



Amazon Web Services

Data Engineering Immersion Day

Lab 4. AWS Lake Formation

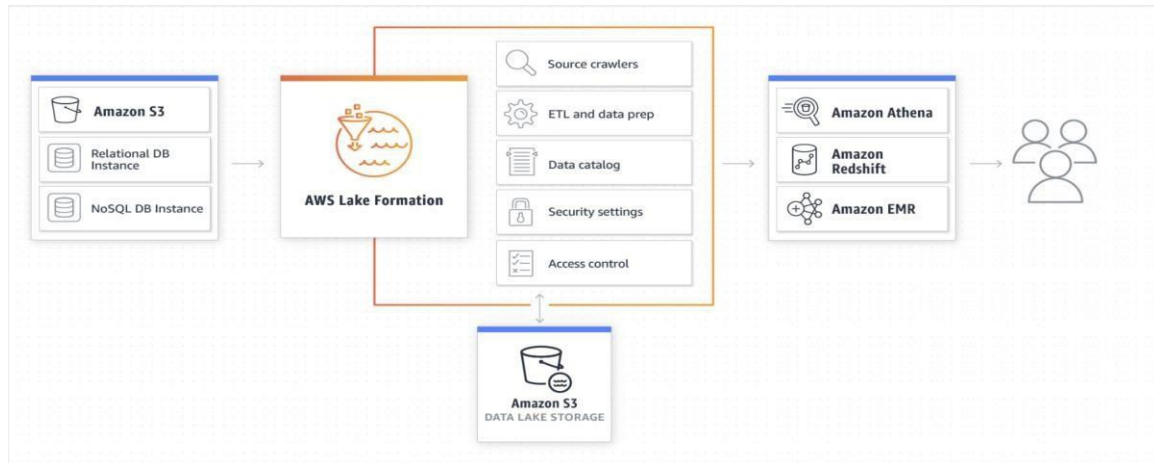
July 2021

Table of Contents

<i>Introduction.....</i>	<i>3</i>
<i>Prerequisites.....</i>	<i>3</i>
<i>Get Started Using the Lab Environment</i>	<i>3</i>
<i>Setup Network Configuration for AWS Glue (for your read)</i>	<i>5</i>
<i>Create an IAM role to use with Lake Formation (for your read)</i>	<i>6</i>
<i>Create Glue JDBC connection for RDS.....</i>	<i>6</i>
<i>Lake Formation – Add Administrator and start workflows using Blueprints.</i>	<i>8</i>
<i>Explore the Underlying Components of a Blueprint</i>	<i>14</i>
<i>Explore workflow results in Athena.....</i>	<i>15</i>
<i>[Optional] Grant fine grain access controls to Data Lake user</i>	<i>17</i>
<i>[Optional] Verify data permissions using Athena</i>	<i>22</i>

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:


Lab 4. AWS Lake Formation




Who are you?

Terms & Conditions:

1. By using the Event Engine for the relevant event, you agree to the Event Terms and Conditions and the AWS Acceptable Use Policy. You acknowledge and agree that are using an AWS-owned account that you can only access for the duration of the relevant event. If you find residual resources or materials in the AWS-owned account, you will make us aware and cease use of the account. AWS reserves the right to terminate the account and delete the contents at any time.
2. You will not: (a) process or run any operation on any data other than test data sets or lab-approved materials by AWS, and (b) copy, import, export or otherwise create derivate works of materials provided by AWS, including but not limited to, data sets.
3. AWS is under no obligation to enable the transmission of your materials through Event Engine and may, in its discretion, edit, block, refuse to post, or remove your materials at any time.
4. Your use of the Event Engine will comply with these terms and all applicable laws, and your access to Event Engine will immediately and automatically terminate if you do not comply with any of these terms or conditions.




This is the 12 digit hash that was given to you or your team.

✓ Accept Terms & Login

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:


 Modules


Environment Setup


[Readme](#)


Outputs:


S3 Bucket name
mod-3fccddd609114925-dmslabs3bucket-1ngcgzzcnd15u 


BusinessAnalystUser
mod-3fccddd609114925-BusinessAnalystUser-MB0XFZLQLOXX 

DMSLabRoleS3 ARN
arn:aws:iam::377243295828:role/mod-3fccddd609114925-DMSLabRoleS3-O2VT1RSN43SG 

Glue Lab Role
mod-3fccddd609114925-GlueLabRole-YLTJA13WW6WT 

S3BucketWorkgroupA
mod-3fccddd609114925-s3bucketworkgroupa-tbon3m1mkunh 

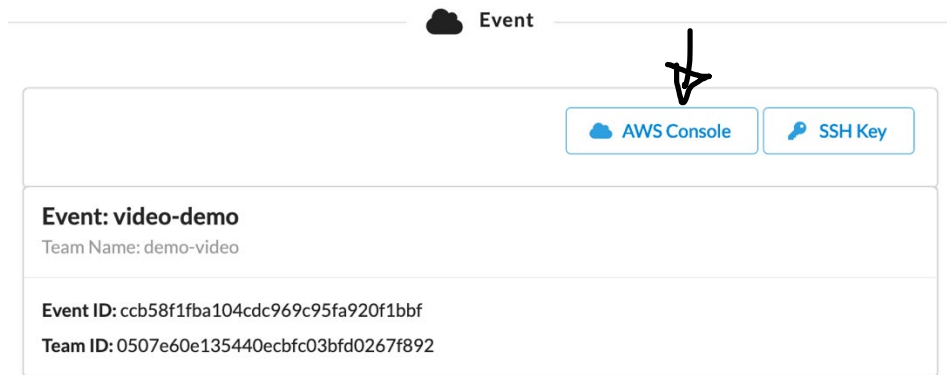
S3BucketWorkgroupB
mod-3fccddd609114925-s3bucketworkgroupb-18ygl8nfp8ead 

