UCX Migration Runbook (Azure & AWS)

UCX Migration Runbook (Azure & AWS)

Common Prerequisites (Both Clouds)

1. **Install Databricks CLI (>= v0.213)**

```shell
brew install databricks
databricks -v
```

2. **Install UCX once on operator machine**

```shell
databricks labs install ucx -p profile
# optional: UCX_FORCE_INSTALL=account databricks labs install ucx
```

   - `UCX_FORCE_INSTALL=user` -> per-user
   - `UCX_FORCE_INSTALL=account` -> across all workspaces in account (requires **account profile**)

3. **Profiles needed**:
   - **Workspace profile** -> day-to-day UCX commands (`-p ws-<name>`).
   - **Account profile** -> group creation at account level, assigning a UC metastore.
   - **Cloud CLI**:
     - **Azure** -> `az login`

- **AWS** -> `aws configure --profile <aws_profile>`

Azure UCX Migration (End-to-End)

Step 0: Pre-requisite: Azure CLI installation

Step 1: Authenticate

- Databricks (workspace):

```shell
databricks auth login -p ws-azure
```

- Databricks (account, for group creation / metastore):

```shell
databricks auth login --account-id <account_id> -p acct-azure
```

- Azure CLI (for credentials/locations):

```shell
az login
az account set --subscription "<SUBSCRIPTION_ID>"
```

Step 2: Assessment

```shell
databricks labs install ucx -p ws-azure
```

```
databricks labs ucx -p ws-azure ensure-assessment-run
```

- Needs **workspace profile**
- Re-run after assigning UC metastore (account-level action)

Step 3: Group Migration

- **Account-level group creation** (requires account profile):

```
Shell
# workflow step, not CLI-only:
databricks labs ucx -p acct-azure create-account-groups   # uses
acct-azure profile
```

- **Workspace group validation & migration** (workspace profile):

```
Shell
databricks labs ucx -p ws-azure validate-groups-permissions
databricks labs ucx -p ws-azure migrate-groups
databricks labs ucx -p ws-azure
remove-workspace-local-backup-groups
```

Step 4: Table Migration

(All workspace profile unless noted)

```
Shell
databricks labs ucx -p ws-azure sync-workspace-info   # or
manual-workspace-info
```

```
databricks labs ucx -p ws-azure create-table-mapping
databricks labs ucx -p ws-azure principal-prefix-access
[--subscription-ids <Azure Subscription ID>]  # requires az login
databricks labs ucx -p ws-azure migrate-credentials       #
creates Access Connectors - requires az login
databricks labs ucx -p ws-azure migrate-locations # requires az
login
databricks labs ucx -p ws-azure create-uber-principal # requires
az login
databricks labs ucx -p ws-azure create-catalogs-schemas
databricks labs ucx create-ucx-catalog
databricks labs ucx -p ws-azure migrate-tables
```

Step 5: Code Migration

```
Shell
databricks labs ucx -p ws-azure lint-local-code
# optional:
# databricks labs ucx -p ws-azure migrate-local-code
# databricks labs ucx -p ws-azure migrate-dbsql-dashboards
```

Step 6: Cluster Remap

```
Shell
databricks labs ucx -p ws-azure cluster-remap
# revert if needed:
databricks labs ucx -p ws-azure revert-cluster-remap
```

Step 7: Debug / Operability

```
Shell
databricks labs ucx -p ws-azure workflows
databricks labs ucx -p ws-azure repair-run --run-id <id>
```

AWS UCX Migration

Step 1: Authenticate

- Databricks (workspace):

```shell
databricks auth login -p ws-aws
```

- Databricks (account, for group creation / metastore):

```shell
databricks auth login --account-id <account_id> -p acct-aws
```

- AWS CLI (for credentials/locations):

```shell
aws configure --profile <aws_profile>
export AWS_PROFILE=<aws_profile>
```

Step 2: Assessment

```shell
databricks labs ucx -p ws-aws ensure-assessment-run
```

- Needs **workspace profile**
- Re-run after metastore assignment (account profile step)

Step 3: Group Migration

- **Account-level group creation** (requires account profile):

```shell
create-account-groups   # uses acct-aws profile
```

- **Workspace group validation & migration**:

```shell
databricks labs ucx -p ws-aws validate-groups-permissions
databricks labs ucx -p ws-aws migrate-groups
databricks labs ucx -p ws-aws
remove-workspace-local-backup-groups
```

