



Guide du développeur

Amazon Managed Blockchain Query



Amazon Managed Blockchain Query: Guide du développeur

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

Les marques et la présentation commerciale d'Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n'est pas d'Amazon, d'une manière susceptible de créer une confusion parmi les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

Table of Contents

Qu'est-ce que Amazon Managed Blockchain (AMB) Query ?	1
Utilisez-vous AMB Query pour la première fois ?	1
Concepts clés	2
Considérations et limites relatives à l'utilisation de la requête Amazon Managed Blockchain (AMB)	2
Configuration	6
Prérequis et considérations	6
Inscrivez-vous pour AWS	6
Création d'un utilisateur IAM avec les autorisations appropriées	7
Installez et configurez le AWS Command Line Interface	7
Utilisez le AWS Management Console pour interroger les blockchains à l'aide d'AMB Query	8
Prise en main	9
Créer une politique IAM	9
Exemples d'utilisation de Go	10
Exemples d'utilisation de Node.js	17
Exemples utilisant Python	21
Exemple utilisant le AWS Management Console	23
Cas d'utilisation des requêtes AMB	24
Consulter les soldes de jetons actuels et historiques	24
Récupérez les données historiques des transactions	24
Obtenez tous les soldes de jetons pour une adresse donnée	24
Lister les événements émis pour une transaction	25
Obtenez tous les jetons émis dans le cadre d'un contrat	25
Listez les contrats et obtenez des informations sur les contrats	26
Référence de l'API de requête AMB	27
Sécurité	28
Chiffrement des données	29
Chiffrement en transit	29
Gestion des identités et des accès	29
Public ciblé	29
Authentification par des identités	30
Gestion de l'accès à l'aide de politiques	31
Comment fonctionne la requête Amazon Managed Blockchain (AMB) avec IAM	33
Exemples de politiques basées sur l'identité	39

Résolution des problèmes	43
Métriques d'utilisation de l'API	45
Statistiques d'utilisation des API sur Amazon CloudWatch	45
Historique de la documentation	47
.....	xlix

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 détecter des jetons fongibles ou non fongibles (NFTs), 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'Ether (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 (NFTs). 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 (0x00).

- Signature Version 4 : signature des appels d'API

Lorsque vous appelez l'AMB Query APIs, 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` s'agit d'un hachage double SHA de la transaction, sans inclure les données des témoins. `transactionHash` 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](#) 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 compte. 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 compte.

Inscrivez-vous pour AWS

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

Si vous en 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 ou un SMS et vous saisirez un code de vérification en utilisant le clavier numérique du téléphone.

Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWS est créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce 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 utilisateur IAM avec les autorisations appropriées

Pour créer et utiliser AMB Query, vous devez créer un principal Gestion des identités et des accès AWS (IAM) (utilisateur ou groupe) doté d'autorisations autorisant les actions nécessaires à Managed Blockchain.

Seuls les principaux IAM peuvent effectuer des demandes d'API de requête AMB. Lorsque vous appelez l'AMB Query APIs, 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.

Pour plus d'informations sur la création d'un utilisateur IAM, consultez la section [Création d'un utilisateur IAM 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 utilisateur IAM](#). Pour un exemple de politique d'autorisation que vous pouvez utiliser pour autoriser un utilisateur à utiliser AMB Query, consultez [Exemples de politiques basées sur l'identité pour la requête Amazon Managed Blockchain \(AMB\)](#).

Installez et configurez le AWS Command Line Interface

Si ce n'est pas déjà fait, installez la dernière interface de AWS ligne de commande (CLI) pour utiliser les AWS ressources d'un terminal. Pour plus d'informations, consultez [Installation ou mise à jour de la version la plus récente de l' AWS CLI](#).

Note

Pour accéder à la CLI, vous avez besoin d'un ID 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, consultez la section [Utilisation d'informations d'identification temporaires avec AWS des ressources](#) dans le Guide de l'utilisateur IAM.

Utilisez le AWS Management Console pour interroger des chaînes de blocs à l'aide de la requête Amazon Managed Blockchain (AMB)

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.

