unternehmens und die geltenden Gesetze und Vorschriften.

Um Datenschutz, Authentifizierung und Zugriffskontrolle zu gewährleisten, verwendet Amazon Managed Blockchain AWS Funktionen und Funktionen des Open-Source-Frameworks, das in Managed Blockchain ausgeführt wird.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Verwendung von AMB Query anwenden können. In den folgenden Themen erfahren Sie, wie Sie AMB Query konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie können auch lernen, wie Sie andere AWS Dienste verwenden können, die Sie bei der Überwachung und Sicherung Ihrer AMB Query-Ressourcen unterstützen.

Topics

Datenverschlüsselung

Datenverschlüsselung verhindert, dass unbefugte Benutzer Daten aus einem Blockchain-Netzwerk und den zugehörigen Datenspeichersystemen lesen. Dazu gehören Daten, die bei der Übertragung durch das Netzwerk möglicherweise abgefangen werden, sogenannte Daten bei der Übertragung.

Verschlüsselung während der Übertragung

Standardmäßig verwendet Managed Blockchain eine HTTPS/TLS Verbindung, um alle Daten zu verschlüsseln, die vom AWS CLI Client an die Dienstendpunkte übertragen werden. AWS

Identitäts- und Zugriffsmanagement für Amazon Managed Blockchain (AMB) Query

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf Ressourcen sicher zu kontrollieren. AWS IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um AMB Query-Ressourcen zu verwenden. IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Themen

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von Ihrer Rolle ab:

- Servicebenutzer – Fordern Sie von Ihrem Administrator Berechtigungen an, wenn Sie nicht auf Features zugreifen können (siehe Fehlerbehebung bei Amazon Managed Blockchain (AMB) Identität und Zugriff abfragen).

- Serviceadministrator – Bestimmen Sie den Benutzerzugriff und stellen Sie Berechtigungsanfragen (siehe So funktioniert Amazon Managed Blockchain (AMB) Query mit IAM).
- IAM-Administrator – Schreiben Sie Richtlinien zur Zugriffsverwaltung (siehe Beispiele für identitätsbasierte Richtlinien für Amazon Managed Blockchain (AMB) Query).

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen sich als IAM-Benutzer authentifizieren oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich als föderierte Identität anmelden, indem Sie Anmeldeinformationen aus einer Identitätsquelle wie AWS IAM Identity Center (IAM Identity Center), Single Sign-On-Authentifizierung oder Anmeldeinformationen verwenden. Google/Facebook Weitere Informationen zum Anmelden finden Sie unter So melden Sie sich bei Ihrem AWS-Konto an im Benutzerhandbuch für AWS-Anmeldung .

AWS Bietet für den programmatischen Zugriff ein SDK und eine CLI zum kryptografischen Signieren von Anfragen. Weitere Informationen finden Sie unter AWS Signature Version 4 for API requests im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, dem sogenannten AWS-Konto Root-Benutzer, der vollständigen Zugriff auf alle AWS-Services Ressourcen hat. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Eine Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Tasks that require root user credentials im IAM-Benutzerhandbuch.

Verbundidentität

Es hat sich bewährt, dass menschliche Benutzer für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen einen Verbund mit einem Identitätsanbieter verwenden müssen.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensverzeichnis, Ihrem Directory Service Web-Identitätsanbieter oder der AWS-Services mithilfe von Anmeldeinformationen aus einer Identitätsquelle zugreift. Verbundene Identitäten übernehmen Rollen, die temporäre Anmeldeinformationen bereitstellen.

Für die zentrale Zugriffsverwaltung empfehlen wir AWS IAM Identity Center. Weitere Informationen finden Sie unter Was ist IAM Identity Center? im AWS IAM Identity Center -Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAM-Benutzer ist eine Identität mit bestimmten Berechtigungen für eine einzelne Person oder Anwendung. Wir empfehlen die Verwendung temporärer Anmeldeinformationen anstelle von IAM-Benutzern mit langfristigen Anmeldeinformationen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Erfordern, dass menschliche Benutzer den Verbund mit einem Identitätsanbieter verwenden müssen, um AWS mithilfe temporärer Anmeldeinformationen darauf zugreifen zu können.

