



Guida per gli sviluppatori

Query su Blockchain gestita da Amazon



Query su Blockchain gestita da Amazon: Guida per gli sviluppatori

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

Table of Contents

Cos'è Amazon Managed Blockchain (AMB) Query?	1
Sei un utente di AMB Query per la prima volta?	1
Concetti chiave	2
Considerazioni e limitazioni per l'utilizzo di Amazon Managed Blockchain (AMB) Query	2
Configurazione	6
Registrati per un Account AWS	6
Crea un utente IAM con le autorizzazioni appropriate	6
Installa e configura AWS Command Line Interface	6
Usa per interrogare Console di gestione AWS le blockchain utilizzando AMB Query	7
Nozioni di base	8
Creazione di una policy IAM	8
Esempi che utilizzano Go	9
Esempi che utilizzano Node.js	16
Esempi che utilizzano Python	19
Esempio di utilizzo di Console di gestione AWS	21
Casi d'uso di AMB Query	23
Interroga i saldi correnti e storici dei token	23
Recupera i dati storici delle transazioni	23
Ottieni tutti i saldi di token per un determinato indirizzo	23
Elenca gli eventi emessi per una transazione	24
Ottieni tutti i token coniati in base a un contratto	24
Elenca i contratti e ottieni informazioni sui contratti	24
Riferimento all'API AMB Query	26
Sicurezza	27
Crittografia dei dati	28
Crittografia dei dati in transito	28
Gestione dell'identità e degli accessi	28
Destinatari	28
Autenticazione con identità	29
Gestione dell'accesso tramite policy	30
In che modo Amazon Managed Blockchain (AMB) Query funziona con IAM	32
Esempi di policy basate su identità	37
Risoluzione dei problemi	42
Metriche di utilizzo delle API	43

Metriche di utilizzo delle API su Amazon CloudWatch	43
Cronologia dei documenti	45
.....	xlvii

Cos'è Amazon Managed Blockchain (AMB) Query?

Amazon Managed Blockchain (AMB) è un servizio completamente gestito progettato per aiutarti a creare applicazioni Web3 resilienti su blockchain pubbliche e private. Usa AMB Access per un accesso istantaneo e senza server a più blockchain. Crea le tue applicazioni predisposte per Web3 senza la necessità di implementare un'infrastruttura blockchain specializzata e di mantenerle connesse alla rete blockchain. Con AMB Query, puoi utilizzare operazioni API intuitive dagli sviluppatori per accedere a dati storici e in tempo reale da più blockchain. I dati blockchain standardizzati possono essere integrati con i servizi AWS, senza richiedere un'infrastruttura blockchain specializzata o ETL (estrazione, trasformazione e caricamento). Tutte le funzionalità AMB si adattano in modo sicuro a build di applicazioni di livello istituzionale e consumer mainstream.

Amazon Managed Blockchain (AMB) Query fornisce l'accesso senza server a set di dati standardizzati e multi-blockchain con operazioni API intuitive per gli sviluppatori. Puoi utilizzare AMB Query per spedire rapidamente applicazioni che richiedono dati da una o più blockchain pubbliche, senza dover sostenere il sovraccarico necessario per analizzare i dati blockchain, tracciare contratti e mantenere un'infrastruttura di indicizzazione specializzata. Sia che tu stia analizzando i saldi storici dei token per token fungibili o token non fungibili (NFTs), visualizzando la cronologia delle transazioni per un determinato indirizzo di portafoglio o eseguendo analisi dei dati sulla distribuzione di criptovalute native come Ether, AMB Query ti consente di accedere ai dati della blockchain.

Sei un utente di AMB Query per la prima volta?

Se utilizzi AMB Query per la prima volta, ti consigliamo di iniziare leggendo le seguenti sezioni:

- [Concetti chiave: Amazon Managed Blockchain \(AMB\) Query](#)
- [Configurazione di Amazon Managed Blockchain \(AMB\) Query](#)
- [Guida introduttiva ad Amazon Managed Blockchain \(AMB\) Query](#)
- [Casi d'uso con Amazon Managed Blockchain \(AMB\) Query](#)

Concetti chiave: Amazon Managed Blockchain (AMB) Query

Note

Questa guida presuppone che tu abbia familiarità con i concetti essenziali della blockchain. Questi concetti includono decentralizzazione, token, contratti, transazioni, portafogli proof-of-work, chiavi pubbliche e private, staking, mining, halvings e altri.

Amazon Managed Blockchain (AMB) Query ti offre un comodo accesso ai dati di rete multi-blockchain, il che semplifica l'estrazione di dati contestuali relativi all'attività blockchain. Puoi usare AMB Query per leggere dati da reti blockchain pubbliche, come Bitcoin Mainnet ed Ethereum Mainnet. Puoi anche ottenere informazioni, come i saldi correnti e storici degli indirizzi, oppure puoi ottenere un elenco di transazioni blockchain per un determinato periodo di tempo. Inoltre, puoi ottenere i dettagli di una determinata transazione, come gli eventi delle transazioni, che puoi analizzare ulteriormente o utilizzare nella logica aziendale per le tue applicazioni.

Considerazioni e limitazioni per l'utilizzo di Amazon Managed Blockchain (AMB) Query

Quando usi AMB Query, considera quanto segue:

- Regioni disponibili

AMB Query è supportato nella regione Stati Uniti orientali (Virginia settentrionale)*us-east-1*.

- Service endpoints (Endpoint del servizio)

AMB Query è accessibile utilizzando il seguente endpoint:

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

- Reti blockchain supportate

AMB Query supporta le seguenti reti blockchain pubbliche:

- **Bitcoin Mainnet:** la rete blockchain pubblica di Bitcoin protetta per proof-of-work consenso e sulla quale viene emessa e negoziata la criptovaluta Bitcoin (BTC). Le transazioni su Mainnet hanno un valore effettivo (ovvero comportano costi reali) e vengono registrate sulla blockchain pubblica.
 - **Bitcoin Testnet:** la testnet per Bitcoin Mainnet. Bitcoin (BTC) su questa rete è separato e distinto da Mainnet BTC e di solito non ha alcun valore.
 - **Ethereum Mainnet:** la rete proof-of-stake principale per la blockchain pubblica di Ethereum. Le transazioni su Mainnet hanno un valore effettivo (ovvero comportano costi reali) e vengono registrate nel registro distribuito.
 - **Sepolia Testnet** — La testnet per la rete principale di Ethereum. Ether (ETH) su questa rete è separato e distinto da Mainnet ETH e di solito non ha alcun valore.
- Token e contratti blockchain supportati

AMB Query supporta i seguenti token contrattuali nativi e standard di Ethereum.