AMB Query 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 certaines conditions préalables. 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

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

Créez une politique IAM pour accéder aux opérations de l'API AMB Query

Pour effectuer des demandes à l'API AMB Query, vous devez utiliser les informations d'identification utilisateur (AWS_ACCESS_KEY_ID et AWS_SECRET_ACCESS_KEY) qui disposent des autorisations IAM appropriées pour Amazon Managed Blockchain (AMB) Query. Dans un terminal sur lequel le est AWS CLI installé, exécutez la commande suivante pour créer une politique IAM permettant d'accéder aux opérations de l'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
```

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

Note

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

Effectuez des demandes d'API 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. AMB Query 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 jetons ERC-721, ERC-1155 et ERC-20.

Les exemples suivants sont créés dans le langage Go et utilisent les opérations de l'API AMB Query. Pour plus d'informations sur Go, consultez la [documentation Go](#). Pour plus d'informations sur l'API de requête AMB, consultez la documentation de [référence de l'API de requête Amazon Managed Blockchain \(AMB\)](#).

Les exemples suivants utilisent les actions `ListTransactions` et `GetTransactionAPI` 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.

Exemple— Effectue l'action de `ListTransactions` l'API en utilisant Go

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

```
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)
}
}

```

Après avoir enregistré le fichier, exécutez le code en utilisant la commande suivante dans le `ListTransactions` répertoire : `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— Effectue l'action de l'GetTransactionAPI 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 `GetTransaction` répertoire.

```

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)
}
}

```

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

Le résultat qui suit ressemble à ce qui suit :

```

{
  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 vous permet d'obtenir le solde des jetons natifs (ETH et BTC), qui peuvent être utilisés pour obtenir le solde actuel d'un compte externe (EOA) à un moment donné.

Exemple— Utilisez l'action GetTokenBalance API pour obtenir le solde d'un jeton natif dans Go

Dans l'exemple suivant, vous utilisez l'GetTokenBalanceAPI pour obtenir un solde d'adresses 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 demandes d'API 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 commande `node --version` et confirmez que vous utilisez la version 14 ou supérieure de Node. Si nécessaire, vous pouvez utiliser la commande `nvm install 14`, puis la commande `nvm use 14` pour installer la version 14.
3. Les variables d'environnement `AWS_ACCESS_KEY_ID` et `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 utilisateur IAM.

```
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 via HTTPS pour accéder aux opérations de l'API 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 qu'[AXIOS](#) et récupérer des données depuis AMB Query.
- Ces exemples utilisent un client HTTP tiers pour Node.js, mais vous pouvez également utiliser le AWS JavaScript SDK pour envoyer des requêtes à AMB Query.
- L'exemple suivant vous montre comment effectuer des demandes d'API AMB Query à l'aide d'Axios et des modules AWS SDK 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"
  }
}
```

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

Vous pouvez utiliser l'GetTokenBalanceAPI pour obtenir le solde de différents jetons (par exemple, ERC20 ERC721, et ERC1155) et de pièces natives (par exemple, ETH et BTC), 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 l'[GetTokenBalanceAPI](#) pour obtenir le solde d'adresses d'un ERC20 jeton, USDC, sur le réseau principal Ethereum.

Pour tester l'GetTokenBalanceAPI, 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 solde ERC20 USDC de l'EOA répertorié 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'
  }
}
```

Effectuez des demandes d'API de requête 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 [kit SDK AWS pour Python \(Boto3\)](#).
3. Installez l'[interface de ligne de AWS commande](#) et exécutez la commande `aws configure` pour définir les variables de votre `Access Key ID` `Secret Access Key`, et `Region`.

Une fois tous les prérequis remplis, vous pouvez utiliser le AWS SDK pour Python via HTTPS pour effectuer des demandes d'API de requête Amazon Managed Blockchain (AMB).

L'exemple Python suivant utilise des modules de boto3 pour envoyer des requêtes associées aux entêtes SigV4 requis à l'opération AMB Query API. `ListTransactionEvents` 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 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')))