WorkgroupManagerUser
mod-3fccddd609114925-WorkgroupManagerUser-5IVE0UQNIBG4 

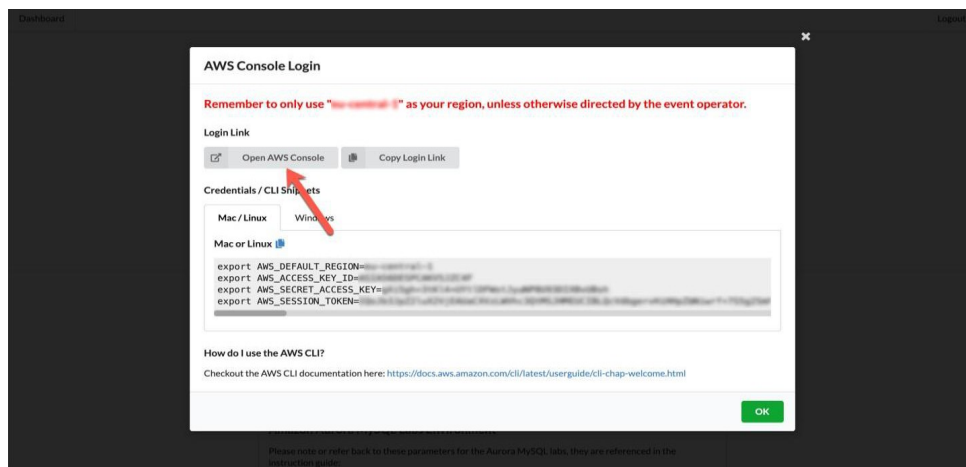
Lab 4. AWS Lake Formation

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

Team Dashboard



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 as 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,

Lab 4. AWS Lake Formation

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

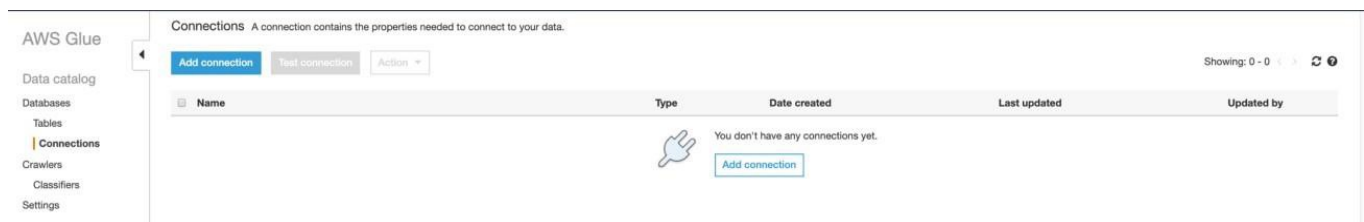
Create an IAM role to use with Lake Formation (for your read)

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**.

The screenshot shows the 'Add connection' form in the AWS Glue console. The form has a progress bar on the left with three steps: 'Connection properties' (selected), 'Connection access', and 'Review all steps'. The main content area is titled 'Set up your connection's properties.' and includes a link 'For more information, see Working with Connections.' The 'Connection name' field contains 'glue-rds-connection'. The 'Connection type' dropdown is set to 'JDBC'. There is an unchecked checkbox for 'Require SSL connection' with the text 'Fail if unable to connect over SSL' below it. The 'Description (optional)' text area contains 'Glue connection to RDS'. A 'Next' button is at the bottom right.

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.

Lab 4. AWS Lake Formation

- 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.

Set up access to your data store.

For more information, see [Working with Connections](#).

JDBC URL ⓘ

JDBC syntax for most database engines is jdbc:protocol://host:port/databaseName.

SQL Server syntax is jdbc:sqlserver://host:port;databaseName=db_name. Oracle syntax is jdbc:oracle:thin://@host:port/service_name. For more variations, see [Working with Connections](#).

Username

Password

VPC

Choose the VPC name that contains your data store.

Subnet

Choose the subnet within your VPC.

Security groups

Choose one or more security groups that allow access to the data store in your VPC. AWS Glue associates these security groups to the ENI attached to your subnet. To allow AWS Glue components to communicate and also prevent access from other networks, at least one chosen security group must specify a self-referencing inbound rule for all TCP ports.

<input type="checkbox"/> Group ID	<input type="text" value="Group name"/>
<input type="checkbox"/> sg-02f37b196bd136979	default
<input checked="" type="checkbox"/> sg-0ed70164f0c305708	updated-dmsstudent-sgdefault-OEYSKU2ZXUTR

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

AWS Glue

Data catalog

Databases

Tables

Connections

Connections A connection contains the properties needed to connect to your data.

Add connection

Test connection

Action ▾

<input type="checkbox"/> Name
<input checked="" type="checkbox"/> glue-rds-connection

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

Test connection

Test connection from your VPC and subnet to data stores and Amazon S3.

IAM role ⓘ

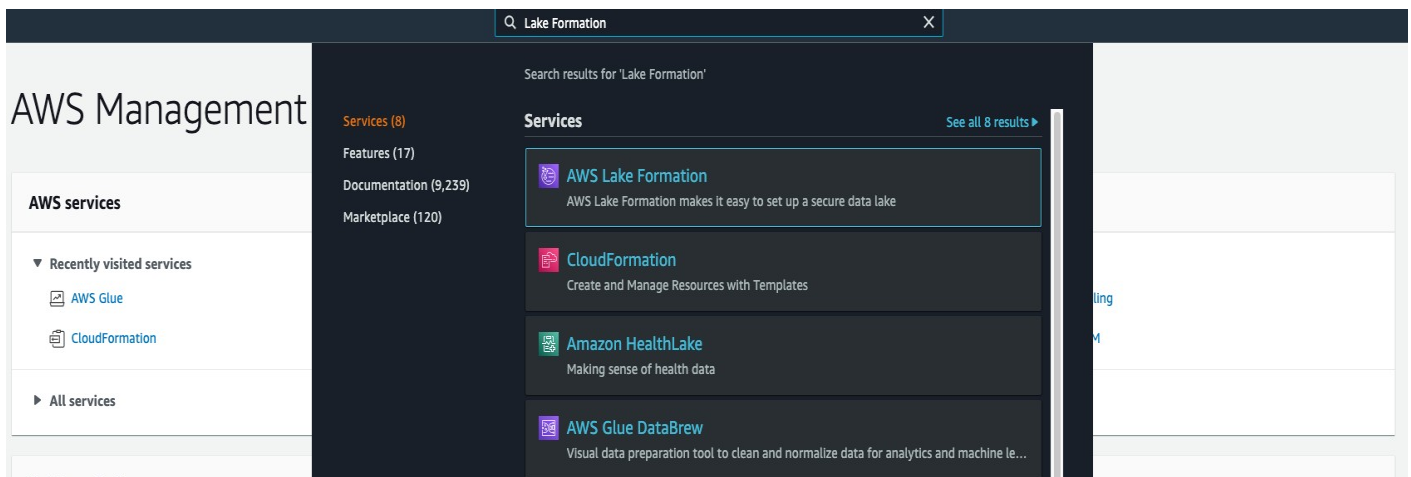
mod-b82e6b0b97d64dfd-LakeFormationWor... ↕

Ensure that this role has permission to access your data store.

