How-To Library

Informatica

Configuring IAM Authentication for Amazon S3 and Amazon S3 V2 Connectors

Abstract

You can use AWS Identity and Access Management (IAM) to control individual and group access to Amazon S3 resources. You can configure AWS IAM to run tasks on the Secure Agent that is installed on the EC2 system. This article describes the guidelines to configure IAM Authentication for Amazon S3 and Amazon S3 V2 Connectors.

Supported Versions

- Informatica Cloud® Data Integration Amazon S3
- Informatica Cloud® Data Integration Amazon S3 V2

Table of Contents

Overview

To control the access of Amazon S3, you can define permissions to the users by configuring AWS Identity and Access Management (IAM). The AWS IAM service provides enhanced security.

Perform the following steps to configure IAM authentication:

1. Create a minimal Amazon S3 bucket policy. You can grant folder-level and object-level access to users.
2. Create the Amazon EC2 Role.
3. Create an Amazon S3 or Amazon S3 V2 connection.
4. Create a mapping.

Create a Minimal Amazon S3 Bucket Policy

The minimal Amazon S3 bucket policy restricts user operations and user access to a particular Amazon S3 bucket by assigning an AWS IAM policy to the users. You can configure the AWS IAM policy through the AWS console.

You can use the following minimum required actions when you use Amazon S3 Connector and Amazon S3 V2 Connector to successfully read data from and write data to Amazon S3 bucket:

- PutObject
- GetObject
- DeleteObject
- ListBucket
- GetBucketPolicy
- ListBucketMultipartUploads. Applicable only for mappings in advanced mode.

Note: Do not add the GetBucketPolicy permission in the Amazon S3 bucket policy when you use Amazon S3 V2 Connector. Amazon S3 V2 Connector does not support the GetBucketPolicy permission.

The following snippet shows a sample Amazon S3 bucket policy for Amazon S3 Connector:

```
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": [ "s3:PutObject", "s3:GetObject","s3:DeleteObject", "s3:ListBucket",
"s3:GetBucketPolicy" ],
"Resource": [ "arn:aws:s3:::<bucket_name>/*", "arn:aws:s3:::<bucket_name>" ]
}
]
}
```

The following snippet shows a sample Amazon S3 bucket policy for Amazon S3 V2 Connector:

```
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": [ "s3:PutObject", "s3:GetObject", "s3:DeleteObject", "s3:ListBucket" ],
"Resource": [ "arn:aws:s3:::<bucket_name>/*", "arn:aws:s3:::<bucket_name>" ]
}
]
}
```

Grant folder-level and object-level access

If you do not want to provide bucket-level access to the users, you can restrict the access by granting folder-level and object-level access. You can enable users to access only particular files or folders within a bucket by granting folder-level and object-level permissions.

For example, the following code snippet shows that the user can read a file `customers.csv` from `SubFolder1`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListBucketAccess",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::<bucket_name>"
            ]
        },
        {
            "Sid": "GetObjectAccessForFileInSubFolder",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::<bucket_name>/<Folder1>/<SubFolder1>/customers.csv"
            ]
        }
    ]
}
```

Additionally, users can read all files within a particular sub-folder. The following snippet shows a sample Amazon S3 bucket policy that allows users to read all files within `SubFolder1`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListBucketAccess",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::<bucket_name>"
            ]
        },
        {
            "Sid": "GetObjectAccessForSubFolder",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::<bucket_name>/Folder1/SubFolder1/*"
            ]
        }
    ]
}
```

You can enable read and write access to users where they can list, read from, write to, and delete data within particular files or folders.

The following snippet shows a sample Amazon S3 bucket policy that demonstrates how users can list, read, write data in multiple parts, and delete files from `SubFolder1` within the `bucket_name`:

```
 {
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListBucketAccess",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::<bucket_name>"
            ]
        },
        {
            "Sid": "S3WriteAccessToSpecificFolder",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:DeleteObject",
                "s3:ListBucketMultipartUploads"
            ],
            "Resource": [
                "arn:aws:s3:::<bucket_name>/<Folder1>/<SubFolder1>/*"
            ]
        }
    ]
}
```

When you enable access to a particular folder, the entire bucket is listed. In order to provide access to a particular object within a folder, you can define a condition to restrict the access to object-level to a particular folder. This will prevent listing of all the folders within the Amazon S3 bucket and provide a very secured access to a particular folder.

