Amazon Web Services

Data Engineering Immersion Day

Lab 4. AWS Lake Formation

July 2021

Table of Contents

Introduction

This lab will give you an understanding of the AWS Lake Formation – a service that makes it easy to set up a secure data lake, as well as Athena for querying the data you import into your data lake.

Prerequisites

1. Make sure you have the Postgres source database information from your Event host. If you are running the lab outside of AWS hosted event, please find the **DMSInstanceEndpoint** parameter value from **dmslab-instructor** CloudFormation **Outputs** tab.
2. Complete Lab1. Hydrating the Data Lake with DMS or Lab1. Copy Source Data
3. Must completed Part A in Lab2.Transforming the Data with Glue

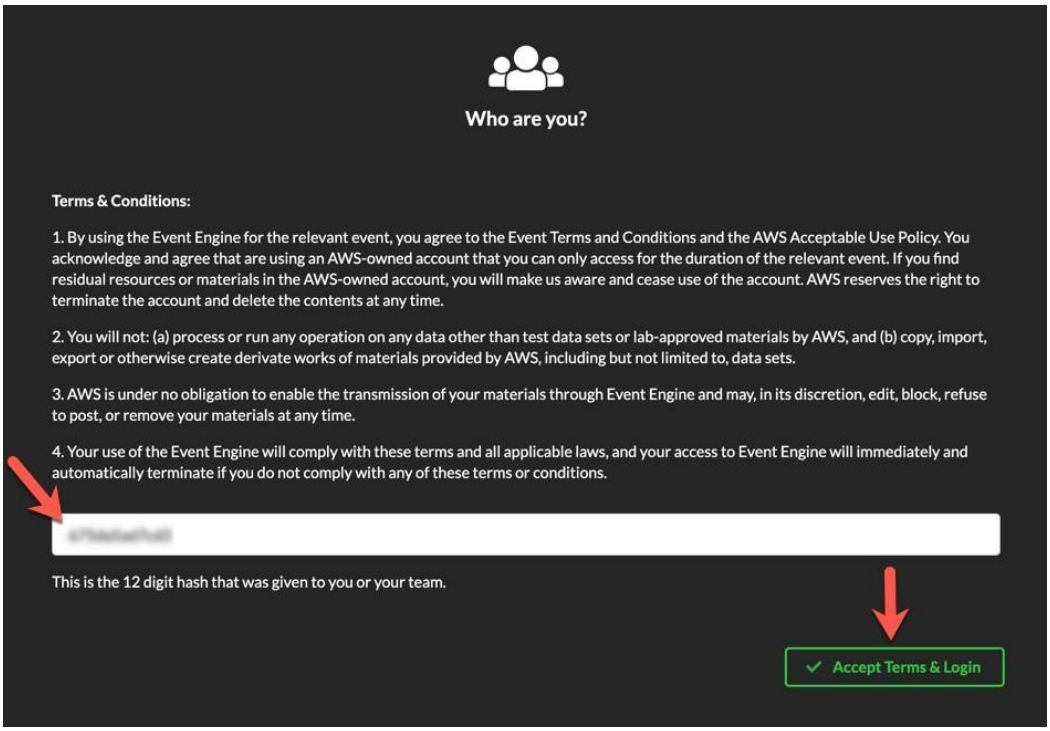
Get Started Using the Lab Environment

Please skip this section if you are running the lab on your own AWS account.

Today, you are attending a formal event and you will have been sent your access details beforehand. If in the future you might want to perform these labs in your own AWS environment by yourself, you can follow instructions on GitHub - https://github.com/awssamples/data-engineering-for-aws-immersion-day.

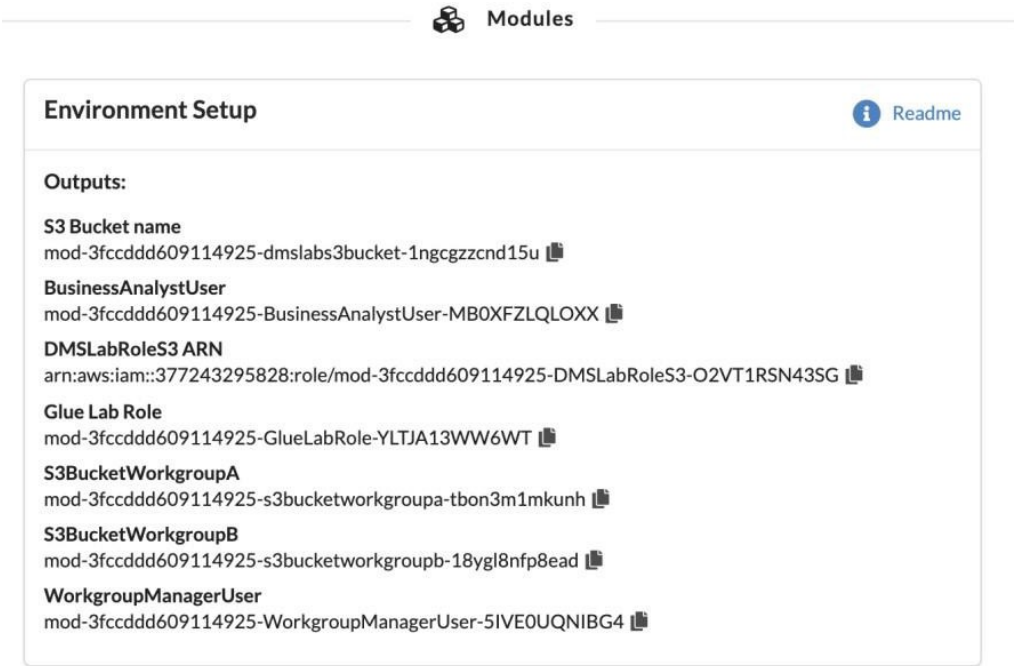
A 12-character access code (or 'hash') is the access code that grants you permission to use a dedicated AWS account for the purposes of this workshop.

1. Go to https://dashboard.eventengine.run/, enter the access code and click Proceed:

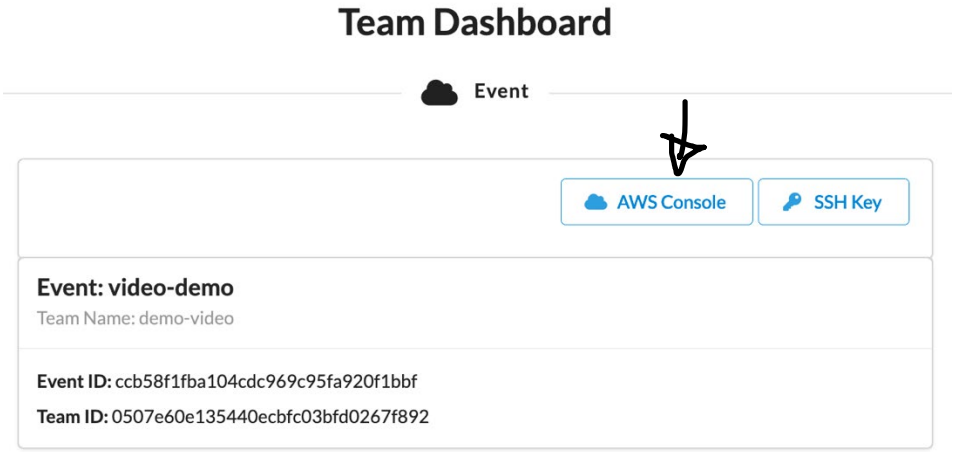


2. On the Team Dashboard web page, you will see a set of connection strings and parameters that you will need during the labs. Best to save them to a text file locally, alternatively you can always go to this page to review them. Replace the parameters with the corresponding values from here were indicated in subsequent labs:

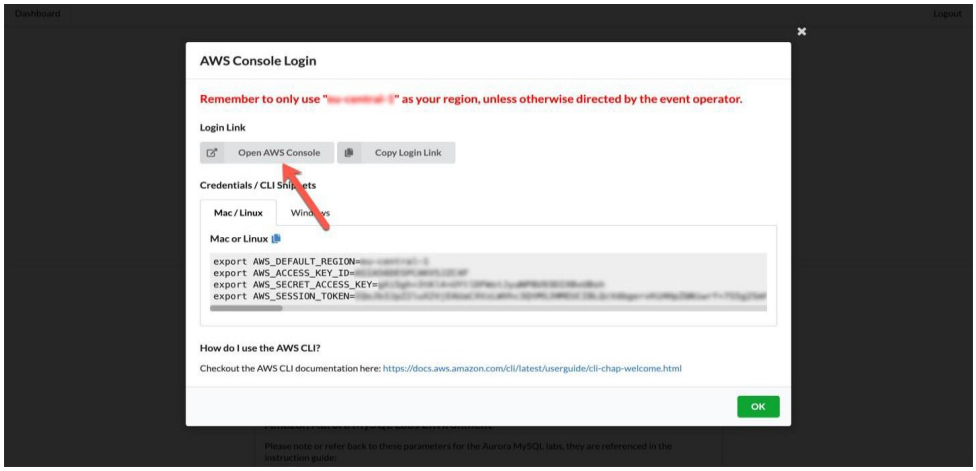
   Because you're at a formal event, some AWS resources have been pre-deployed for your convenience, for example:



3. On the Team Dashboard, please click AWS Console to log into the AWS Management Console:



4. Click Open Console. For the purposes of this workshop, you will not need to use command line and API access credentials.



Once you have completed these steps, you can continue with the rest of this lab.

Setup Network Configuration for AWS Glue (for your read)

If you use Amazon Virtual Private Cloud (Amazon VPC) to host your AWS resources, you can establish a private connection between your VPC and AWS Glue. You use this connection to enable AWS Glue to communicate with the resources in your VPC without going through the public internet.

Amazon VPC is an AWS service that you can use to launch AWS resources in a virtual network that you define. With a VPC, you have control over your network settings, such the IP address range, subnets, route tables, and network gateways. To connect your VPC to AWS Glue, you define an interface VPC endpoint for AWS Glue. When you use a VPC interface endpoint,

communication between your VPC and AWS Glue is conducted entirely and securely within the AWS network.

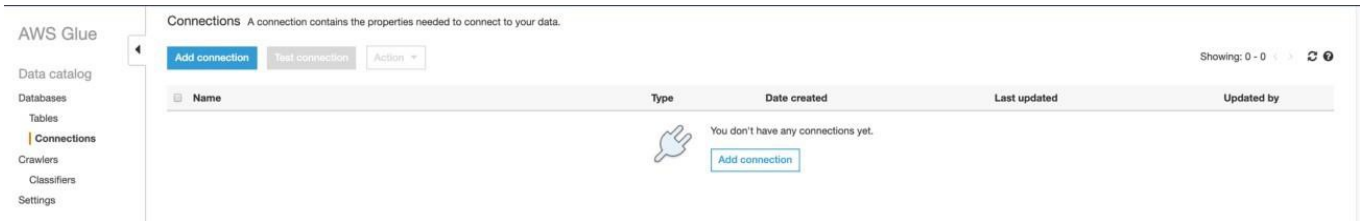
Create an IAM role to use with Lake Formation (for your read)

With AWS Lake Formation, you can import your data using *workflows*. A workflow defines the data source and schedule to import data into your data lake. You can easily define workflows using *blueprints*, or templates, that Lake Formation provides.