Eine IAM-Gruppe spezifiziert eine Sammlung von IAM-Benutzern und erleichtert die Verwaltung von Berechtigungen für große Gruppen von Benutzern. Weitere Informationen finden Sie unter Anwendungsfälle für IAM-Benutzer im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität mit spezifischen Berechtigungen, die temporäre Anmeldeinformationen bereitstellt. Sie können eine Rolle übernehmen, indem Sie von einer Benutzer- zu einer IAM-Rolle (Konsole) wechseln AWS CLI oder einen AWS API-Vorgang aufrufen. Weitere Informationen finden Sie unter Methoden, um eine Rolle zu übernehmen im IAM-Benutzerhandbuch.

IAM-Rollen sind nützlich für den Verbundbenutzer-Zugriff, temporäre IAM-Benutzerberechtigungen, kontoübergreifenden Zugriff, serviceübergreifenden Zugriff und Anwendungen, die auf Amazon EC2 laufen. Weitere Informationen finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie definiert Berechtigungen, wenn sie mit einer Identität oder Ressource verknüpft sind. AWS bewertet diese Richtlinien, wenn ein Principal eine Anfrage stellt. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu JSON-Richtliniendokumenten finden Sie unter Übersicht über JSON-Richtlinien im IAM-Benutzerhandbuch.

Mit Hilfe von Richtlinien legen Administratoren fest, wer Zugriff auf was hat, indem sie definieren, welches Prinzipal welche Aktionen auf welchen Ressourcenund unter welchen Bedingungendurchführen darf.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator erstellt IAM-Richtlinien und fügt sie zu Rollen hinzu, die die Benutzer dann übernehmen können. IAM-Richtlinien definieren Berechtigungen unabhängig von der Methode, die zur Ausführung der Operation verwendet wird.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität (Benutzer, Gruppe oder Rolle) anfügen können. Diese Richtlinien steuern, welche Aktionen Identitäten für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können Inline-Richtlinien (direkt in eine einzelne Identität eingebettet) oder verwaltete Richtlinien (eigenständige Richtlinien, die mit mehreren Identitäten verbunden sind) sein. Informationen dazu, wie Sie zwischen verwalteten und Inline-Richtlinien wählen, finden Sie unter Choose between managed policies and inline policies im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele hierfür sind Vertrauensrichtlinien für IAM-Rollen und Amazon S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben.

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Weitere Richtlinientypen

AWS unterstützt zusätzliche Richtlinientypen, mit denen die maximalen Berechtigungen festgelegt werden können, die durch gängigere Richtlinientypen gewährt werden:

- Berechtigungsgrenzen – Eine Berechtigungsgrenze legt die maximalen Berechtigungen fest, die eine identitätsbasierte Richtlinie einer IAM-Entität erteilen kann. Weitere Informationen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im -IAM-Benutzerhandbuch.
- Richtlinien zur Dienstkontrolle (SCPs) — Geben Sie die maximalen Berechtigungen für eine Organisation oder Organisationseinheit in an AWS Organizations. Weitere Informationen finden Sie unter Service-Kontrollrichtlinien im AWS Organizations -Benutzerhandbuch.

- Richtlinien zur Ressourcenkontrolle (RCPs) — Legen Sie die maximal verfügbaren Berechtigungen für Ressourcen in Ihren Konten fest. Weitere Informationen finden Sie im AWS Organizations Benutzerhandbuch unter Richtlinien zur Ressourcenkontrolle (RCPs).
- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die als Parameter übergeben werden, wenn Sie eine temporäre Sitzung für eine Rolle oder einen Verbundbenutzer erstellen. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn für eine Anfrage mehrere Arten von Richtlinien gelten, sind die daraus resultierenden Berechtigungen schwieriger zu verstehen. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie unter Bewertungslogik für Richtlinien im IAM-Benutzerhandbuch.

So funktioniert Amazon Managed Blockchain (AMB) Query mit IAM

Bevor Sie IAM verwenden, um den Zugriff auf AMB Query zu verwalten, sollten Sie sich darüber informieren, welche IAM-Funktionen für die Verwendung mit AMB Query verfügbar sind.

