



Amazon Web Services Data Engineering Immersion Day

Lab 4. AWS Lake Formation

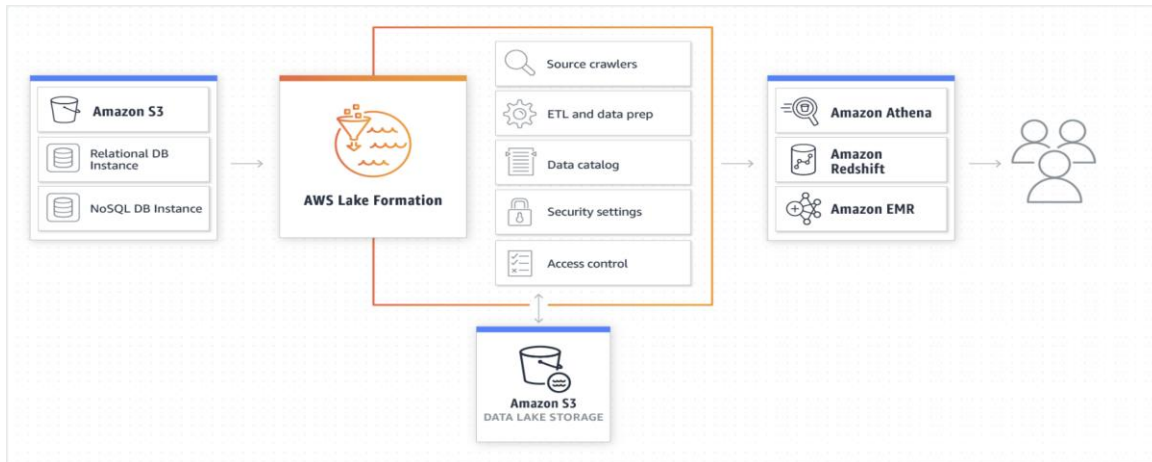
August 2020

Table of Contents

<i>Introduction</i>	<i>3</i>
<i>Prerequisites.....</i>	<i>3</i>
<i>Get Started Using the Lab Environment.....</i>	<i>4</i>
<i>Setup Network Configuration for AWS Glue</i>	<i>6</i>
<i>Create an IAM role to use with Lake Formation:</i>	<i>6</i>
<i>Create Glue JDBC connection for RDS.....</i>	<i>7</i>
<i>Lake Formation - Add Administrator and start workflows using Blueprints.</i>	<i>9</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>21</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.



Today, you are attending a formal AWS event. If in the future you might want to perform these labs in your own AWS environment by yourself, you can follow instructions here - <https://aws-dataengineering-day.workshop.aws/en/1200.html>

Prerequisites

1. Make sure you have the Postgres source database information from your Event Dashboard handy. If you are running the lab outside of AWS hosted event, please find the **DMSInstanceEndpoint** parameter value from **dmslab-instructor** [CloudFormation Outputs](#) tab.

RDS DB Info [Readme](#)

Outputs:
No outputs defined

2. Completed Lab 1. Hydrating the Data Lake with DMS or Lab2. Copy RDS Source Data - Prelab
3. Must completed Part A in Lab 2. ETL with AWS Glue

