aws

Guide du développeur

# Amazon Managed Blockchain Query

# Amazon Managed Blockchain Query: Guide du développeur

# Table of Contents

# Qu'est-ce que Amazon Managed Blockchain (AMB) Query ?

Amazon Managed Blockchain (AMB) est un service entièrement géré conçu pour vous aider à créer des applications Web3 résilientes sur des chaînes de blocs publiques et privées. Utilisez AMB Access pour un accès instantané et sans serveur à plusieurs chaînes de blocs. Créez vos applications prêtes pour le Web3 sans avoir à déployer une infrastructure blockchain spécialisée et à les maintenir connectées au réseau blockchain. Avec AMB Query, vous pouvez utiliser des opérations d'API conviviales pour les développeurs pour accéder aux données historiques et en temps réel de plusieurs chaînes de blocs. Les données de blockchain standardisées peuvent être intégrées aux services AWS, sans nécessiter d'infrastructure de blockchain spécialisée ou d'ETL (extraction, transformation et chargement). Toutes les fonctionnalités d'AMB s'adaptent en toute sécurité aux versions d'applications destinées aux institutions et aux grands consommateurs.

Amazon Managed Blockchain (AMB) Query fournit un accès sans serveur à des ensembles de données standardisés comportant plusieurs chaînes de blocs avec des opérations d'API conviviales pour les développeurs. Vous pouvez utiliser AMB Query pour expédier rapidement des applications qui nécessitent des données provenant d'une ou de plusieurs chaînes de blocs publiques, sans avoir à surcharger l'analyse des données de la chaîne de blocs, le suivi des contrats et la maintenance d'une infrastructure d'indexation spécialisée. Que vous analysiez les soldes historiques de jetons pour les jetons fongibles ou non fongibles (NFT), que vous consultiez l'historique des transactions pour une adresse de portefeuille donnée ou que vous analysiez des données sur la distribution de cryptomonnaies natives telles que l'Ether, AMB Query vous donne accès aux données de la blockchain.

## Utilisez-vous AMB Query pour la première fois ?

Si vous utilisez AMB Query pour la première fois, nous vous recommandons de commencer par lire les sections suivantes :

- Concepts clés : requête Amazon Managed Blockchain (AMB)
- Configuration de la requête Amazon Managed Blockchain (AMB)
- Commencer à utiliser Amazon Managed Blockchain (AMB) Query
- Cas d'utilisation avec Amazon Managed Blockchain (AMB) Query

# Concepts clés : requête Amazon Managed Blockchain (AMB)

> **Note**
>
> Ce guide part du principe que vous connaissez les concepts essentiels de la blockchain. Ces concepts incluent la décentralisation, les jetons, les contrats, les transactions, les portefeuilles proof-of-work, les clés publiques et privées, le staking, le minage, la réduction de moitié, etc.

Amazon Managed Blockchain (AMB) Query vous permet d'accéder facilement aux données du réseau multi-chaînes de blocs, ce qui vous permet d'extraire plus facilement des données contextuelles liées à l'activité de la blockchain. Vous pouvez utiliser AMB Query pour lire les données des réseaux de blockchain publics, tels que Bitcoin Mainnet et Ethereum Mainnet. Vous pouvez également obtenir des informations, telles que les soldes actuels et historiques des adresses, ou vous pouvez obtenir une liste des transactions de blockchain pour une période donnée. En outre, vous pouvez obtenir les détails d'une transaction donnée, tels que les événements de transaction, que vous pouvez analyser plus en détail ou utiliser dans la logique métier de vos applications.

## Considérations et limites relatives à l'utilisation de la requête Amazon Managed Blockchain (AMB)

Lorsque vous utilisez AMB Query, tenez compte des points suivants :

- Régions disponibles

  La requête AMB est prise en charge dans la `us-east-1` région USA Est (Virginie du Nord).

- Points de terminaison de service

  AMB Query est accessible à l'aide du point de terminaison suivant :

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

- Réseaux de blockchain pris en charge

  AMB Query prend en charge les réseaux de blockchain publics suivants :

- Bitcoin Mainnet — Le réseau public de blockchain Bitcoin sécurisé par proof-of-work consensus et sur lequel la cryptomonnaie Bitcoin (BTC) est émise et échangée. Les transactions sur Mainnet ont une valeur réelle (c'est-à-dire qu'elles entraînent des coûts réels) et sont enregistrées sur la blockchain publique.
- Bitcoin Testnet — Le réseau de test pour le réseau principal Bitcoin. Le Bitcoin (BTC) sur ce réseau est séparé et distinct du BTC sur le réseau principal et n'a généralement aucune valeur.
- Ethereum Mainnet — Le réseau proof-of-stake principal de la blockchain publique Ethereum. Les transactions sur Mainnet ont une valeur réelle (c'est-à-dire qu'elles entraînent des coûts réels) et sont enregistrées dans le registre distribué.
- Sepolia Testnet — Le réseau de test pour le réseau principal Ethereum. L'éther (ETH) sur ce réseau est séparé et distinct de l'ETH sur le réseau principal et n'a généralement aucune valeur.

- Tokens et contrats de blockchain pris en charge

  AMB Query prend en charge les jetons de contrat Ethereum natifs et standard suivants.

  - Jetons natifs de la blockchain publique
    - Bitcoin (BTC) — Il s'agit du jeton natif des blockchains liées au Bitcoin.
    - Ether (ETH) — Il s'agit du jeton natif des blockchains liées à Ethereum.

  - Normes des contrats Ethereum
    - Norme de jeton ERC-20 — L'ERC-20 est une norme pour les jetons fongibles. Il possède une propriété qui rend chaque jeton ERC-20 exactement identique (en type et en valeur) à un autre jeton ERC-20 émis, ce qui signifie qu'un jeton est et sera toujours égal à tous les autres jetons. Pour plus d'informations, consultez le standard de jeton ERC-20 sur Ethereum.org.
    - Norme de jeton non fongible ERC-721 — L'ERC-721 est une norme pour les jetons non fongibles (NFT). Ce type de jeton est unique et peut avoir une valeur différente de celle d'un autre jeton issu du même contrat, peut-être en raison de son âge, de sa rareté ou d'autres propriétés. Pour plus d'informations, consultez le standard de jeton ERC-721 sur Ethereum.org.

      Norme multi-jetons ERC-1155 — L'ERC-1155 est une norme qui crée une interface de contrat capable de représenter et de contrôler un certain nombre de types de jetons fongibles et non fongibles. De cette façon, le jeton ERC-1155 peut fonctionner de la même manière que les jetons ERC-20 et ERC-721, voire fonctionner comme les deux en même temps. Le jeton ERC-1155 améliore les fonctionnalités des normes ERC-20 et ERC-721, le rendant ainsi plus

efficace, tout en corrigeant les erreurs de mise en œuvre évidentes. Pour plus d'informations, consultez le standard de jeton ERC-1155 sur Ethereum.org.

- Finalité

  Dans les blockchains, la finalité signifie qu'il est peu probable que les transactions valides soient annulées. Pour le réseau principal Bitcoin, AMB Query considère qu'une transaction est définitive après 6 blocs. Pour le Bitcoin Testnet, il considère qu'une transaction est définitive après 6 blocs ou 60 minutes, selon la première éventualité. Pour les réseaux Ethereum pris en charge, AMB Query considère qu'une transaction est définitive après 64 blocs.

  Les opérations d'API relatives au solde des jetons et aux contrats d'AMB Query ne renvoient que des données ayant atteint la finalité. Cependant, les opérations de l'API de transaction et d'événement transactionnel d'AMB Query peuvent renvoyer des données pour les transactions confirmées sur le réseau blockchain même si elles n'ont pas encore atteint leur finalité.

- Adresse NULL non prise en charge

  AMB Query ne prend pas en charge l'adresse NULL (`0x0000000000000000000000000000000000000000`).

- Signature Version 4 : signature des appels d'API

  Lorsque vous appelez les API de requête AMB, vous pouvez le faire via une connexion HTTPS authentifiée à l'aide du processus de signature Signature Version 4. Cela signifie que seuls les principaux IAM autorisés du AWS compte peuvent effectuer des appels à l'API AMB Query. Pour ce faire, des AWS informations d'identification (un identifiant de clé d'accès et une clé d'accès secrète) doivent être fournies avec l'appel.

  > **Important**
  >
  > N'intégrez pas les informations d'identification du client dans les applications destinées aux utilisateurs.

- AMB Query prend en charge les identifiants de transaction Bitcoin et les hachages de transactions

  Pour les réseaux Bitcoin, les opérations de l'API AMB Query prennent en charge à la fois l'identifiant de transaction (`transactionId`) et le hachage de transaction (`transactionHash`).