```

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': '0xdead0000000000000000000420694206942*****',
      'transactionHash':
      '0x125714bb4db48757007fff2671b37637bbfd6d47b3a4757ebbd0c522*****',
      'value': '410241996771871894771826174755464'
    }
  ]
}

```

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

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

Exemple

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 GetTokenBalance comme 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.

AMB Query 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](#)
- [Récupérez les données historiques des transactions](#)
- [Obtenez tous les soldes de jetons pour une adresse donnée](#)
- [Lister les événements émis pour une transaction](#)
- [Obtenez tous les jetons émis dans le cadre d'un contrat](#)
- [Listez les contrats et obtenez des informations sur les contrats](#)

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 détenus par des tiers (). EOAs Par exemple, vous pouvez utiliser l'opération `GetTokenBalance` API pour obtenir le solde d'adresses du ERC20 jeton, 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 ERC20 jetons sur la blockchain Ethereum, comme un événement de transfert ou un événement de retrait du ERC20 contrat.

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 (NFTs) émis conformément à la norme ERC721 contractuelle sur la blockchain Ethereum en utilisant l'[ListTokenBalances](#) opération API.

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 dépoyeur 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 APIs l'interrogation de jetons, de transactions et de 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 conformité AWS](#). 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](#)
- [Gestion des identités et des accès pour Amazon Managed Blockchain \(AMB\) Query](#)

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 HTTPS/TLS connexion pour chiffrer toutes les données transmises du AWS CLI client aux points de terminaison du AWS service.

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

Gestion des identités et des accès AWS (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser les ressources AMB Query. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques

- [Public ciblé](#)
- [Authentification par des identités](#)
- [Gestion de l'accès à l'aide de politiques](#)
- [Comment fonctionne la requête Amazon Managed Blockchain \(AMB\) avec IAM](#)
- [Exemples de politiques basées sur l'identité pour la requête Amazon Managed Blockchain \(AMB\)](#)
- [Résolution des problèmes d'identité et d'accès aux requêtes Amazon Managed Blockchain \(AMB\)](#)

Public ciblé

La façon dont vous utilisez Gestion des identités et des accès AWS (IAM) varie en fonction de votre rôle :

- Utilisateur du service : demandez des autorisations à votre administrateur si vous ne pouvez pas accéder aux fonctionnalités (voir [Résolution des problèmes d'identité et d'accès aux requêtes Amazon Managed Blockchain \(AMB\)](#))

- Administrateur du service : déterminez l'accès des utilisateurs et soumettez les demandes d'autorisation (voir [Comment fonctionne la requête Amazon Managed Blockchain \(AMB\) avec IAM](#))
- Administrateur IAM : rédigez des politiques pour gérer l'accès (voir [Exemples de politiques basées sur l'identité pour la requête Amazon Managed Blockchain \(AMB\)](#))

Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié en tant qu'utilisateur IAM ou en assumant un rôle IAM.

Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en tant qu'identité fédérée à l'aide d'informations d'identification provenant d'une source d'identité telle que AWS IAM Identity Center (IAM Identity Center), d'une authentification unique ou d'informations d'identification. Google/Facebook Pour plus d'informations sur la connexion, consultez [Connexion à votre Compte AWS](#) dans le Guide de l'utilisateur Connexion à AWS .

Pour l'accès par programmation, AWS fournit un SDK et une CLI pour signer les demandes de manière cryptographique. Pour plus d'informations, consultez [Signature AWS Version 4 pour les demandes d'API](#) dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une seule identité de connexion appelée utilisateur Compte AWS root qui dispose d'un accès complet à toutes Services AWS les ressources. Il est vivement déconseillé d'utiliser l'utilisateur racine pour vos tâches quotidiennes. Pour les tâches qui requièrent des informations d'identification de l'utilisateur racine, consultez [Tâches qui requièrent les informations d'identification de l'utilisateur racine](#) dans le Guide de l'utilisateur IAM.

Identité fédérée

Il est recommandé d'obliger les utilisateurs humains à utiliser la fédération avec un fournisseur d'identité pour accéder à Services AWS l'aide d'informations d'identification temporaires.

Une identité fédérée est un utilisateur provenant de l'annuaire de votre entreprise, de votre fournisseur d'identité Web ou Directory Service qui y accède à Services AWS l'aide d'informations d'identification provenant d'une source d'identité. Les identités fédérées assument des rôles qui fournissent des informations d'identification temporaires.

Pour une gestion des accès centralisée, nous vous recommandons d'utiliser AWS IAM Identity Center. Pour plus d'informations, consultez [Qu'est-ce que IAM Identity Center ?](#) dans le Guide de l'utilisateur AWS IAM Identity Center .

Utilisateurs et groupes IAM

Un [utilisateur IAM](#) est une identité qui dispose d'autorisations spécifiques pour une seule personne ou application. Nous vous recommandons d'utiliser ces informations d'identification temporaires au lieu des utilisateurs IAM avec des informations d'identification à long terme. Pour plus d'informations, voir [Exiger des utilisateurs humains qu'ils utilisent la fédération avec un fournisseur d'identité pour accéder à AWS l'aide d'informations d'identification temporaires](#) dans le guide de l'utilisateur IAM.

[Les groupes IAM](#) spécifient une collection d'utilisateurs IAM et permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Pour plus d'informations, consultez [Cas d'utilisation pour les utilisateurs IAM](#) dans le Guide de l'utilisateur IAM.

Rôles IAM

Un [rôle IAM](#) est une identité dotée d'autorisations spécifiques qui fournit des informations d'identification temporaires. Vous pouvez assumer un rôle en [passant d'un rôle utilisateur à un rôle IAM \(console\)](#) ou en appelant une opération AWS CLI ou AWS API. Pour plus d'informations, consultez [Méthodes pour endosser un rôle](#) dans le Guide de l'utilisateur IAM.