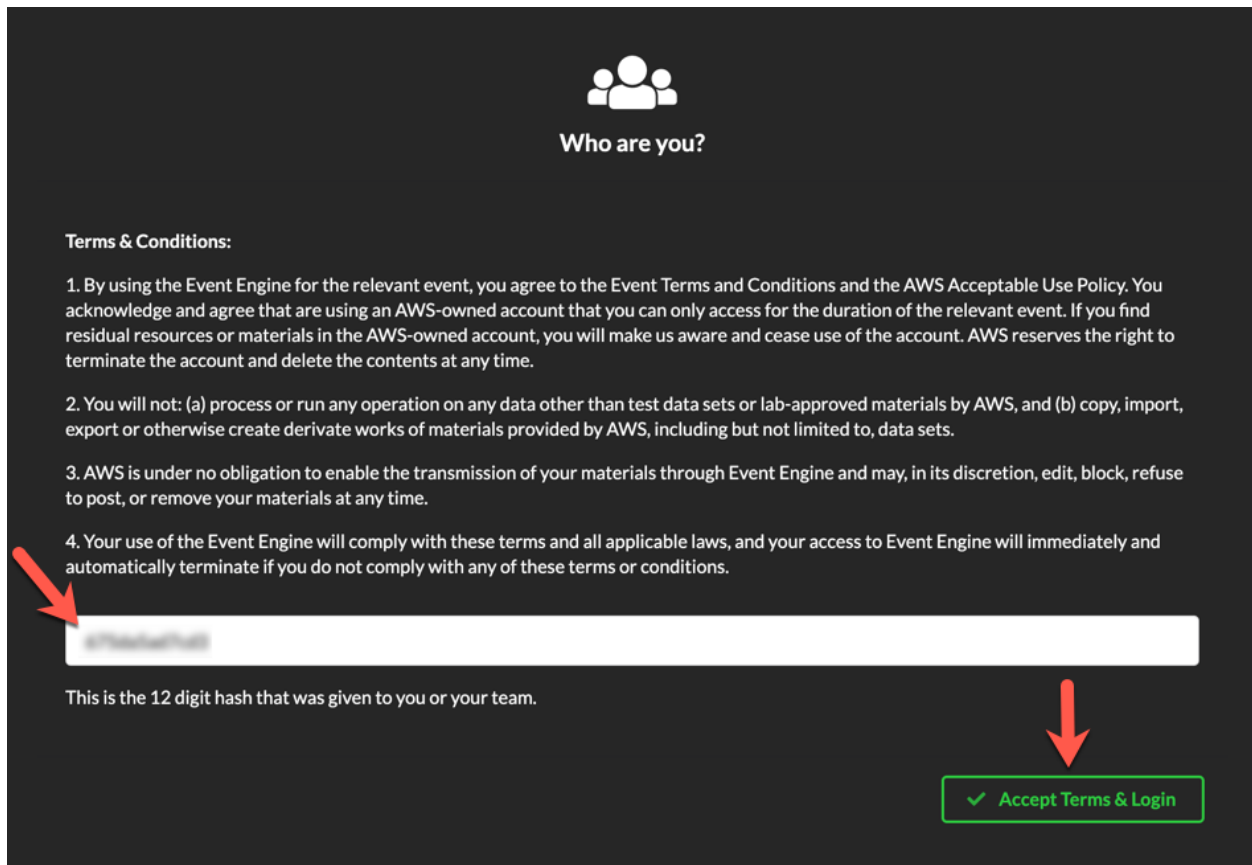
Get Started Using the Lab Environment

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/aws-samples/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:



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 where indicated in subsequent labs:


Lab 4. AWS Lake Formation


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-YLTJA13VW6WT 


S3BucketWorkgroupA
mod-3fccddd609114925-s3bucketworkgroupa-tbon3m1mkunh 



S3BucketWorkgroupB
mod-3fccddd609114925-s3bucketworkgroupb-18ygl8nfp8ead 

WorkgroupManagerUser
mod-3fccddd609114925-WorkgroupManagerUser-5IVE0UQNIBG4 

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

Team Dashboard

 Event

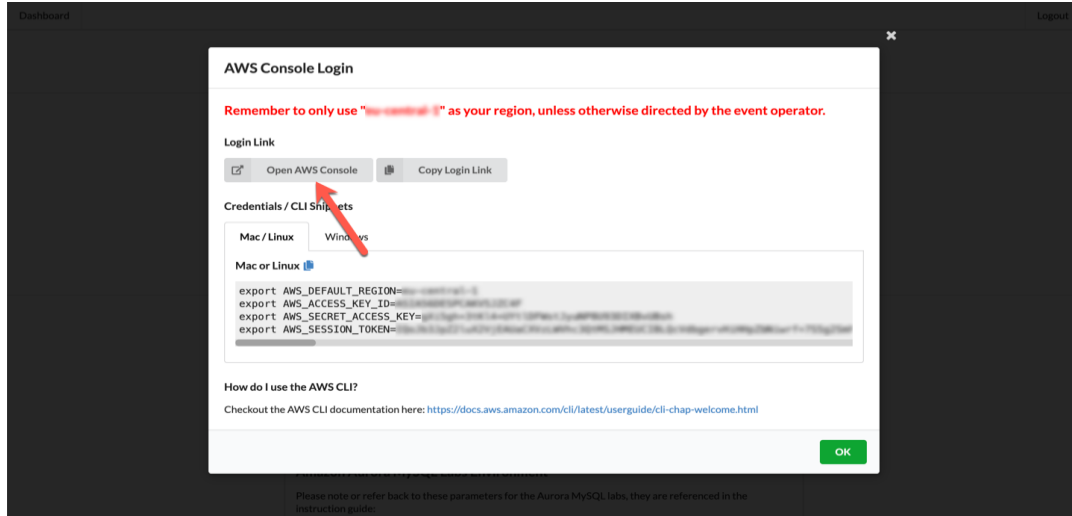
 

Event: Data Engineering Immersion Day - Test
Team Name:

Event ID: d2302d4ae9ff4ea2857846b74f7de7e2
Team ID: 1c2f7ad7ec044b0b8276f917c5983133

Lab 4. AWS Lake Formation

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

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

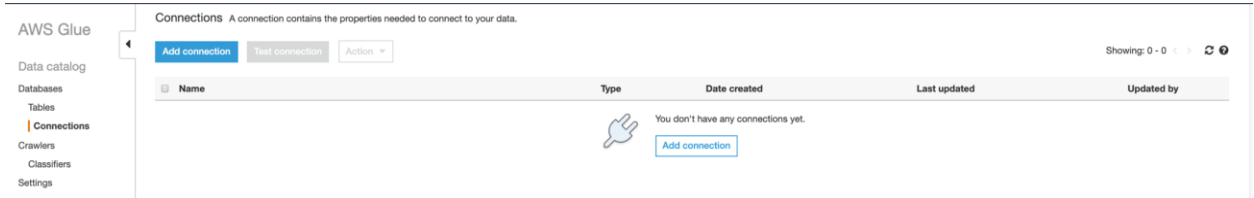
Create an IAM role to use with Lake Formation:

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.

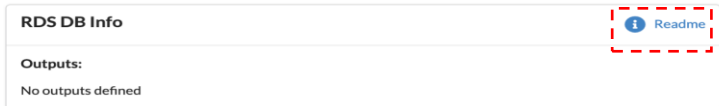
When you create a workflow, you must assign it an AWS Identity and Access Management (IAM) role that enables Lake Formation to set up the necessary resources on your behalf to ingest the data. In this lab, we've pre-created an IAM role for you, called `<random>-LakeFormationWorkflowRole-<random>`

Create Glue JDBC connection for RDS

1. Navigate to the AWS Glue console: <https://console.aws.amazon.com/glue/home?region=us-east-1>
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 **RDS_Server_Name** from **RDS DB Info** dashboard.
 
 - b. If you are running the lab outside of AWS event, find the **DMSInstanceEndpoint** value on the **dmslab-instructor CloudFormation** 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.