`transactionId`Il s'agit d'un hachage double SHA de la transaction, sans inclure les données des témoins. `transactionHash`Il s'agit d'un hachage double SHA de la transaction, y compris les données du témoin (également connu sous le nom d'identifiant de transaction témoin).

Lorsque vous invoquez les opérations de `ListTransactionEvents`l'API `GetTransaction`or pour les réseaux Bitcoin, vous pouvez spécifier le `transactionId` ou le`transactionHash`. De plus, toutes les opérations de requête AMB sur les réseaux Bitcoin qui renvoient a `transactionId` ou a `transactionHash` incluront les deux valeurs dans la réponse.

# Configuration de la requête Amazon Managed Blockchain (AMB)

Avant d'utiliser Amazon Managed Blockchain (AMB) Query pour la première fois, suivez les étapes décrites dans cette section pour créer un AWS . La section suivante explique comment commencer à utiliser AMB Query.

## Prérequis et considérations

Avant d'utiliser Amazon Web Services pour la première fois, vous devez disposer d'un AWS .

### Inscrivez-vous pour AWS

Lorsque vous vous inscrivez à Amazon Web Services (AWS), votre AWS le compte est automatiquement ouvert pour tous Services AWS, y compris Amazon Managed Blockchain (AMB) Query. Seuls les services que vous utilisez vous sont facturés.

Si vous avez un Compte AWS déjà, passez à l'étape suivante. Si vous n'avez pas de Compte AWS, utilisez la procédure suivante pour en créer un.

Pour créer un AWS compte

1. Ouvrez l'https://portal.aws.amazon.com/billing/inscription.
2. Suivez les instructions en ligne.

   Dans le cadre de la procédure d‘inscription, vous recevrez un appel téléphonique et vous saisirez un code de vérification en utilisant le clavier numérique du téléphone.

   Lorsque vous vous inscrivez à un Compte AWS, une Utilisateur racine d'un compte AWSest créé. L'utilisateur root a accès à tous Services AWS et les ressources du compte. La meilleure pratique de sécurité consiste à attribuer un accès administratif à un utilisateur, et à utiliser uniquement l‘utilisateur racine pour effectuer les tâches nécessitant un accès utilisateur racine.

# Création d'un IAM utilisateur avec les autorisations appropriées

Pour créer et utiliser AMB Query, vous devez créer un AWS Identity and Access Management (IAM) principal (utilisateur ou groupe) doté d'autorisations autorisant les actions nécessaires à Managed Blockchain.

Seuls IAM les principaux peuvent effectuer des API demandes de AMB requête. Lorsque vous appelez la AMB requêteAPIs, vous pouvez le faire via une HTTPS connexion authentifiée à l'aide du processus de signature Signature version 4. Cela signifie que seuls IAM les directeurs autorisés du AWS le compte peut effectuer des API appels AMB Query. Pour ce faire, AWS des informations d'identification (un identifiant de clé d'accès et une clé d'accès secrète) doivent être fournies avec l'appel.

Pour plus d'informations sur la création d'un IAM utilisateur, voir Création d'un IAM utilisateur dans votre AWS compte. Pour plus d'informations sur la façon d'associer une politique d'autorisations à un utilisateur, consultez la section Modification des autorisations d'un IAM utilisateur. Pour un exemple de politique d'autorisation que vous pouvez utiliser pour autoriser un utilisateur à utiliser AMB Query, consultezExemples de politiques basées sur l'identité pour Amazon Managed Blockchain () AMB Query.

# Installez et configurez AWS Command Line Interface

Si ce n'est pas déjà fait, installez le dernier AWS Interface de ligne de commande (CLI) avec laquelle travailler AWS ressources provenant d'un terminal. Pour plus d'informations, voir Installation ou mise à jour de la dernière version du AWS CLI.

> **Note**
>
> Pour CLI y accéder, vous avez besoin d'un identifiant de clé d'accès et d'une clé d'accès secrète. Utilisation des informations d'identification temporaires au lieu des clés d'accès à long terme si possible. Les informations d'identification temporaires incluent un ID de clé d'accès, une clé d'accès secrète et un jeton de sécurité qui indique la date d'expiration des informations d'identification. Pour plus d'informations, voir Utilisation d'informations d'identification temporaires avec AWS ressources du guide de IAM l'utilisateur.

# Utilisez la commande AWS Management Console pour interroger des chaînes de blocs à l'aide d'Amazon Managed Blockchain (AMB) Query

Vous pouvez accéder à Amazon Managed Blockchain (AMB) Query et effectuer des requêtes sur les réseaux de blockchain pris en charge à l'aide du AWS Management Console. Les étapes suivantes indiquent comment procéder :

1. Ouvrez la console Amazon Managed Blockchain à l'adresse https://console.aws.amazon.com/managedblockchain/.
2. Choisissez l'éditeur de requête dans la section Requête.
3. Choisissez parmi l'un des réseaux Blockchain pris en charge.
4. Choisissez le type de requête que vous souhaitez exécuter.
5. Entrez les paramètres appropriés pour le type de requête que vous avez sélectionné et exécutez la requête.

AMBQuery exécutera votre requête et vous verrez les résultats dans la fenêtre des résultats de la requête.

# Commencer à utiliser Amazon Managed Blockchain (AMB) Query

Utilisez les step-by-step didacticiels de cette section pour apprendre à effectuer des tâches à l'aide d'Amazon Managed Blockchain (AMB) Query. Ces procédures nécessitent un certain nombre de prérequis. Si vous utilisez AMB Query pour la première fois, vous pouvez consulter la section Configuration de ce guide. Pour de plus amples informations, veuillez consulter Configuration de la requête Amazon Managed Blockchain (AMB).

> **Note**
>
> Certaines variables de ces exemples ont été délibérément masquées. Remplacez-les par vos propres modèles valides avant d'exécuter ces exemples.

Rubriques

- Création d'une IAM politique pour accéder aux API opérations de AMB requête
- Effectuez des API demandes de requête Amazon Managed Blockchain (AMB) à l'aide de Go
- Effectuez des API demandes de requête Amazon Managed Blockchain (AMB) à l'aide du fichier Node.js
- Effectuer des API requêtes Amazon Managed Blockchain (AMB) à l'aide de Python
- Utilisez la requête Amazon Managed Blockchain (AMB) sur AWS Management Console pour exécuter l' GetTokenBalance opération

## Création d'une IAM politique pour accéder aux API opérations de AMB requête

Pour effectuer des API demandes de AMB requête, vous devez utiliser les informations d'identification utilisateur (AWS_ACCESS_ KEY _ID et AWS_SECRET _ ACCESS _KEY) disposant des IAM autorisations appropriées pour Amazon Managed Blockchain (AMB) Query. Dans un terminal doté du AWS CLI installé, exécutez la commande suivante pour créer une IAM politique d'accès aux API opérations de AMB requête :

```
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
```

Après avoir créé la politique, associez-la au rôle IAM d'un utilisateur pour qu'elle prenne effet. Dans le volet AWS Management Console, accédez au IAM service et associez la politique `AmazonManagedBlockchainQueryAccess` au rôle attribué à l'IAMutilisateur qui utilisera le service. Pour plus d'informations, consultez Création d'un rôle et attribution à un IAM utilisateur.

> **Note**
>
> AWS vous recommande de donner accès à des API opérations spécifiques plutôt que d'utiliser le joker*. Pour de plus amples informations, veuillez consulter Accès à des API actions de requête Amazon Managed Blockchain (AMB) spécifiques.

# Effectuez des API demandes de requête Amazon Managed Blockchain (AMB) à l'aide de Go

Avec Amazon Managed Blockchain (AMB) Query, vous pouvez créer des applications qui dépendent d'un accès instantané aux données de la blockchain une fois qu'elles sont confirmées sur la blockchain, même si elles n'ont pas encore atteint leur finalité. AMBQuery permet plusieurs cas d'utilisation, tels que le remplissage de l'historique des transactions d'un portefeuille, la fourniture d'informations contextuelles sur une transaction en fonction de son hachage de transaction ou l'obtention du solde d'un jeton natif ainsi que de ERC -721, ERC -1155 et -20 jetons. ERC

Les exemples suivants sont créés dans le langage Go et utilisent les API opérations de AMB requête. Pour plus d'informations sur Go, consultez la documentation Go. Pour plus d'informations sur la AMB requêteAPI, consultez la documentation de API référence des requêtes Amazon Managed Blockchain (AMB).

Les exemples suivants utilisent les `GetTransaction` API actions `ListTransactions` et pour obtenir d'abord une liste de toutes les transactions pour une adresse externe donnée (EOA) sur le réseau principal Ethereum, puis l'exemple suivant récupère les détails des transactions pour une seule transaction dans la liste.

