Enable AWS Glue Access

Contents:

If you have integrated with an EMR cluster version 5.8.0 or later, you can configure your Hive instance to use AWS Glue Data Catalog for storage and access to Hive metadata.

> **Tip:** For metastores that are used across a set of services, accounts, and applications, AWS Glue is the recommended method of access.

For more information on AWS Glue, see *https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-hive-metastore-glue.html*.

This section describes how to enable integration with your AWS Glue deployment.

Supported Deployment

AWS Glue tables can be read under the following conditions:

- The Trifacta platform uses S3 as the base storage layer.
- The Trifacta platform is integrated with an EMR cluster:
  - EMR version 5.8.0 or later
  - EMR cluster has been configured with HiveServer2
- The Hive deployment must be integrated with AWS Glue.

  > **NOTE:** Hive connections are supported when S3 is the backend datastore.

- For HiveServer2 connectivity, the Trifacta node has direct access to the Master node of the EMR cluster.

EMR Settings

When you create the EMR cluster, please verify the following in the AWS Glue Data Catalog settings:

- **`Use for Hive table metadata`**
- **`Use for Spark table metadata`**

Deploy Credentials JAR to S3

To enable integration between the Trifacta platform and AWS Glue, a JAR file for managing the Trifacta credentials for AWS access must be deployed to S3 in a location that is accessible to the EMR cluster.

When the EMR cluster is launched with the followng custom bootstrap action, the cluster does one of the following:

- Interacts with AWS Glue using the credentials specified in `trifacta-conf.json`

- If `aws.mode` = `user`, then the credentials registered by the user are used to connect to AWS Glue.

Steps:

1. From the installation of the Trifacta platform, retrieve the following file:

```
[TRIFACTA_INSTALL_DIR]/aws/glue-credential-provider/build/libs/trifacta-aws-glue-credential-provider.
jar
```

2. Upload this JAR file to an S3 bucket location where the EMR cluster can access it:
   a. **Via AWS Console S3 UI:** See *http://docs.aws.amazon.com/cli/latest/reference/s3/index.html*.
   b. **Via AWS command line:**

```
aws s3 cp trifacta-aws-glue-credential-provider.jar s3://<YOUR-BUCKET>/
```

3. Create a bootstrap action script named `configure_glue_lib.sh`. The contents must be the following:

```
sudo aws s3 cp s3://<YOUR-BUCKET>/trifacta-aws-glue-credential-provider.jar  /usr/share/aws/emr/emrfs
/auxlib/
sudo aws s3 cp s3://<YOUR-BUCKET>/trifacta-aws-glue-credential-provider.jar  /usr/lib/hive/auxlib/
```

4. This script must be uploaded into S3 in a location that can be accessed from the EMR cluster. Retain the full path to this location.
5. Add a bootstrap action to EMR cluster configuration.
   a. **Via AWS Console S3 UI:** Create the bootstrap action to point to the script that you uploaded on S3.
   b. **Via AWS command line:**
      i. Upload the `configure_glue_lib.sh` file to the accessible S3 bucket.
      ii. In the command line cluster creation script, add a custom bootstrap action. Example:

```
--bootstrap-actions '[
{"Path":"s3://<YOUR-BUCKET>/configure_glue_lib.sh","Name":"Custom action"}
]'
```

Authentication

Authentication methods are required permissions are based on the AWS authentication mode:

```
"aws.mode": "system",
```

aws.mode value	Permissions	Doc
`system`	IAM role assigned to the cluster must provide access to AWS Glue.	See *Configure for AWS*.
`user`	The user role must provide access to AWS Glue.	Se below for an example fine-grained access control. See *Configure AWS Per-User Authentication*.

Example fine-grain access control for IAM policy:

If you are using IAM roles to provide access to AWS Glue, you can review the following fine-grained access control, which includes the permissions required to access AWS Glue tables. Please add this to the Permissions section of your AWS Glue Catalog Settings page.

NOTE: Please verify that access is granted in the IAM policy to the default database for AWS Glue, as noted below.

```
{
    "Sid" : "accessToAllTables",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : [  "arn:aws:iam::<accountId>:role/glue-read-all" ]
    },
    "Action" : [ "glue:GetDatabases", "glue:GetDatabase", "glue:GetTables", "glue:GetTable", "glue:
GetUserDefinedFunctions", "glue:GetPartitions" ],
    "Resource" : [ "arn:aws:glue:us-west-2:<accountId>:catalog", "arn:aws:glue:us-west-2:<accountId>:database
/default", "arn:aws:glue:us-west-2:<accountId>:database/global_temp", "arn:aws:glue:us-west-2:<accountId>:
database/mydb", "arn:aws:glue:us-west-2:<accountId>:table/mydb/*" ]
}
```

Limitations

- Access is read-only. Publishing to Glue hosted on EMR is not supported.

Enable

Please verify the following have been enabled and configured.

1. Your deployment has been configured to meet the Supported Deployment guidelines above.
2. You must integrate the platform with Hive.

   NOTE: For the Hive hostname and port number, use the Master public DNS values. For more information, see *https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-hive-metastore-glue.html*.

   For more information, see *Configure for Hive*.
3. If you are using it, the custom SQL query feature must be enabled. For more information, see *Enable Custom SQL Query*.

Create Connection

You can create one or more connections to databases in your AWS Glue deployment.

Key fields:

Field	Description
EMR Master Node DNS	This DNS value can be retrieved from the EMR console.
Port	The port number through which to connect to the DNS master node
Connection String Options	No values need to be provided here.

- See *Create Connection Window*.
- See *Connections Page*.

Use

After the integration has been made between the platform and AWS Glue, you can import datasets.

- Browse for datasets through AWS Glue. See *AWS Glue Browser*.
- Import using custom SQL queries. For more information, see *Create Dataset with SQL*.