- **Token nativi per la blockchain pubblica**
 - **Bitcoin (BTC):** questo è il token nativo delle blockchain legate a Bitcoin.
 - **Ether (ETH):** questo è il token nativo delle blockchain legate a Ethereum.
- **Standard contrattuali di Ethereum**
 - **Token Standard ERC-20** — L'ERC-20 è uno standard per token fungibili. Ha una proprietà che rende ogni token ERC-20 esattamente uguale (per tipo e valore) a un altro token ERC-20 coniato, il che significa che un token è e sarà sempre uguale a tutti gli altri token. [Per ulteriori informazioni, consulta lo standard dei token ERC-20 su Ethereum.org.](#)
 - **Standard di token non fungibili ERC-721** — L'ERC-721 è uno standard per token non fungibili (NFTs). Questo tipo di token è unico e può avere un valore diverso rispetto a un altro token dello stesso contratto, probabilmente a causa della sua età, rarità o altre proprietà. Per ulteriori informazioni, consulta lo standard dei [token ERC-721](#) su Ethereum.org.

Standard multi-token ERC-1155 — L'ERC-1155 è uno standard che crea un'interfaccia contrattuale in grado di rappresentare e controllare qualsiasi numero di tipi di token fungibili e non fungibili. [In questo modo, il token ERC-1155 può funzionare allo stesso modo dei token ERC-20 ed ERC-721, anche funzionando contemporaneamente.](#) Il token ERC-1155 migliora la funzionalità degli standard ERC-20 ed ERC-721, rendendolo più efficiente e correggendo al contempo gli errori di implementazione evidenti. [Per ulteriori informazioni, consulta lo standard dei token ERC-1155 su Ethereum.org.](#)

- Finalità

Nelle blockchain, la finalità significa che è improbabile che le transazioni valide vengano annullate. Per la rete principale di Bitcoin, AMB Query considera una transazione definitiva dopo 6 blocchi. Per Bitcoin Testnet, considera definitiva una transazione dopo 6 blocchi o 60 minuti, a seconda dell'evento che si verifica per primo. Per le reti Ethereum supportate, AMB Query considera una transazione definitiva dopo 64 blocchi.


Le operazioni API relative al saldo dei token e ai contratti di AMB Query restituiscono solo i dati che hanno raggiunto la finalità. Tuttavia, le operazioni API relative alle transazioni e agli eventi di transazione di AMB Query possono restituire dati per transazioni confermate sulla rete blockchain anche se non sono ancora state completate.

- Indirizzo NULL non supportato

AMB Query non supporta l'indirizzo NULL
(0x00).

- Signature (versione 4): firma delle chiamate API

Quando si effettuano chiamate a AMB Query APIs, è possibile farlo tramite una connessione HTTPS autenticata utilizzando il processo di [firma Signature Version 4](#). Ciò significa che solo i principali IAM autorizzati presenti nell' AWS account possono effettuare chiamate API AMB Query. Per fare ciò, è necessario fornire AWS delle credenziali (un ID della chiave di accesso e una chiave di accesso segreta) con la chiamata.

 Important

Non incorporate le credenziali dei client nelle applicazioni rivolte agli utenti.

- AMB Query supporta gli identificatori e gli hash delle transazioni Bitcoin

Per le reti Bitcoin, le operazioni dell'API AMB Query supportano sia l'identificatore di transazione (`transactionId`) che l'hash della transazione (`transactionHash`). `transactionHash` tratta di un hash a doppio SHA della transazione, esclusi i dati dei testimoni. `transactionId` tratta di un hash a doppio SHA della transazione che include i dati dei testimoni (noto anche come ID della transazione testimone).

Quando richiami le operazioni [GetTransaction](#) [ListTransactionEvents](#) API per le reti Bitcoin, puoi specificare il o il. `transactionId` `transactionHash` Inoltre, tutte le operazioni AMB Query sulle reti Bitcoin che restituiscono a `transactionId` o a `transactionHash` includeranno entrambi i valori come parte della risposta.

Configurazione di Amazon Managed Blockchain (AMB) Query

Prima di utilizzare Amazon Managed Blockchain (AMB) Query per la prima volta, segui i passaggi in questa sezione per creare un AWS account. La sezione seguente illustra come iniziare a usare AMB Query.

Registrati per un Account AWS

Per iniziare AWS, hai bisogno di un Account AWS. Per informazioni sulla creazione di un Account AWS, vedi Guida [introduttiva a un Account AWS](#) nella Guida Gestione dell'account AWS di riferimento.

Crea un utente IAM con le autorizzazioni appropriate

Per creare e utilizzare AMB Query, devi creare un principale AWS Identity and Access Management (IAM) (utente o gruppo) con autorizzazioni che consentano le azioni Managed Blockchain necessarie.

Solo i presidi IAM possono effettuare richieste API AMB Query. Quando si effettuano chiamate alle API AMB Query, è possibile farlo tramite una connessione HTTPS autenticata utilizzando il processo di [firma Signature](#) Version 4. Ciò significa che solo i principali IAM autorizzati presenti nell' AWS account possono effettuare chiamate all'API AMB Query. Per fare ciò, è necessario fornire AWS delle credenziali (un ID della chiave di accesso e una chiave di accesso segreta) con la chiamata.

Per informazioni su come creare un utente IAM, consulta [Creazione di un utente IAM nel tuo AWS account](#). Per ulteriori informazioni su come allegare una politica di autorizzazioni a un utente, consulta [Modifica delle autorizzazioni per un utente IAM](#). Per un esempio di politica di autorizzazioni che puoi utilizzare per concedere a un utente il permesso di lavorare con AMB Query, vedi. [Esempi di policy basate sull'identità per Amazon Managed Blockchain \(AMB\) Query](#)

Installa e configura AWS Command Line Interface

Se non l'hai già fatto, installa l' AWS Command-Line interfaccia (CLI) più recente per utilizzare AWS le risorse di un terminale. Per ulteriori informazioni, consulta [Installare o aggiornare la versione più recente della AWS CLI](#).

Note

Per l'accesso alla CLI, sono necessari un ID chiave di accesso e una chiave di accesso segreta. Utilizza credenziali temporanee al posto delle chiavi di accesso a lungo termine quando possibile. Le credenziali temporanee includono un ID della chiave di accesso, una chiave di accesso segreta e un token di sicurezza che ne indica la scadenza. Per ulteriori informazioni, consulta [Using temporary credenziali with AWS resources](#) nella IAM User Guide.

Usa il Console di gestione AWS per interrogare le blockchain utilizzando Amazon Managed Blockchain (AMB) Query

Puoi accedere ad Amazon Managed Blockchain (AMB) Query ed effettuare query sulle reti blockchain supportate utilizzando il Console di gestione AWS. I seguenti passaggi mostrano come eseguire questa operazione:

1. Apri la console Amazon Managed Blockchain all'indirizzo <https://console.aws.amazon.com/managedblockchain/>.
2. Scegli Query editor dalla sezione Query.
3. Scegli tra una delle reti Blockchain supportate.
4. Scegli il tipo di query che desideri eseguire.
5. Immettete i parametri pertinenti per il tipo di interrogazione selezionato ed Esegui la query.

AMB Query eseguirà la tua interrogazione e vedrai i risultati nella finestra dei risultati della query.