Example — Effectuez l'**`ListTransactions`**APIaction en utilisant Go

Copiez le code suivant dans un fichier nommé `listTransactions.go` dans le ListTransactionsrépertoire.

```
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)
    }
}
```

Après avoir enregistré le fichier, exécutez le code en utilisant la commande suivante dans le ListTransactionsrépertoire :`go run listTransactions.go`.

Le résultat qui suit ressemble à ce qui suit :

```
{
  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 — Effectuez l'**`GetTransaction`**APIaction en utilisant Go

Cet exemple utilise un hachage de transaction issu de la sortie précédente. Copiez le code suivant dans un fichier nommé `GetTransaction.go` dans le GetTransactionrépertoire.

```
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)
    }
}
```

Après avoir enregistré le fichier, exécutez le code en utilisant la commande suivante dans le GetTransactionrépertoire :`go run GetTransaction.go`.

Le résultat qui suit ressemble à ce qui suit :

```
{
  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
  }
}
```

Cela vous `GetTokenBalance` API permet d'obtenir le solde des jetons natifs (ETHetBTC), qui peuvent être utilisés pour obtenir le solde actuel d'un compte externe (EOA) à un moment donné.

Example — Utilisez l'**`GetTokenBalance`**APIaction pour obtenir le solde d'un jeton natif dans Go

Dans l'exemple suivant, vous utilisez le `GetTokenBalance` API pour obtenir le solde d'une adresse Ether (ETH) sur le réseau principal Ethereum. Copiez le code suivant dans un fichier nommé `GetTokenBalanceEth.go` dans le GetTokenBalancerépertoire.

```
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)
    }
}
```

Après avoir enregistré le fichier, exécutez le code en utilisant la commande suivante dans le GetTokenBalancerépertoire :`go run GetTokenBalanceEth.go`.

Le résultat qui suit ressemble à ce qui suit :

```
{
  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"
  }
}
```

# Effectuez des API demandes de requête Amazon Managed Blockchain (AMB) à l'aide du fichier Node.js

Pour exécuter ces exemples de nœuds, les conditions préalables suivantes s'appliquent :

1. Le gestionnaire de version de nœud (nvm) et Node.js doivent être installés sur votre machine. Vous trouverez les instructions d'installation pour votre système d'exploitation ici.
2. Utilisez la `node --version` commande et confirmez que vous utilisez la version 14 ou supérieure de Node. Si nécessaire, vous pouvez utiliser la `nvm install 14` commande, puis la `nvm use 14` commande pour installer la version 14.
3. Les variables `AWS_ACCESS_KEY_ID` d'environnement `AWS_SECRET_ACCESS_KEY` doivent contenir les informations d'identification associées au compte.

   Exportez ces variables sous forme de chaînes sur votre client à l'aide des commandes suivantes. Remplacez les valeurs surlignées ci-dessous par les valeurs appropriées du compte IAM utilisateur.

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

> **Note**
>
> - Une fois toutes les conditions requises remplies, vous pouvez envoyer des demandes signées HTTPS pour accéder aux API opérations de requête Amazon Managed Blockchain (AMB) et effectuer des demandes à l'aide du module https natif dans Node.js, ou vous pouvez utiliser une bibliothèque tierce telle que Query AXIOSet récupérer des données depuis AMB Query.
> - Ces exemples utilisent un HTTP client tiers pour Node.js, mais vous pouvez également utiliser AWS JavaScript SDKpour faire des demandes à AMB Query.
> - L'exemple suivant vous montre comment effectuer des API requêtes AMB Query à l'aide d'Axios et du AWS SDKmodules pour SigV4.

Copiez le `package.json` fichier suivant dans le répertoire de travail de votre environnement local :

```
{
  "name": "amb-query-examples",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "@aws-crypto/sha256-js": "^4.0.0",
    "@aws-sdk/credential-provider-node": "^3.360.0",
    "@aws-sdk/protocol-http": "^3.357.0",
    "@aws-sdk/signature-v4": "^3.357.0",
    "axios": "^1.4.0"
  }
}
```

Example — Récupérez le solde historique des jetons à partir d'une adresse externe spécifique (EOA) à l'aide de AMB Query **`GetTokenBalance`** API

Vous pouvez utiliser le `GetTokenBalance` API pour obtenir le solde de différents jetons (par exemple, ERC2 0ERC721, etERC1155) et de pièces natives (par exemple, ETH etBTC), que vous pouvez utiliser pour obtenir le solde actuel d'un compte externe (EOA) sur la base d'un historique `timestamp` (horodatage Unix - secondes). Dans cet exemple, vous utilisez le `GetTokenBalance`APIpour obtenir un solde d'adresses d'un jeton ERC2 0USDC, sur le réseau principal Ethereum.

Pour tester le `GetTokenBalance`API, copiez le code suivant dans un fichier nommé `token-balance.js` et enregistrez le fichier dans le même répertoire de travail :

```
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);
```

Pour exécuter le code, ouvrez un terminal dans le même répertoire que vos fichiers et exécutez la commande suivante :

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

Cette commande exécute le script en transmettant les arguments définis dans le code pour demander le USDC solde ERC2 nul des valeurs EOA répertoriées sur le réseau principal Ethereum. La réponse est similaire à ce qui suit :

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

# Effectuer des API requêtes Amazon Managed Blockchain (AMB) à l'aide de Python

Pour exécuter ces exemples Python, les conditions préalables suivantes s'appliquent :

1. Python doit être installé sur votre machine. Vous trouverez les instructions d'installation pour votre système d'exploitation ici.
2. Installez le AWSSDKpour Python (Boto3).
3. Installer le AWS Interface de ligne de commande et exécutez la commande `aws configure` pour définir les variables de votre `Access Key IDSecret Access Key`, et`Region`.

Après avoir rempli tous les prérequis, vous pouvez utiliser le AWS SDKpour que Python envoie HTTPS des API requêtes Amazon Managed Blockchain (AMB).

L'exemple Python suivant utilise des modules de boto3 pour envoyer des requêtes associées aux en-têtes SigV4 requis à l'opération Query. AMB `ListTransactionEvents` API Cet exemple permet de récupérer une liste d'événements émis par une transaction donnée sur le réseau principal Ethereum.

Copiez le `list-transaction-events.py` fichier suivant dans le répertoire de travail de votre environnement local :

```
import json
from botocore.auth import SigV4Auth
from botocore.awsrequest import AWSRequest
from botocore.session import Session
from botocore.httpsession import URLLib3Session

def signed_request(url, method,  params, service, region):

    session = Session()
    sigv4 = SigV4Auth(session.get_credentials(), service, region)
    data = json.dumps(params)
    request = AWSRequest(method, url, data=data)
    sigv4.add_auth(request)
    http_session = URLLib3Session()
    response = http_session.send(request.prepare())

    return(response)
```

```
url = 'https://managedblockchain-query.us-east-1.amazonaws.com/list-transaction-events'
method = 'POST'
params = {
'network': 'ETHEREUM_MAINNET',
'transactionHash': '0x125714bb4db48757007fff2671b37637bbfd6d47b3a4757ebbd0c5222984f905'
}
service = 'managedblockchain-query'
region = 'us-east-1'

# Call the listTransactionEvents operation. This operation will return transaction
 details for
# all transactions that are con#rmed on the blockchain, even if they have not reached
# #nality.
listTransactionEvents = signed_request(url, method, params, service, region)

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

Pour exécuter l'exemple de code sur`ListTransactionEvents`, enregistrez le fichier dans votre répertoire de travail, puis exécutez la commande`python3 list-transaction-events.py`. Cette commande exécute le script en transmettant les arguments définis dans le code pour demander les événements associés au hachage de transaction donné sur le réseau principal Ethereum. La réponse est similaire à ce qui suit :

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

# Utilisez la requête Amazon Managed Blockchain (AMB) sur AWS Management Console pour exécuter l' GetTokenBalance opération

L'exemple suivant montre comment obtenir le solde d'un jeton sur le réseau principal Ethereum à l'aide de la requête Amazon Managed Blockchain (AMB) sur le AWS Management Console

Example

1. Ouvrez la console Amazon Managed Blockchain à l'adresse https://console.aws.amazon.com/managedblockchain/.
2. Choisissez l'éditeur de requête dans la section Requête.
3. Choisissez ETHEREUM_ MAINNET comme réseau Blockchain.
4. Choisissez GetTokenBalancecomme type de requête.
5. Entrez votre adresse Blockchain pour le jeton.
6. Entrez l'adresse du contrat pour le jeton.
7. Entrez l'ID de jeton facultatif pour le jeton.
8. Choisissez la date limite pour le solde du jeton.
9. Entrez l'option À l'heure pour le solde du jeton.
10. Choisissez Exécuter la requête.