IAM-Funktionen, die Sie mit Amazon Managed Blockchain (AMB) Query verwenden können

IAM-Feature	Unterstützung für AMB Query
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Nein
Bedingungsschlüssel für die Richtlinie	Nein
ACLs	Nein
ABAC (Tags in Richtlinien)	Nein
Temporäre Anmeldeinformationen	Ja

IAM-Feature	Unterstützung für AMB Query
Prinzipalberechtigungen	Ja
Servicerollen	Nein
Serviceverknüpfte Rollen	Nein

Einen allgemeinen Überblick darüber, wie AMB Query und andere AWS Dienste mit den meisten IAM-Funktionen funktionieren, finden Sie im AWS IAM-Benutzerhandbuch unter Dienste, die mit IAM funktionieren.

Identitätsbasierte Richtlinien für AMB Query

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für AMB Query

Beispiele für identitätsbasierte AMB Query-Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Amazon Managed Blockchain (AMB) Query

Ressourcenbasierte Richtlinien in AMB Query

Unterstützt ressourcenbasierte Richtlinien: Nein

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und

Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Weitere Informationen finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Richtlinienaktionen für AMB Query

Unterstützt Richtlinienaktionen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Nehmen Sie Aktionen in eine Richtlinie auf, um Berechtigungen zur Ausführung des zugehörigen Vorgangs zu erteilen.

Eine Liste der AMB-Query-Aktionen finden Sie unter Von Amazon Managed Blockchain (AMB) Query definierte Aktionen in der Service Authorization Reference.

Richtlinienaktionen in AMB Query verwenden vor der Aktion das folgende Präfix:

```
managedblockchain-query:
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

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

Beispiele für identitätsbasierte AMB Query-Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Amazon Managed Blockchain (AMB) Query

Richtlinienressourcen für AMB Query

Unterstützt Richtlinienressourcen: Nein

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Als Best Practice geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen (ARN) an. Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

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

Eine Liste der AMB-Query-Ressourcentypen und ihrer ARNs Eigenschaften finden Sie unter Resources Defined by Amazon Managed Blockchain (AMB) Query in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen Sie den ARN jeder Ressource angeben können, finden Sie unter Von Amazon Managed Blockchain (AMB) Query definierte Aktionen.

Beispiele für identitätsbasierte AMB Query-Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Amazon Managed Blockchain (AMB) Query

Bedingungsschlüssel für Richtlinien für AMB Query

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Nein

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Condition` gibt an, wann Anweisungen auf der Grundlage definierter Kriterien ausgeführt werden. Sie können bedingte Ausdrücke erstellen, die Bedingungsoperatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der

Anforderung übereinstimmt. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter Kontextschlüssel für AWS globale Bedingungen im IAM-Benutzerhandbuch.

Eine Liste der AMB-Query-Bedingungsschlüssel finden Sie unter Bedingungsschlüssel für Amazon Managed Blockchain (AMB) -Abfragen in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Von Amazon Managed Blockchain (AMB) Query definierte Aktionen.

Beispiele für identitätsbasierte AMB Query-Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Amazon Managed Blockchain (AMB) Query

ACLs in AMB Query

Unterstützt ACLs: Nein

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

ABAC mit AMB Query

Unterstützt ABAC (Tags in Richtlinien): Nein

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen, auch als Tags bezeichnet, definiert werden. Sie können Tags an IAM-Entitäten und AWS -Ressourcen anhängen und dann ABAC-Richtlinien entwerfen, um Operationen zu ermöglichen, wenn das Tag des Prinzipals mit dem Tag auf der Ressource übereinstimmt.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Tag-Informationen an, indem Sie die Schlüssel `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter Definieren von Berechtigungen mit ABAC-Autorisierung im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe Attributbasierte Zugriffskontrolle (ABAC) verwenden im IAM-Benutzerhandbuch.

Verwenden temporärer Anmeldeinformationen mit AMB Query

Unterstützt temporäre Anmeldeinformationen: Ja

Temporäre Anmeldeinformationen ermöglichen kurzfristigen Zugriff auf AWS Ressourcen und werden automatisch erstellt, wenn Sie einen Verbund verwenden oder die Rollen wechseln. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter Temporäre Anmeldeinformationen in IAM und AWS-Services , die mit IAM funktionieren im IAM-Benutzerhandbuch.

Serviceübergreifende Prinzipalberechtigungen für AMB Query

Unterstützt Forward Access Sessions (FAS): Ja

Forward Access Sessions (FAS) verwenden die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen an AWS-Service nachgeschaltete Dienste zu stellen. Einzelheiten zu den Richtlinien für FAS-Anforderungen finden Sie unter Zugriffssitzungen weiterleiten.

Servicerollen für AMB Query

Unterstützt Servicerollen: Nein

Eine Servicerolle ist eine IAM-Rolle, die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.

> **Warning**
>
> Durch das Ändern der Berechtigungen für eine Servicerolle kann die AMB Query-Funktionalität beeinträchtigt werden. Bearbeiten Sie Servicerollen nur, wenn AMB Query dazu eine Anleitung bietet.

Dienstbezogene Rollen für AMB Query

Unterstützt serviceverknüpfte Rollen: Ja

Eine dienstverknüpfte Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene

Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie in der Tabelle nach einem Service mit einem `Yes` in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

Beispiele für identitätsbasierte Richtlinien für Amazon Managed Blockchain (AMB) Query

Standardmäßig sind Benutzer und Rollen nicht berechtigt, AMB Query-Ressourcen zu erstellen oder zu ändern. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von IAM-Richtlinien (Konsole) im IAM-Benutzerhandbuch.

Einzelheiten zu den von AMB Query definierten Aktionen und Ressourcentypen, einschließlich des Formats ARNs für die einzelnen Ressourcentypen, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Managed Blockchain (AMB) -Abfragen in der Service Authorization Reference.

Themen

- Best Practices für Richtlinien
- Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer
- Zugreifen auf bestimmte Amazon Managed Blockchain (AMB) Query API-Aktionen

Best Practices für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand AMB Query-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Beachten Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Richtlinien und Empfehlungen:

- Erste Schritte mit AWS verwalteten Richtlinien und Umstellung auf Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für

viele gängige Anwendungsfälle gewähren, um damit zu beginnen, Ihren Benutzern und Workloads Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien oder Von AWS verwaltete Richtlinien für Auftragsfunktionen im IAM-Benutzerhandbuch.

- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. CloudFormation B. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung mit IAM Access Analyzer im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Sicherer API-Zugriff mit MFA im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Best Practices für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der API oder. 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": "*"
        }
    ]
}
```