Guida introduttiva ad Amazon Managed Blockchain (AMB) Query

Utilizza step-by-step i tutorial in questa sezione per imparare a eseguire attività utilizzando Amazon Managed Blockchain (AMB) Query. Queste procedure richiedono alcuni prerequisiti. Se non conosci AMB Query, puoi consultare la sezione Configurazione di questa guida. Per ulteriori informazioni, consulta [Configurazione di Amazon Managed Blockchain \(AMB\) Query](#).

Note

Alcune variabili in questi esempi sono state deliberatamente offuscate. Sostituiscile con altre valide prima di eseguire questi esempi.

Argomenti

- [Crea una policy IAM per accedere alle operazioni dell'API AMB Query](#)
- [Effettua richieste API Amazon Managed Blockchain \(AMB\) Query utilizzando Go](#)
- [Effettua richieste API Amazon Managed Blockchain \(AMB\) Query utilizzando Node.js](#)
- [Effettua richieste API Amazon Managed Blockchain \(AMB\) Query utilizzando Python](#)
- [Usa Amazon Managed Blockchain \(AMB\) Query su Console di gestione AWS per eseguire l'operazione GetTokenBalance](#)

Crea una policy IAM per accedere alle operazioni dell'API AMB Query

Per effettuare richieste API AMB Query, devi utilizzare le credenziali utente (AWS_ACCESS_KEY_ID e AWS_SECRET_ACCESS_KEY) che dispongono delle autorizzazioni IAM appropriate per Amazon Managed Blockchain (AMB) Query. In un terminale su cui è AWS CLI installato, esegui il seguente comando per creare una policy IAM per accedere alle operazioni dell'API AMB Query:

```
cat <<EOT > ~/amb-query-access-policy.json
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
    "Sid" : "AMBQueryAccessPolicy",
    "Effect": "Allow",
    "Action": [
        "managedblockchain-query:*"
    ],
    "Resource": "*"
}
]
}
EOT
aws iam create-policy --policy-name AmazonManagedBlockchainQueryAccess --policy-
document file://$HOME/amb-query-access-policy.json
```

Dopo aver creato la policy, associala al ruolo di un utente IAM per renderla effettiva. Nella Console di gestione AWS, accedi al servizio IAM e collega la policy AmazonManagedBlockchainQueryAccess al ruolo assegnato all'utente IAM che utilizzerà il servizio. Per ulteriori informazioni, consulta [Creazione di un ruolo e assegnazione a un utente IAM](#).

Note

AWS consiglia di consentire l'accesso a operazioni API specifiche anziché utilizzare la * wild-card. Per ulteriori informazioni, consulta [Accesso a specifiche azioni dell'API Amazon Managed Blockchain \(AMB\) Query](#).

Effettua richieste API Amazon Managed Blockchain (AMB) Query utilizzando Go

Con Amazon Managed Blockchain (AMB) Query, puoi creare applicazioni che dipendono dall'accesso istantaneo ai dati della blockchain una volta confermati sulla blockchain, anche se non sono ancora stati raggiunti. AMB Query consente diversi casi d'uso, come la compilazione della cronologia delle transazioni di un portafoglio, la fornitura di informazioni contestuali su una transazione in base all'hash della transazione o l'ottenimento del saldo di un token nativo e dei token ERC-721, ERC-1155 ed ERC-20.

I seguenti esempi sono creati nel linguaggio Go e utilizzano le operazioni dell'API AMB Query. Per ulteriori informazioni su Go, consulta la [documentazione di Go](#). Per ulteriori informazioni sull'API AMB Query, consulta la documentazione di [riferimento sull'API Query di Amazon Managed Blockchain \(AMB\)](#).

Gli esempi seguenti utilizzano le azioni `ListTransactions` e le `GetTransaction` API per ottenere prima un elenco di tutte le transazioni per un determinato indirizzo di proprietà esterna (EOA) sulla rete principale di Ethereum, quindi l'esempio successivo recupera i dettagli della transazione per una singola transazione dall'elenco.

Example— Effettua l'azione API usando Go `ListTransactions`

Copia il codice seguente in un file denominato `listTransactions.go` nella `ListTransactions` directory.

```
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 := "0x0000bf26964af9d7eed9e03e53415d*****"
    network := managedblockchainquery.QueryNetworkEthereumMainnet
    sortOrder := managedblockchainquery.SortOrderAscending
    fromTime := time.Date(1971, 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)
}
}

```

Dopo aver salvato il file, esegui il codice utilizzando il seguente comando all'interno della `ListTransactions` directory: `go run listTransactions.go`.

L'output che segue è simile al seguente:

```

{
  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— Esegui l'azione dell'GetTransactionAPI utilizzando Go

Questo esempio utilizza un hash di transazione dell'output precedente. Copia il codice seguente in un file denominato `GetTransaction.go` nella `GetTransactiondirectory`.

```

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 confirmed on the blockchain, even if they have not
// reached finality.
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)
}
}

```

Dopo aver salvato il file, esegui il codice utilizzando il seguente comando all'interno della GetTransactiondirectory:go run GetTransaction.go.

L'output che segue è simile al seguente:

```

{
  Transaction: {
    BlockHash: "0x000005c6a71d1afbc005a652b6ceca71cd516d97b0fc514c2a1d0f2ca3912345",
    BlockNumber: "11111111",
    CumulativeGasUsed: "5555555",
    EffectiveGasPrice: "444444444444",
    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
}
}
```

L'GetTokenBalanceAPI consente di ottenere il saldo dei token nativi (ETH e BTC), che possono essere utilizzati per ottenere il saldo corrente di un conto di proprietà esterna (EOA) in un determinato momento.

Example— Usa l'azione GetTokenBalance API per ottenere il saldo di un token nativo in Go

Nell'esempio seguente, utilizzi l'GetTokenBalanceAPI per ottenere un saldo in Ether (ETH) di un indirizzo sulla rete principale di Ethereum. Copia il codice seguente in un file denominato GetTokenBalanceEth.go nella GetTokenBalancedirectory.

```
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)
}
}
```

Dopo aver salvato il file, esegui il codice utilizzando il seguente comando all'interno della `GetTokenBalancedirectory`: `go run GetTokenBalanceEth.go`.

L'output che segue è simile al seguente:

```
{
  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"
  }
}
```

Effettua richieste API Amazon Managed Blockchain (AMB) Query utilizzando Node.js

Per eseguire questi esempi di nodi, si applicano i seguenti prerequisiti:

1. È necessario che il node version manager (nvm) e Node.js siano installati sul computer. [Puoi trovare le istruzioni di installazione per il tuo sistema operativo qui.](#)
2. Usa il `node --version` comando e conferma che stai usando la versione 14 o successiva di Node. Se necessario, è possibile utilizzare il `nvm install 14` comando, seguito dal `nvm use 14` comando per installare la versione 14.
3. Le variabili `AWS_ACCESS_KEY_ID` di ambiente `AWS_SECRET_ACCESS_KEY` devono contenere le credenziali associate all'account.

Esporta queste variabili come stringhe sul tuo client utilizzando i seguenti comandi. Sostituisci i valori evidenziati di seguito con i valori appropriati dell'account utente IAM.

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

Note

- Dopo aver completato tutti i prerequisiti, puoi inviare richieste firmate tramite HTTPS per accedere alle operazioni dell'API Amazon Managed Blockchain (AMB) Query ed effettuare richieste utilizzando il [modulo https nativo in Node.js](#), oppure puoi utilizzare una libreria di terze parti come [AXIOS](#) e recuperare dati da AMB Query.
- Questi esempi utilizzano un client HTTP di terze parti per Node.js, ma puoi anche utilizzare l' AWS JavaScript SDK per effettuare richieste a AMB Query.
- L'esempio seguente mostra come effettuare richieste API AMB Query utilizzando Axios e i moduli AWS SDK per SigV4.

Copiate il seguente `package.json` file nella directory di lavoro del vostro ambiente locale:

```
{  
  "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— Recupera il saldo storico dei token da uno specifico indirizzo di proprietà esterna (EOA) utilizzando l'API AMB Query GetTokenBalance

Puoi utilizzare l'GetTokenBalanceAPI per ottenere il saldo di vari token (ad esempio, e ERC1155) e monete native (ad esempio ERC20 ERC721, ETH e BTC), che puoi utilizzare per ottenere il saldo corrente di un account di proprietà esterna (EOA) in base a uno storico timestamp (timestamp Unix: secondi). In questo esempio, si utilizza l'[GetTokenBalanceAPI](#) per ottenere il saldo degli indirizzi di un ERC20 token, USDC, sulla rete principale di Ethereum.

Per testare l'GetTokenBalanceAPI, copia il codice seguente in un file denominato token-balance.js e salva il file nella stessa directory di lavoro:

```
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);

```

Per eseguire il codice, aprite un terminale nella stessa directory dei file ed eseguite il seguente comando:

```

npm i
node token-balance.js

```

Questo comando esegue lo script, passando gli argomenti definiti nel codice per richiedere il saldo ERC20 USDC dell'EOA elencato sulla rete principale di Ethereum. La risposta è simile a quella riportata di seguito.

```

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

```

Effettua richieste API Amazon Managed Blockchain (AMB) Query utilizzando Python

Per eseguire questi esempi in Python, si applicano i seguenti prerequisiti:

1. Devi avere Python installato sulla tua macchina. Puoi trovare le istruzioni di installazione per il tuo sistema operativo [qui](#).
2. Installa l'[SDK AWS per Python \(Boto3\)](#).
3. Installa l'[interfaccia a riga di AWS comando](#) ed esegui il comando `aws configure` per impostare le variabili per, e. Access Key ID Secret Access Key Region

Dopo aver completato tutti i prerequisiti, puoi utilizzare l' AWS SDK per Python su HTTPS per effettuare richieste API Amazon Managed Blockchain (AMB) Query.

Il seguente esempio di Python utilizza i moduli di boto3 per inviare richieste apposte con le intestazioni SigV4 richieste all'operazione AMB Query API. `ListTransactionEvents` Questo esempio recupera un elenco di eventi emessi da una determinata transazione sulla rete principale di Ethereum.

Copia il seguente `list-transaction-events.py` file nella directory di lavoro del tuo ambiente locale:

```
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 confirmed on the blockchain, even if they have not reached
# finality.
listTransactionEvents = signed_request(url, method, params, service, region)

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

Per eseguire il codice di esempio su `ListTransactionEvents`, salvate il file nella directory di lavoro, quindi eseguite il comando `python3 list-transaction-events.py`. Questo comando esegue lo script, passando gli argomenti definiti nel codice per richiedere gli eventi associati all'hash della transazione specificato sulla rete principale di Ethereum. La risposta è simile a quella riportata di seguito.