Les rôles IAM sont utiles pour l'accès des utilisateurs fédérés, les autorisations temporaires des utilisateurs IAM, les accès intercompte, les accès entre services et les applications exécutées sur Amazon EC2. Pour plus d'informations, consultez [Accès intercompte aux ressources dans IAM](#) dans le Guide de l'utilisateur IAM.

Gestion de l'accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique définit les autorisations lorsqu'elles sont associées à une identité ou à une ressource. AWS évalue ces politiques lorsqu'un directeur fait une demande. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations les documents de politique JSON, consultez [Vue d'ensemble des politiques JSON](#) dans le Guide de l'utilisateur IAM.

À l'aide de politiques, les administrateurs précisent qui a accès à quoi en définissant 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. Un administrateur IAM crée des politiques IAM et les ajoute aux rôles, que les utilisateurs peuvent ensuite assumer. Les politiques IAM définissent les autorisations quelle que soit la méthode que vous utilisez pour exécuter l'opération.

Politiques basées sur l'identité

Les stratégies basées sur l'identité sont des documents de stratégie d'autorisations JSON que vous attachez à une identité (utilisateur, groupe ou rôle). Ces politiques contrôlent les actions que peuvent exécuter ces identités, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez [Définition d'autorisations IAM personnalisées avec des politiques gérées par le client](#) dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être des politiques intégrées (intégrées directement dans une seule identité) ou des politiques gérées (politiques autonomes associées à plusieurs identités). Pour découvrir comment choisir entre des politiques gérées et en ligne, consultez [Choix entre les politiques gérées et les politiques en ligne](#) dans le Guide de l'utilisateur IAM.

Politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Les exemples incluent les politiques de confiance de rôle IAM et les stratégies de compartiment Amazon S3. 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. Vous devez [spécifier un principal](#) dans une politique basée sur les ressources.

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

Autres types de politique

AWS prend en charge des types de politiques supplémentaires qui peuvent définir les autorisations maximales accordées par les types de politiques les plus courants :

- Limites d'autorisations : une limite des autorisations définit le nombre maximum d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM. Pour plus d'informations, consultez [Limites d'autorisations pour des entités IAM](#) dans le Guide de l'utilisateur IAM.

- Politiques de contrôle des services (SCPs) — Spécifiez les autorisations maximales pour une organisation ou une unité organisationnelle dans AWS Organizations. Pour plus d'informations, consultez [Politiques de contrôle de service](#) dans le Guide de l'utilisateur AWS Organizations .
- Politiques de contrôle des ressources (RCPs) : définissez le maximum d'autorisations disponibles pour les ressources de vos comptes. Pour plus d'informations, voir [Politiques de contrôle des ressources \(RCPs\)](#) dans le guide de AWS Organizations l'utilisateur.
- Politiques de session : politiques avancées que vous passez en tant que paramètre lorsque vous créez par programmation une session temporaire pour un rôle ou un utilisateur fédéré. Pour plus d'informations, consultez [Politiques de session](#) dans le Guide de l'utilisateur IAM.

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 déterminer s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, consultez la section [Logique d'évaluation des politiques](#) dans le guide de l'utilisateur IAM.

Comment fonctionne la requête Amazon Managed Blockchain (AMB) avec IAM

Avant d'utiliser IAM pour gérer l'accès à AMB Query, découvrez quelles fonctionnalités IAM peuvent être utilisées avec AMB Query.

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

Fonctionnalité IAM	Support des requêtes AMB
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

Fonctionnalité IAM	Support des requêtes AMB
ABAC (étiquettes dans les politiques)	Non
Informations d'identification temporaires	Oui
Autorisations de principal	Oui
Rôles du service	Non
Rôles liés à un service	Non

Pour obtenir une vue d'ensemble de la façon dont AMB Query et les autres AWS services fonctionnent avec la plupart des fonctionnalités IAM, consultez la section [AWS Services compatibles avec IAM dans le Guide de l'utilisateur IAM](#).

Politiques basées sur l'identité pour AMB Query

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

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez [Définition d'autorisations IAM personnalisées avec des politiques gérées par le client](#) dans le Guide de l'utilisateur IAM.

Avec les politiques IAM basées sur l'identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Pour découvrir tous les éléments que vous utilisez dans une politique JSON, consultez [Références des éléments de politique JSON IAM](#) dans le Guide de l'utilisateur IAM.

Exemples de politiques basées sur l'identité pour AMB Query

Pour consulter des exemples de politiques basées sur l'identité AMB Query, consultez. [Exemples de politiques basées sur l'identité pour la requête Amazon Managed Blockchain \(AMB\)](#)

Politiques basées sur les ressources dans AMB Query

Prend en charge les politiques basées sur les ressources : non

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Par exemple, les politiques de confiance de rôle IAM et les politiques de compartiment Amazon S3 sont des 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 permettre un accès intercompte, vous pouvez spécifier un compte entier ou des entités IAM dans un autre compte en tant que principal dans une politique basée sur les ressources. Pour plus d'informations, consultez [Accès intercompte aux ressources dans IAM](#) dans le Guide de l'utilisateur IAM.

Actions de politique pour AMB Query

Prend en charge les actions de politique : oui

Les administrateurs peuvent utiliser les politiques AWS JSON 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 `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Intégration d'actions dans une politique afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des actions de requête AMB, consultez la section [Actions définies par la requête Amazon Managed Blockchain \(AMB\) 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 AMB Query, consultez. [Exemples de politiques basées sur l'identité pour la requête Amazon Managed Blockchain \(AMB\)](#)

Ressources relatives aux politiques pour AMB Query

Prend en charge les ressources de politique : non