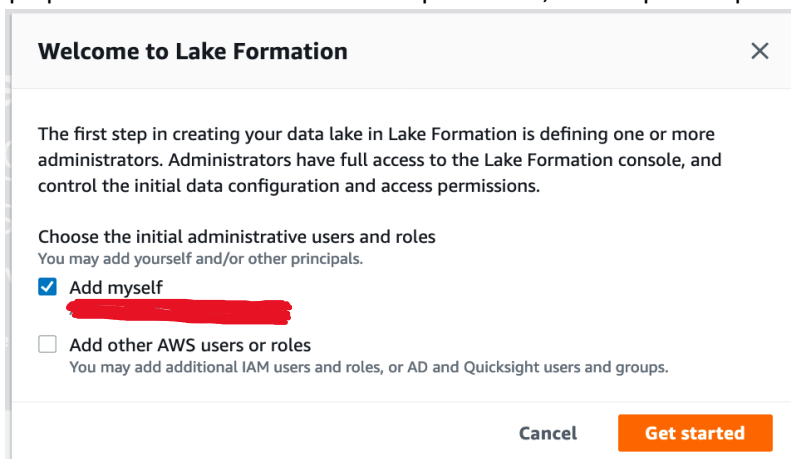
[Create IAM role.](#)

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**

Lab 4. AWS Lake Formation

Welcome to Lake Formation



The first step in creating your data lake in Lake Formation is defining one or more administrators. Administrators have full access to the Lake Formation console, and control the initial data configuration and access permissions.

Choose the initial administrative users and roles

You may add yourself and/or other principals.

☒ Add myself

AWS account: 913536263025

☐ Add other AWS users or roles

Select additional IAM users and roles to be data lake administrators.

Cancel

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

The screenshot shows the AWS Lake Formation console. On the left, the navigation pane has 'Databases' selected under 'Data catalog'. The main area is titled 'Databases (0/1)' and contains a table with the following columns: Name, Owner account ID, Shared resource, and Shared resource owner. A single row is visible with the name 'ticketdata'. An 'Actions' dropdown menu is open for the 'ticketdata' row, showing options: Database, Delete, Edit, Create resource link, Permissions, Grant, Revoke, Verify permissions, and View permissions. The 'Grant' option is highlighted. At the top right of the main area, there are buttons for 'Actions', 'View tables', and 'Create database'.

Part A of Lab 2. ETL with AWS Glue

Lab 4. AWS Lake Formation

- 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**.

Grant permissions: ticketdata ✕
Choose the access permissions to grant.

IAM users and roles
Add one or more IAM users or roles.

Choose IAM principals to add

mod-b82e6b0b97d64dfd-LakeFormationWorkflowRole-163KGGWZCGXIZ ✕
Role

TeamRole ✕
Role

Active Directory users and groups (EMR beta only)
Enter one or more Active Directory users or groups.

Ex: arn:aws:iam::<AccountId>:saml-provider/<SamlProviderName>:user/<UserName>

Database permissions
Choose the specific access permissions to grant.

☐ Create table ☐ Alter ☐ Drop

☒ Super
This permission is the union of the individual permissions above and supersedes them. [See here](#) 🔗

Grantable permissions
Choose the permissions that may be granted to others.

☐ Create table ☐ Alter ☐ Drop

☒ Super
This permission allows the principal to grant any of the above permissions and supersedes those grantable permissions.

Cancel

Grant

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

AWS Lake Formation ✕

Dashboard

▼ Data catalog

Databases

Tables

Settings

▼ Register and ingest

Data lake locations

Blueprints

Crawlers 🔗

Jobs 🔗

▼ Permissions

Admins and database creators

Data permissions

Data locations

External data filtering

AWS Lake Formation > Databases

Databases (0/1)

Find databases

Name	Owner account ID	Shared resource	Shared resource owner	Description
ticketdata		-	-	-

Actions

Database

Delete

Edit

Create resource link

Permissions

Grant

Revoke

Verify permissions

View permissions

View tables

Create database

< 1 >

Lab 4. AWS Lake Formation

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.

Edit database

Database details

Name
ticketdata

Location - *optional*
Choose an Amazon S3 path for this database, which eliminates the need to grant data location permissions on catalog table paths that are this location's children
e.g.: s3://bucket/prefix/ Browse

Description - *optional*
Enter a description

Default permissions for newly created tables
This setting maintains existing AWS Glue Data Catalog behavior. You can still set individual permissions, which will take effect when you revoke the Super permission from IAMAllowedPrincipals. See [Changing Default Settings for Your Data Lake](#).

☐ Use only IAM access control for new tables in this database

Cancel Save

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

AWS Lake Formation X

AWS Lake Formation > Blueprints

▼ Blueprint overview
Blueprints enable data ingestion from common sources using automated workflows.

Database blueprints
Ingest data from MySQL, PostgreSQL, Oracle, and SQL server databases to your data lake, either as bulk load snapshot, or incrementally load new data over time.

Log file blueprints
Ingest data from popular log file formats from AWS CloudTrail, Classic Load Balancer, and Application Load Balancer logs

Use blueprint

Workflows
Workflows are instances of ingestion blueprints in Lake Formation.

Name	Created on	Last updated	Last run status
No available workflows			

Use blueprint

- 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**

Lab 4. AWS Lake Formation

AWS Lake Formation > Blueprints > Use a blueprint

Use a blueprint

Blueprint type
Configure a blueprint to create a workflow.

☒ **Database snapshot**
Bulk load data to your data lake from MySQL, PostgreSQL, Oracle, and Microsoft SQL Server databases.

☐ **Incremental database**
Load new data to your data lake from MySQL, PostgreSQL, Oracle, and SQL Server databases.

☐ **AWS CloudTrail**
Bulk load data from AWS CloudTrail sources.

☐ **Classic Load Balancer logs**
Load data from Classic Load Balancer logs.

☐ **Application Load Balancer logs**
Load data from Application Load Balancer logs.

Import source
Configure the workflow source.

Database connection
Choose the connection to the data source. [Create a connection in AWS Glue](#)

glue-rds-connection

Source data path
Enter the path from which to ingest data. For JDBC databases with schema support, enter database/schema/table (case sensitive). Substitute the percent (%) wildcard for schema or table.

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.