```
{
  'events':
  [
    {
      'contractAddress': '0x95ad61b0a150d79219dcf64e1e6cc01f*****',
      'eventType': 'ERC20_TRANSFER',
      'from': '0xab5801a7d398351b8be11c439e05c5b3*****',
      'network': 'ETHEREUM_MAINNET',
      'to': '0xdead00000000000000000000420694206942*****',
      'transactionHash':
      '0x125714bb4db48757007fff2671b37637bbfd6d47b3a4757ebbd0c522*****',
      'value': '410241996771871894771826174755464'
    }
  ]
}
```

Usa Amazon Managed Blockchain (AMB) Query su Console di gestione AWS per eseguire l'operazione `GetTokenBalance`

L'esempio seguente mostra come ottenere il saldo di un token sulla rete principale di Ethereum utilizzando Amazon Managed Blockchain (AMB) Query su Console di gestione AWS

Example

1. Apri la console Amazon Managed Blockchain all'indirizzo <https://console.aws.amazon.com/managedblockchain/>.
2. Scegli Query editor dalla sezione Query.
3. Scegli ETHEREUM_MAINNET come rete Blockchain.
4. Scegli GetTokenBalance come tipo di query.
5. Inserisci il tuo indirizzo Blockchain per il token.
6. Inserisci l'indirizzo del contratto per il token.
7. Inserisci l'ID token opzionale per il token.
8. Scegli la data di scadenza per il saldo del token.
9. Inserisci l'opzione At time per il saldo del token.
10. Scegli Esegui query.

AMB Query eseguirà la tua interrogazione e vedrai i risultati nella finestra dei risultati della query.

Casi d'uso con Amazon Managed Blockchain (AMB) Query

Questo argomento fornisce un elenco dei casi d'uso di AMB Query.

Argomenti

- [Interroga i saldi correnti e storici dei token](#)
- [Recupera i dati storici delle transazioni](#)
- [Ottieni tutti i saldi di token per un determinato indirizzo](#)
- [Elenca gli eventi emessi per una transazione](#)
- [Ottieni tutti i token conati in base a un contratto](#)
- [Elenca i contratti e ottieni informazioni sui contratti](#)

Interroga i saldi correnti e storici dei token

L'[GetTokenBalance](#) API ottiene il saldo dei token supportati (ERC20, ERC721, ERC1155) e delle monete native (ETH, BTC) per ottenere il saldo attuale o storico utilizzando un timestamp universale (timestamp Unix, in secondi) di account di proprietà esterna (). EOAs Ad esempio, puoi utilizzare l'operazione `GetTokenBalance` API per ottenere un saldo degli indirizzi del token, USDC, sulla ERC20 rete principale di Ethereum. Puoi anche recuperare in batch i saldi di token e monete native utilizzando l'operazione `BatchGetTokenBalance`

Per ulteriori informazioni, consulta la [Amazon Managed Blockchain \(AMB\) Query Reference Guide](#).

Recupera i dati storici delle transazioni

Con Amazon Managed Blockchain (AMB) Query, puoi recuperare dati storici da blockchain pubbliche come Ethereum e Bitcoin. Questa funzionalità consente diversi casi d'uso, come il recupero della cronologia delle transazioni su un portafoglio blockchain o la fornitura di informazioni contestuali su una transazione in base all'hash della transazione. È possibile utilizzare l'operazione [ListTransactions](#) API per ottenere un elenco di transazioni per un determinato indirizzo di proprietà esterna (EOA) sulla rete principale di Ethereum, quindi è possibile utilizzare l'operazione [GetTransaction](#) API per recuperare i dettagli della transazione per una singola transazione dall'elenco.

Per ulteriori informazioni, consulta la [Amazon Managed Blockchain \(AMB\) Query Reference Guide](#).

Ottieni tutti i saldi di token per un determinato indirizzo

Puoi utilizzare l'operazione [ListTokenBalances](#) API per ottenere saldi su portafogli, interfacce utente, utilità web3 e altro ancora. Questa operazione API restituisce un elenco di tutti i saldi di un indirizzo tra token (ERC20, ERC1155) e monete native (ETH ERC721, BTC) su una determinata blockchain pubblica utilizzando un'unica operazione API. Ad esempio, puoi fornire un indirizzo di proprietà esterna (EOA) e una rete (la rete principale di Ethereum) e nella risposta puoi ricevere un elenco di token e saldi di monete nativi.

Per ulteriori informazioni, consulta la [Amazon Managed Blockchain \(AMB\) Query Reference Guide](#).

Elenca gli eventi emessi per una transazione

Puoi utilizzare l'operazione [ListTransactionEvents](#) API per recuperare un elenco di eventi contrattuali emessi come risultato di una determinata transazione, identificati dal relativo hash (identificatore della transazione). Ad esempio, puoi utilizzarla [ListTransactionEvents](#) per recuperare gli eventi risultanti di una transazione che richiama una funzione di un contratto ERC20 token sulla Blockchain di Ethereum, come un evento di trasferimento o un evento di ritiro dal contratto. ERC20

Per ulteriori informazioni, consulta la [Amazon Managed Blockchain \(AMB\) Query Reference Guide](#).

Ottieni tutti i token conati in base a un contratto

Puoi utilizzare l'operazione [ListTokenBalances](#) API per restituire un elenco di tutti i token supportati (ERC20, ERC721, ERC1155) conati da un contratto quando viene fornito l'indirizzo del contratto come input. Ad esempio, puoi recuperare informazioni relative ai token non fungibili (NFTs) conati dallo standard ERC721 contrattuale sulla blockchain di Ethereum utilizzando l'operazione API [ListTokenBalances](#)

Per ulteriori informazioni, consulta la [Amazon Managed Blockchain \(AMB\) Query Reference Guide](#).

Elenca i contratti e ottieni informazioni sui contratti

Puoi utilizzare l'operazione [ListAssetContracts](#) API per elencare i contratti ERC-721, ERC-1155 o ERC-20 distribuiti da un determinato indirizzo. Inoltre, se disponi dell'indirizzo del contratto, puoi utilizzare l'operazione [GetAssetContract](#) API per recuperare le proprietà del contratto, come l'indirizzo del implementatore del tipo di contratto e i metadati del token pertinenti.

Per ulteriori informazioni, consulta la [Amazon Managed Blockchain \(AMB\) Query Reference Guide](#).

Riferimento all'API di interrogazione Amazon Managed Blockchain (AMB)

Amazon Managed Blockchain (AMB) Query fornisce operazioni API per interrogare le blockchain supportate. Ciò include APIs l'interrogazione di token, transazioni e contratti. Per ulteriori informazioni, consulta l'[AMB Query](#) API Reference.

Sicurezza in Amazon Managed Blockchain (AMB) Query

La sicurezza del cloud ha AWS la massima priorità. In qualità di AWS cliente, puoi beneficiare di data center e architetture di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra te e te. AWS Il [modello di responsabilità condivisa](#) lo descrive sia come sicurezza del cloud che come sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel Cloud AWS. AWS fornisce inoltre servizi che è possibile utilizzare in modo sicuro. I revisori di terze parti testano e verificano regolarmente l'efficacia della sicurezza come parte dei [programmi di conformitàAWS](#). Per maggiori informazioni sui programmi di conformità che si applicano ad Amazon Managed Blockchain (AMB) Query, consulta [AWS Services in Scope by Compliance Program](#).
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. L'utente è anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti dell'azienda e le leggi e le normative applicabili.

Per fornire protezione dei dati, autenticazione e controllo degli accessi, Amazon Managed Blockchain utilizza AWS le caratteristiche e le caratteristiche del framework open source in esecuzione in Managed Blockchain.

Questa documentazione ti aiuta a capire come applicare il modello di responsabilità condivisa quando usi AMB Query. I seguenti argomenti mostrano come configurare AMB Query per soddisfare gli obiettivi di sicurezza e conformità. Puoi anche imparare a utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere le tue risorse AMB Query.

Argomenti

- [Crittografia dei dati](#)
- [Gestione delle identità e degli accessi per Amazon Managed Blockchain \(AMB\) Query](#)

Crittografia dei dati

La crittografia dei dati aiuta a impedire agli utenti non autorizzati di leggere i dati da una rete blockchain e dai sistemi di archiviazione dati associati. Ciò include i dati che potrebbero essere intercettati mentre viaggiano nella rete, noti come dati in transito.

Crittografia dei dati in transito

Per impostazione predefinita, Managed Blockchain utilizza una HTTPS/TLS connessione per crittografare tutti i dati trasmessi dal AWS CLI client agli endpoint del servizio. AWS

Gestione delle identità e degli accessi per Amazon Managed Blockchain (AMB) Query

AWS Identity and Access Management (IAM) è un software Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle risorse. AWS Gli amministratori IAM controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse AMB Query. IAM è un software Servizio AWS che puoi utilizzare senza costi aggiuntivi.

Argomenti

- [Destinatari](#)
- [Autenticazione con identità](#)
- [Gestione dell'accesso tramite policy](#)
- [In che modo Amazon Managed Blockchain \(AMB\) Query funziona con IAM](#)
- [Esempi di policy basate sull'identità per Amazon Managed Blockchain \(AMB\) Query](#)
- [Risoluzione dei problemi relativi all'identità e all'accesso ad Amazon Managed Blockchain \(AMB\) Query](#)

Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia in base al tuo ruolo:

- Utente del servizio: richiedi le autorizzazioni all'amministratore se non riesci ad accedere alle funzionalità (consulta [Risoluzione dei problemi relativi all'identità e all'accesso ad Amazon Managed Blockchain \(AMB\) Query](#))