Les administrateurs peuvent utiliser les politiques AWS JSON 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 de politique JSON `Resource` indique le ou les objets auxquels l'action s'applique. Il est recommandé de définir une ressource à l'aide de son [Amazon Resource Name \(ARN\)](#). Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, 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 caractéristiques ARNs, consultez la section [Resources Defined by Amazon Managed Blockchain \(AMB\) Query](#) dans le Service Authorization Reference. Pour savoir avec quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez [Actions définies par Amazon Managed Blockchain \(AMB\) Query](#).

Pour consulter des exemples de politiques basées sur l'identité AMB Query, consultez. [Exemples de politiques basées sur l'identité pour la requête Amazon Managed Blockchain \(AMB\)](#)

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 les politiques AWS JSON 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` indique à quel moment les instructions s'exécutent en fonction de critères définis. 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. Pour voir toutes les clés de condition AWS globales, voir les clés de [contexte de condition AWS globales](#) dans le guide de l'utilisateur IAM.

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

Pour consulter des exemples de politiques basées sur l'identité AMB Query, consultez. [Exemples de politiques basées sur l'identité pour la requête Amazon Managed Blockchain \(AMB\)](#)

ACLs dans 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. ACLs sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

ABAC avec requête AMB

Prise en charge d'ABAC (balises dans les politiques) : non

Le contrôle d'accès par attributs (ABAC) est une stratégie d'autorisation qui définit les autorisations en fonction des attributs appelés balises. Vous pouvez associer des balises aux entités et aux AWS ressources IAM, puis concevoir des politiques ABAC pour autoriser les opérations lorsque la balise du principal correspond à la balise de la ressource.

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'informations sur ABAC, consultez [Définition d'autorisations avec l'autorisation ABAC](#) dans le Guide de l'utilisateur IAM. Pour accéder à un didacticiel décrivant les étapes de configuration de l'ABAC, consultez [Utilisation du contrôle d'accès par attributs \(ABAC\)](#) dans le Guide de l'utilisateur IAM.

Utilisation d'informations d'identification temporaires avec AMB Query

Prend en charge les informations d'identification temporaires : oui

Les informations d'identification temporaires fournissent un accès à court terme aux AWS ressources et sont automatiquement créées lorsque vous utilisez la fédération ou que vous changez de rôle. 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 [Informations d'identification de sécurité temporaires dans IAM](#) et [Services AWS compatibles avec IAM](#) dans le Guide de l'utilisateur IAM.

Autorisations principales interservices pour AMB Query

Prend en charge les sessions d'accès direct (FAS) : oui

Les sessions d'accès direct (FAS) utilisent les autorisations du principal appelant et Service AWS, combinées Service AWS à la demande d'envoi de demandes aux services en aval. Pour plus de détails sur la politique relative à la transmission de demandes FAS, consultez la section [Sessions de transmission d'accès](#).

Rôles de service pour AMB Query

Prend en charge les rôles de service : Non

Un rôle de service est un [rôle IAM](#) qu'un service endosse pour accomplir des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer un rôle de service à partir d'IAM. Pour plus d'informations, consultez [Création d'un rôle pour la délégation d'autorisations à un Service AWS](#) dans le Guide de l'utilisateur IAM.

Warning

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

Rôles liés à un service pour AMB Query

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 endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

Pour plus d'informations sur la création ou la gestion des rôles liés à un service, consultez [Services AWS 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 la requête Amazon Managed Blockchain (AMB)

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou à modifier les ressources AMB Query. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM.

Pour apprendre à créer une politique basée sur l'identité IAM à l'aide de ces exemples de documents de politique JSON, consultez [Création de politiques IAM \(console\)](#) dans le Guide de l'utilisateur IAM.

Pour plus de détails sur les actions et les types de ressources définis par AMB Query, y compris le ARNs format de chaque type de ressource, consultez la section [Actions, ressources et clés de condition pour la requête Amazon Managed Blockchain \(AMB\)](#) dans le Service Authorization Reference.

Rubriques

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

Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer des ressources AMB Query dans votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS 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 des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez [politiques gérées par AWS](#) ou [politiques gérées par AWS pour les activités professionnelles](#) dans le Guide de l'utilisateur IAM.
- Accordez les autorisations de moindre privilège : lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule 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 d'IAM pour appliquer des autorisations, consultez [politiques et autorisations dans IAM](#) dans le Guide de l'utilisateur IAM.
- Utilisez des conditions dans les politiques IAM pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service AWS, tel que CloudFormation. Pour plus d'informations, consultez [Conditions pour éléments de politique JSON IAM](#) dans le Guide de l'utilisateur IAM.
- Utilisez l'Analyseur d'accès IAM pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : l'Analyseur d'accès IAM valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez [Validation de politiques avec IAM Access Analyzer](#) dans le Guide de l'utilisateur IAM.
- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue. Compte AWS Pour exiger la MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez [Sécurisation de l'accès aux API avec MFA](#) dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez [Bonnes pratiques de sécurité dans IAM](#) dans le Guide de l'utilisateur IAM.

Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Cet exemple montre comment créer une politique qui permet aux utilisateurs IAM d'afficher les politiques en ligne et gérées attachées à leur identité d'utilisateur. Cette politique inclut les autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide de l'API AWS CLI or 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": "*"
    }
  ]
}
```