Choose an Amazon S3 location in region us-east-1

< S3

- ☐ aws-athena-query-results-us-east-1-861525167008 >
- ☐ aws-glue-scripts-861525167008-us-east-1 >
- ☐ aws-glue-temporary-861525167008-us-east-1 >
- ☐ cf-templates-1am9ivtpy9915-us-east-1 >
- ☐ kinesis-pre-lab-processedS3bucket-1rjdj6en5pjxa >
- ☐ kinesis-pre-lab-raws3bucket-r8zx4qoouthk >
- ☐ lf-data-lake-861525167008 >
- ☐ lf-workshop-861525167008 >
- ☒ mod-3fccddd609114925-dmslabs3bucket-4f4ndmet5tmw >
- ☐ mod-3fccddd609114925-s3bucketworkgroupa-1m6lh4qussvia >
- ☐ mod-3fccddd609114925-s3bucketworkgroupb-10lkurw7b6mu >

Cancel Select

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

Lab 4. AWS Lake Formation

Import target

Configure the target of the workflow.

Target database
Choose a database in the AWS Glue Data Catalog. [Create database](#)

ticketdata

▼

↻

Target storage location
Choose a data lake location or other Amazon S3 path.

s3://mod-3fccddd609114925-dmslabs3bucket-4kj97hqsyfii/lakeformation

Browse

Data format
Choose the output data format.

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**

Import options

Configure the workflow.

Workflow name

RDS-S3-Glue-Workflow

Name may contain letters (A-Z), numbers (0-9), hyphens (-), or underscores (_), and must be less than 256 characters long.

IAM role

mod-3fccddd609114925-LakeFormationWorkflowRole-9LD1VGID97PY

▼

Table prefix
The table prefix that is used for catalog tables that are created.

lakeformation

Table prefixes may contain lower case letters (a-z), numbers (0-9), hyphens (-), or underscores (_).

Maximum capacity - optional
Sets the number of data processing units (DPUs) that can be allocated when this job runs. A DPU is a relative measure of processing power that consists of 4 vCPUs of compute capacity and 16 GB of memory.

Enter a maximum capacity

Concurrency - optional
Sets the maximum number of concurrent runs that are allowed for this job. An error is returned when this threshold is reached. The default is 5.

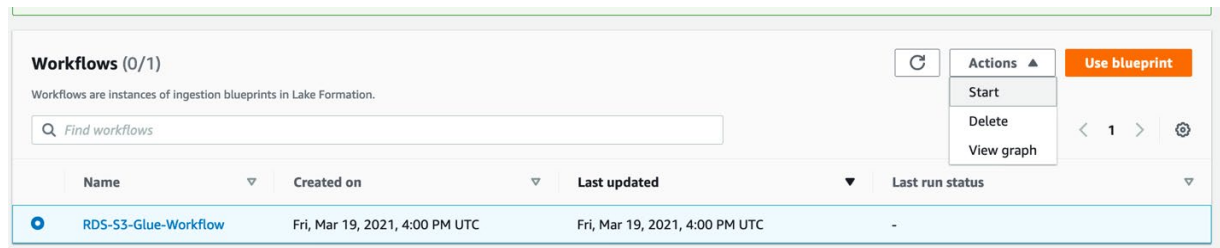
5

Cancel

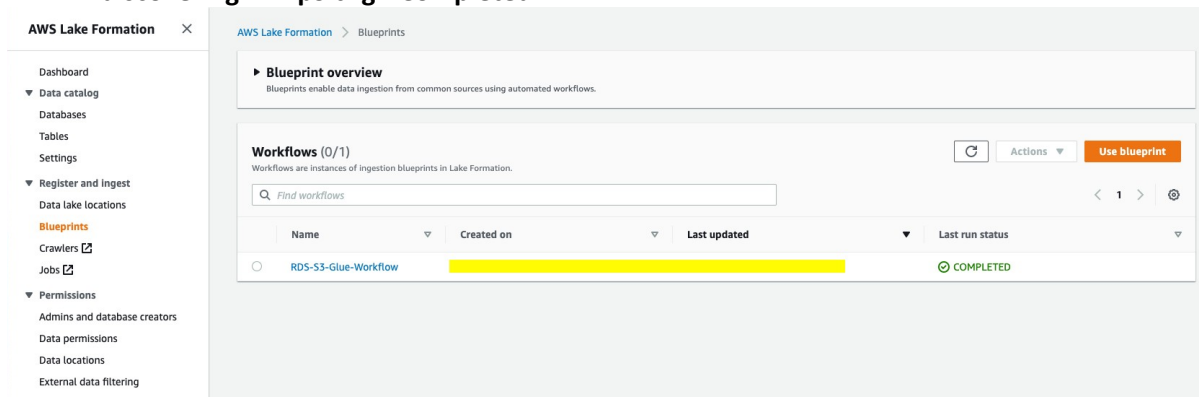
Create

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.

Lab 4. AWS Lake Formation



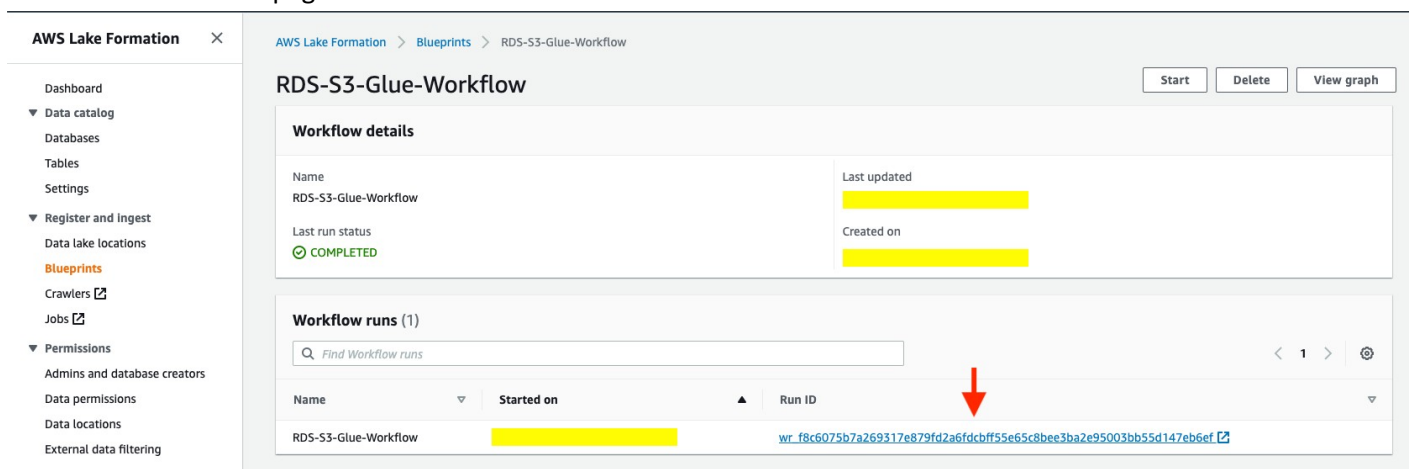
10. Once the workflow starts executing, you will see the status changes from **running** -> **discovering** -> **importing** -> **Completed**