- Amministratore del servizio: determina l'accesso degli utenti e invia le richieste di autorizzazione (consulta [In che modo Amazon Managed Blockchain \(AMB\) Query funziona con IAM](#))
- Amministratore IAM: scrivi policy per gestire l'accesso (consulta [Esempi di policy basate sull'identità per Amazon Managed Blockchain \(AMB\) Query](#))

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi autenticarti come utente IAM o assumendo un ruolo IAM. Utente root dell'account AWS

Puoi accedere come identità federata utilizzando credenziali provenienti da una fonte di identità come Centro identità AWS IAM (IAM Identity Center), autenticazione Single Sign-On o credenziali. Google/Facebook Per ulteriori informazioni sull'accesso, consulta [Come accedere all' Account AWS](#) nella Guida per l'utente di Accedi ad AWS .

Per l'accesso programmatico, AWS fornisce un SDK e una CLI per firmare crittograficamente le richieste. Per ulteriori informazioni, consulta [AWS Signature Version 4 per le richieste API](#) nella Guida per l'utente di IAM.

Account AWS utente root

Quando si crea un Account AWS, si inizia con un'identità di accesso denominata utente Account AWS root che ha accesso completo a tutte Servizi AWS le risorse. Consigliamo vivamente di non utilizzare l'utente root per le attività quotidiane. Per le attività che richiedono le credenziali dell'utente root, consulta [Attività che richiedono le credenziali dell'utente root](#) nella Guida per l'utente IAM.

Identità federata

Come procedura ottimale, richiedi agli utenti umani di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente della directory aziendale, del provider di identità Web o Directory Service che accede Servizi AWS utilizzando le credenziali di una fonte di identità. Le identità federate assumono ruoli che forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, si consiglia di utilizzare Centro identità AWS IAM. Per ulteriori informazioni, consulta [Che cos'è il Centro identità IAM?](#) nella Guida per l'utente di Centro identità AWS IAM .

Utenti e gruppi IAM

Un [utente IAM](#) è una identità che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ti consigliamo di utilizzare credenziali temporanee invece di utenti IAM con credenziali a lungo termine. Per ulteriori informazioni, consulta [Richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee](#) nella Guida per l'utente IAM.

Un [gruppo IAM](#) specifica una raccolta di utenti IAM e semplifica la gestione delle autorizzazioni per gestire gruppi di utenti di grandi dimensioni. Per ulteriori informazioni, consulta [Casi d'uso per utenti IAM](#) nella Guida per l'utente di IAM.

Ruoli IAM

Un [ruolo IAM](#) è un'identità con autorizzazioni specifiche che fornisce credenziali temporanee. Puoi assumere un ruolo [passando da un ruolo utente a un ruolo IAM \(console\)](#) o chiamando un'operazione AWS CLI o AWS API. Per ulteriori informazioni, consulta [Metodi per assumere un ruolo](#) nella Guida per l'utente di IAM.

I ruoli IAM sono utili per l'accesso degli utenti federati, le autorizzazioni utente IAM temporanee, l'accesso multi-account, l'accesso multi-servizio e le applicazioni in esecuzione su Amazon EC2. Per maggiori informazioni, consultare [Accesso a risorse multi-account in IAM](#) nella Guida per l'utente IAM.

Gestione dell'accesso tramite policy

Puoi controllare l'accesso AWS creando policy e associandole a AWS identità o risorse. Una policy definisce le autorizzazioni quando è associata a un'identità o a una risorsa. AWS valuta queste politiche quando un preside effettua una richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per maggiori informazioni sui documenti delle policy JSON, consulta [Panoramica delle policy JSON](#) nella Guida per l'utente IAM.

Utilizzando le policy, gli amministratori specificano chi ha accesso a cosa definendo quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Un amministratore IAM crea le policy IAM e le aggiunge ai ruoli, che gli utenti possono quindi assumere. Le policy IAM definiscono le autorizzazioni indipendentemente dal metodo utilizzato per eseguirle.

Policy basate sull'identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile collegare a un'identità (utente, gruppo o ruolo). Tali policy controllano le operazioni autorizzate per l'identità, nonché le risorse e le condizioni in cui possono essere eseguite. Per informazioni su come creare una policy basata su identità, consultare [Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente](#) nella Guida per l'utente IAM.

Le policy basate su identità possono essere policy in linea (con embedding direttamente in una singola identità) o policy gestite (policy autonome collegate a più identità). Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta [Scegliere tra policy gestite e policy in linea](#) nella Guida per l'utente di IAM.

Policy basate sulle risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi includono le policy di trust dei ruoli IAM e le policy dei bucket di Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. In una policy basata sulle risorse è obbligatorio [specificare un'entità principale](#).

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non è possibile utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi che possono impostare le autorizzazioni massime concesse dai tipi di policy più comuni:

- Limiti delle autorizzazioni: imposta il numero massimo di autorizzazioni che una policy basata su identità ha la possibilità di concedere a un'entità IAM. Per ulteriori informazioni, consulta [Limiti delle autorizzazioni per le entità IAM](#) nella Guida per l'utente di IAM.
- Politiche di controllo del servizio (SCPs): specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa in AWS Organizations. Per ulteriori informazioni, consultare [Policy di controllo dei servizi](#) nella Guida per l'utente di AWS Organizations.
- Politiche di controllo delle risorse (RCPs): imposta le autorizzazioni massime disponibili per le risorse nei tuoi account. Per ulteriori informazioni, consulta [Politiche di controllo delle risorse \(RCPs\)](#) nella Guida per l'AWS Organizations utente.

- Policy di sessione: policy avanzate passate come parametro quando si crea una sessione temporanea per un ruolo o un utente federato. Per maggiori informazioni, consultare [Policy di sessione](#) nella Guida per l'utente IAM.

Più tipi di policy

Quando a una richiesta si applicano più tipi di policy, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire o meno una richiesta quando sono coinvolti più tipi di policy, consulta [Logica di valutazione delle policy](#) nella IAM User Guide.

In che modo Amazon Managed Blockchain (AMB) Query funziona con IAM

Prima di utilizzare IAM per gestire l'accesso a AMB Query, scopri quali funzionalità IAM sono disponibili per l'uso con AMB Query.

Funzionalità IAM che puoi utilizzare con Amazon Managed Blockchain (AMB) Query

Funzionalità IAM	Supporto AMB Query
Policy basate sull'identità	Sì
Policy basate su risorse	No
Operazioni di policy	Sì
Risorse relative alle policy	No
Chiavi di condizione delle policy	No
ACLs	No
ABAC (tag nelle policy)	No
Credenziali temporanee	Sì
Autorizzazioni del principale	Sì
Ruoli di servizio	No
Ruoli collegati al servizio	No

Per avere una visione di alto livello di come AMB Query e altri AWS servizi funzionano con la maggior parte delle funzionalità IAM, consulta [AWS i servizi che funzionano con IAM nella IAM User Guide](#).

Politiche basate sull'identità per AMB Query

Supporta le policy basate sull'identità: sì

Le policy basate sull'identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta [Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente](#) nella Guida per l'utente di IAM.

Con le policy basate sull'identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta [Guida di riferimento agli elementi delle policy JSON IAM](#) nella Guida per l'utente IAM.

Esempi di politiche basate sull'identità per AMB Query

Per visualizzare esempi di politiche basate sull'identità di AMB Query, vedere. [Esempi di policy basate sull'identità per Amazon Managed Blockchain \(AMB\) Query](#)

Politiche basate sulle risorse all'interno di AMB Query

Supporta le policy basate su risorse: no

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli IAM e le policy di bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. In una policy basata sulle risorse è obbligatorio [specificare un'entità principale](#). I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso multi-account, è possibile specificare un intero account o entità IAM in un altro account come entità principale in una policy basata sulle risorse. Per ulteriori informazioni, consulta [Accesso a risorse multi-account in IAM](#) nella Guida per l'utente IAM.

Azioni politiche per AMB Query

Supporta le operazioni di policy: si

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso in una policy. Includere le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco di azioni di query AMB, consulta [Actions Defined by Amazon Managed Blockchain \(AMB\) Query](#) nel Service Authorization Reference.

Le azioni politiche in AMB Query utilizzano il seguente prefisso prima dell'azione:

```
managedblockchain-query:
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

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

Per visualizzare esempi di politiche basate sull'identità di AMB Query, vedere. [Esempi di policy basate sull'identità per Amazon Managed Blockchain \(AMB\) Query](#)

Risorse politiche per AMB Query

Supporta le risorse di policy: No

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento JSON `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Come best practice, specifica una risorsa utilizzando il suo [nome della risorsa Amazon](#)

[\(ARN\)](#). Per le azioni che non supportano le autorizzazioni a livello di risorsa, si utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

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

Per visualizzare un elenco dei tipi di risorse AMB Query e relativi ARNs, consulta [Resources Defined by Amazon Managed Blockchain \(AMB\) Query](#) nel Service Authorization Reference. Per sapere con quali azioni puoi specificare l'ARN di ogni risorsa, consulta [Actions Defined by Amazon Managed Blockchain \(AMB\) Query](#).

Per visualizzare esempi di politiche basate sull'identità di AMB Query, consulta [Esempi di policy basate sull'identità per Amazon Managed Blockchain \(AMB\) Query](#)

Chiavi relative alle condizioni delle policy per AMB Query

Supporta le chiavi di condizione delle policy specifiche del servizio: No

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento Condition specifica quando le istruzioni vengono eseguite in base a criteri definiti. È possibile compilare espressioni condizionali che utilizzano [operatori di condizione](#), ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di [contesto delle condizioni AWS globali nella Guida](#) per l'utente IAM.

Per visualizzare un elenco delle chiavi di condizione di AMB Query, consulta [Condition Keys for Amazon Managed Blockchain \(AMB\) Query](#) nel Service Authorization Reference. Per sapere con quali azioni e risorse puoi utilizzare una chiave di condizione, consulta [Actions Defined by Amazon Managed Blockchain \(AMB\) Query](#).

Per visualizzare esempi di politiche basate sull'identità di AMB Query, consulta [Esempi di policy basate sull'identità per Amazon Managed Blockchain \(AMB\) Query](#)

ACLs in AMB Query

Supporti ACLs: no

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

ABAC con AMB Query

Supporta ABAC (tag nelle policy): No

Il controllo degli accessi basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base ad attributi chiamati tag. È possibile allegare tag a entità e AWS risorse IAM, quindi progettare politiche ABAC per consentire le operazioni quando il tag del principale corrisponde al tag sulla risorsa.

Per controllare l'accesso basato su tag, fornire informazioni sui tag nell'[elemento condizione](#) di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Sì. Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per maggiori informazioni su ABAC, consulta [Definizione delle autorizzazioni con autorizzazione ABAC](#) nella Guida per l'utente di IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta [Utilizzo del controllo degli accessi basato su attributi \(ABAC\)](#) nella Guida per l'utente di IAM.

Utilizzo di credenziali temporanee con AMB Query

Supporta le credenziali temporanee: sì

Le credenziali temporanee forniscono un accesso a breve termine alle AWS risorse e vengono create automaticamente quando si utilizza la federazione o si cambia ruolo. AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta [Credenziali di sicurezza temporanee in IAM](#) e [Servizi AWS compatibili con IAM](#) nella Guida per l'utente IAM.

Autorizzazioni principali multiservizio per AMB Query

Supporta l'inoltro delle sessioni di accesso (FAS): sì

Le sessioni di accesso inoltrato (FAS) utilizzano le autorizzazioni del principale che chiama un Servizio AWS, combinate con la richiesta di effettuare richieste Servizio AWS ai servizi downstream. Per i dettagli delle policy relative alle richieste FAS, consulta [Forward access sessions](#).

Ruoli di servizio per AMB Query

Supporta i ruoli di servizio: no

Un ruolo di servizio è un [ruolo IAM](#) che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta [Create a role to delegate permissions to an Servizio AWS](#) nella Guida per l'utente IAM.

Warning

La modifica delle autorizzazioni per un ruolo di servizio potrebbe interrompere la funzionalità di AMB Query. Modifica i ruoli di servizio solo quando AMB Query fornisce indicazioni in tal senso.

Ruoli collegati ai servizi per AMB Query

Supporta i ruoli collegati ai servizi: no

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un Servizio AWS. Il servizio può assumere il ruolo per eseguire un'operazione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati al servizio, ma non modificarle.

Per ulteriori informazioni su come creare e gestire i ruoli collegati ai servizi, consulta [Servizi AWS supportati da IAM](#). Trova un servizio nella tabella che include un Yes nella colonna Service-linked role (Ruolo collegato ai servizi). Scegli il collegamento Sì per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Esempi di policy basate sull'identità per Amazon Managed Blockchain (AMB) Query

Per impostazione predefinita, gli utenti e i ruoli non sono autorizzati a creare o modificare risorse AMB Query. Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta [Creazione di policy IAM \(console\)](#) nella Guida per l'utente di IAM.

Per dettagli sulle azioni e sui tipi di risorse definiti da AMB Query, incluso il formato di ARNs per ogni tipo di risorsa, consulta [Actions, Resources and Condition Keys for Amazon Managed Blockchain \(AMB\) Query](#) nel Service Authorization Reference.

Argomenti

- [Best practice per le policy](#)
- [Consentire agli utenti di visualizzare le loro autorizzazioni](#)
- [Accesso a specifiche azioni dell'API Amazon Managed Blockchain \(AMB\) Query](#)

Best practice per le policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare le risorse AMB Query nel tuo account. Queste azioni possono comportare costi aggiuntivi per l'Account AWS. Quando si creano o modificano policy basate sull'identità, seguire queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le policy gestite che concedono le autorizzazioni per molti casi d'uso comuni. AWS Sono disponibili nel tuo Account AWS. Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per maggiori informazioni, consulta [Policy gestite da AWS](#) o [Policy gestite da AWS per le funzioni dei processi](#) nella Guida per l'utente di IAM.
- Applicazione delle autorizzazioni con privilegio minimo - Quando si impostano le autorizzazioni con le policy IAM, concedere solo le autorizzazioni richieste per eseguire un'attività. È possibile farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegio minimo. Per maggiori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta [Policy e autorizzazioni in IAM](#) nella Guida per l'utente di IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso - Per limitare l'accesso ad azioni e risorse è possibile aggiungere una condizione alle policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio CloudFormation. Per

maggiori informazioni, consultare la sezione [Elementi delle policy JSON di IAM: condizione](#) nella Guida per l'utente di IAM.