AMBQuery exécutera votre requête et vous verrez les résultats dans la fenêtre des résultats de la requête.

# Cas d'utilisation avec Amazon Managed Blockchain (AMB) Query

Cette rubrique fournit une liste des cas d'utilisation des requêtes AMB.

Rubriques



## Consulter les soldes de jetons actuels et historiques

L'`GetTokenBalance`API obtient le solde des jetons pris en charge (ERC20, ERC721, ERC1155) et des pièces natives (ETH, BTC) pour obtenir le solde actuel ou historique en utilisant un horodatage universel (horodatage Unix, en secondes) des comptes externes (EOA). Par exemple, vous pouvez utiliser l'opération `GetTokenBalance` API pour obtenir le solde d'adresses du jeton ERC20, USDC, sur le réseau principal Ethereum. Vous pouvez également récupérer par lots les soldes de jetons et de pièces natives à l'aide de l'opération `BatchGetTokenBalance` API.

Pour plus d'informations, consultez le guide de référence des requêtes Amazon Managed Blockchain (AMB).

## Récupérez les données historiques des transactions

Avec Amazon Managed Blockchain (AMB) Query, vous pouvez récupérer des données historiques à partir de chaînes de blocs publiques telles que Ethereum et Bitcoin. Cette fonctionnalité permet plusieurs cas d'utilisation, tels que la récupération de l'historique des transactions sur un portefeuille blockchain ou la fourniture d'informations contextuelles sur une transaction en fonction de son hachage de transaction. Vous pouvez utiliser l'opération `ListTransactions`API pour obtenir une liste des transactions pour une adresse externe donnée (EOA) sur le réseau principal Ethereum, puis

vous pouvez utiliser l'opération `GetTransaction`API pour récupérer les détails des transactions pour une seule transaction dans la liste.

Pour plus d'informations, consultez le guide de référence des requêtes Amazon Managed Blockchain (AMB).

## Obtenez tous les soldes de jetons pour une adresse donnée

Vous pouvez utiliser l'opération `ListTokenBalances`API pour obtenir des soldes sur les portefeuilles, les interfaces utilisateur, les utilitaires Web3, etc. Cette opération d'API renvoie une liste de tous les soldes d'une adresse entre les jetons (ERC20, ERC721, ERC1155) et les pièces natives (ETH, BTC) sur une blockchain publique donnée en utilisant une seule opération d'API. Par exemple, vous pouvez fournir une adresse externe (EOA) et un réseau (le réseau principal Ethereum), et vous pouvez recevoir une liste de jetons et de soldes de pièces natifs dans la réponse.

Pour plus d'informations, consultez le guide de référence des requêtes Amazon Managed Blockchain (AMB).

## Lister les événements émis pour une transaction

Vous pouvez utiliser l'opération `ListTransactionEvents`API pour récupérer une liste des événements de contrat émis à la suite d'une transaction donnée, identifiés par son hachage (identifiant de transaction). Par exemple, vous pouvez l'utiliser `ListTransactionEvents`pour récupérer les événements résultants d'une transaction qui appelle une fonction d'un contrat de jeton ERC20 sur la blockchain Ethereum, comme un événement de transfert ou un événement de retrait du contrat ERC20.

Pour plus d'informations, consultez le guide de référence des requêtes Amazon Managed Blockchain (AMB).

## Obtenez tous les jetons émis dans le cadre d'un contrat

Vous pouvez utiliser l'opération `ListTokenBalances`API pour renvoyer une liste de tous les jetons pris en charge (ERC20, ERC721, ERC1155) émis par un contrat lorsque l'adresse du contrat est transmise en entrée. Par exemple, vous pouvez récupérer des informations relatives aux jetons non fongibles (NFT) émis selon la norme de contrat ERC721 sur la blockchain Ethereum en utilisant l'opération API. `ListTokenBalances`

Pour plus d'informations, consultez le guide de référence des requêtes Amazon Managed Blockchain (AMB).

## Listez les contrats et obtenez des informations sur les contrats

Vous pouvez utiliser l'opération `ListAssetContracts`API pour répertorier les contrats ERC-721, ERC-1155 ou ERC-20 déployés par une adresse donnée. En outre, si vous avez l'adresse du contrat, vous pouvez utiliser l'opération `GetAssetContract`API pour récupérer les propriétés du contrat, telles que l'adresse du déployeur du type de contrat et les métadonnées du jeton pertinentes.

Pour plus d'informations, consultez le guide de référence des requêtes Amazon Managed Blockchain (AMB).

# Référence de l'API de requête Amazon Managed Blockchain (AMB)

Amazon Managed Blockchain (AMB) Query fournit des opérations d'API pour interroger les chaînes de blocs prises en charge. Cela inclut les API pour interroger les jetons, les transactions et les contrats. Pour plus d'informations, consultez la référence de l'API de requête AMB.

# Sécurité dans la requête Amazon Managed Blockchain (AMB)

La sécurité du cloud AWS est une priorité absolue. En tant que AWS client, vous bénéficiez de centres de données et d'architectures réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le modèle de responsabilité partagée décrit cela comme étant à la fois la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS Cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de conformitéAWS. Pour en savoir plus sur les programmes de conformité qui s'appliquent à Amazon Managed Blockchain (AMB) Query, consultez la section AWS Services concernés par le programme de conformité.
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris la sensibilité de vos données, les exigences de votre entreprise, et la législation et la réglementation applicables.

Pour assurer la protection des données, l'authentification et le contrôle d'accès, Amazon Managed Blockchain utilise les AWS fonctionnalités et les fonctionnalités du framework open source exécuté dans Managed Blockchain.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation d'AMB Query. Les rubriques suivantes vous montrent comment configurer AMB Query pour répondre à vos objectifs de sécurité et de conformité. Vous pouvez également apprendre à utiliser d'autres AWS services qui vous aident à surveiller et à sécuriser vos ressources AMB Query.

Rubriques

# Chiffrement des données

Le chiffrement des données permet d'empêcher les utilisateurs non autorisés de lire les données d'un réseau blockchain et des systèmes de stockage de données associés. Cela inclut les données susceptibles d'être interceptées lorsqu'elles circulent sur le réseau, appelées données en transit.

## Chiffrement en transit

Par défaut, Managed Blockchain utilise une connexion HTTPS/TLS pour chiffrer toutes les données transmises du AWS CLI client aux points de terminaison du service. AWS

# Gestion des identités et des accès pour Amazon Managed Blockchain (AMB) Query

AWS Identity and Access Management (IAM) est un Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès à AWS ressources. IAMles administrateurs contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser les ressources de AMB requête. IAMest un Service AWS que vous pouvez utiliser sans frais supplémentaires.

Rubriques



## Public ciblé

Comment utilisez-vous AWS Identity and Access Management (IAM) diffère en fonction du travail que vous effectuez dans AMB Query.

Utilisateur du service : si vous utilisez le service AMB Query pour effectuer votre travail, votre administrateur vous fournit les informations d'identification et les autorisations dont vous avez besoin. Au fur et à mesure que vous utilisez de plus en plus de fonctionnalités de AMB requête pour effectuer

votre travail, vous aurez peut-être besoin d'autorisations supplémentaires. En comprenant bien la gestion des accès, vous saurez demander les autorisations appropriées à votre administrateur. Si vous ne pouvez pas accéder à une fonctionnalité dans AMB Query, consultezRésolution des problèmes liés à l'identité et à l'accès aux requêtes Amazon Managed Blockchain (AMB).

Administrateur du service — Si vous êtes responsable des ressources de AMB Query dans votre entreprise, vous avez probablement un accès complet à AMB Query. C'est à vous de déterminer les fonctionnalités et les ressources de AMB requête auxquelles les utilisateurs de votre service doivent accéder. Vous devez ensuite envoyer des demandes à votre IAM administrateur pour modifier les autorisations des utilisateurs de votre service. Consultez les informations de cette page pour comprendre les concepts de base deIAM. Pour en savoir plus sur la façon dont votre entreprise peut utiliser IAM AMB Query, consultezComment fonctionne Amazon Managed Blockchain (AMB) Query avec IAM.

IAMadministrateur — Si vous êtes IAM administrateur, vous souhaiterez peut-être en savoir plus sur la manière dont vous pouvez rédiger des politiques pour gérer l'accès à AMB Query. Pour consulter des exemples de politiques basées sur l'identité de AMB requête que vous pouvez utiliser dansIAM, consultez. Exemples de politiques basées sur l'identité pour Amazon Managed Blockchain () AMB Query

## Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS en utilisant vos informations d'identification. Vous devez être authentifié (connecté) à AWS) en tant que Utilisateur racine d'un compte AWS, en tant qu'IAMutilisateur ou en assumant un IAM rôle.