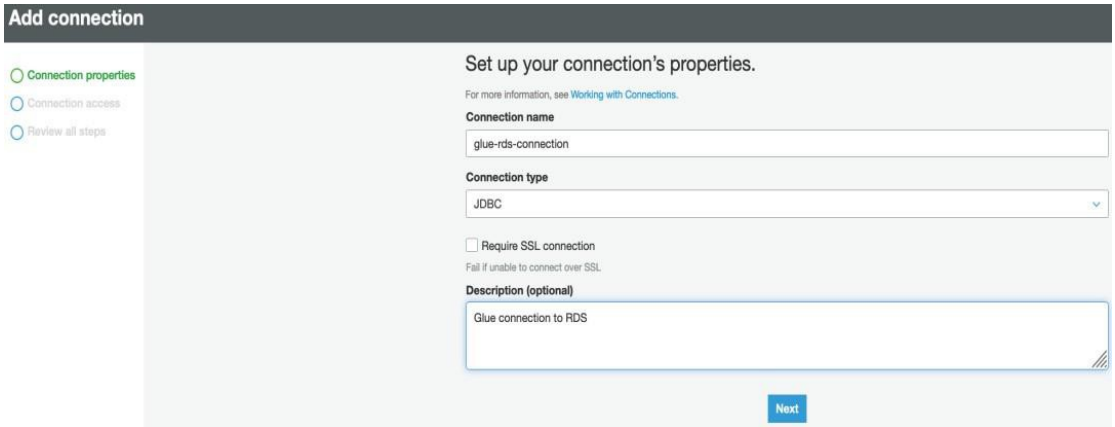
When you create a workflow, you must assign it an AWS Identity and Access Management (IAM) role that enables Lake Formation to set up the necessary resources on your behalf to ingest the data. In this lab,
we've pre-created an IAM role for you, called **<random>-LakeFormationWorkflowRole<random>**

Create Glue JDBC connection for RDS

1. Navigate to the AWS Glue console: https://console.aws.amazon.com/glue/home
2. On the AWS Glue menu, select **Connections**.



3. Click **Add Connection**.
4. Enter **glue-rds-connection** as the connection name.
5. Choose **JDBC** for connection type.
6. Optionally, enter the description. This should also be descriptive and easily recognized and Click **Next**.

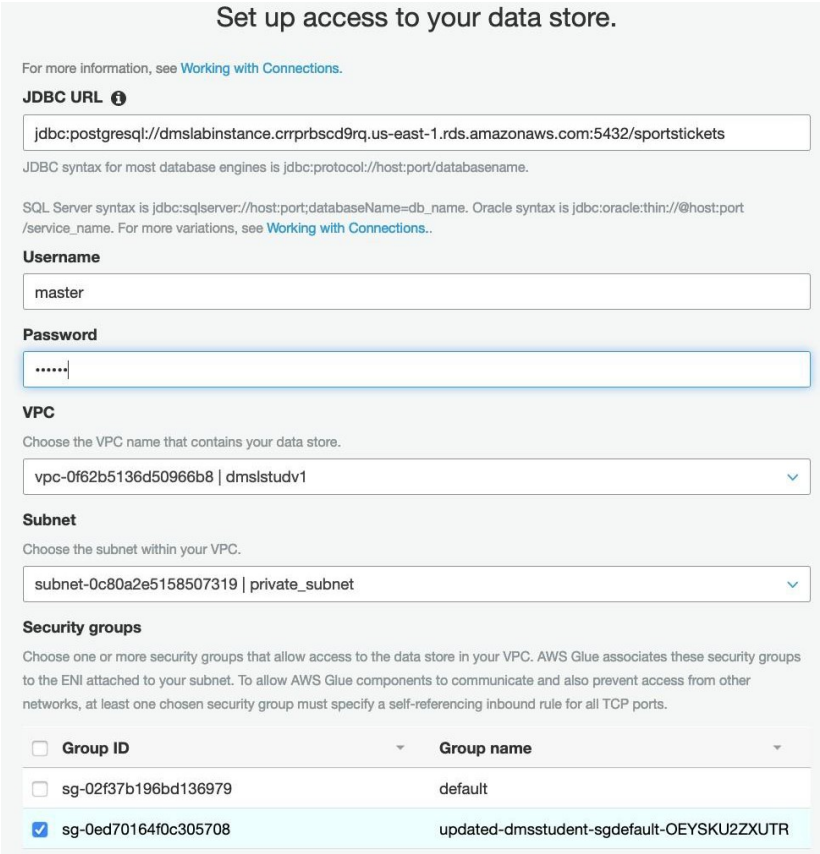


7. Input **JDBC URL** with the format of
   jdbc:postgresql://<RDS_Server_Name>:5432/sportstickets

   a. Get the **Database Endpoint** from your Event Engine Team dashboard.

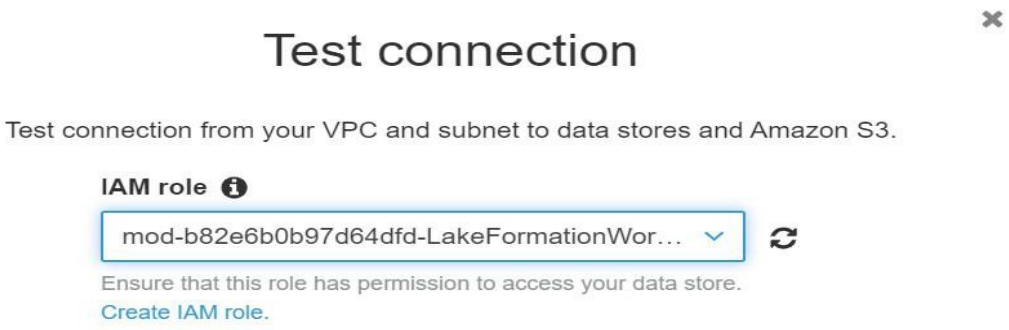
b. If you are running the lab outside of AWS event, find the **DMSInstanceEndpoint** value on the CloudFormation stack **dmslab-instructor Outputs** tab.

8. Enter **master** as username, **master123** as **Password**
9. For **VPC**, select the pre-created VPC ending with **dmslstudv1**
10. For **Subnet,** choose one of **private_subnet**
11. Select the **security group** with **sgdefault** in the name.



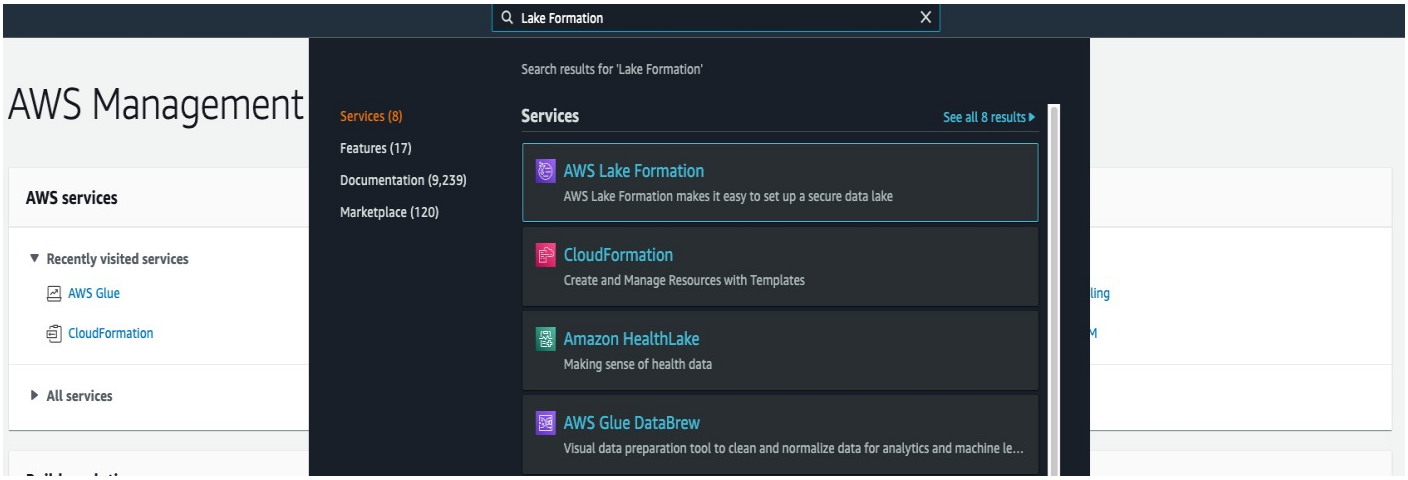
12. Click **Next to** complete the **glue-rds-connection** setup. To test it, select the connection, and choose **Test connection**.

13. Choose the pre-created **IAM role** (looks like **<random>-LakeFormationWorkflowRole<random>**), then click **Test Connection.**

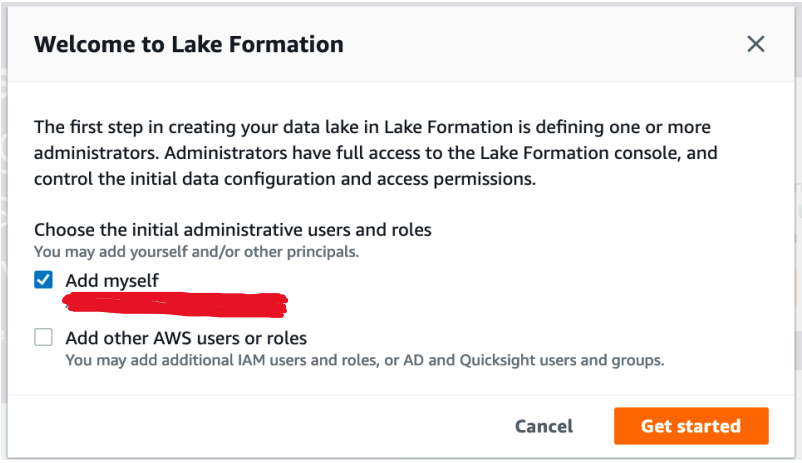


Lake Formation – Add Administrator and start workflows using Blueprints.