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;

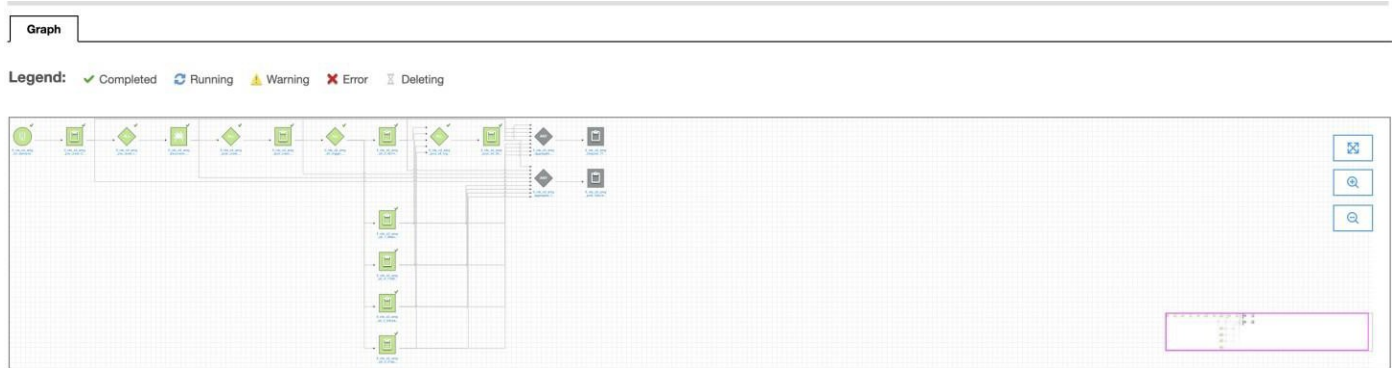
1. On the **Lake Formation console**, in the navigation pane, choose **Blueprints**
2. In the **Workflow section**, click on the **Workflow name**. This will direct you to the Workflow run page. Click on the **Run Id**.



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**.

Lab 4. AWS Lake Formation

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**

AWS Lake Formation > Databases

Databases (0/1)

Find databases

Actions View tables Create database

Name	Owner account ID	Shared resource	Shared resource owner	Amazon S3 path	Description
ticketdata		-	-	-	-

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**

AWS Lake Formation > Tables

Tables (25)

Find table by properties

Database: ticketdata Clear filter

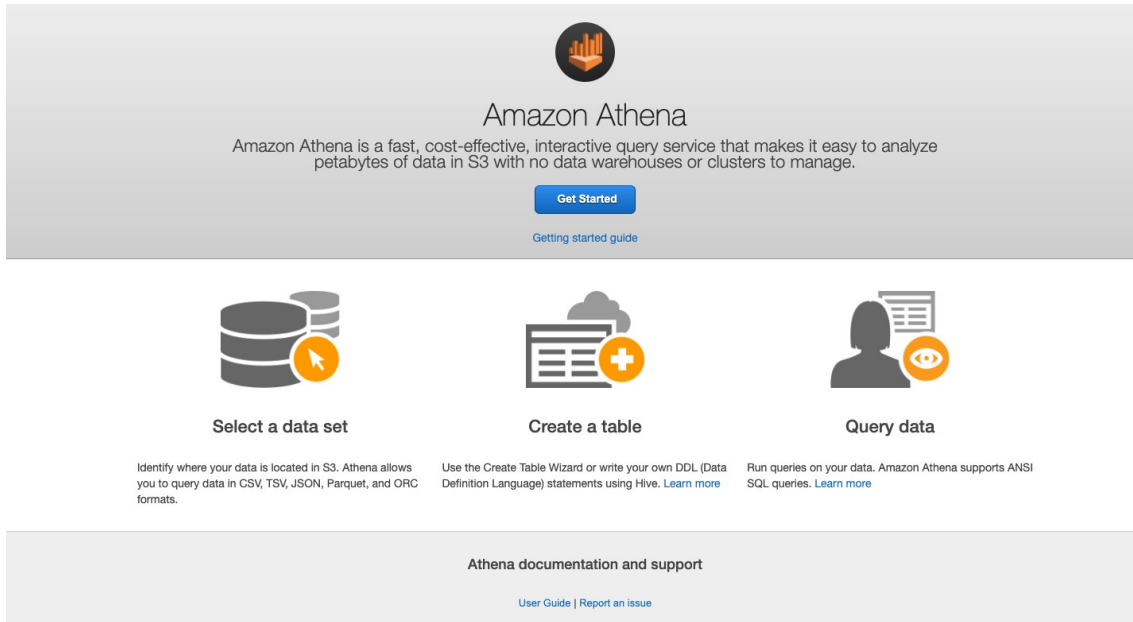
Actions View data Create table using a crawler Create table

Name	Database	Owner account ID	Shared resource	Shared resource owner	Location	Classification
sport_division	ticketdata	-	-	-	s3://dmsl...	csv
seat	ticketdata	-	-	-	s3://dmsl...	csv
ticket_purchase_hist	ticketdata	-	-	-	s3://dmsl...	csv
player	ticketdata	-	-	-	s3://dmsl...	csv
nfl_data	ticketdata	-	-	-	s3://dmsl...	csv
parquet_sport_team	ticketdata	-	-	-	s3://dmsl...	parquet
sport_location	ticketdata	-	-	-	s3://dmsl...	csv
parquet_sporting_event_ticket	ticketdata	-	-	-	s3://dmsl...	parquet
parquet_sporting_event	ticketdata	-	-	-	s3://dmsl...	parquet
parquet_person	ticketdata	-	-	-	s3://dmsl...	parquet
nfl_stadium_data	ticketdata	-	-	-	s3://dmsl...	csv
person	ticketdata	-	-	-	s3://dmsl...	csv
sporting_event_ticket_info	ticketdata	-	-	-	-	-
sport_league	ticketdata	-	-	-	s3://dmsl...	csv
lakeformation_sportstickets_dms_sample_player	ticketdata	-	-	-	s3://dmsl...	PARQUET
sporting_event_info	ticketdata	-	-	-	-	-