Zugreifen auf bestimmte Amazon Managed Blockchain (AMB) Query API-Aktionen

> **Note**
>
> Um auf die AMB-Abfrage zugreifen zu können, um API-Aufrufe zu tätigen, benötigen Sie Benutzeranmeldedaten (`AWS_ACCESS_KEY_IDundAWS_SECRET_ACCESS_KEY`), die über die entsprechenden IAM-Berechtigungen für AMB Query verfügen.

Example IAM-Richtlinie für den Zugriff auf alle Amazon Managed Blockchain (AMB) -Abfragen APIs

In diesem Beispiel wird einem IAM-Benutzer AWS-Konto Zugriff auf alle AMB-Abfragen gewährt. APIs

JSON

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

Example IAM-Richtlinie für den Zugriff auf Amazon Managed Blockchain (AMB) Query und **`ListTransactions GetTransaction`** APIs

Dieses Beispiel gewährt einem IAM-Benutzer AWS-Konto Zugriff auf die AMB-Abfrage und `ListTransaction GetTransaction` APIs

> **Note**
>
> Sie können das APIs im Beispiel durch andere ersetzen oder hinzufügen, APIs um Zugriff auf andere oder mehrere zu gewähren. APIs Eine Liste von AMB Query APIs finden Sie im Amazon Managed Blockchain (AMB) Query API Reference Guide.

JSON

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

Fehlerbehebung bei Amazon Managed Blockchain (AMB) Identität und Zugriff abfragen

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit AMB Query und IAM auftreten können.

Themen

- Ich bin nicht berechtigt, eine Aktion in AMB Query auszuführen

Ich bin nicht berechtigt, eine Aktion in AMB Query auszuführen

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der IAM-Benutzer `mateojackson` versucht, über die Konsole Details zu einer fiktiven *`my-example-widget`*-Ressource anzuzeigen, jedoch nicht über `managedblockchain-query::`*`GetWidget`*-Berechtigungen verfügt.