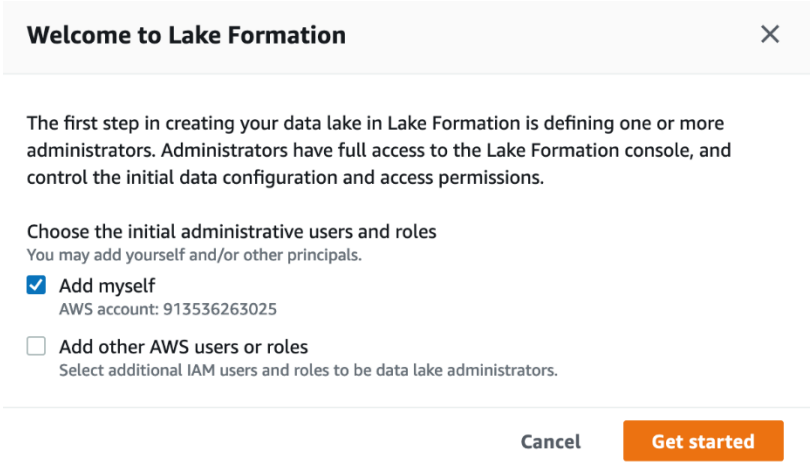
Navigate to the AWS Lake Formation service



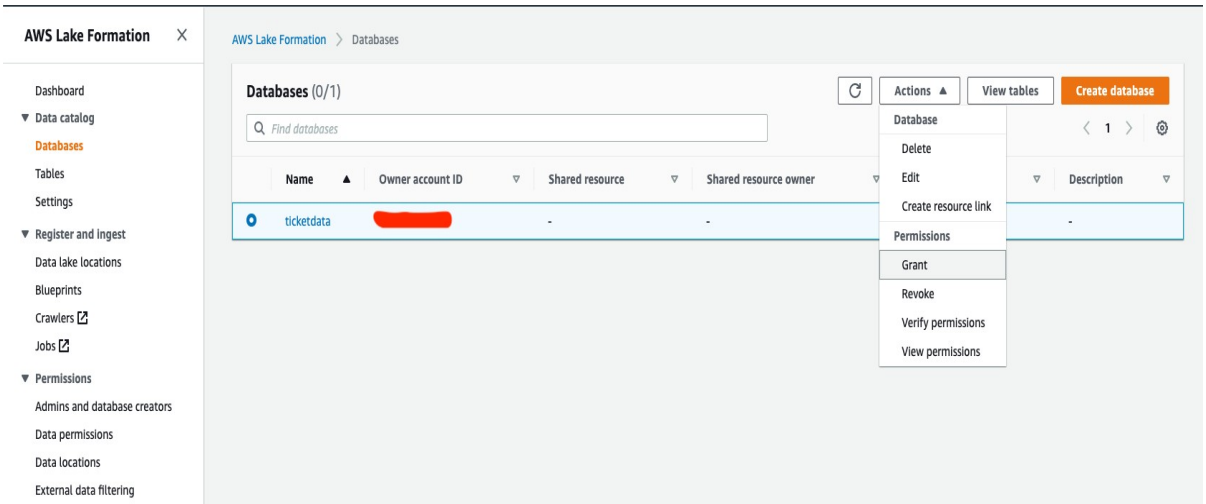
1. If you are logging into the lake formation console for the first time, you will see the window pop up. In order to do that follow Steps 2 and 3, else skip to Step 4.



2. Add **myself** as the Lake Formation Administrator and Click **Get started**

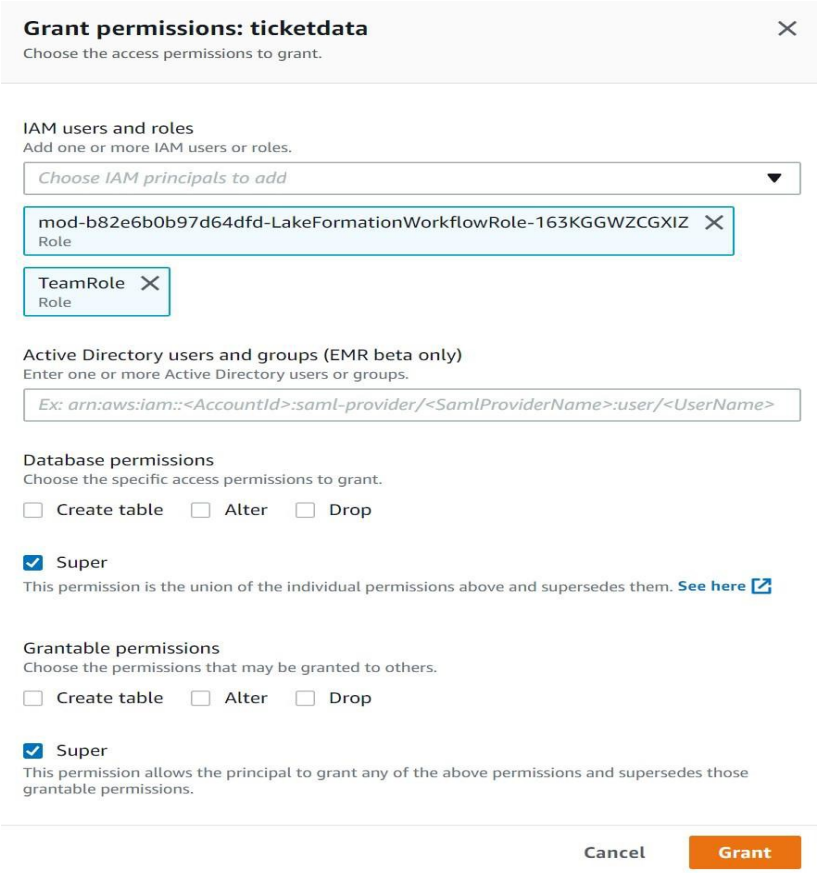


3. Navigate to **Databases** on left pane. Select **ticketdata** and click on **Actions**, select **Grant** to grant permissions. If you can't see any databases, make sure to complete

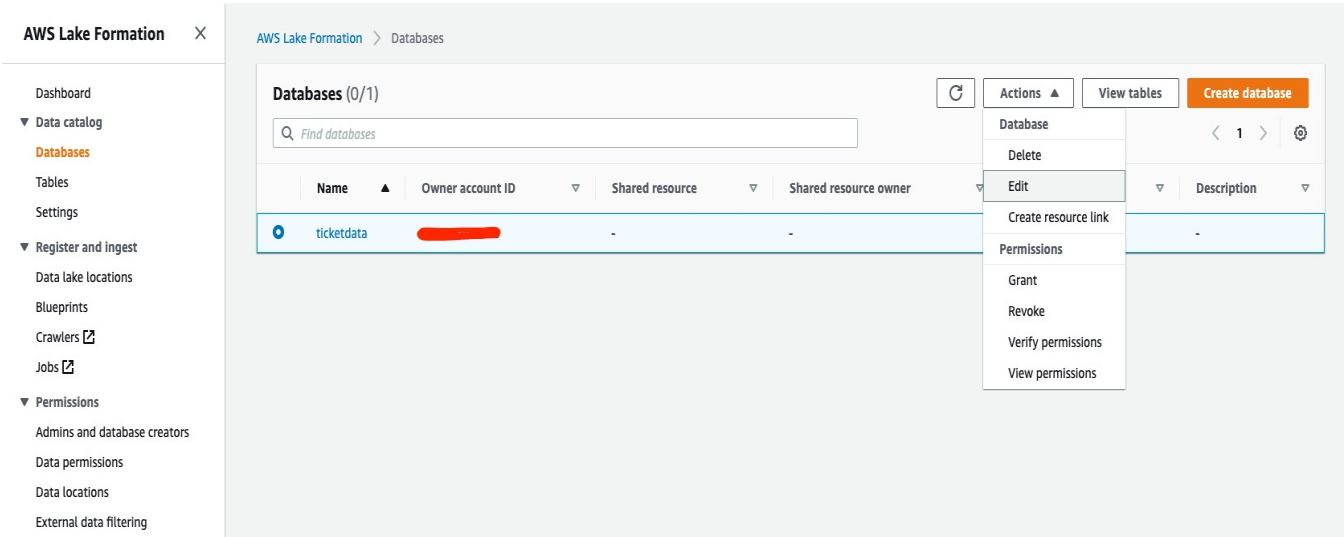


Part A of Lab 2. ETL with AWS Glue

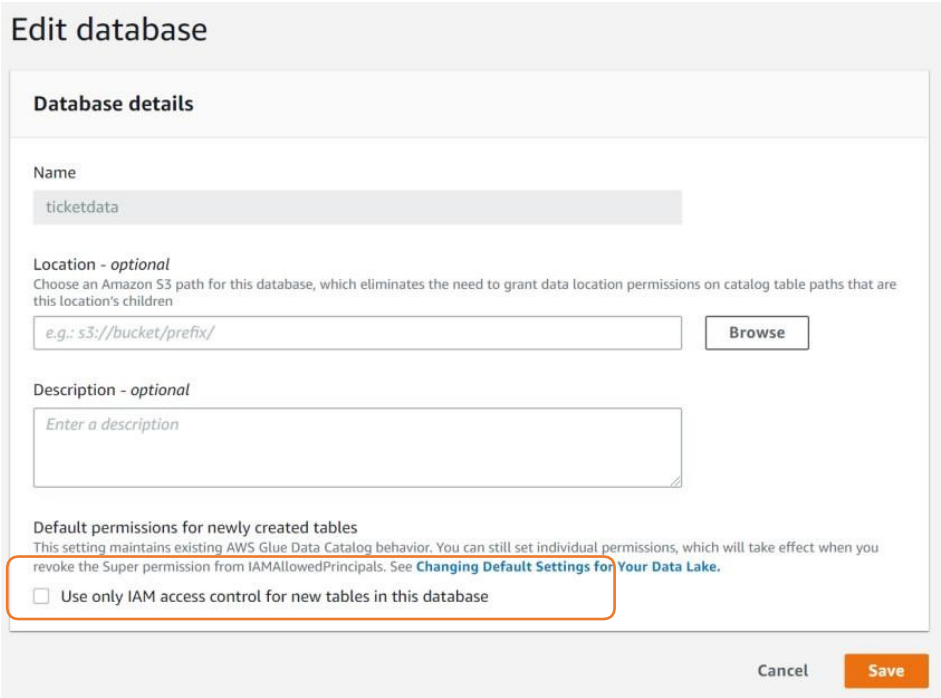
4. Under "**IAM Users and Roles**", select two roles: the Lake Formation role that was precreated: **<random>-LakeFormationWorkflowRole-<random>** and **TeamRole**. Grant **super** permissions for **Database permissions and Grantable permissions.**



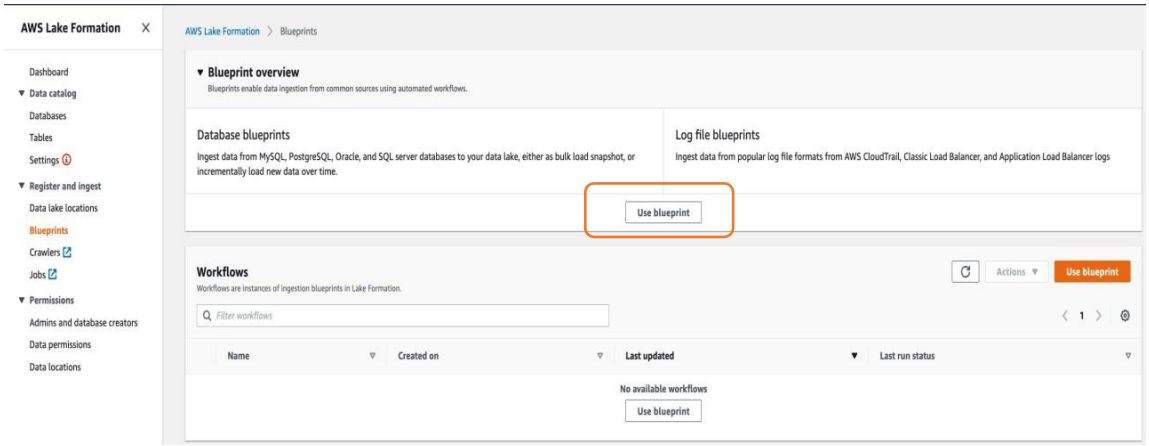
5. Select **Actions->Edit** on the **ticketdata** database



6. Clear the checkbox **Use only IAM access control** and click **Save**. Changing the default security setting so that access to Data Catalog resources (databases and tables) is managed by Lake Formation permissions.