- Utilizzo dello strumento di analisi degli accessi IAM per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali - Lo strumento di analisi degli accessi IAM convalida le policy nuove ed esistenti in modo che aderiscano al linguaggio (JSON) della policy IAM e alle best practice di IAM. Lo strumento di analisi degli accessi IAM offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per maggiori informazioni, consultare [Convalida delle policy per il Sistema di analisi degli accessi IAM](#) nella Guida per l'utente di IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungere le condizioni MFA alle policy. Per maggiori informazioni, consultare [Protezione dell'accesso API con MFA](#) nella Guida per l'utente di IAM.

Per maggiori informazioni sulle best practice in IAM, consulta [Best practice di sicurezza in IAM](#) nella Guida per l'utente di IAM.

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono collegate alla relativa identità utente.

Questa policy include le autorizzazioni per completare questa azione sulla console o utilizzando programmaticamente l'API o. AWS CLI AWS

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "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": "*"
}
]
}

```

Accesso a specifiche azioni dell'API Amazon Managed Blockchain (AMB) Query

Note

Per accedere ad AMB Query ed effettuare chiamate API, sono necessarie credenziali utente (AWS_ACCESS_KEY_ID e AWS_SECRET_ACCESS_KEY) che dispongano delle autorizzazioni IAM appropriate per AMB Query.

Example Policy IAM per accedere a tutte le query di Amazon Managed Blockchain (AMB) APIs

Questo esempio concede a un utente IAM l' Account AWS accesso a tutte le AMB Query. APIs

JSON

```

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

```

```

    "Resource": "*"
  }
]
}

```

Example Policy IAM per accedere ad Amazon Managed Blockchain (AMB) Query **ListTransactions** e **GetTransaction** APIs

Questo esempio concede a un utente IAM il tuo Account AWS accesso ad AMB Query e ListTransaction GetTransaction APIs

Note

È possibile sostituire o aggiungere quello dell' APIs esempio con altro APIs per consentire l'accesso ad altri o più. APIs Per un elenco di AMB Query APIs, consulta la 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": "*"
    }
  ]
}

```

Risoluzione dei problemi relativi all'identità e all'accesso ad Amazon Managed Blockchain (AMB) Query

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con AMB Query e IAM.

Argomenti

- [Non sono autorizzato a eseguire un'azione in AMB Query](#)

Non sono autorizzato a eseguire un'azione in AMB Query

Se ricevi un errore che indica che non sei autorizzato a eseguire un'operazione, le tue policy devono essere aggiornate per poter eseguire l'operazione.

L'errore di esempio seguente si verifica quando l'utente IAM mateojackson prova a utilizzare la console per visualizzare i dettagli relativi a una risorsa *my-example-widget* fittizia ma non dispone di autorizzazioni `managedblockchain-query::GetWidget` fittizie.

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

In questo caso, la policy per l'utente mateojackson deve essere aggiornata per consentire l'accesso alla risorsa *my-example-widget* utilizzando l'azione `managedblockchain-query::GetWidget`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Metriche di utilizzo dell'API Amazon Managed Blockchain (AMB) Query su Amazon CloudWatch

Metriche di utilizzo delle API su Amazon CloudWatch

I parametri di utilizzo delle API pubblicati CloudWatch corrispondono alle quote del servizio Amazon Managed Blockchain (AMB) Query. Puoi configurare allarmi per avvisarti quando il tuo utilizzo si avvicina a una quota di servizio. Per ulteriori informazioni sull' CloudWatch integrazione con le quote di servizio, consulta i [parametri di utilizzo di AWS](#) nella Amazon CloudWatch User Guide.

AMB Query pubblica le seguenti metriche API nel AWS/Usage namespace, con il nome del servizio. Amazon Managed Blockchain Query

Metrica	Description
CallCount	Il numero totale di chiamate effettuate a un'API in AMB Query. SUM rappresenta il numero totale di chiamate all'API durante il periodo specificato.

Amazon Managed Blockchain (AMB) Query pubblica i parametri di utilizzo nel AWS/Usage namespace con le seguenti dimensioni.

Dimensione	Descrizione
Servizio	Il nome del servizio che contiene la risorsa. AWS Amazon Managed Blockchain Query sarà sempre il valore di questa dimensione.
Tipo	Il tipo di entità segnalata. API sarà sempre il valore di questa dimensione.

Dimensione	Descrizione
Risorsa	Il tipo di risorse segnalate. Il nome dell' operazione AMB Query API utilizzata sarà il valore di questa dimensione.
Classe	La classe della risorsa segnalata. Non sarà sempre il valore di questa dimensione.

Cronologia dei documenti per la AMB Query User Guide

La tabella seguente descrive le versioni della documentazione per AMB Query.

Modifica	Descrizione	Data
AMB Query supporta gli identificatori e gli hash delle transazioni Bitcoin	Per le reti Bitcoin, le operazioni dell'API AMB Query supportano sia l'identificatore di transazione (<code>transactionId</code>) che l'hash della transazione (<code>transactionHash</code>).	21 marzo 2024
Support per i parametri di utilizzo delle API su Amazon CloudWatch	AMB Query ha aggiunto il supporto per le metriche di utilizzo delle API su Amazon CloudWatch. Queste metriche di utilizzo corrispondono alle quote del servizio AMB Query.	8 febbraio 2024
Support per le transazioni che non hanno raggiunto il termine	AMB Query ha aggiunto il supporto per le transazioni che non hanno raggiunto la scadenza. Rimuove inoltre il supporto per la proprietà <code>transactionStatus</code> dalla risposta dell'operazione <code>GetTransaction</code> . Utilizzare invece le proprietà <code>executionStatus</code> e <code>confirmationStatus</code> per determinare lo stato della transazione.	1 febbraio 2024
Obsolescenza della proprietà nel tipo di dati <code>Transaction</code>	Amazon Managed Blockchain (AMB) Query ha reso obsoleta la proprietà	20 dicembre 2023

nel tipo di dati `Transaction`. È necessario utilizzare i campi `executionStatus` e `confirmationStatus` e per determinare se la transazione è `status` o `FINAL FAILED`.

[Support per Sepolia Testnet](#)

Amazon Managed Blockchain (AMB) Query ora supporta le query su Ethereum Sepolia Testnet.

19 ottobre 2023

[Support per contratti patrimoniali](#)

È possibile utilizzare l'operazione [ListAssetContracts](#) API per elencare gli elenchi distribuiti da un determinato indirizzo. Inoltre, se disponi dell'indirizzo del contratto, puoi utilizzare l'operazione [GetAssetContract](#) API per recuperare i dettagli del contratto.

16 ottobre 2023

[Support per Bitcoin Testnet](#)

Amazon Managed Blockchain (AMB) Query ora supporta le query su Bitcoin Testnet.

16 ottobre 2023

[Versione iniziale](#)

Versione iniziale del servizio AMB Query.

27 luglio 2023

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.