Lab 4. AWS Lake Formation

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	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 ⓘ

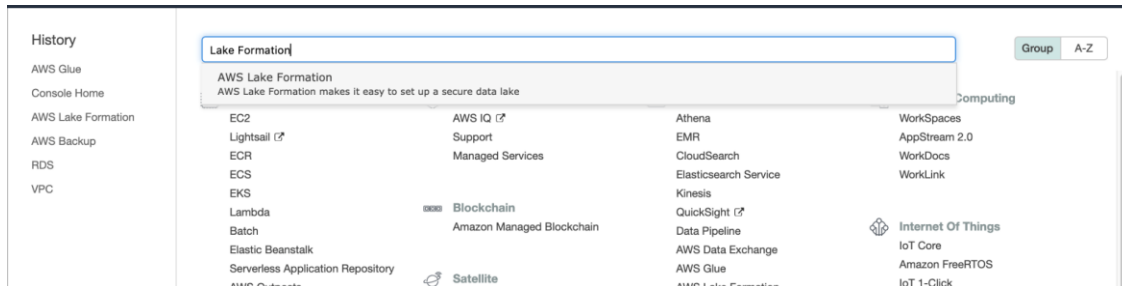
↻

Ensure that this role has permission to access your data store.
[Create IAM role.](#)

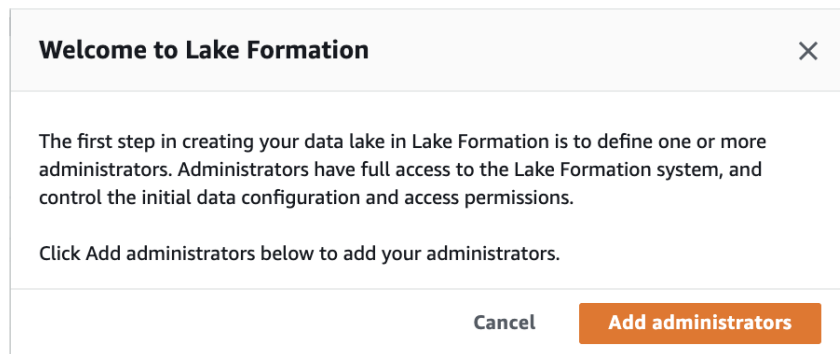
Lake Formation - Add Administrator and start workflows using Blueprints.

Navigate to the AWS Lake Formation service:

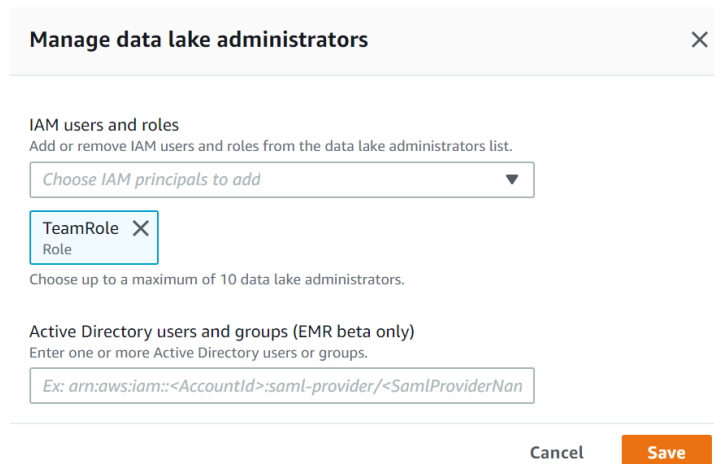
<https://console.aws.amazon.com/lakeformation/home?region=us-east-1#databases>



1. If you are logging into the lake formation console for the first time then you must add administrators first in order to do that follow Steps 2 and 3. Else skip to Step 4.
2. Click **Add administrators**