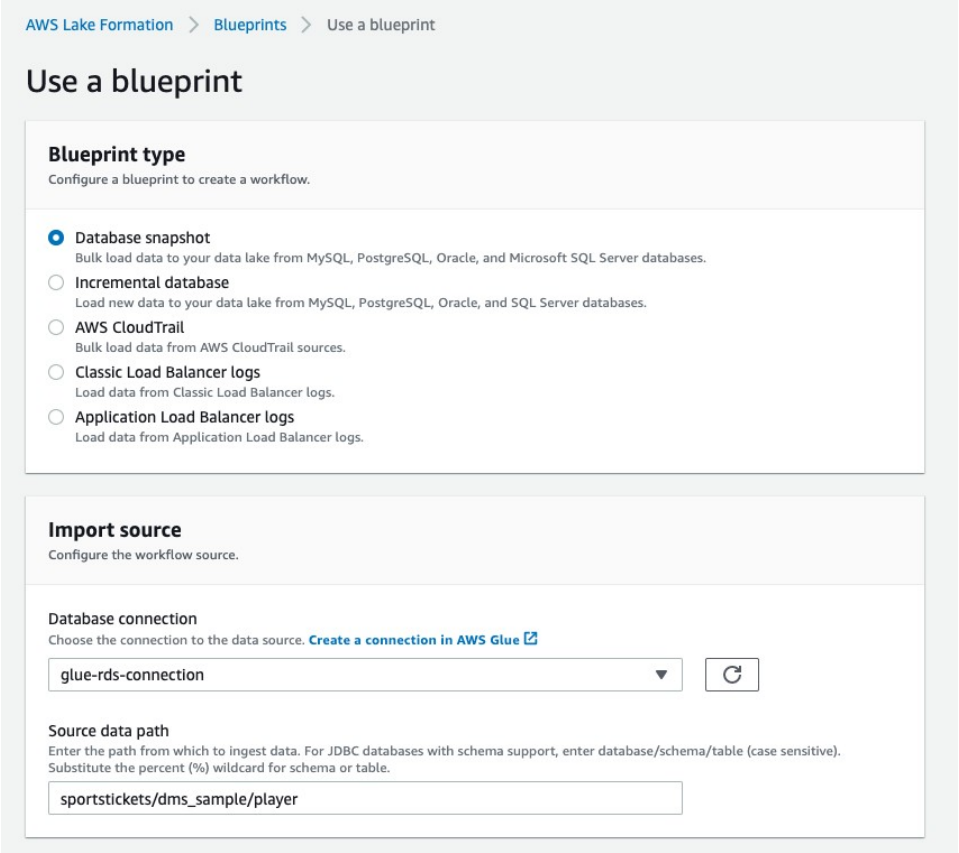
7. On the left pane navigate to **Blueprints** and click **Use blueprints**.



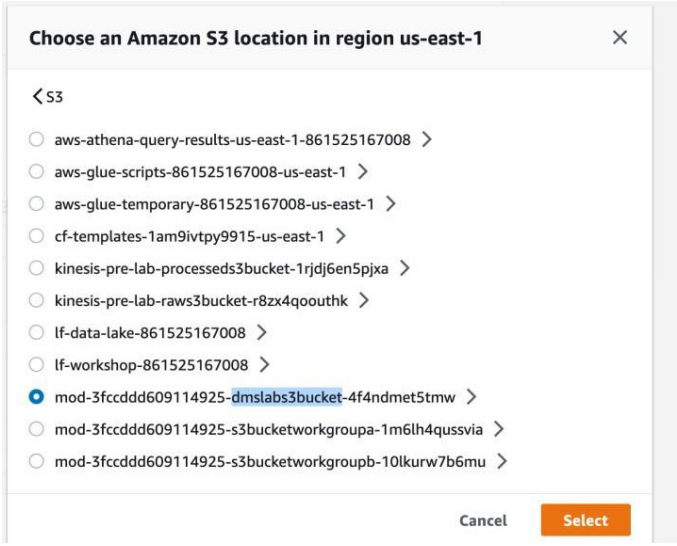
- For **Blueprint Type**, select **Database snapshot**
- Under **Import Source**

    a For **Database Connection** choose **glue-rds-connection**

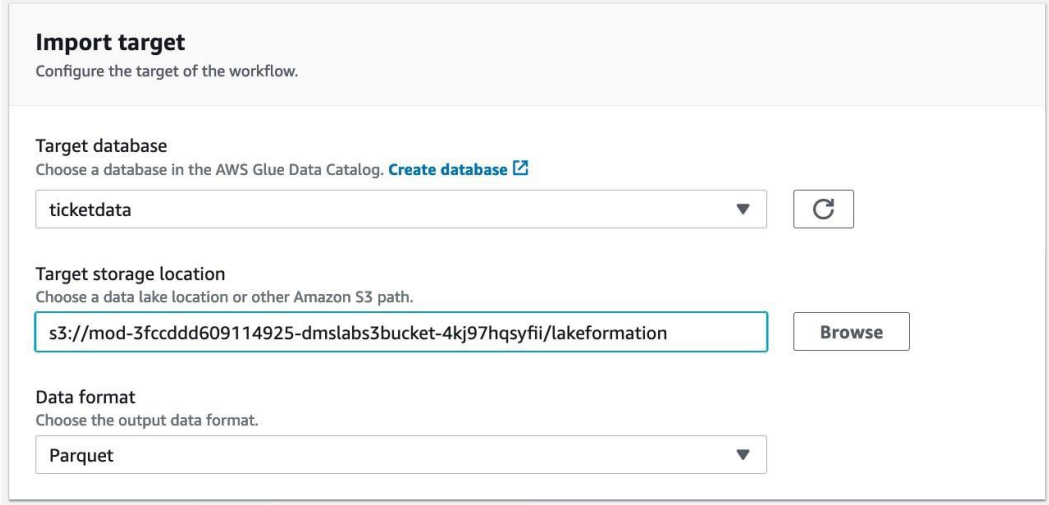
    b For **Source Data Path** enter **sportstickets/dms_sample/player**



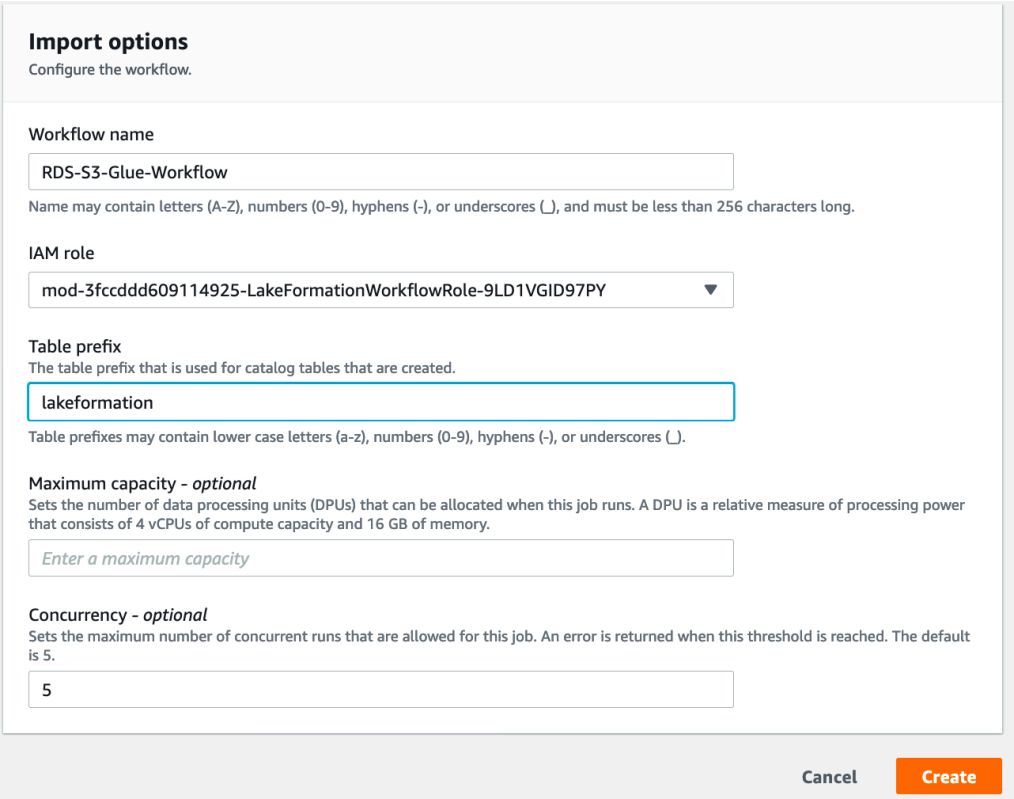
- Under **Import Target**
  - **i.** For **Target Database**, choose **ticketdata**
  - ii. For **Target storage location** browse and select the **xxx-dmslabS3bucket-xxx** created in the previous lab.



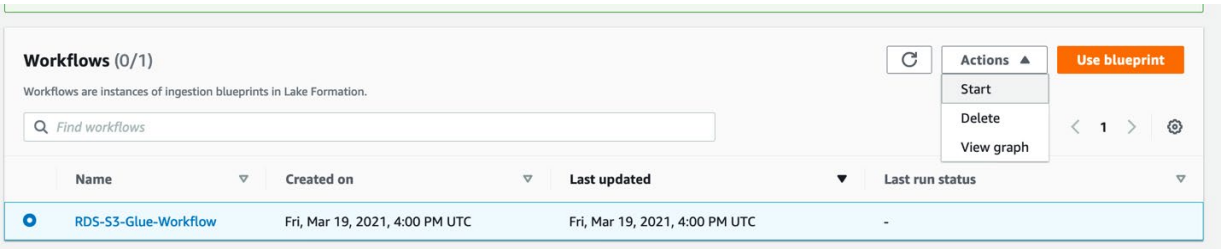
  - iii. Add **/lakeformation** at the end of the bucket url path, e.g. s3://xxx-dmslabs3bucket-xxx**/lakeformation**
  - iv. For **Data Format** choose **Parquet**



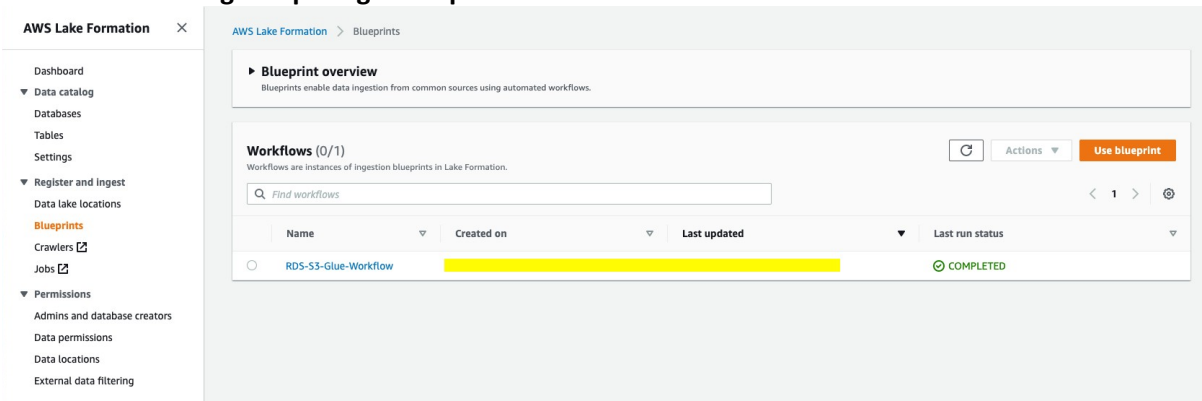
- For **Import Frequency**, Select **Run On Demand**
- For **Import Options**:
  - **i** Give a Workflow Name **RDS-S3-Glue-Workflow**
  - **ii** For the **IAM role** contains **LakeFormationWorkflowRole**
  - **iii** For **Table prefix** type **lakeformation**



8. Leave other options as default, click **Create**, and wait for the console to report that the workflow was successfully created.
9. Once the blueprint gets created, select it and click **Action -> Start.** There may be a delay of 5-10 seconds for the blueprint showing up. You may have to **hit refresh** button.