Vous pouvez vous connecter à AWS en tant qu'identité fédérée en utilisant les informations d'identification fournies par le biais d'une source d'identité. AWS IAM Identity Center Les utilisateurs (IAMIdentity Center), l'authentification unique de votre entreprise et vos informations d'identification Google ou Facebook sont des exemples d'identités fédérées. Lorsque vous vous connectez en tant qu'identité fédérée, votre administrateur a préalablement configuré la fédération d'identité à l'aide de IAM rôles. Lorsque vous accédez AWS en utilisant la fédération, vous assumez indirectement un rôle.

Selon le type d'utilisateur que vous êtes, vous pouvez vous connecter au AWS Management Console ou le AWS portail d'accès. Pour plus d'informations sur la connexion à AWS, voir Comment se connecter à votre Compte AWS dans le .Connexion à AWS Guide de l'utilisateur.

Si vous accédez AWS programmatiquement, AWS fournit un kit de développement logiciel (SDK) et une interface de ligne de commande (CLI) pour signer cryptographiquement vos demandes à l'aide

de vos informations d'identification. Si vous n'utilisez pas AWS outils, vous devez signer vous-même les demandes. Pour plus d'informations sur l'utilisation de la méthode recommandée pour signer vous-même les demandes, voir Signature AWS APIdemandes dans le guide de IAM l'utilisateur.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être fournir des informations de sécurité supplémentaires. Par exemple, AWS vous recommande d'utiliser l'authentification multifactorielle (MFA) pour renforcer la sécurité de votre compte. Pour en savoir plus, consultez la section Authentification multifactorielle dans le AWS IAM Identity Center Guide de l'utilisateur et utilisation de l'authentification multifactorielle (MFA) dans AWS dans le guide de l'utilisateur IAM.

## Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez avec une seule identité de connexion qui donne un accès complet à tous Services AWS et les ressources du compte. Cette identité s'appelle Compte AWS utilisateur root et est accessible en vous connectant avec l'adresse e-mail et le mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur racine et utilisez-les pour effectuer les tâches que seul l'utilisateur racine peut effectuer. Pour obtenir la liste complète des tâches qui nécessitent que vous vous connectiez en tant qu'utilisateur root, consultez la section Tâches nécessitant des informations d'identification utilisateur root dans le guide de IAM l'utilisateur.

## Identité fédérée

La meilleure pratique consiste à obliger les utilisateurs humains, y compris les utilisateurs nécessitant un accès administrateur, à utiliser la fédération avec un fournisseur d'identité pour accéder Services AWS en utilisant des informations d'identification temporaires.

Une identité fédérée est un utilisateur de l'annuaire des utilisateurs de votre entreprise, un fournisseur d'identité Web, le AWS Directory Service, le répertoire Identity Center ou tout utilisateur accédant Services AWS en utilisant les informations d'identification fournies par le biais d'une source d'identité. Lorsque les identités fédérées accèdent Comptes AWS, ils assument des rôles, et les rôles fournissent des informations d'identification temporaires.

Pour une gestion centralisée des accès, nous vous recommandons d'utiliser AWS IAM Identity Center. Vous pouvez créer des utilisateurs et des groupes dans IAM Identity Center, ou vous pouvez vous connecter et synchroniser avec un ensemble d'utilisateurs et de groupes dans votre propre source d'identité afin de les utiliser dans tous vos Comptes AWS et applications. Pour plus

d'informations sur IAM Identity Center, consultez Qu'est-ce qu'IAMIdentity Center ? dans le AWS IAM Identity Center Guide de l'utilisateur.

## Utilisateurs et groupes IAM

Un IAMutilisateur est une identité au sein de votre Compte AWS qui dispose d'autorisations spécifiques pour une seule personne ou une seule application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des IAM utilisateurs dotés d'informations d'identification à long terme, telles que des mots de passe et des clés d'accès. Toutefois, si vous avez des cas d'utilisation spécifiques qui nécessitent des informations d'identification à long terme auprès des IAM utilisateurs, nous vous recommandons de faire pivoter les clés d'accès. Pour plus d'informations, voir Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification à long terme dans le Guide de IAM l'utilisateur.

Un IAMgroupe est une identité qui définit un ensemble d'IAMutilisateurs. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez nommer un groupe IAMAdminset lui donner les autorisations nécessaires pour administrer IAM des ressources.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. Pour en savoir plus, voir Quand créer un IAM utilisateur (au lieu d'un rôle) dans le Guide de IAM l'utilisateur.

## IAMrôles

Un IAMrôle est une identité au sein de votre Compte AWS qui dispose d'autorisations spécifiques. Il est similaire à un IAM utilisateur, mais n'est pas associé à une personne en particulier. Vous pouvez assumer temporairement un IAM rôle dans AWS Management Console en changeant de rôle. Vous pouvez assumer un rôle en appelant un AWS CLI or AWS APIopération ou en utilisant une option personnaliséeURL. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez la section Utilisation IAM des rôles dans le Guide de IAM l'utilisateur.

IAMles rôles dotés d'informations d'identification temporaires sont utiles dans les situations suivantes :

- Accès utilisateur fédéré : pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour plus d'informations sur les rôles pour la fédération, voir Création d'un rôle pour un fournisseur d'identité tiers dans le guide de IAM l'utilisateur. Si vous utilisez IAM Identity Center, vous configurez un ensemble d'autorisations. Pour contrôler les accès auxquels vos identités peuvent accéder après leur authentification, IAM Identity Center met en corrélation l'ensemble d'autorisations avec un rôle dans. IAM Pour plus d'informations sur les ensembles d'autorisations, voir Ensembles d'autorisations dans le AWS IAM Identity Center Guide de l'utilisateur.
- Autorisations IAM utilisateur temporaires : un IAM utilisateur ou un rôle peut assumer un IAM rôle afin d'obtenir temporairement différentes autorisations pour une tâche spécifique.
- Accès entre comptes : vous pouvez utiliser un IAM rôle pour autoriser une personne (un mandant fiable) d'un autre compte à accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Cependant, avec certains Services AWS, vous pouvez associer une politique directement à une ressource (au lieu d'utiliser un rôle comme proxy). Pour connaître la différence entre les rôles et les politiques basées sur les ressources pour l'accès entre comptes, voir Accès aux ressources entre comptes IAM dans le guide de l'IAMutilisateur.
- Accès multiservices — Certains Services AWS utiliser des fonctionnalités dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, un rôle de service ou un rôle lié au service.
  - Sessions d'accès transmises (FAS) : lorsque vous utilisez un IAM utilisateur ou un rôle pour effectuer des actions dans AWS, vous êtes considéré comme un directeur. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FASutilise les autorisations du principal appelant un Service AWS, combiné à la demande Service AWS pour adresser des demandes aux services en aval. FASles demandes ne sont effectuées que lorsqu'un service reçoit une demande nécessitant des interactions avec d'autres Services AWS ou des ressources à compléter. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur les politiques relatives FAS aux demandes, voir Transférer les sessions d'accès.
  - Rôle de service — Un rôle de service est un IAMrôle qu'un service assume pour effectuer des actions en votre nom. Un IAM administrateur peut créer, modifier et supprimer un rôle de service de l'intérieurIAM. Pour plus d'informations, voir Création d'un rôle pour déléguer des autorisations à un Service AWS dans le guide de l'utilisateur IAM.

  - Rôle lié à un service — Un rôle lié à un service est un type de rôle lié à un service Service AWS. Le service peut assumer le rôle d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS et appartiennent au service. Un IAM administrateur peut consulter, mais pas modifier les autorisations pour les rôles liés à un service.
- Applications exécutées sur Amazon EC2 : vous pouvez utiliser un IAM rôle pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une EC2 instance et qui créent AWS CLI or AWS APIdemandes. Cela est préférable au stockage des clés d'accès dans l'EC2instance. Pour attribuer un AWS pour attribuer un rôle à une EC2 instance et le mettre à la disposition de toutes ses applications, vous créez un profil d'instance attaché à l'instance. Un profil d'instance contient le rôle et permet aux programmes exécutés sur l'EC2instance d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez la section Utilisation d'un IAM rôle pour accorder des autorisations aux applications exécutées sur des EC2 instances Amazon dans le Guide de IAM l'utilisateur.

Pour savoir s'il faut utiliser IAM des rôles ou des IAM utilisateurs, voir Quand créer un IAM rôle (au lieu d'un utilisateur) dans le guide de IAM l'utilisateur.