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

Note

Pour accéder à la requête AMB pour effectuer des appels d'API, vous aurez besoin d'informations d'identification utilisateur (AWS_ACCESS_KEY_ID et AWS_SECRET_ACCESS_KEY) disposant des autorisations IAM appropriées pour AMB Query.

Exemple Politique IAM pour accéder à toutes les requêtes Amazon Managed Blockchain (AMB) APIs

Cet exemple accorde à un utilisateur IAM l' Compte AWS accès à toutes les requêtes AMB. APIs

JSON

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

Exemple Politique IAM pour accéder à la requête Amazon Managed Blockchain (AMB) et **ListTransactions GetTransaction** APIs

Cet exemple accorde à un utilisateur IAM l' Compte AWS accès à la requête AMB et **ListTransaction GetTransaction** APIs

Note

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

JSON

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

Résolution des problèmes d'identité et d'accès aux requêtes Amazon Managed Blockchain (AMB)

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec AMB Query et IAM.

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 quand l'utilisateur IAM `mateojackson` tente d'utiliser la console pour afficher des informations détaillées sur une ressource `my-example-widget` fictive, mais ne dispose pas des autorisations `managedblockchain-query::GetWidget` 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 correspondent 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. Nonesera 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 (<code>transactionId</code>) et le hachage de transaction (<code>transactionHash</code>).	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 <code>status</code> propriété dans la réponse de l' <code>GetTransaction</code> opération. Vous utiliserez plutôt les <code>executionStatus</code> propriétés <code>confirmationStatus</code> et pour déterminer le statut de la transaction.	1er 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 execution Status champs confirmationStatus et pour déterminer si status la transaction est FINAL ouFAILED.	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.