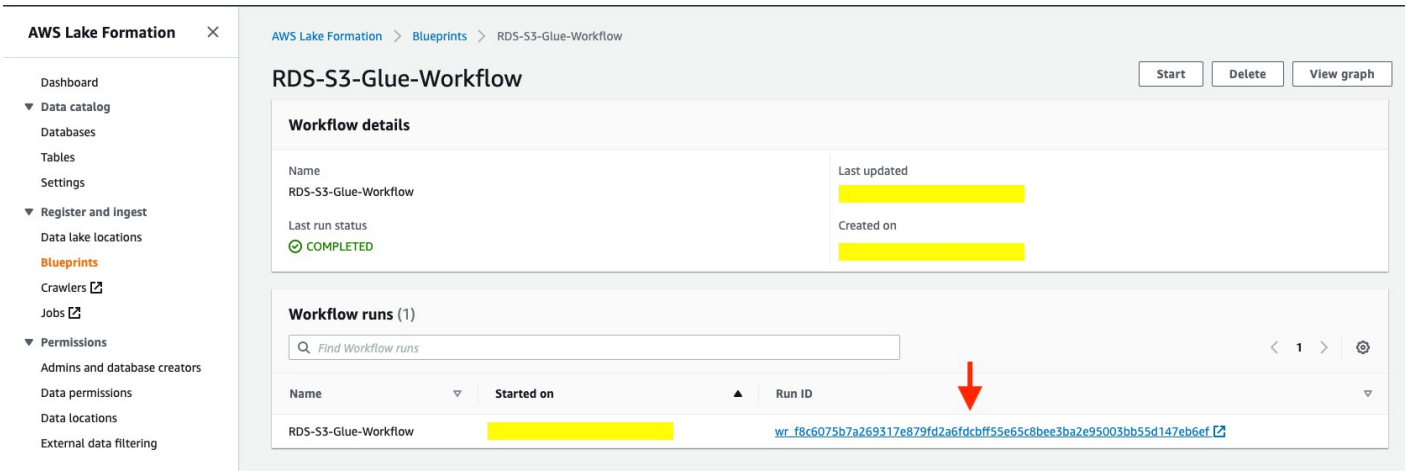
10. Once the workflow starts executing, you will see the status changes from **running -> discovering ->importing ->Completed**



Explore the Underlying Components of a Blueprint

The Lake Formation blueprint creates a Glue Workflow under the hood which contains Glue ETL jobs – both python shell and pyspark, Glue crawlers and triggers. It will take somewhere between 20-30 mins to finish its first execution. In the meantime, let us drill down to see what it creates for us;

1. On the **Lake Formation console**, in the navigation pane, choose **Blueprints**
2. In the **Workflow section**, click on the **Workflow name**. This will direct you to the Workflow run page. Click on the **Run Id.**

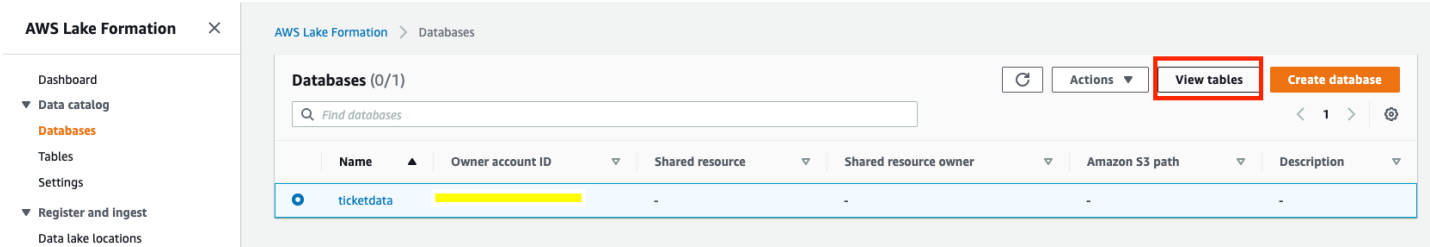


3. Here you can see the graphical representation of the Glue workflow built by Lake Formation blueprint. Highlighting and clicking on individual components will display the details of those components (name, description, job run id, start time, execution time)
4. To understand what all Glue Jobs got created as a part of this workflow, in the navigation pane, click on **Jobs**.

5. Every job comes with history, details, script and metrics tab. Review each of these tabs for any of the python shell or pyspark jobs.

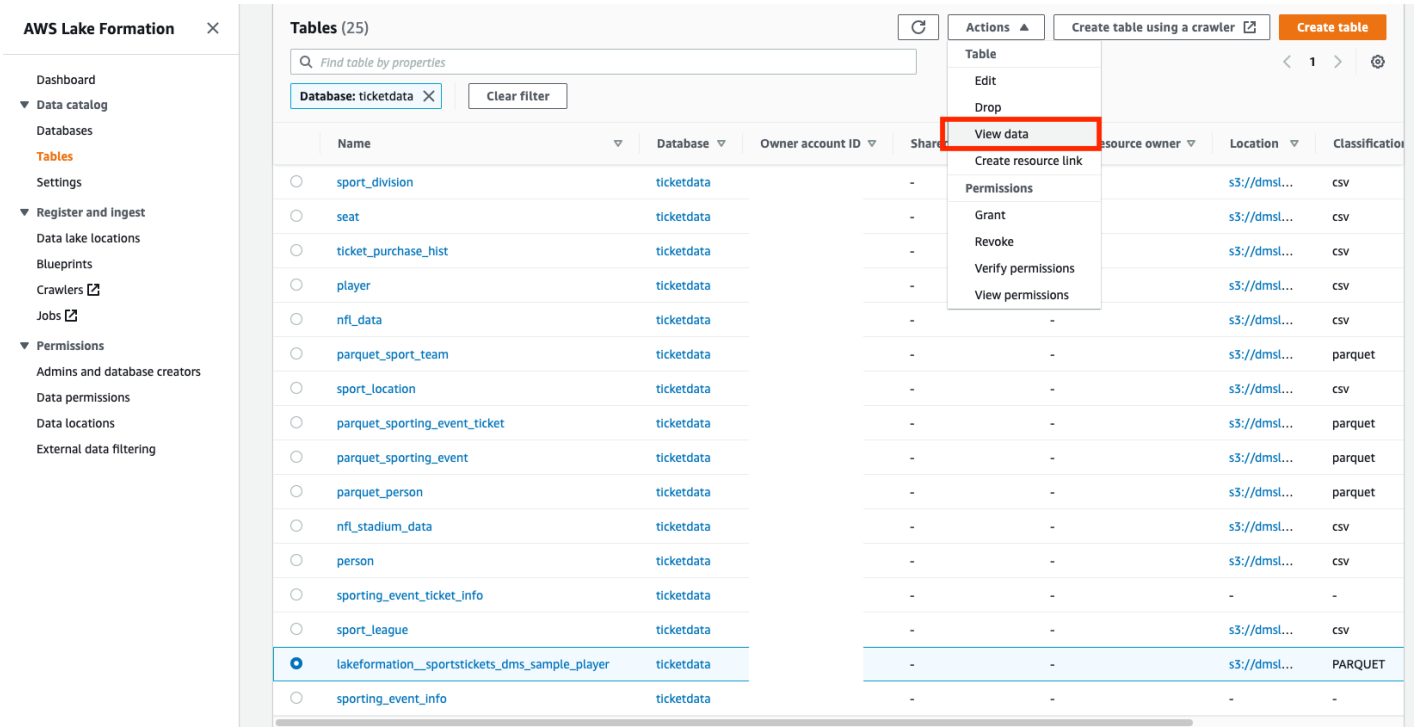
Explore workflow results in Athena

1. Navigate to the **Lake Formation** Console:
2. Navigate to **Databases** on the left panel and select **ticketdata**
3. Click on **View tables**



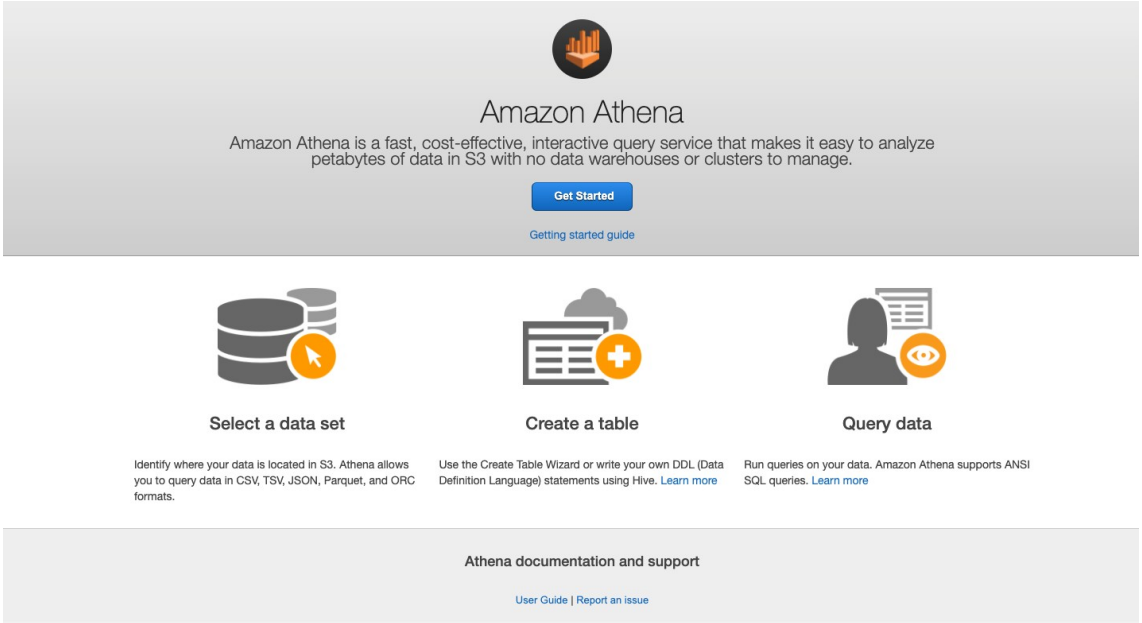
4. Select table **lakeformation_sportstickets_dms_sample_player**. As per our configuration above, Lake Formation tables were prefixed with **lakeformation_**

5. And Click **Action** -> **View Data**

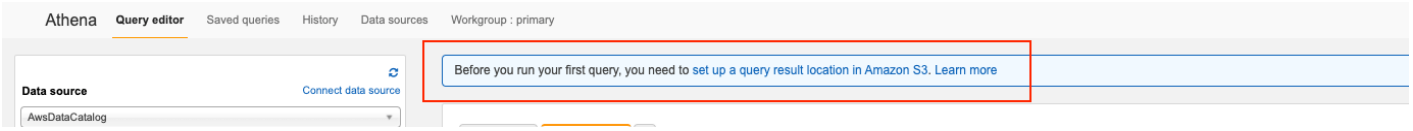


This will now take you to **Athena** console.

