Databricks on AWS

Last updated: December 1, 2023

Migration guide for cluster-scoped init scripts from DBFS to Unity Catalog volumes

For Databricks Runtime 13.3 LTS and above with Unity Catalog, migrate init scripts from DBFS to Unity Catalog volumes. This guidance is for customers who use shared or single-user access modes with Unity Catalog enabled.

In general, you must do the following:

1. Create a managed volume and check that the cluster creator can access the volume.
2. Copy all your init scripts and files referenced by the init scripts from DBFS to your volume.
    a. [Optional] Modify init scripts to reference files on volumes if necessary.
3. Update cluster configuration and cluster policies to reference the init scripts on volumes.

Follow these steps to migrate init scripts from DBFS to Unity Catalog volumes.

1) Create a managed volume

Create a managed Unity Catalog volume and grant cluster creators READ_VOLUME permission on it.

- For UC shared clusters: Allow the init scripts to run from the created volume.

You can create a managed volume using SQL or the Databricks CLI. If you use the CLI, install it and configure the appropriate authentication method. Perform the following steps:

```
databricks volumes create <catalog> <schema> <volume> MANAGED
```

Grant cluster creators permissions to read files on your volume:

```
databricks grants update volume <catalog>.<schema>.<volume> --json '{
  "changes": [
     {
       "principal": "account users",
       "add": ["READ_VOLUME"]
     }
  ]
}
'
```

2) Copy all of your init scripts and files referenced by the init scripts from DBFS to your Unit Catalog volume.

Identify all objects (such as clusters, jobs, and DLT pipelines) that use init scripts on DBFS. Use the script detection notebook to prepare a list of individual init scripts and a list of their referenced files on DBFS.

To copy the init scripts and files referenced by the init scripts from DBFS to volumes you can upload the files via UI or use the Databricks CLI to perform the copying.

If using the CLI, do the following steps:

3) Copy file from the DBFS directly to a volume:

```
databricks fs cp 'dbfs:/FileStore/init-script.sh'
'dbfs:/Volumes/<catalog>/<schema>/<volume>/'
```

If you are accessing files such as configuration files, libraries, and other scripts from within the init scripts, you must copy them to the volume and update the paths. Use the same path string used for accessing volume content from the Spark cluster: /Volumes/<catalog>/<schema>/<volume>. For example:

```
#!/bin/bash

ls /Volumes/main/default/configs/
```

4) Update cluster configurations and cluster policies to reference the init scripts on cloud storage

Change the init script paths (more information) in clusters, jobs, Delta Live Tables pipelines, and cluster policies to point to the created volume instead of DBFS. (If you use the init scripts detection notebook (AWS, Azure, GCP), click links in the generated HTML tables):

- **Clusters**: edit your existing clusters: remove init scripts that use files on DBFS, and add init scripts with source "Volumes", providing the path to the file on Volume, like `/Volumes/main/default/init-scripts/init_script.sh`.
  Note: If you are using cluster policies, you must update both the cluster policy and the configurations of any other clusters using the policy. Cluster policy changes do not propagate to other clusters using that policy.
- **Jobs**: Edit the cluster definitions in each task that uses a dedicated job cluster and in each shared job cluster.

- **If using Cluster policies**: Edit the cluster policy. In the policy definition, search for blocks like the following, where N is the item number:

```
{
  "init_scripts.N.dbfs.destination": {
    "type": "fixed",
    "value": "dbfs:/FileStore/init-scripts/empty_init_script.sh"
  }
}
```

and replace them with the following, adjusting the file path:

```
{
  "init_scripts.N.volumes.destination": {
      "type": "fixed",
      "value": "/Volumes/main/default/init-scripts/init_script.sh"
  }
}
```

- **Delta Live Tables pipelines**: Ope (pipeline settings, select the "JSON" tab, and in the cluster definition(s), change entries in the `init_scripts` array as follows. Currently, init scripts from volumes are supported only in the PREVIEW channel!

```
{
 "dbfs": {
   "destination": "dbfs:/FileStore/init-scripts/empty_init_script.sh"
 }
}
```

to

```
{
 "volumes": {
   "destination": "/Volumes/main/default/init-scripts/init_script.sh"
 }
}
```