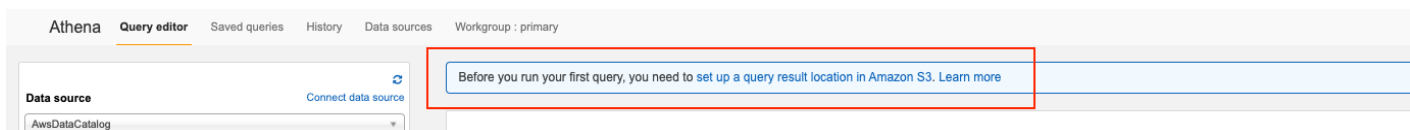
Lab 4. AWS Lake Formation

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**

Settings

Settings apply by default to all new queries. [Learn more](#)

Query result location and encryption

Workgroup: **primary**

Query result location [Select](#)
The S3 path requires a trailing slash. Example: s3://query-results-bucket/folder/

Encrypt query results ☐ [?](#)

Autocomplete ☐ [?](#)

Query engine version

Athena occasionally releases a new engine version to provide improved performance, functionality, and code fixes. [Learn more](#)

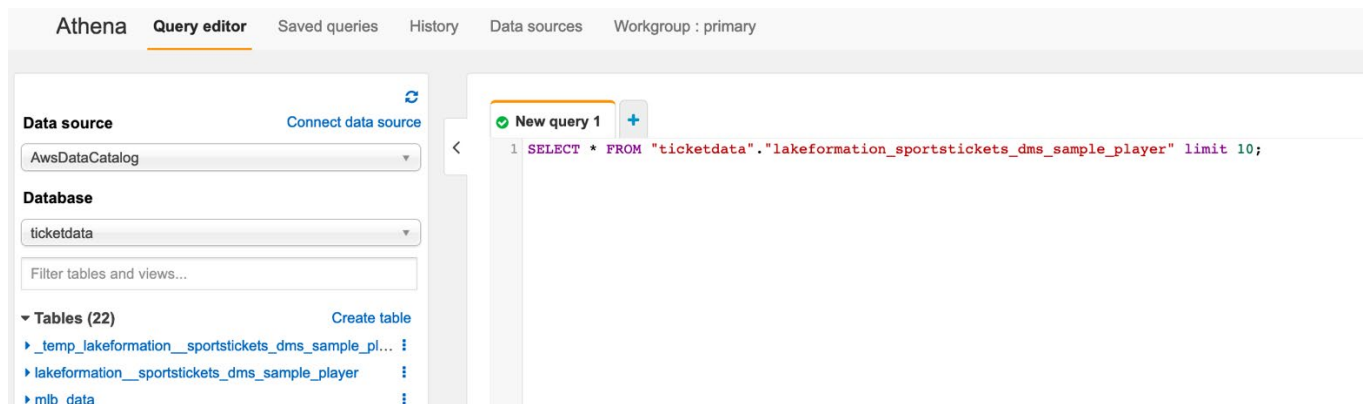
Upgrade query engines Let Athena choose when to automatically upgrade all of your workgroups manually set on Athena engine version 1 to Athena engine version 2.

[Set workgroups to automatically upgrade](#) [?](#)

[Cancel](#) [Save](#)

Lab 4. AWS Lake Formation

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:

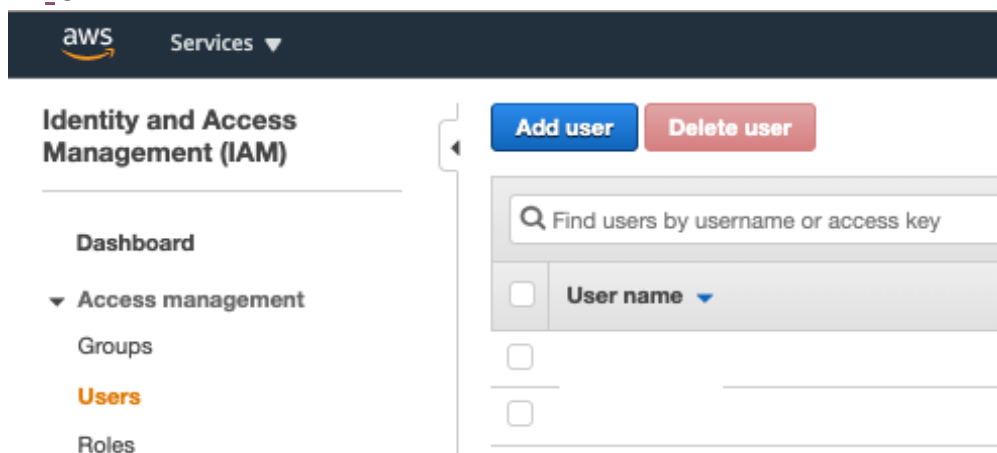
```
SELECT count(*) as recordcount FROM "ticketdata"."lakeformation_sportstickets_dms_sample_player" limit 10;
```

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!**

Lab 4. AWS Lake Formation

Add user

1 2 3 4 5

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name* datalake_user

[Add another user](#)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

- Access type* ☒ **Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.
- ☒ **AWS Management Console access**
Enables a **password** that allows users to sign-in to the AWS Management Console.

Console password* ☐ Autogenerated password
☒ Custom password

☐ Show password

Require password reset ☐ User must create a new password at next sign-in
Users automatically get the `IAMUserChangePassword` policy to allow them to change their own password.