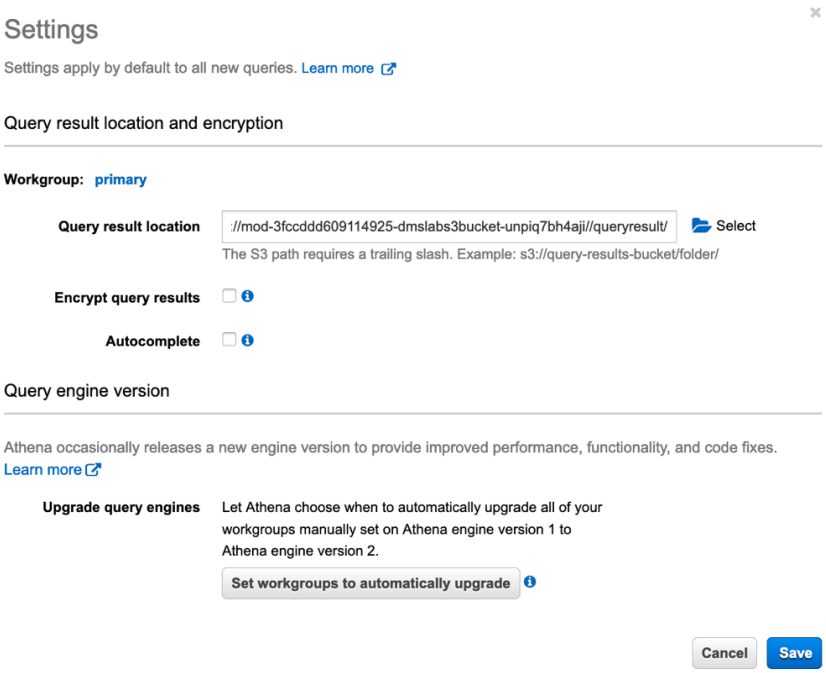
If you see a "Get Started" page, it's because it's the first time we're using Athena in this AWS Account. To proceed, click **Get Started**



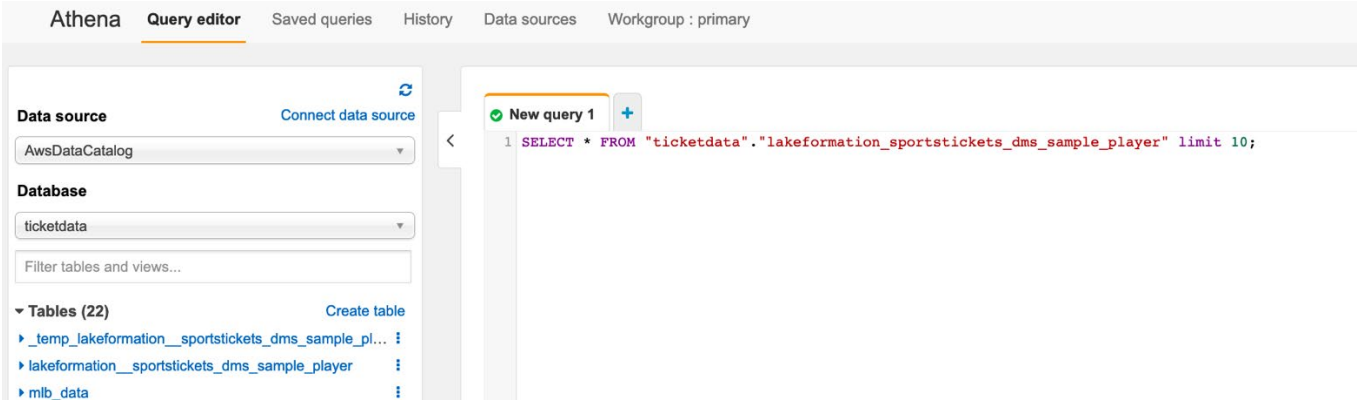
Then click **set up a query result location in Amazon S3** at the top



In the pop-up window in the **Query result location** field, enter your s3 bucket location followed by /, so that it looks like **s3:///xxx-dmslabs3bucket-xxx/queryresult/** and click **Save**



On Athena Console, you can run some queries using query editor:



To select some rows from the table, try running:

```
SELECT * FROM "ticketdata"."lakeformation_sportstickets_dms_sample_player" limit 10;
```

To get a row count, run:

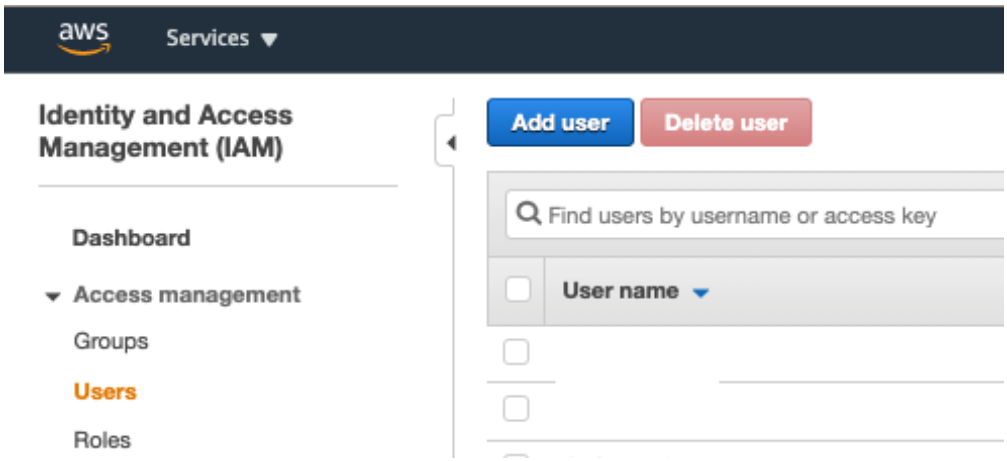
```
SELECT count(*) as recordcount FROM "ticketdata"."lakeformation_sportstickets_dms_sample_player" limit 10;
```

Congratulations!!! You have completed lake formation lab. To explore more fine grain data lake security feature, continue to next section.

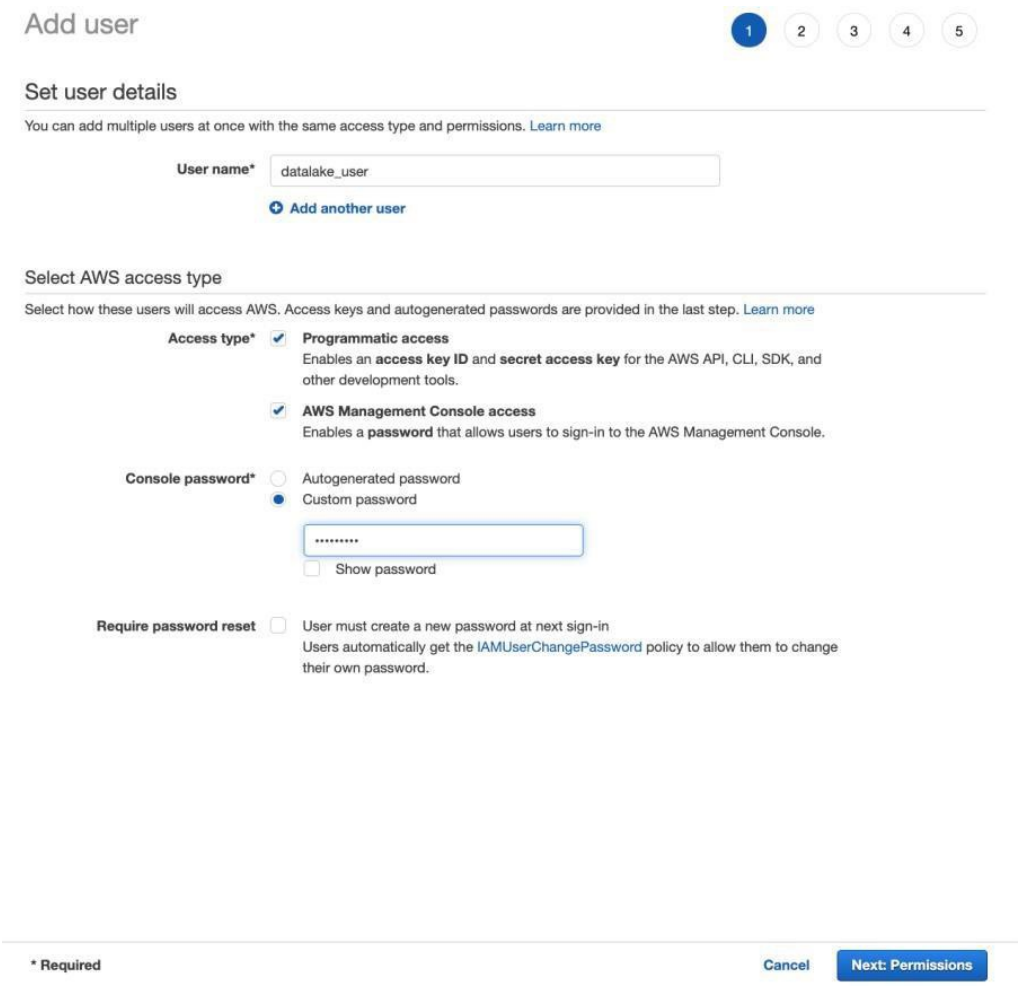
[Optional] Grant fine grain access controls to Data Lake user

Before we start the querying the data, let us create an IAM User **datalake_user** and grant column level access on the table created by the Lake formation workflow above, to **datalake_user.**

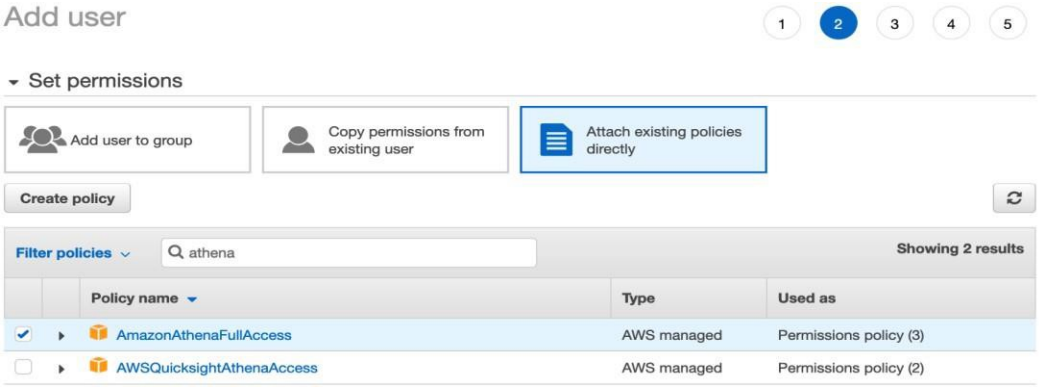
1. Navigate to **IAM Console** and click on **Add User**.