## Gestion des accès à l’aide de politiques

Vous contrôlez l'accès dans AWS en créant des politiques et en les associant à AWS identités ou ressources. Une politique est un objet dans AWS qui, lorsqu'elle est associée à une identité ou à une ressource, définit leurs autorisations. AWS évalue ces politiques lorsqu'un principal (utilisateur, utilisateur root ou session de rôle) fait une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques sont stockées dans AWS sous forme de JSON documents. Pour plus d'informations sur la structure et le contenu des documents de JSON politique, voir Présentation des JSON politiques dans le guide de IAM l'utilisateur.

Les administrateurs peuvent utiliser AWS JSONpolitiques pour spécifier qui a accès à quoi. C’est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d’aucune autorisation. Pour autoriser les utilisateurs à effectuer des actions sur les ressources dont ils ont besoin, un IAM administrateur peut créer des IAM politiques. L'administrateur peut ensuite ajouter les IAM politiques aux rôles, et les utilisateurs peuvent assumer les rôles.

IAMles politiques définissent les autorisations pour une action, quelle que soit la méthode que vous utilisez pour effectuer l'opération. Par exemple, supposons que vous disposiez d’une politique qui

autorise l'action `iam:GetRole`. Un utilisateur appliquant cette politique peut obtenir des informations sur le rôle auprès du AWS Management Console, le AWS CLI, ou le AWS API.

### Politiques basées sur l'identité

Les politiques basées sur l'identité sont JSON des documents de politique d'autorisation que vous pouvez joindre à une identité, telle qu'un IAM utilisateur, un groupe d'utilisateurs ou un rôle. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour savoir comment créer une politique basée sur l'identité, voir Création de IAM politiques dans le Guide de l'IAMutilisateur.

Les politiques basées sur l'identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez associer à plusieurs utilisateurs, groupes et rôles dans votre Compte AWS. Les politiques gérées incluent AWS politiques gérées et politiques gérées par le client. Pour savoir comment choisir entre une politique gérée ou une politique intégrée, voir Choisir entre des politiques gérées et des politiques intégrées dans le Guide de l'IAMutilisateur.

### Politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents JSON de stratégie que vous attachez à une ressource. Les politiques de confiance dans les IAM rôles et les politiques relatives aux compartiments Amazon S3 sont des exemples de politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou Services AWS.

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser AWS politiques gérées à partir IAM d'une stratégie basée sur les ressources.

### Listes de contrôle d'accès (ACLs)

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLssont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format du document JSON de stratégie.

Amazon S3 AWS WAF, et Amazon VPC sont des exemples de services qui prennent en chargeACLs. Pour en savoir plusACLs, consultez la présentation de la liste de contrôle d'accès (ACL) dans le guide du développeur Amazon Simple Storage Service.

## Autres types de politique

AWS prend en charge d'autres types de politiques moins courants. Ces types de politiques peuvent définir le nombre maximum d'autorisations qui vous sont accordées par des types de politiques plus courants.

- Limites d'autorisations — Une limite d'autorisations est une fonctionnalité avancée dans laquelle vous définissez le maximum d'autorisations qu'une politique basée sur l'identité peut accorder à une IAM entité (IAMutilisateur ou rôle). Vous pouvez définir une limite d'autorisations pour une entité. Les autorisations en résultant représentent la combinaison des politiques basées sur l'identité d'une entité et de ses limites d'autorisation. Les politiques basées sur les ressources qui spécifient l'utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d'autorisations. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour plus d'informations sur les limites d'autorisations, voir Limites d'autorisations pour les IAM entités dans le Guide de IAM l'utilisateur.
- Politiques de contrôle des services (SCPs) : SCPs JSON politiques qui spécifient les autorisations maximales pour une organisation ou une unité organisationnelle (UO) dans AWS Organizations. AWS Organizations est un service de regroupement et de gestion centralisée de plusieurs Comptes AWS que votre entreprise possède. Si vous activez toutes les fonctionnalités d'une organisation, vous pouvez appliquer des politiques de contrôle des services (SCPs) à l'un ou à l'ensemble de vos comptes. Les SCP limites d'autorisations pour les entités figurant dans les comptes des membres, y compris chaque Utilisateur racine d'un compte AWS. Pour plus d'informations sur les Organizations et SCPs consultez les politiques de contrôle des services dans le AWS Organizations Guide de l'utilisateur.
- Politiques de séance : les politiques de séance sont des politiques avancées que vous utilisez en tant que paramètre lorsque vous créez par programmation une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de séance en résultant sont une combinaison des politiques basées sur l'identité de l'utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d'une politique basée sur les ressources. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour plus d'informations, consultez la section Politiques de session dans le guide de IAM l'utilisateur.

### Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS détermine s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, voir la logique d'évaluation des politiques dans le guide de IAM l'utilisateur.

## Comment fonctionne Amazon Managed Blockchain (AMB) Query avec IAM

Avant de commencer IAM à gérer l'accès à AMB Query, découvrez quelles IAM fonctionnalités peuvent être utilisées avec AMB Query.

IAMfonctionnalités que vous pouvez utiliser avec Amazon Managed Blockchain (AMB) Query

| IAMfonctionnalité | AMBSupport pour les requêtes |
|---|---|
| Politiques basées sur l'identité | Oui |
| Politiques basées sur les ressources | Non |
| Actions de politique | Oui |
| Ressources de politique | Non |
| Clés de condition d'une politique | Non |
| ACLs | Non |
| ABAC(balises dans les politiques) | Non |
| Informations d'identification temporaires | Oui |
| Autorisations de principal | Oui |
| Fonctions du service | Non |
| Rôles liés à un service | Non |

Pour obtenir une vue d'ensemble de la façon dont AMB Query et d'autres AWS les services fonctionnent avec la plupart IAM des fonctionnalités, voir AWS services compatibles avec IAM le Guide de l'IAMutilisateur.

## Politiques basées sur l'identité pour Query AMB

Prend en charge les politiques basées sur l'identité : oui

Les politiques basées sur l'identité sont JSON des documents de politique d'autorisation que vous pouvez joindre à une identité, telle qu'un IAM utilisateur, un groupe d'utilisateurs ou un rôle. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour savoir comment créer une politique basée sur l'identité, voir Création de IAM politiques dans le Guide de l'IAMutilisateur.

Avec les politiques IAM basées sur l'identité, vous pouvez spécifier les actions et les ressources autorisées ou refusées ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Vous ne pouvez pas spécifier le principal dans une politique basée sur une identité, car celle-ci s'applique à l'utilisateur ou au rôle auquel elle est attachée. Pour en savoir plus sur tous les éléments que vous pouvez utiliser dans une JSON politique, consultez la référence aux éléments de IAM JSON politique dans le Guide de IAM l'utilisateur.

Exemples de politiques basées sur l'identité pour Query AMB

Pour consulter des exemples de politiques basées sur l'identité des AMB requêtes, consultez. Exemples de politiques basées sur l'identité pour Amazon Managed Blockchain () AMB Query

## Politiques basées sur les ressources dans Query AMB

Prend en charge les politiques basées sur les ressources : non

Les politiques basées sur les ressources sont des documents JSON de stratégie que vous attachez à une ressource. Les politiques de confiance dans les IAM rôles et les politiques relatives aux compartiments Amazon S3 sont des exemples de politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou Services AWS.

Pour activer l'accès entre comptes, vous pouvez spécifier un compte entier ou IAM des entités d'un autre compte comme principal dans une politique basée sur les ressources. L'ajout d'un principal entre comptes à une politique basée sur les ressources ne représente qu'une partie de l'instauration de la relation d'approbation. Lorsque le principal et la ressource sont différents Comptes AWS, un IAM administrateur du compte sécurisé doit également accorder à l'entité principale (utilisateur ou rôle) l'autorisation d'accéder à la ressource. Pour ce faire, il attache une politique basée sur une identité à l'entité. Toutefois, si une politique basée sur des ressources accorde l'accès à un principal dans le même compte, aucune autre politique basée sur l'identité n'est requise. Pour plus d'informations, voir Accès aux ressources entre comptes IAM dans le Guide de IAM l'utilisateur.

## Actions de politique pour AMB Query

Prend en charge les actions de politique : oui

Les administrateurs peuvent utiliser AWS JSONpolitiques pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'`Action`élément d'une JSON politique décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès dans une politique. Les actions politiques portent généralement le même nom que les actions associées AWS APIopération. Il existe certaines exceptions, telles que les actions avec autorisation uniquement qui n'ont pas d'opération correspondante. API Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d'actions dans une politique afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des actions de AMB requête, consultez la section Actions définies par Amazon Managed Blockchain (AMB) Query dans le Service Authorization Reference.

Les actions de politique dans AMB Query utilisent le préfixe suivant avant l'action :

```
managedblockchain-query:
```

Pour indiquer plusieurs actions dans une seule déclaration, séparez-les par des virgules.

