# Purpose

The purpose of this document is to record the process used to migrate from Databricks internal metastore to AWS Glue metastore. The goal is to ease the process for customers, make the transition smooth, and provide a path for data validation to retain integrity of the data.

# High level overview

* Databricks setup assumptions
  + AWS
  + Using internal metastore (beyond default)
  + Databases/Tables are specified as S3 paths
  + Delta is used extensively
  + Compatibility with Athena is desired
* Preparing the workspace for Glue
* Migrating metadata from internal metastore to glue
  + Differences between internal metastore and Glue
  + Migration script
    - <https://field-eng.cloud.databricks.com/#notebook/1403167/command/1403168>
    - Script courtesy of Nick Karpov/Darshan Pandya
* Validation
* Permanently setting Glue as the metastore

# Cluster migration

* <https://docs.databricks.com/data/metastores/aws-glue-metastore.html>
  + This document goes through setting up Glue and permissions
* Glue Catalog clusters need
  + IAM role that has permissions for Glue
  + This setting: spark.databricks.hive.metastore.glueCatalog.enabled true
* While still doing the migration, leave some clusters with that setting disabled to allow for metadata migration.

# Metadata migration

* Prerequisites
  + <https://docs.databricks.com/data/metastores/aws-glue-metastore.html#databricks-limitations>
  + Default schema in Databricks metastore is not accessible
    - Will need to create the “default” database manually
    - Set the location of default to be the S3 location of the DBFS root bucket /tables/ (TODO - get better URL)
* Migration
  + Exporting the DDL from the internal metastore
    - Use a cluster that is not glue catalog enabled and run the following notebook to extract the DDL
      * <https://field-eng.cloud.databricks.com/#notebook/1403167/command/1403168>
    - In the resulting string, transform DBFS paths to S3 paths
    - Manually create the default database and have it point to the same path on the S3 databricks root bucket
  + Importing the DDL
    - Take the resulting script generated in the above and run it on a cluster that is connected to Glue.
* Athena integration - <https://docs.databricks.com/delta/presto-integration.html>
  + Generate manifest
  + Generate external table that uses manifest (separate from delta table - different name)
  + Setup manifest updates

# Validation

* Take note of the core data tables and do a count using a cluster that is using the Databricks internal metastore
* Do a count on the same tables within a Glue cluster
* Ensure that the counts match to make sure all partitions are accounted for
* No additional checking is needed as the underlying data is the same (same S3 bucket and same S3 files)

# Productionalizing

* Use either a global init script or cluster policies (recommended)
* Global Init script
  + Use the following init script to permanently enable glue on all clusters
  + <https://demo.cloud.databricks.com/#notebook/5856841/command/5856843>
* Cluster policies (recommended)
  + In all cluster policies set the following spark config
    - spark.databricks.hive.metastore.glueCatalog.enabled true
  + Only allow create cluster using cluster policies
* Init script vs cluster policies
  + Init script
    - No need to configure policies
    - Applicable to all clusters
    - Must ensure all clusters use a IAM role that has access to Glue
    - Unable to use Databricks internal metastore without affecting everyone
    - Settings are not transparent (i.e. not visible from the UI)
  + Cluster policies
    - Need to make sure jobs using automated clusters set this setting as well
    - Admins can still use Databricks internal metastore by not using a policy while normal users are forced to use a policy
    - Settings are transparent