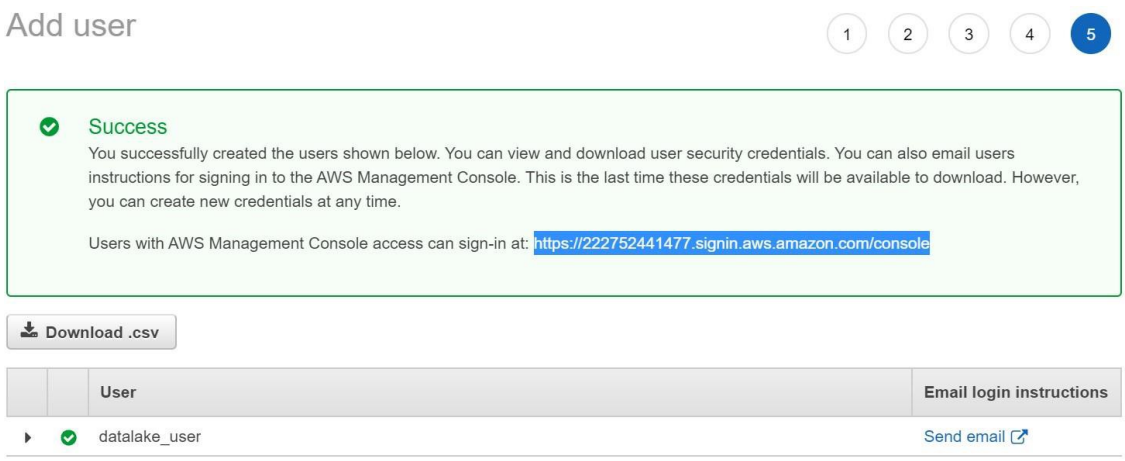
2. Create a user named **datalake_user** and give it a password: **Master123!**



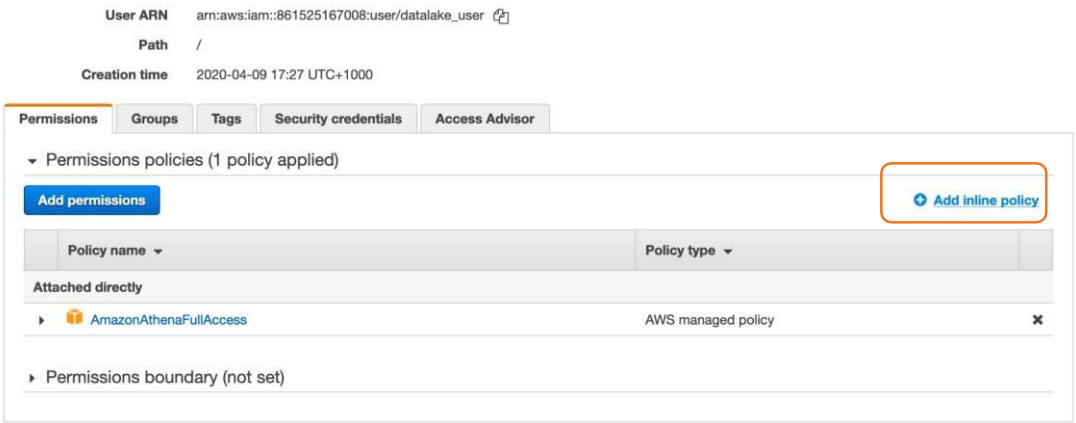
3. Next click on **Permissions**
4. Choose **Attach existing policies directly** and search for **AthenaFullAccess**



5. Keep navigating to the next steps until reached the end. Review the details and click on "**Create User**".
6. On the final screen, write down the sign-in link and hit **Close**



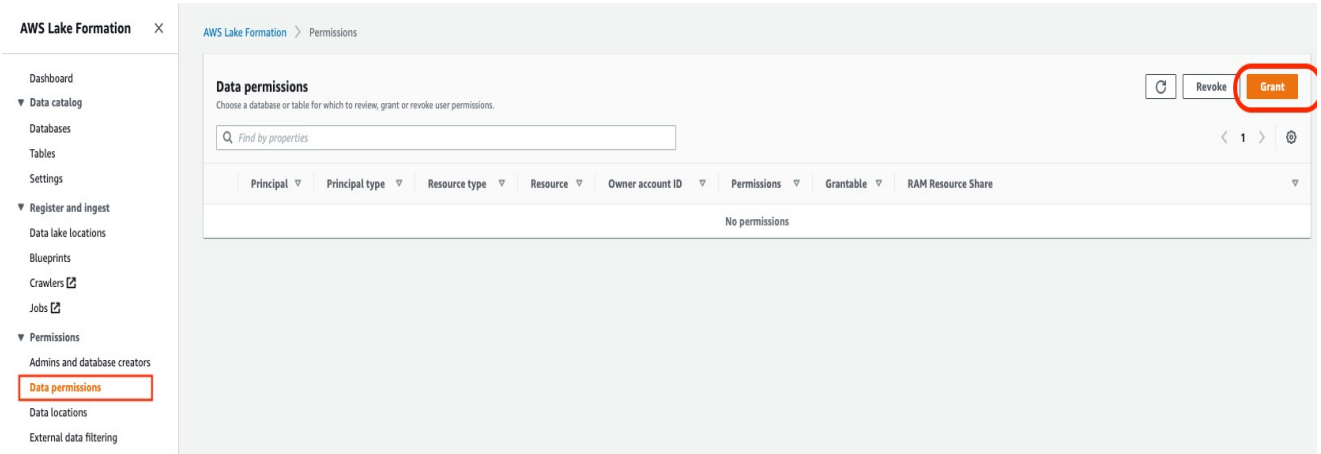
7. Click on the **datalake_user** user, and **add inline policy** and switch to the **JSON** tab



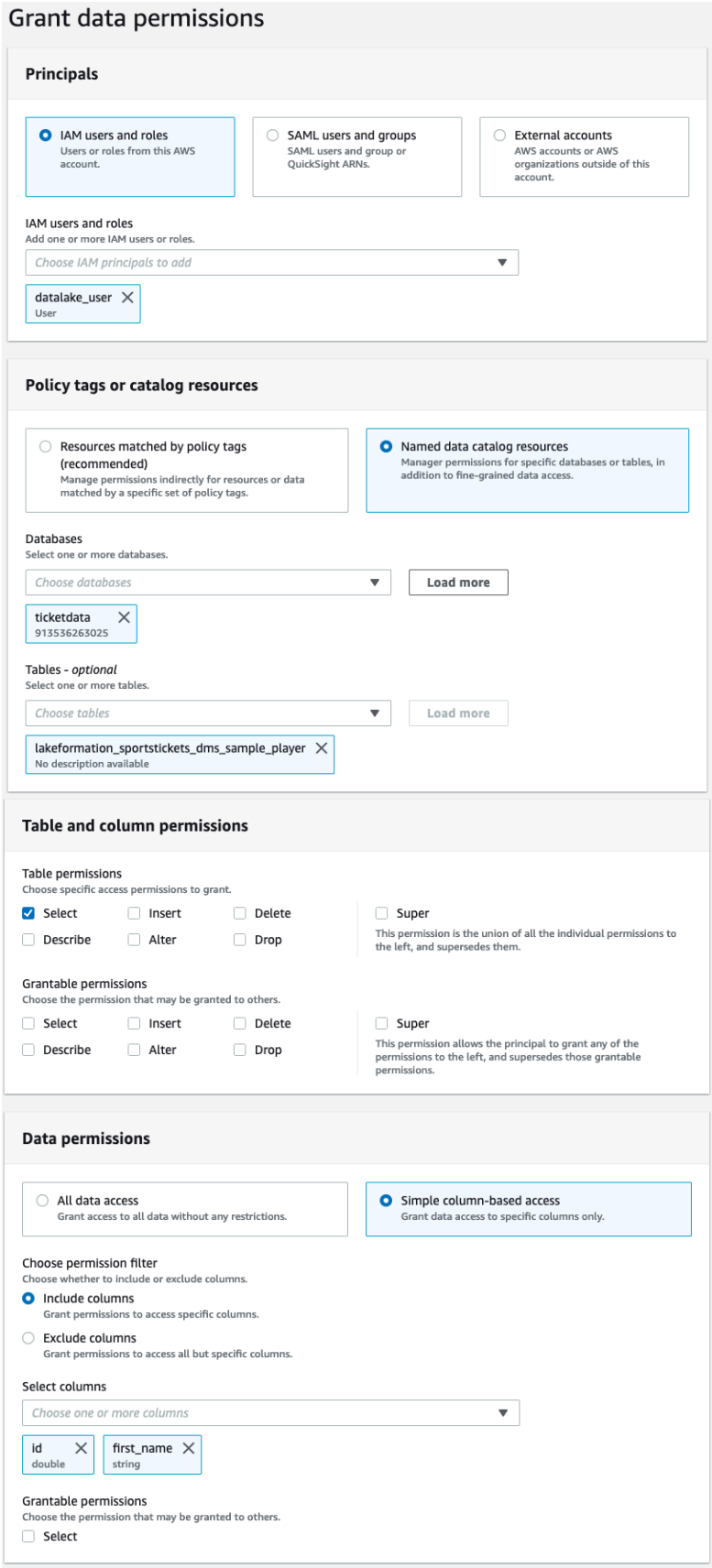
Use the following json snippet replacing <your_dmslabs3bucket_unique_name> with the name of your dmslabs3bucket, e.g. **mod-08b80667356c4f8a- dmslabs3bucketnh54wqg771lk**