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

Pour consulter des exemples de politiques basées sur l'identité des AMB requêtes, consultez. Exemples de politiques basées sur l'identité pour Amazon Managed Blockchain () AMB Query

## Ressources relatives aux politiques pour AMB Query

Prend en charge les ressources politiques : Non

Les administrateurs peuvent utiliser AWS JSONpolitiques pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Resource` JSON de stratégie indique le ou les objets auxquels s'applique l'action. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de spécifier une ressource en utilisant son Amazon Resource Name (ARN). Vous pouvez le faire pour des actions qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

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

Pour consulter la liste des types de ressources AMB Query et leurs caractéristiquesARNs, consultez la section Resources Defined by Amazon Managed Blockchain (AMB) Query dans le Service Authorization Reference. Pour savoir avec quelles actions vous pouvez spécifier pour chaque ressource, consultez Actions définies par Amazon Managed Blockchain (AMB) Query. ARN

Pour consulter des exemples de politiques basées sur l'identité des AMB requêtes, consultez. Exemples de politiques basées sur l'identité pour Amazon Managed Blockchain () AMB Query

## Clés de conditions de politique pour AMB Query

Prend en charge les clés de condition de politique spécifiques au service : Non

Les administrateurs peuvent utiliser AWS JSONpolitiques pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Condition` (ou le bloc `Condition`) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément `Condition` est facultatif. Vous pouvez créer des expressions

conditionnelles qui utilisent des opérateurs de condition, tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande.

Si vous spécifiez plusieurs `Condition` éléments dans une instruction ou plusieurs clés dans un seul `Condition` élément, AWS les évalue à l'aide d'une `AND` opération logique. Si vous spécifiez plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une `OR` opération logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l'instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Par exemple, vous pouvez autoriser un IAM utilisateur à accéder à une ressource uniquement si celle-ci est étiquetée avec son nom IAM d'utilisateur. Pour plus d'informations, consultez IAMla section Éléments de politique : variables et balises dans le Guide de IAM l'utilisateur.

AWS prend en charge les clés de condition globales et les clés de condition spécifiques au service. Pour tout voir AWS clés de condition globales, voir AWS clés contextuelles des conditions globales dans le guide de IAM l'utilisateur.

Pour consulter la liste des clés de condition de AMB requête, consultez la section Clés de condition pour Amazon Managed Blockchain (AMB) Query dans le Service Authorization Reference. Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, consultez Actions Defined by Amazon Managed Blockchain (AMB) Query.

Pour consulter des exemples de politiques basées sur l'identité des AMB requêtes, consultez. Exemples de politiques basées sur l'identité pour Amazon Managed Blockchain () AMB Query

## ACLsdans AMB Query

Supports ACLs : Non

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLssont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format du document JSON de stratégie.

## ABACavec AMB Query

Supports ABAC (balises dans les politiques) : Non

Le contrôle d'accès basé sur les attributs (ABAC) est une stratégie d'autorisation qui définit les autorisations en fonction des attributs. Entrée AWS, ces attributs sont appelés balises. Vous pouvez associer des tags à IAM des entités (utilisateurs ou rôles) et à de nombreuses AWS ressources. Le

balisage des entités et des ressources est la première étape deABAC. Vous concevez ensuite des ABAC politiques pour autoriser les opérations lorsque le tag du principal correspond à celui de la ressource à laquelle il essaie d'accéder.

ABACest utile dans les environnements qui se développent rapidement et aide dans les situations où la gestion des politiques devient fastidieuse.

Pour contrôler l'accès basé sur des étiquettes, vous devez fournir les informations d'étiquette dans l'élément de condition d'une politique utilisant les clés de condition `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

Si un service prend en charge les trois clés de condition pour tous les types de ressources, alors la valeur pour ce service est Oui. Si un service prend en charge les trois clés de condition pour certains types de ressources uniquement, la valeur est Partielle.

Pour plus d'informationsABAC, voir Qu'est-ce que c'est ABAC ? dans le guide de IAM l'utilisateur. Pour consulter un didacticiel présentant les étapes de configurationABAC, voir Utiliser le contrôle d'accès basé sur les attributs (ABAC) dans le guide de l'IAMutilisateur.

## Utilisation d'informations d'identification temporaires avec AMB Query

Prend en charge les informations d'identification temporaires : oui

Momentanée Services AWS ne fonctionnent pas lorsque vous vous connectez à l'aide d'informations d'identification temporaires. Pour plus d'informations, y compris lesquelles Services AWS utilisez des informations d'identification temporaires, voir Services AWS qui fonctionnent avec IAM le Guide de l'IAMutilisateur.

Vous utilisez des informations d'identification temporaires si vous vous connectez au AWS Management Console en utilisant n'importe quelle méthode, à l'exception du nom d'utilisateur et du mot de passe. Par exemple, lorsque vous accédez AWS à l'aide du lien d'authentification unique (SSO) de votre entreprise, ce processus crée automatiquement des informations d'identification temporaires. Vous créez également automatiquement des informations d'identification temporaires lorsque vous vous connectez à la console en tant qu'utilisateur, puis changez de rôle. Pour plus d'informations sur le changement de rôle, consultez la section Passage à un rôle (console) dans le guide de IAM l'utilisateur.

Vous pouvez créer manuellement des informations d'identification temporaires à l'aide du AWS CLI or AWS API. Vous pouvez ensuite utiliser ces informations d'identification temporaires pour accéder AWS. AWS recommande de générer dynamiquement des informations d'identification

temporaires au lieu d'utiliser des clés d'accès à long terme. Pour plus d'informations, consultez la section Informations d'identification de sécurité temporaires dans IAM.

## Autorisations principales interservices pour Query AMB

Prend en charge les sessions d'accès transféré (FAS) : Oui

Lorsque vous utilisez un IAM utilisateur ou un rôle pour effectuer des actions dans AWS, vous êtes considéré comme un directeur. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FASutilise les autorisations du principal appelant un Service AWS, combiné à la demande Service AWS pour adresser des demandes aux services en aval. FASles demandes ne sont effectuées que lorsqu'un service reçoit une demande nécessitant des interactions avec d'autres Services AWS ou des ressources à compléter. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur les politiques relatives FAS aux demandes, voir Transférer les sessions d'accès.

## Rôles de service pour AMB Query

Supporte les rôles de service : Non

Un rôle de service est un IAMrôle qu'un service assume pour effectuer des actions en votre nom. Un IAM administrateur peut créer, modifier et supprimer un rôle de service de l'intérieurIAM. Pour plus d'informations, voir Création d'un rôle pour déléguer des autorisations à un Service AWS dans le guide de l'utilisateur IAM.

> **Warning**
>
> La modification des autorisations associées à un rôle de service peut interrompre la fonctionnalité de AMB requête. Modifiez les rôles de service uniquement lorsque AMB Query fournit des instructions à cet effet.

## Rôles liés à un service pour Query AMB

Prend en charge les rôles liés à un service : non

Un rôle lié à un service est un type de rôle de service lié à un Service AWS. Le service peut assumer le rôle d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS et appartiennent au service. Un IAM administrateur peut consulter, mais pas modifier les autorisations pour les rôles liés à un service.

Pour plus de détails sur la création ou la gestion des rôles liés à un service, voir AWS services qui fonctionnent avec IAM. Recherchez un service dans le tableau qui inclut un `Yes` dans la colonne Rôle lié à un service. Choisissez le lien Oui pour consulter la documentation du rôle lié à ce service.

# Exemples de politiques basées sur l'identité pour Amazon Managed Blockchain () AMB Query

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou à modifier des ressources de AMB requête. Ils ne peuvent pas non plus effectuer de tâches en utilisant le AWS Management Console, AWS Command Line Interface (AWS CLI), ou AWS API. Pour autoriser les utilisateurs à effectuer des actions sur les ressources dont ils ont besoin, un IAM administrateur peut créer des IAM politiques. L'administrateur peut ensuite ajouter les IAM politiques aux rôles, et les utilisateurs peuvent assumer les rôles.

Pour savoir comment créer une politique IAM basée sur l'identité à l'aide de ces exemples de documents de JSON stratégie, consultez la section Création de IAM politiques dans le guide de l'IAMutilisateur.

Pour plus de détails sur les actions et les types de ressources définis par AMB Query, y compris le ARNs format de la requête Actions, Resources, and Condition Keys for Amazon Managed Blockchain (AMB) Query dans le Service Authorization Reference.