* Required

[Cancel](#)

[Next: Permissions](#)

3. Next click on **Permissions**

4. Choose **Attach existing policies directly** and search for **AthenaFullAccess**

Add user

1 2 3 4 5

Set permissions

Add user to group

Copy permissions from existing user

Attach existing policies directly

[Create policy](#)

Filter policies		Showing 2 results	
Policy name		Type	Used as
<input checked="" type="checkbox"/>	AmazonAthenaFullAccess	AWS managed	Permissions policy (3)
<input type="checkbox"/>	AWSQuicksightAthenaAccess	AWS managed	Permissions policy (2)

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**

Lab 4. AWS Lake Formation

Add user

1 2 3 4 5

Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://222752441477.signin.aws.amazon.com/console>

[Download .csv](#)

User	Email login instructions
<input checked="" type="checkbox"/> datalake_user	Send email

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

[Add user](#) [Delete user](#)

Showing 4 results

<input type="checkbox"/>	User name	Groups	Access key age	Password age	Last activity	MFA
<input checked="" type="checkbox"/>	datalake_user	None	None	Today	None	Not enabled
<input type="checkbox"/>	EC2-user	None	None	None	None	Not enabled

User ARN `arn:aws:iam::861525167008:user:datalake_user`

Path `/`

Creation time `2020-04-09 17:27 UTC+1000`

Permissions **Groups** **Tags** **Security credentials** **Access Advisor**

Permissions policies (1 policy applied)

[Add permissions](#) [Add inline policy](#)

Policy name	Policy type
<input checked="" type="checkbox"/> AmazonAthenaFullAccess	AWS managed policy

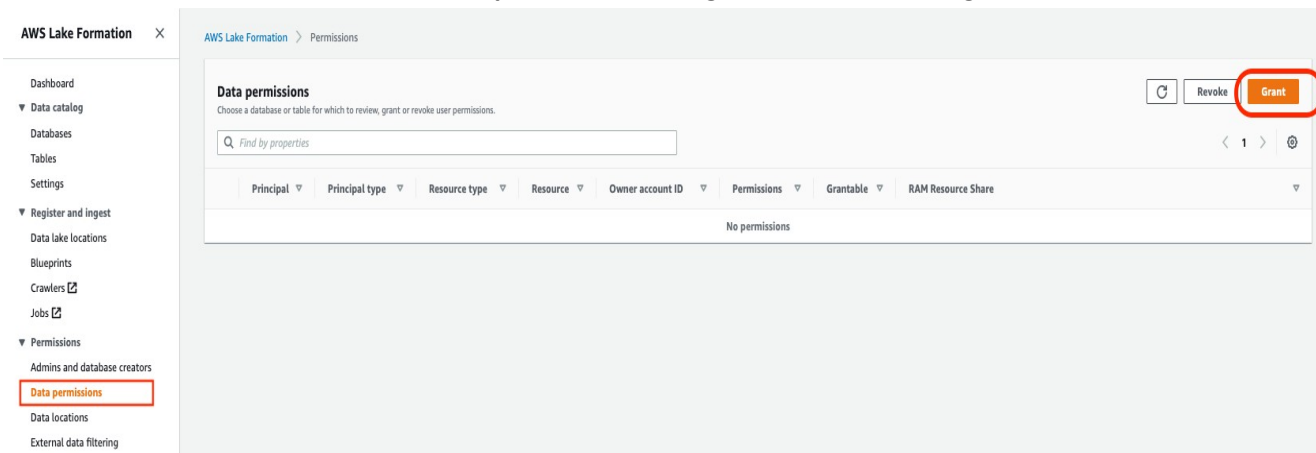
Permissions boundary (not set)

