📖 **05-metastore-creation.md**

🔗Creating a Cloud Metastore

This module includes all prerequisites for setting up the Cloud Metastore for the Serverless Spark Lab-

🔗1. Prerequisites

🔗1.1. GCP Project Details

Note the project number and project ID.
We will need this for the rest of the lab

🔗1.2. IAM Roles needed to create metastore

🔗Grant the following permissions

- Dataproc Editor
- Service Account User

🔗2. Declaring Variables

🔗2.1 Set the PROJECT_ID in Cloud Shell

Open Cloud shell or navigate to shell.cloud.google.com
Run the below

```
gcloud config set project $PROJECT_ID
```

🔗2.2 Verify the PROJECT_ID in Cloud Shell

Next, run the following command in cloud shell to ensure that the current project is set correctly:

```
gcloud config get-value project
```

2.3 Declare the variables

Based on the prereqs and checklist, declare the following variables in cloud shell by replacing with your values:

```
METASTORE_NAME=                       # Name of the metastore Service to be created.
REGION=                               # Region in which metastore service is to be created.
VPC=                                  # The VPC network to be used.
port=9083                             # Change the port number as per the Requirement.
tier=Developer                        # Change the tier as per the Requirement.
Metastore_Version=3.1.2               # Change the metastore version as per the Requirement,
```

3. Create a metastore

3.1. Create a metastore through the google cloud shell

```
gcloud metastore services create $METASTORE_NAME \
    --location=$REGION \
    --network=$VPC \
    --port=$port \
    --tier=$tier \
    --hive-metastore-version=$Metastore_Version
```

3.2. Create a metastore through the GCP console

Navigate to the Dataproc Service in your GCP project and click on Metastore Click **+CREATE>Composer 2**

Dataproc

Dataproc Metastore   ➕ CREATE   🗑 DELETE

Jobs on clusters ^

⊹ Clusters

☰ Jobs

⯐ Workflows

ılı Auto-scaling policies

Serverless ^

☰ Batches

▊▋ Sessions

Utilities ^

⊞ Component exchange

◈ Metastore

⋈ Workbench

≡ Filter Filter services   ❓

☐	Service name ↑	Status	Created	Metastore version	Location	Network configuration	URI
☐		✅ ACTIVE	17 May 2022, 14:26:03	3.1.2	us-central1		thrift://10.140.192.15:9083

← **Create service**

Service info

Service name *
service-f86d   ❓

Data location *
us-central1   ▼   ❓

Metastore version
3.1.2   ▼   ❓

Release channel
Stable   ▼   ❓

Port
9083   ❓

Service tier *
Developer   ▼   ❓

Network configuration

Use a network on which the service can be accessed. The network can belong to the
same project or a different project.

🔘 Expose service in 1 VPC network.

☐ Use shared VPC network

Network *
default   ▼   ❓

Next, fill in the following values in the metastore creation window :

- **Service name** - A unique identifier for your environment

- **Data location** - The region where you want to create the metastore

- **Metastore Version** - #default

- **Release channel** - #default

- **Port** - #default

- **Service tier** - #default

- **Network Configuration** - select the network and subnetwork with Private Google Access Enabled

- Next under **Endpoint protocol** select one of the below options:
  Thrift gRPC

- Click on **ADD LABELS** to attach the labels.

- Next, click on **Create** to create the Metastore.

4. Metastore logs

To view the metastore logs, click the 'View Logs' button on the metastore page and the logs will be shown as below:

Operations
Logging

Logs Explorer   **OPTIONS** ▾   ⊙ REFINE SCOPE   Project   ⊖ SHARE LINK   ◐ LEARN

Query   Recent (1)   Saved (0)   Suggested (4)   Library   🗑 Clear query   💾 Save   Stream logs   **Run query**

⊙ Last 1 hour   🔍 Search all fields   Dataproc Metastore serv... +2 ▾   Log name ▾   Severity ▾   +1 filter   ⚪ Show query

⚪ Log fields   ⚪ Histogram   ⊞ Create metric   🔔 Create alert   ⊙ Jump to now   More actions ▾

Query results ~500 log entries   ⬇ Download   ⛶

SEVERITY	TIMESTAMP ↑	IST ▾	SUMMARY ✎ EDIT
› ⓘ	2022-05-17 15:44:36.822 IST		map[class:listener.RequestErrorListener log:Request: get_databases, latency: 25ms thread:pool-15-thread-170]
› ⓘ	2022-05-17 15:44:50.411 IST		map[class:HiveMetaStore.audit log:ugi=hive ip=10.140.129.202 cmd=Cleaning up thread local RawStore... thread:pool-15-thread-170]
› ⓘ	2022-05-17 15:44:50.411 IST		map[class:metastore.HiveMetaStore log:176: Done cleaning up thread local RawStore thread:pool-15-thread-170]
› ⓘ	2022-05-17 15:44:50.411 IST		map[class:metastore.HiveMetaStore log:176: Cleaning up thread local RawStore... thread:pool-15-thread-170]
› ⓘ	2022-05-17 15:44:50.411 IST		map[class:HiveMetaStore.audit log:ugi=hive ip=10.140.129.202 cmd=Done cleaning up thread local RawStore thread:pool-15-thread-170]
› ⓘ	2022-05-17 15:45:04.815 IST		map[class:metastore.HiveMetaStore log:177: source:127.0.0.1 get_database: default thread:pool-15-thread-171]
› ⓘ	2022-05-17 15:45:04.816 IST		map[class:HiveMetaStore.audit log:ugi=hive ip=127.0.0.1 cmd=source:127.0.0.1 get_database: default thread:pool-15-thread-171]
› ⚠	2022-05-17 15:45:04.818 IST		map[class:metastore.ObjectStore log:datanucleus.autoStartMechanismMode is set to unsupported value null . Setting it to value: ignored thread:pool-15-thread-171]
› ⓘ	2022-05-17 15:45:04.818 IST		map[class:metastore.HiveMetaStore log:177: Opening raw store with implementation class:org.apache.hadoop.hive.metastore.ObjectStore thread:pool-15-thread-171]
› ⓘ	2022-05-17 15:45:04.819 IST		map[class:metastore.ObjectStore log:ObjectStore, initialize called thread:pool-15-thread-171]
› ⓘ	2022-05-17 15:45:04.832 IST		map[class:metastore.MetaStoreDirectSql log:Using direct SQL, underlying DB is MYSQL thread:pool-15-thread-171]
› ⓘ	2022-05-17 15:45:04.832 IST		map[class:metastore.ObjectStore log:Initialized ObjectStore thread:pool-15-thread-171]
› ⓘ	2022-05-17 15:45:04.838 IST		map[class:listeners.SyncIamListener log:Going forward with RemoveIamPolicy execution, received root url: https://metastore.googleapis.com thread:pool-15-thread-171]