Step 4: Table Migration

(All workspace profile unless noted)

```shell
databricks labs ucx -p ws-aws sync-workspace-info   # or
manual-workspace-info
databricks labs ucx -p ws-aws create-table-mapping
databricks labs ucx -p ws-aws principal-prefix-access   #
requires AWS CLI profile
databricks labs ucx -p ws-aws migrate-credentials --aws-profile
<aws_profile>   # converts Instance Profiles -> UC storage
credentials
databricks labs ucx -p ws-aws migrate-locations
databricks labs ucx -p ws-aws create-uber-principal
databricks labs ucx -p ws-aws create-catalogs-schemas
databricks labs ucx -p ws-aws migrate-tables
```

Step 5: Debug / Operability

```shell
databricks labs ucx -p ws-aws workflows
databricks labs ucx -p ws-aws repair-run --run-id <id>
```

In case UCX doesnt work:
https://www.databricks.com/blog/migrating-tables-hive-metastore-unity-catalog-metastore

Issues and Solution Bank

UCX Issues and Solutions Bank

---

1. Workspace Listing Task Failed

Description:
During UCX assessment, the job run fails at the `Workspace_listing` task with `Invalid request path`.

Root Cause:
The UCX assessment workflow depends on workspace REST APIs. If one API call fails, the dashboard cannot populate completely.

Error Message:

```
None
Invalid request path
```

Solution:

1. Open the failed UCX job run in the Databricks Jobs UI.
2. Delete only the failed `Workspace_listing` task.
3. Repair the run using CLI:

```
Shell
databricks labs ucx -p <ws> repair-run --run-id <id>
```

4. Verify that the Assessment dashboard populates successfully.

---

2. UCX for Both eHMS and HMS

Description:
Customer uses both Databricks-provided Hive Metastore and an external HMS (eHMS).

Root Cause:
UCX can only manage one metastore per installation.

Solution:

1. Run separate UCX installations — one for Databricks HMS and one for eHMS.
2. Ensure each installation points to the correct metastore in `config.yml`.

---

3. UCX Cannot Run Successfully on Second Instance

Description:
UCX installation fails on a second workspace with “OAuth refresh token invalid”.

Root Cause:
OAuth device-flow tokens are workspace-scoped and not reused between workspaces.

Error Message:

```
None
ValueError: default auth: databricks-cli: cannot get access
token: token refresh failed
```

Solution:

1. Create a new workspace profile with host and PAT.
2. Update `~/.databricks/config` with the new profile.
3. Re-install UCX:

```
Shell
databricks labs install ucx -p <ws-2>
```

---

4. Customer Using Sedona Library

Description:
Sedona or geospatial libraries throw errors on UC clusters.

Root Cause:
Sedona JAR files conflict with Py4J security when Unity Catalog is enabled.

Solution:

1. Run Sedona workloads on Single-User clusters.
2. Disable Py4J security:

```shell
spark.databricks.pyspark.enablePy4JSecurity=false
```

---

5. Init Script Fails After UC Upgrade

Description:
Cluster-scoped init scripts in Volumes path fail with “Permission denied”.

Root Cause:
UC migration adds metastore allowlists; Volumes path may be blocked.

Solution:

1. Enable cluster logging and inspect stderr.
2. Add the Volumes path to the metastore allowlist.
3. Ensure cluster owner is set correctly.
4. Restart the cluster.

---

6. Azure WASBS Paths Unsupported

Description:
UCX throws `UNSUPPORTED_FILE_SCHEME` when tables use `wasbs://` mounts.

Root Cause:
Unity Catalog supports only `abfss://` (ADLS Gen2).

Error Message:

```
None
UPGRADE_NOT_SUPPORTED.UNSUPPORTED_FILE_SCHEME
```

Solution:

1. Remount storage with `abfss://`.
2. For non-Gen2 data, clone tables via:

```
SQL
CREATE TABLE new_tbl CLONE old_tbl;
```

3. Re-run table migration.

---

7. UCX Assessment Not Showing All Tables

Description:
Assessment dashboard misses mounted databases.

Root Cause:
Default UCX config limits scope to DBFS root.

Solution:

1. Edit config:

```shell
databricks labs ucx -p <ws> open-remote-config
```

2. Add `included_databases`.
3. Re-run:

```shell
databricks labs ucx -p <ws> ensure-assessment-run
```

---

8. Unable to Authenticate Login

Description:
`databricks auth login --host` fails locally due to blocked device-flow.