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:Put*",
        "s3:Get*",
        "s3:List*"
      ],
      "Resource": [
        "arn:aws:s3:::<your_dmslabs3bucket_unique_name>/*"
      ]
    }
  ]
}
```

8. Give a name **athena_access** to the policy, then **Create Policy**. IAM user with required policies have been created.
9. Next, Navigate to **Lake Formation console**, under **Permissions** choose **Data permissions**.
10. Choose **Grant**, and in the **Grant permissions** dialog box, do the following:



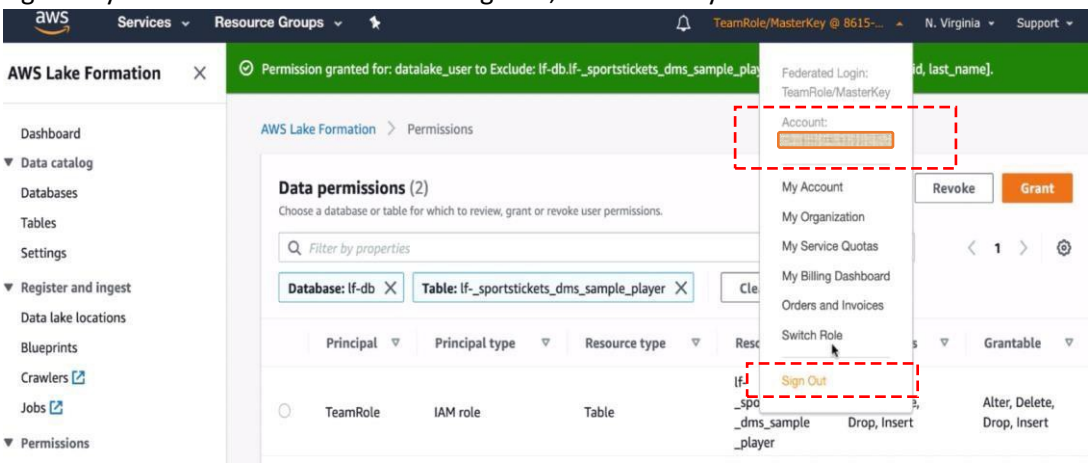
11. Once the **Grant permissions** window opens up:
    a. For **IAM user and roles**, choose **datalake_user.**
    b. Under **Policy tags or catalog resources**, choose **Named data catalog resources**
    c. For **Database**, choose **ticketdata**
    d. The **Table** list populates.
    e. For **Table**, choose **lakeformation_sportstickets_dms_sample_player**.
    f. For Columns, select **Include Columns** and choose **id**, **first_name**
    g. For **Table permissions**, choose **Select**.
    h. Under Data Permissions, choose **Simple column-based access** and select columns **id** and **first_name** to be included.
    i. Choose **Grant**


Grant data permissions
Principals
IAM users and roles
Users or roles from this AWS account.
SAML users and groups
SAML users and group or QuickSight ARNs.
External accounts
AWS accounts or AWS organizations outside of this account.
IAM users and roles
Add one or more IAM users or roles.
Choose IAM principals to add
datalake_user
User
Policy tags or catalog resources
Resources matched by policy tags (recommended)
Manage permissions indirectly for resources or data matched by a specific set of policy tags.
Named data catalog resources
Manager permissions for specific databases or tables, in addition to fine-grained data access.
Databases
Select one or more databases.
Choose databases
Load more
ticketdata
913536263025
Tables - optional
Select one or more tables.
Choose tables
Load more
lakeformation_sportstickets_dms_sample_player
No description available
Table and column permissions
Table permissions
Choose specific access permissions to grant.
Select
Insert
Delete
Describe
Alter
Drop
Super
This permission is the union of all the individual permissions to the left, and supersedes them.
Grantable permissions
Choose the permission that may be granted to others.
Select
Insert
Delete
Describe
Alter
Drop
Super
This permission allows the principal to grant any of the permissions to the left, and supersedes those grantable permissions.
Data permissions
All data access
Grant access to all data without any restrictions.
Simple column-based access
Grant data access to specific columns only.
Choose permission filter
Choose whether to include or exclude columns.
Include columns
Grant permissions to access specific columns.
Exclude columns
Grant permissions to access all but specific columns.
Select columns
Choose one or more columns
id
double
first_name
string
Grantable permissions
Choose the permission that may be granted to others.
Select

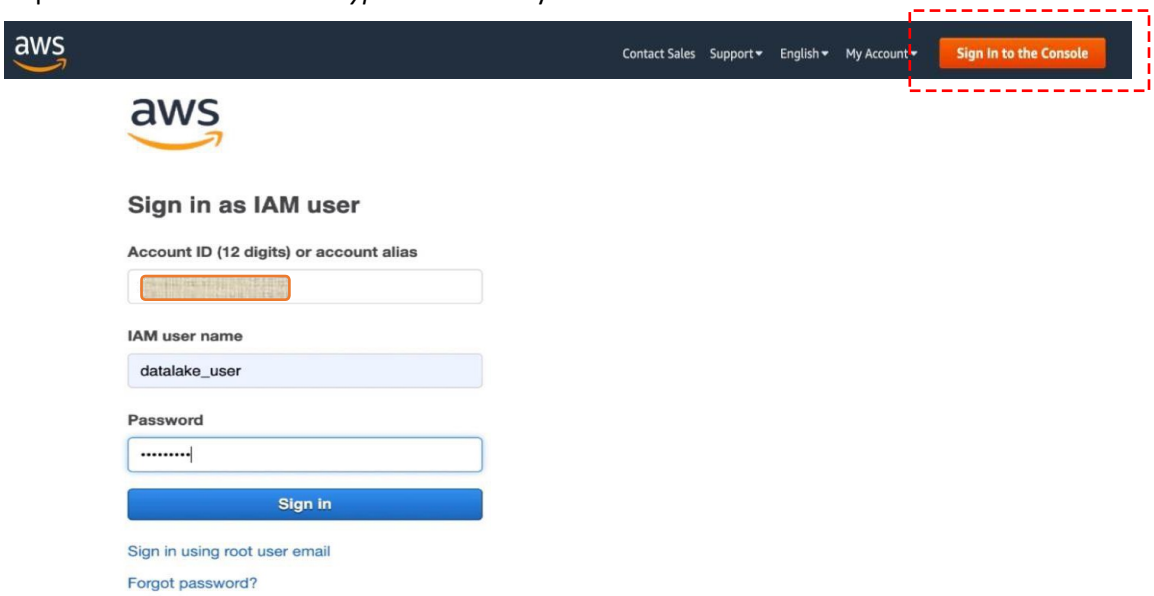
[Optional] Verify data permissions using Athena

Using Athena, let us now explore the data set as the **datalake_user**.

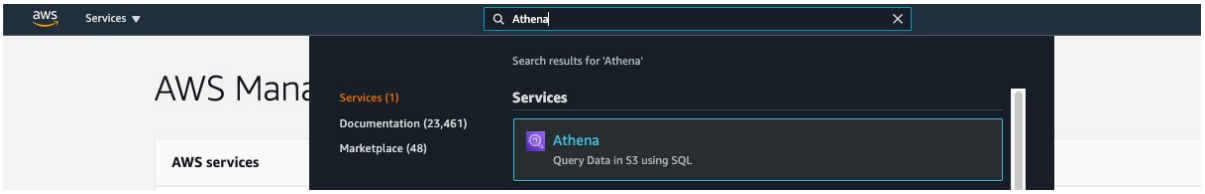
Sign out your AWS Account. Before doing that, write down your **Account ID**.



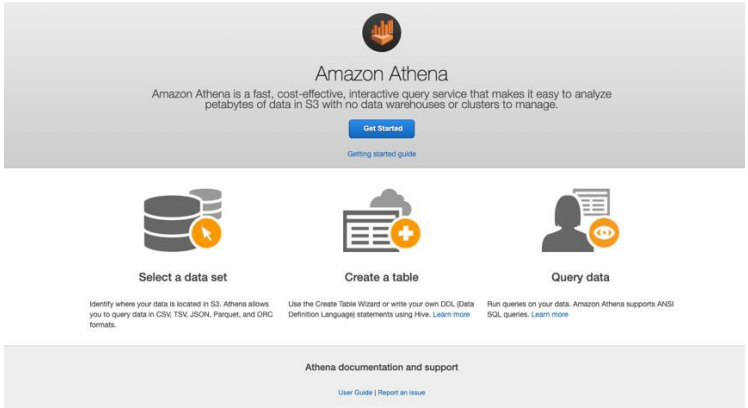
On the same web page, sign back in as the IAM user **datalake_user**, using **Master123!** as password. Note: remove *hyphens* '-' from your Account ID



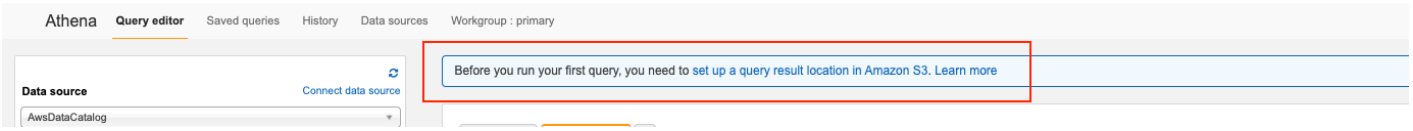
1. Make sure to change the region to **the appropriate AWS region**
2. Navigate to **Athena console**



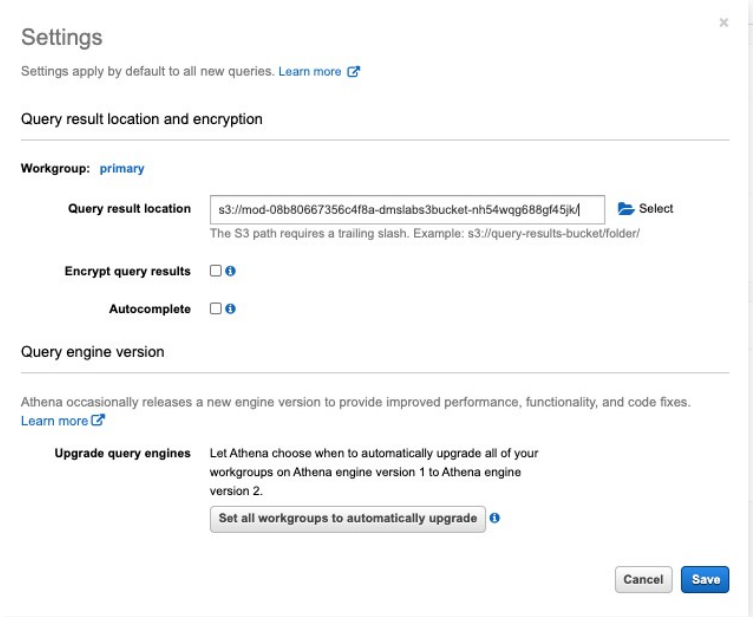
3. If you see a "Get Started" page, if it's the first time to use Athena in this AWS Account. To proceed, click **Get Started**



Then click **set up a query result location in Amazon S3** at the top.



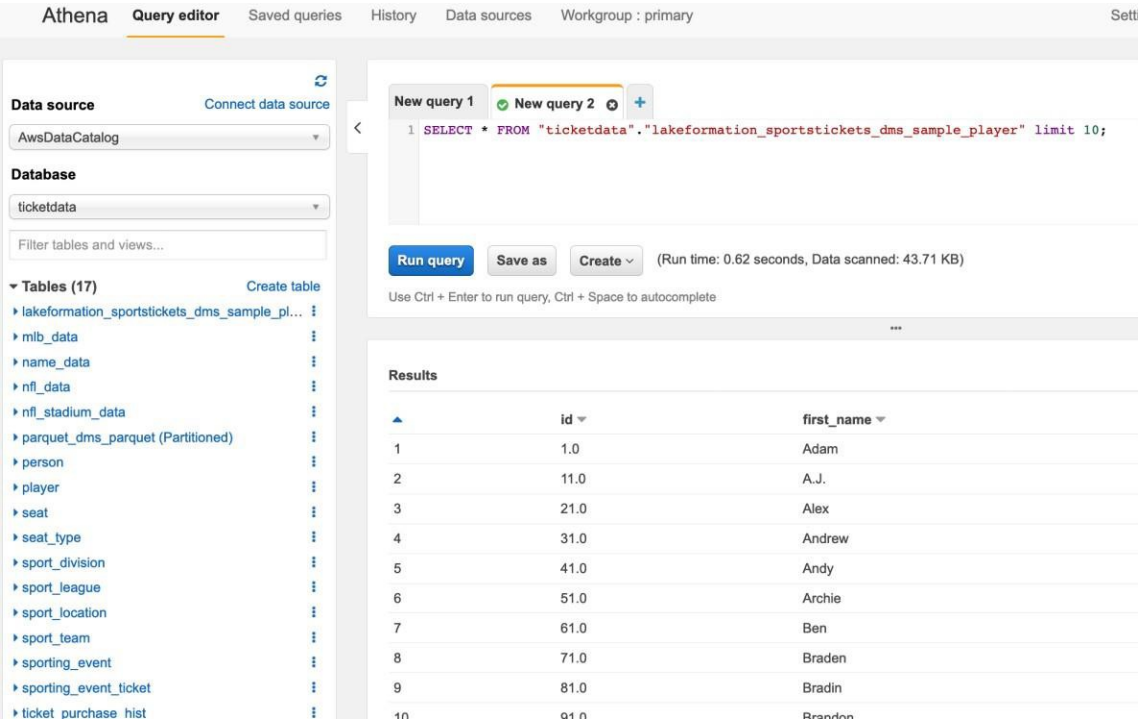
In the pop-up window, enter your s3 bucket location followed by '/' in the **Query result location** box. It looks like **s3://xxx-dmslabs3bucket-xxx/** and click **Save**



4. Next, ensure database **ticketdata** is selected.
5. Now run a select query:

```
SELECT * FROM "ticketdata"."lakeformation_sportstickets_dms_sample_player" limit 10;
```

6. You will notice that the **datalake_user** can **only see** the columns **id, first_name** in the 'select *' query result. The datalake_user cannot see **last_name**, **sports_team_id**, **full_name** columns in the table**.**



This explains that using AWS Lake Formation, you can provide granular access at table and column level to IAM users.

Congratulations!! You have successfully completed this lab!