Rubriques

- Bonnes pratiques en matière de politiques
- Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations
- Accès à des API actions de requête Amazon Managed Blockchain (AMB) spécifiques

## Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer des ressources de AMB requête dans votre compte. Ces actions peuvent entraîner des coûts pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez les directives et recommandations suivantes :

- Commencez avec AWS politiques gérées et évolution vers des autorisations de moindre privilège — Pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez le AWS politiques gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons

de réduire davantage les autorisations en définissant AWS des politiques gérées par le client spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez .AWS politiques gérées ou AWS politiques gérées pour les fonctions professionnelles dans le guide de IAM l'utilisateur.

- Appliquer les autorisations du moindre privilège : lorsque vous définissez des autorisations à IAM l'aide de politiques, accordez uniquement les autorisations nécessaires à l'exécution d'une tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation IAM pour appliquer des autorisations, consultez la section Politiques et autorisations du Guide de IAM l'utilisateur. IAM
- Utilisez des conditions dans IAM les politiques pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques pour limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez rédiger une condition de politique pour spécifier que toutes les demandes doivent être envoyées en utilisantSSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un Service AWS, comme AWS CloudFormation. Pour plus d'informations, voir Éléments IAM JSON de politique : Condition dans le guide de IAM l'utilisateur.
- Utilisez IAM Access Analyzer pour valider vos IAM politiques afin de garantir des autorisations sécurisées et fonctionnelles. IAM Access Analyzer valide les politiques nouvelles et existantes afin qu'elles respectent le langage des politiques (JSON) et IAM les IAM meilleures pratiques. IAMAccess Analyzer fournit plus de 100 vérifications des politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez la section Validation des politiques d'IAMAccess Analyzer dans le guide de IAM l'utilisateur.
- Exiger une authentification multifactorielle (MFA) — Si vous avez un scénario qui nécessite IAM des utilisateurs ou un utilisateur root dans votre Compte AWS, activez MFA pour plus de sécurité. Pour exiger le MFA moment où les API opérations sont appelées, ajoutez MFA des conditions à vos politiques. Pour plus d'informations, consultez la section Configuration de l'APIaccès MFA protégé dans le Guide de l'IAMutilisateur.

Pour plus d'informations sur les meilleures pratiques en matière de sécuritéIAM, consultez la section Bonnes pratiques en matière de sécurité IAM dans le Guide de IAM l'utilisateur.

## Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Cet exemple montre comment créer une politique qui permet aux IAM utilisateurs de consulter les politiques intégrées et gérées associées à leur identité d'utilisateur. Cette politique inclut les

autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide du AWS CLI or AWS API.

```
{
    "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": "*"
        }
    ]
}
```

## Accès à des API actions de requête Amazon Managed Blockchain (AMB) spécifiques

> **Note**
>
> Pour accéder à la AMB requête et API passer des appels, vous aurez besoin d'informations d'identification utilisateur (`AWS_ACCESS_KEY_IDetAWS_SECRET_ACCESS_KEY`) disposant des IAM autorisations appropriées pour AMB Query.

Example IAMPolitique d'accès à toutes les requêtes Amazon Managed Blockchain (AMB) APIs

Cet exemple accorde à un IAM utilisateur le statut de votre Compte AWS accès à toutes les AMB requêtesAPIs.

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

Example IAMPolitique d'accès à Amazon Managed Blockchain (AMB) Query **`ListTransactions`** et **`GetTransaction`** APIs

Cet exemple accorde à un IAM utilisateur le statut de votre Compte AWS accès à la AMB requête `ListTransaction` et `GetTransaction` APIs

> **Note**
>
> APIsDans l'exemple, vous pouvez remplacer ou ajouter le par un autre APIs pour donner accès à un autre ou à plusieursAPIs. Pour obtenir la liste des AMB requêtesAPIs, consultez le guide de API référence des requêtes Amazon Managed Blockchain (AMB).

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

# Résolution des problèmes liés à l'identité et à l'accès aux requêtes Amazon Managed Blockchain (AMB)

Utilisez les informations suivantes pour diagnostiquer et résoudre les problèmes courants que vous pouvez rencontrer lors de l'utilisation de AMB Query etIAM.

Rubriques

- Je ne suis pas autorisé à effectuer une action dans AMB Query

## Je ne suis pas autorisé à effectuer une action dans AMB Query

Si vous recevez une erreur qui indique que vous n'êtes pas autorisé à effectuer une action, vos politiques doivent être mises à jour afin de vous permettre d'effectuer l'action.

L'exemple d'erreur suivant se produit lorsque l'`mateojackson`IAMutilisateur essaie d'utiliser la console pour afficher les détails d'une *`my-example-widget`* ressource fictive mais ne dispose pas des `managedblockchain-query:`*`:GetWidget`* autorisations fictives.

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

Dans ce cas, la politique qui s'applique à l'utilisateur `mateojackson` doit être mise à jour pour autoriser l'accès à la ressource *`my-example-widget`* à l'aide de l'action `managedblockchain-query::`*`GetWidget`*.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

# Métriques d'utilisation de l'API de requête Amazon Managed Blockchain (AMB) sur Amazon CloudWatch

## Statistiques d'utilisation des API sur Amazon CloudWatch

Les métriques d'utilisation de l'API publiées pour CloudWatch correspondre aux quotas du service de requête Amazon Managed Blockchain (AMB). Vous pouvez configurer des alarmes pour vous avertir lorsque votre utilisation approche d'un quota de service. Pour plus d'informations sur CloudWatch l'intégration avec les quotas de service, consultez les métriques d'utilisation d'AWS dans le guide de CloudWatch l'utilisateur Amazon.

AMB Query publie les métriques d'API suivantes dans l'espace de `AWS/Usage` noms, avec le nom du `Amazon Managed Blockchain Query` service.

| Métrique | Description |
| --- | --- |
| `CallCount` | Le nombre total d'appels effectués vers une API dans AMB Query. SUM représente le nombre total d'appels à l'API pendant la période spécifiée. |

Amazon Managed Blockchain (AMB) Query publie des métriques d'utilisation dans l'espace de `AWS/Usage` noms avec les dimensions suivantes.

| Dimension | Description |
| --- | --- |
| Service | Nom du AWS service contenant la ressource. `Amazon Managed Blockchain Query` sera toujours la valeur de cette dimension. |
| Type | Type d'entité signalée. `API`sera toujours la valeur de cette dimension. |

| Dimension | Description |
| --- | --- |
| Ressource | Type de ressources signalées. Le nom de l'opération AMB Query API utilisée sera la valeur de cette dimension. |
| Classe | Classe de la ressource signalée. None sera toujours la valeur de cette dimension. |

# Historique du document pour le guide de l'utilisateur AMB Query

Le tableau suivant décrit les versions de documentation pour AMB Query.

| Modification | Description | Date |
| --- | --- | --- |
| AMB Query prend en charge les identifiants de transaction Bitcoin et les hachages de transactions | Pour les réseaux Bitcoin, les opérations de l'API AMB Query prennent en charge à la fois l'identifiant de transaction (`transactionId` ) et le hachage de transaction (`transactionHash` ). | 21 mars 2024 |
| Support pour les métriques d'utilisation des API sur Amazon CloudWatch | AMB Query a ajouté la prise en charge des métriques d'utilisation de l'API sur CloudWatch. Ces mesures d'utilisation correspondent aux quotas du service AMB Query. | 8 février 2024 |
| Support pour les transactions non finalisées | AMB Query a ajouté le support pour les transactions qui n'ont pas atteint leur finalité. Cela supprime également la prise en charge de la `status` propriété dans la réponse de l'`GetTransaction` opération . Vous utiliserez plutôt les `executionStatus` propriétés `confirmationStatus` et pour déterminer le statut de la transaction. | 1 février 2024 |

| | | |
|---|---|---|
| Obsolète de la `status` propriété dans le type de données Transaction | Amazon Managed Blockchain (AMB) Query a rendu cette `status` propriété obsolète dans le type de données Transaction. Vous devez utiliser les `executionStatus` champs `confirmationStatus` et pour déterminer si `status` la transaction est `FINAL` ou`FAILED`. | 20 décembre 2023 |
| Support pour Sepolia Testnet | Amazon Managed Blockchain (AMB) Query prend désormais en charge les requêtes sur le réseau de test Ethereum Sepolia. | 19 octobre 2023 |
| Support pour les contrats d'actifs | Vous pouvez utiliser l'opération `ListAssetContracts` API pour répertorier les déploiements effectués par une adresse donnée. De plus, si vous avez l'adresse du contrat, vous pouvez utiliser l'opération `GetAssetContract` API pour récupérer les détails du contrat. | 16 octobre 2023 |
| Support pour Bitcoin Testnet | Amazon Managed Blockchain (AMB) Query prend désormais en charge les requêtes sur le Bitcoin Testnet. | 16 octobre 2023 |
| Première version | Version initiale du service AMB Query. | 27 juillet 2023 |

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.