The following snippet shows a sample Amazon S3 bucket policy that demonstrates how users can access a particular object within a subfolder:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowListingOfUserFolder",
            "Action": [
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::infa.qa.minimum.access.bucket"
            ],
            "Condition": {
                "StringLike": {
                    "s3:prefix": [
                        "SubFolder_1/*"
                    ]
                }
            }
        },
        {
            "Sid": "S3OperationsFolderLevel",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:DeleteObject"
            ],
            "Resource": "arn:aws:s3:::infa.qa.minimum.access.bucket/SubFolder_1/*"
        }
    ]
}
```

The `StringLike` condition enables access to a particular sub-folder without listing the contents of all the folders within the S3 bucket. You can perform the operations at a folder-level instead of bucket-level.

For mappings in advanced mode, the StringLike condition is restricted to bucket-level access only. You can still access the objects within the folder, but the entire contents are listed. This is done by providing the `AllowListBucketMultipartUploads` permission at bucket level.

The following snippet shows a sample Amazon S3 bucket policy that demonstrates how users can access a particular object within a subfolder for mappings in advanced mode:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowListingOfUserFolder",
            "Action": [
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::infa.qa.minimum.access.bucket"
            ],
            "Condition": {
                "StringLike": {
                    "s3:prefix": [
                        "SubFolder_1/*"
                    ]
                }
            }
        },
        {
            "Sid": "AllowListBucketMultipartUploads",
            "Action": [
                "s3:ListBucketMultipartUploads"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::infa.qa.minimum.access.bucket"
            ]
        },
        {
            "Sid": "S3OperationsFolderLevel",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:DeleteObject",
                "s3:PutObjectTagging"
            ],
            "Resource": "arn:aws:s3:::infa.qa.minimum.access.bucket/SubFolder_1/*"
        }
    ]
}
```

Create the Amazon EC2 Role

You must create an Amazon EC2 Role to provide users access to the Amazon S3 resources. You can use the Amazon EC2 Role when you create an EC2 system in the Amazon S3 bucket.

1. Log in to the **AWS Console**.
2. Click **Dashboard** from the left panel.

   The **AWS Service** dashboard page appears.
3. Click **IAM**.

   The **Welcome to Identity and Access Management** page appears.
4. Click **Policies** from the left panel.

   The **Policy** page appears.
5. Click **Create Policy** or select the required existing Amazon S3 Policy.

   You can edit or review the policy.
6. Select **Role** from the left panel and click **Create role**.

   The **Create role** page appears.
7. Select **EC2** under the **Choose the service that will use this role** section.
8. Select the required **Amazon EC2** role type under the **Select your use case** section.
9. Click **Next: Permission**.
10. Select the required **Amazon S3 Policy** in the **Attach Permission Policies** page.
11. Click **Next: Review**.
12. Specify the name of the role in the **Create role** review page.
13. Click **Create Role**.
14. Review the Role ARN, Instance Profile ARNs, and Policy values in the **Summary** page.

After you create the Amazon EC2 Role, create an EC2 instance. Assign the Amazon EC2 Role to the EC2 instance.
For more information about creating an EC2 instance and assigning an Amazon EC2 Role to the Amazon EC2 instance, see the Amazon S3 documentation.

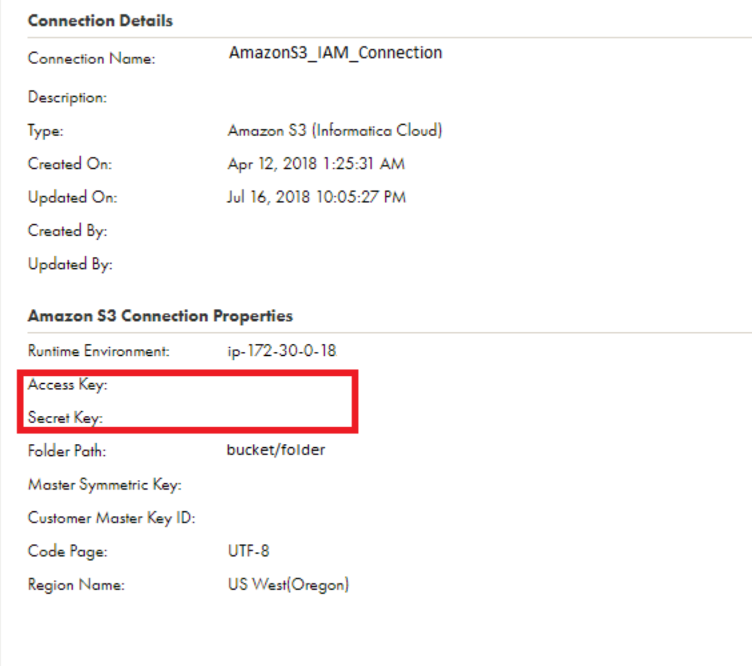
Create a Connection

You can create an Amazon S3 or Amazon S3 V2 connection. Specify the connection properties for configuring the IAM authentication to control secure access of Amazon S3 resources.

Create an Amazon S3 Connection

When you set up an Amazon S3 connection, you must configure the connection properties.