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

In diesem Fall muss die Richtlinie für den Benutzer `mateojackson` aktualisiert werden, damit er mit der `managedblockchain-query::`*`GetWidget`*-Aktion auf die *`my-example-widget`*-Ressource zugreifen kann.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Amazon Managed Blockchain (AMB) API-Nutzungsmetriken bei Amazon abfragen CloudWatch

API-Nutzungsmetriken bei Amazon CloudWatch

Die veröffentlichten API-Nutzungsmetriken CloudWatch entsprechen den Service-Kontingenten für Amazon Managed Blockchain (AMB) Query. Sie können Alarme so konfigurieren, dass Sie benachrichtigt werden, wenn sich Ihre Nutzung einem Servicekontingent nähert. Weitere Informationen zur CloudWatch Integration mit Servicekontingenten finden Sie unter AWS-Nutzungsmetriken im CloudWatch Amazon-Benutzerhandbuch.

AMB Query veröffentlicht die folgenden API-Metriken im `AWS/Usage` Namespace mit dem `Amazon Managed Blockchain Query` Servicenamen.

Metrik	Description
`CallCount`	Die Gesamtzahl der Aufrufe an eine API in AMB Query. SUM steht für die Gesamtzahl der API-Aufrufe während des angegebenen Zeitraums.

Amazon Managed Blockchain (AMB) Query veröffentlicht Nutzungsmetriken im `AWS/Usage` Namespace mit den folgenden Dimensionen.

Dimension	Beschreibung
Service	Der Name des AWS Dienstes, der die Ressource enthält. `Amazon Managed Blockchain Query` wird immer der Wert für diese Dimension sein.
Typ	Der Typ der Entität, über die berichtet wird. `API`wird immer der Wert für diese Dimension sein.

Dimension	Beschreibung
Ressource	Die Art der Ressourcen, die gemeldet werden. Der Name des verwendeten AMB Query API-Vorgangs wird der Wert für diese Dimension sein.
Klasse	Die Klasse der Ressource, über die berichtet wird. `None`wird immer der Wert für diese Dimension sein.

Dokumentenverlauf für das AMB Query User Guide

In der folgenden Tabelle werden die Dokumentationsversionen für AMB Query beschrieben.

Änderung	Beschreibung	Datum
AMB Query unterstützt Bitcoin-Transaktions-Identifikatoren und Transaktions-Hashes	Für Bitcoin-Netzwerke unterstützen AMB Query API-Operationen sowohl die Transaktions-ID (`transactionId`) als auch den Transaktions-Hash (`transactionHash`).	21. März 2024
Support für API-Nutzungsmetriken bei Amazon CloudWatch	AMB Query hat Unterstützung für API-Nutzungsmetriken hinzugefügt. CloudWatch Diese Nutzungsmetriken entsprechen den AMB Query-Dienstkontingenten.	8. Februar 2024
Support für Transaktionen, die noch nicht abgeschlossen sind	AMB Query hat Unterstützung für Transaktionen hinzugefügt, die noch nicht abgeschlossen sind. Außerdem wird die Unterstützung für die `status` Eigenschaft aus der Antwort des `GetTransaction` Vorgangs entfernt. Stattdessen verwenden Sie die `executionStatus` Eigenschaften `confirmationStatus` und, um den Status der Transaktion zu ermitteln.	1. Februar 2024

Die `status` Eigenschaft im Datentyp „Transaktion“ ist veraltet	Amazon Managed Blockchain (AMB) Query hat die `status` Eigenschaft im Datentyp Transaction als veraltet eingestuft. Sie müssen die `executionStatus` Felder `confirmationStatus` und verwenden, um festzuste llen, ob der Wert `status` der Transaktion oder ist. `FINAL` `FAILED`	20. Dezember 2023
Support für Sepolia Testnet	Amazon Managed Blockchain (AMB) Query unterstützt jetzt Abfragen im Ethereum Sepolia Testnet.	19. Oktober 2023
Support für Vermögens kontrakte	Sie können den `ListAsset Contracts` API-Vorga ng verwenden, um eine Liste aufzulisten, die von einer bestimmten Adresse bereitges tellt wurden. Wenn Sie über die Vertragsadresse verfügen, können Sie außerdem den `GetAssetContract` API-Vorgang verwenden, um die Vertragsdetails abzurufen.	16. Oktober 2023
Support für Bitcoin Testnet	Amazon Managed Blockchain (AMB) Query unterstützt jetzt Abfragen im Bitcoin-Testnet.	16. Oktober 2023
Erstversion	Erste Version des AMB Query-Dienstes.	27. Juli 2023

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.