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

Lab 4. AWS Lake Formation

```
{
  "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**

Lab 4. AWS Lake Formation

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 X
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

ticketdata X
913536263025

Load more

Tables - optional
Select one or more tables.

Choose tables

lakeformation_sportstickets_dms_sample_player X
No description available

Load more

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 X
double

first_name X
string

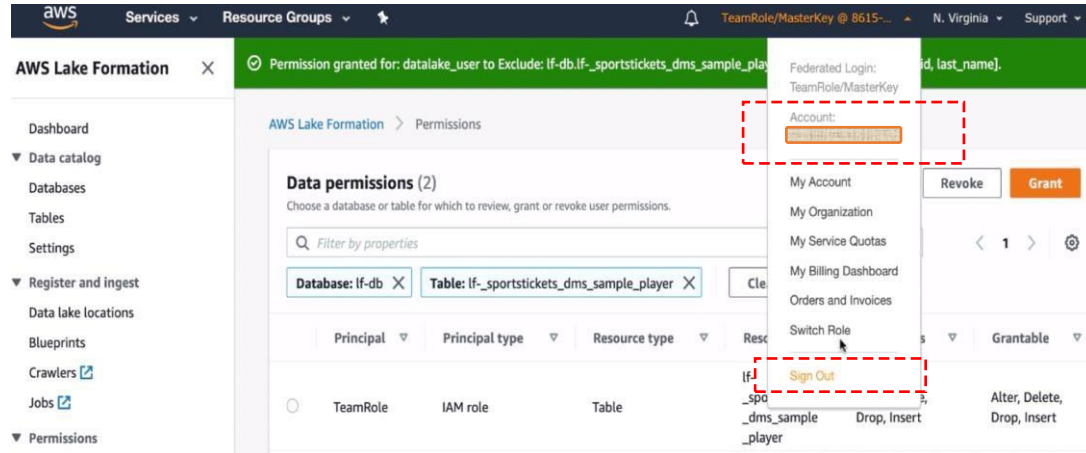
Grantable permissions
Choose the permission that may be granted to others.

☐ **Select**

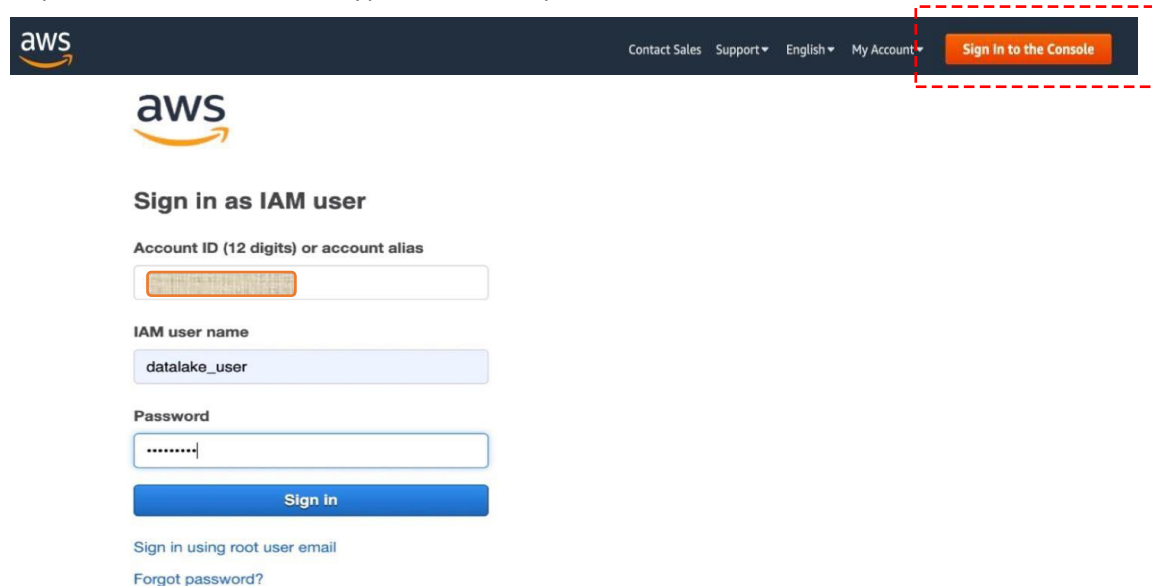
[Optional] Verify data permissions using Athena

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

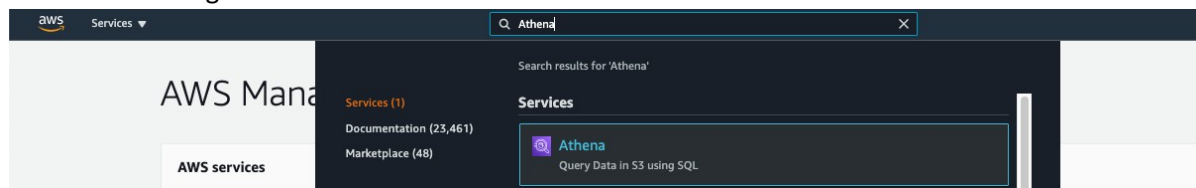
Sign out your AWS Account. Before doing that, write down your **Account ID**.



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

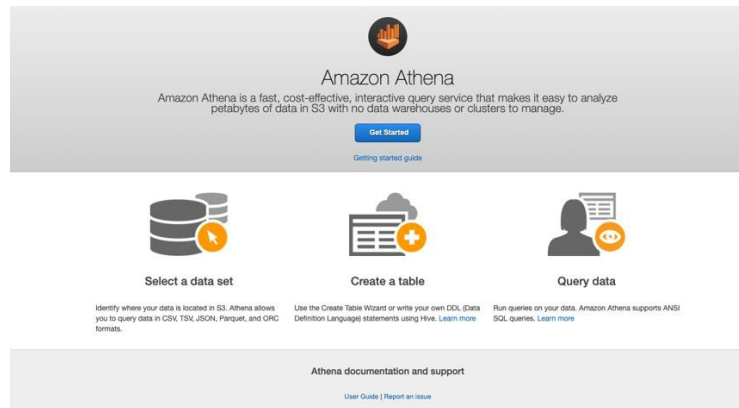


1. Make sure to change the region to **the appropriate AWS region**
2. Navigate to **Athena console**

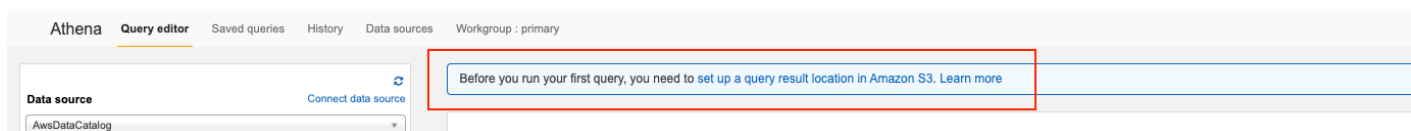


3. If you see a "Get Started" page, if it's the first time to use Athena in this AWS Account. To proceed, click **Get Started**

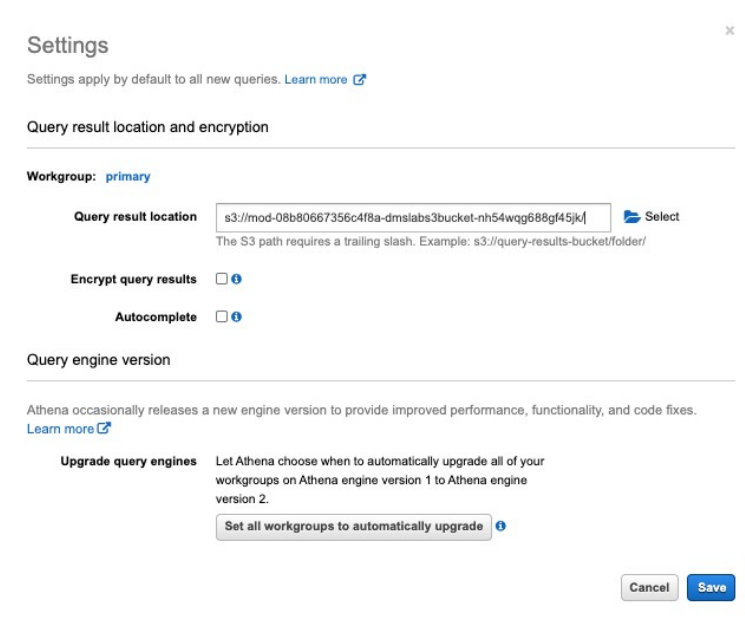
Lab 4. AWS Lake Formation



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.

Lab 4. AWS Lake Formation

The screenshot displays the AWS Athena Query Editor. On the left, the 'Data source' is set to 'AwsDataCatalog' and the 'Database' is 'ticketdata'. A list of 17 tables is shown, including 'lakeformation_sportstickets_dms_sample_pl...', 'mlb_data', 'name_data', 'nfl_data', 'nfl_stadium_data', 'parquet_dms_parquet (Partitioned)', 'person', 'player', 'seat', 'seat_type', 'sport_division', 'sport_league', 'sport_location', 'sport_team', 'sporting_event', 'sporting_event_ticket', and 'ticket_purchase_hist'. The main query editor shows a SQL query: `SELECT * FROM "ticketdata"."lakeformation_sportstickets_dms_sample_player" limit 10;`. Below the query, the 'Run query' button is highlighted, and the status indicates '(Run time: 0.62 seconds, Data scanned: 43.71 KB)'. The 'Results' section shows a table with 10 rows and 3 columns: 'id', 'first_name', and 'last_name' (partially visible). The first row is (1, Adam, ...), the second is (2, A.J., ...), and so on, up to (10, Braden, ...).

	id	first_name
1	1.0	Adam
2	11.0	A.J.
3	21.0	Alex
4	31.0	Andrew
5	41.0	Andy
6	51.0	Archie
7	61.0	Ben
8	71.0	Braden
9	81.0	Bradin
10	91.0	Braden

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!