Root Cause:
Local environment cannot open browser-based auth flow.

Solution:

1. Use Databricks Web Terminal to authenticate.
2. Or create a PAT and add manually to `~/.databricks/config`.

---

9. Crawl Group Issue

Description:
Group crawl task skipped during assessment.

Root Cause:
Old UCX versions skip dependency tasks.

Solution:

1. Upgrade UCX to latest.
2. Re-run assessment:

```shell
databricks labs ucx -p <ws> ensure-assessment-run
```

---

10. Regular Pip Not Allowed

Description:
Environment blocks `pip install`.

Root Cause:
Corporate proxy or no Internet egress.

Solution:

1. Set internal PyPI mirror:

```shell
pip config set global.index-url <internal_url>
```

2. Retry UCX installation.

---

11. No Internet Allowed on Host

Description:
CLI cannot download UCX packages.

Root Cause:
Network restrictions or no egress NAT.

Solution:

1. Download UCX release ZIP manually from GitHub.
2. Install offline:

```shell
databricks labs install ucx --offline
```

3. Use internal PyPI mirror for dependencies.

---

12. User Lacks IAM Roles (AWS)

Description:
UCX job clusters fail with multiple instance profiles.

Root Cause:
Conflicting IAM roles.

Solution:

1. Use “No Isolation Shared” cluster temporarily.
2. Verify IAM policy alignment.

---

13. Outdated CLI – No Such Command ‘labs’

Description:
Old Databricks CLI does not recognize `labs`.

Error Message:

```none
databricks: 'labs' is not a databricks command
```

Solution:

Check is databricks cli version is older than 0.213

```shell
databricks -v
brew upgrade databricks
databricks labs install ucx
```

---

14. Quota Limits Hit During UCX Jobs

Description:
UCX job creation fails with quota errors.

Root Cause:
Workspace compute quota exceeded.

Solution:

1. Lower max workers in UCX job cluster (e.g. 2–4).
2. Retry workflow.

---

15. Principal-Prefix-Access (Azure)

Description:
Fails when Azure CLI not authenticated or subscription IDs missing.

Error Message:

```
d.l.ucx.principal-prefix-access] ValueError: In order to obtain AAD token, Please run azure cli to authenticate.
ValueError: In order to obtain AAD token, Please run azure cli to authenticate.
ValueError: Please enter subscription ids to scan storage accounts in.
```

Solution:

```shell
az login
az account set --subscription <subscription-id>
set DATABRICKS_AUTH_TYPE=azure-cli
databricks labs ucx -p ws-azure principal-prefix-access
--subscription-ids <subscription-id>
```

---

16. Principal-Prefix-Access (AWS)

Description:
Fails if AWS CLI not configured or profile missing.

Error Message:

```none
ValueError: Missing AWS credentials/profile
```

Solution:

```shell
aws configure --profile <aws_profile>
export AWS_PROFILE=<aws_profile>
databricks labs ucx -p ws-aws principal-prefix-access
--aws-profile <aws_profile>
```

---

19. Migrate-Locations PermissionDenied

Description:
Fails when user lacks `CREATE EXTERNAL LOCATION`.

Error Message:

```
None
PermissionDenied: User does not have CREATE EXTERNAL LOCATION on
Metastore
```

Solution:

1. Grant user `CREATE EXTERNAL LOCATION` privilege.
2. Re-run:

```
Shell
databricks labs ucx -p <ws> migrate-locations
```

---

20. Create-Catalogs-Schemas (Windows Path Parsing)

Description:
Windows shell misinterprets `abfss://` URIs.

Error Message:

```
None
'abfss:' is not recognized as an internal or external command
```

Solution:
Use PowerShell or escape colon:

```
Shell
abfss^://container@storageaccount.dfs.core.windows.net/path
```

---

21. CLI Path Misconfiguration

Description:
Databricks CLI cannot locate config or binary.

Error Message:

```
None
auth_type = DATABRICKS_CLI
```

Solution:

1. Set env variable:

```
Shell
set
DATABRICKS_CLI_PATH=C:\Users\<user>\AppData\Roaming\Python\Script
s
```

2. Verify CLI:

```
Shell
databricks workspace ls
```

---

22. Users Still Query Hive Metastore After Cutover

Description:
Queries continue hitting `hive_metastore` after UC migration.

Root Cause:
Hive Metastore access not disabled.

Solution:
Disable Hive Metastore access in workspace admin UI or enable HMS federation only for legacy tables.