To run a mapping on Secure Agent installed on an EC2 system, you must not provide the **Access Key ID** and **Secret Access Key** when you create an Amazon S3 connection.
The following image shows the sample values in the Amazon S3 connection properties:



Connection Details

Connection Name:	AmazonS3_IAM_Connection
Description:	
Type:	Amazon S3 (Informatica Cloud)
Created On:	Apr 12, 2018 1:25:31 AM
Updated On:	Jul 16, 2018 10:05:27 PM
Created By:	
Updated By:	

Amazon S3 Connection Properties

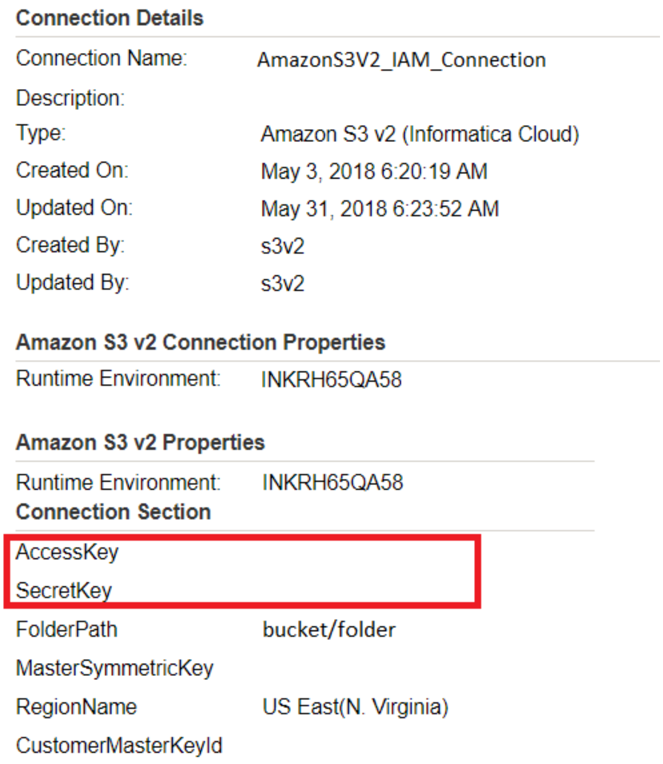
Runtime Environment:	ip-172-30-0-18
Access Key:	
Secret Key:	
Folder Path:	bucket/folder
Master Symmetric Key:	
Customer Master Key ID:	
Code Page:	UTF-8
Region Name:	US West(Oregon)

Create an Amazon S3 V2 Connection

When you set up an Amazon S3 V2 connection, you must configure the connection properties.

To run a mapping on Secure Agent installed on an EC2 system, you must not provide the **Access Key ID** and **Secret Access Key** when you create an Amazon S3 V2 connection.

The following image shows the sample values in the Amazon S3 V2 connection properties:



Connection Details

Connection Name:	AmazonS3V2_IAM_Connection
Description:	
Type:	Amazon S3 v2 (Informatica Cloud)
Created On:	May 3, 2018 6:20:19 AM
Updated On:	May 31, 2018 6:23:52 AM
Created By:	s3v2
Updated By:	s3v2

Amazon S3 v2 Connection Properties

Runtime Environment:	INKRH65QA58

Amazon S3 v2 Properties

Runtime Environment:	INKRH65QA58

Connection Section

AccessKey	
SecretKey	
FolderPath	bucket/folder
MasterSymmetricKey	
RegionName	US East(N. Virginia)
CustomerMasterKeyId	

Create a Mapping

Create a mapping to read data from a source and write data to an Amazon S3 target. Configure AWS IAM authentication for secure and controlled access to Amazon S3 resources when you run the mapping.

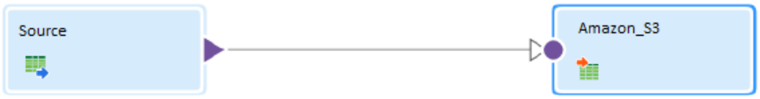
1. In Data Integration, click **New** > **Mappings** > **Create**.

   The **New Mapping** dialog box appears.
2. Enter a name, location, and description for the mapping.
3. On the Source transformation, specify a name and description in the general properties.
4. On the **Source** tab, perform the following steps to provide the source details to read data from the source:
   1. In the **Connection** field, select the source connection.
   2. In the **Source Type** field, select the type of the source.
   3. In the **Object** field, select the required object.
   4. In the **Advanced Properties** section, provide the appropriate values.
5. On the **Fields** tab, map the source fields to the target fields.
6. On the Target transformation, specify a name and description in the general properties.
7. On the **Target** tab, perform the following steps to provide the target details to write data to the Amazon S3 target:
   1. In the **Connection** field, select the required Amazon S3 target connection.
   2. In the **Target Type** field, select the type of the target.

    3. In the **Object** field, select the required object.
    4. In the **Operation** field, select the required operation.
    5. In the **Advanced Properties** section, provide appropriate values for the advanced target properties.

8. Map the source and the Amazon S3 target.

   The following image shows a sample mapping:



9. Click **Save** > **Run** to validate the mapping.

   In **Monitor**, you can monitor the status of the logs after you run the task.

10. Click **Action** > **New Mapping Task** on the left corner of the task wizard.

    The **Mapping Task** page appears.

11. Provide a name of the mapping task and select the runtime environment.

    The mapping that you created is selected automatically.

12. Click **Save** > **Run** to run the mapping task.

Author

Salam Subhashree

Acknowledgements

The author would like to acknowledge Shivaprasad Yallappagoudar, for his technical assistance with this article.