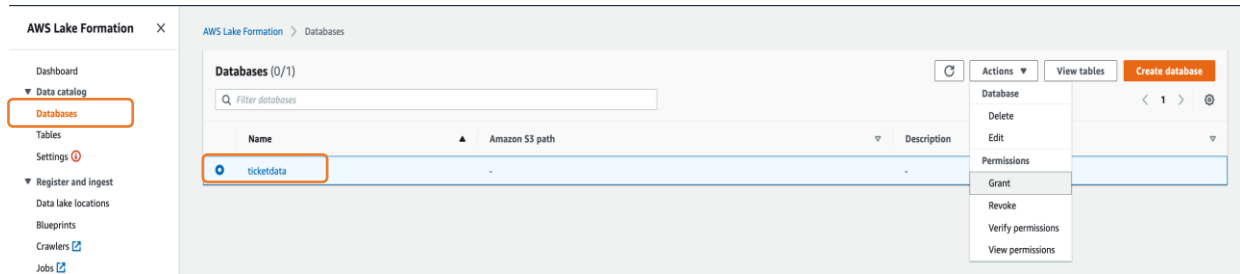


3. Add **TeamRole** Role as the Lake Formation Administrator and Click **Save**



Lab 4. AWS Lake Formation

4. 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**



5. Under “IAM Users and Roles”, select two roles: the Lake Formation role that was pre-created: **<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

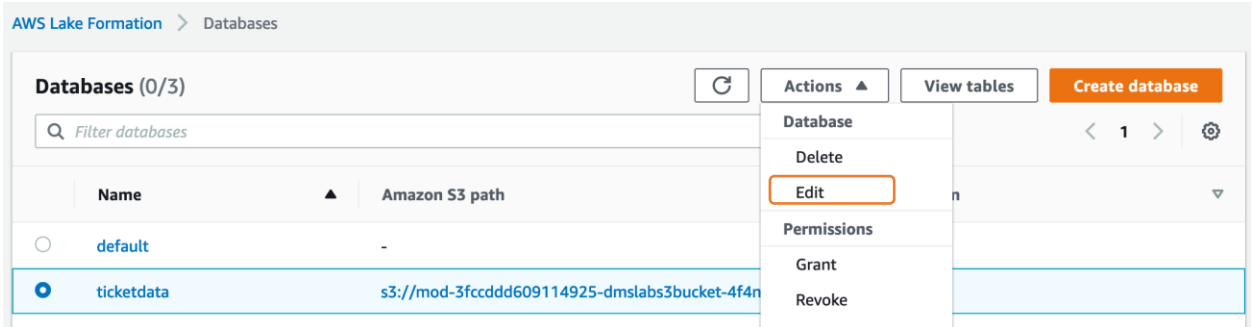
☒ **Super**
This permission allows the principal to grant any of the above permissions and supersedes those grantable permissions.

Cancel

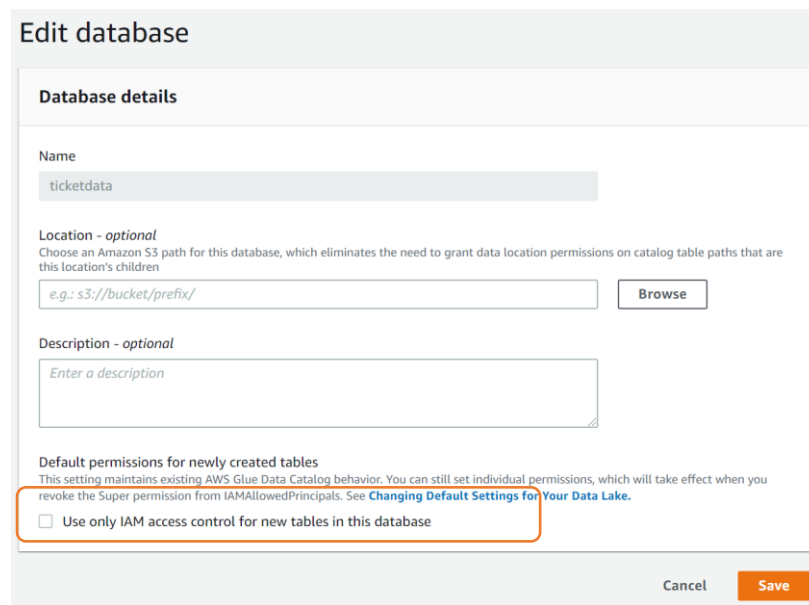
Grant

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

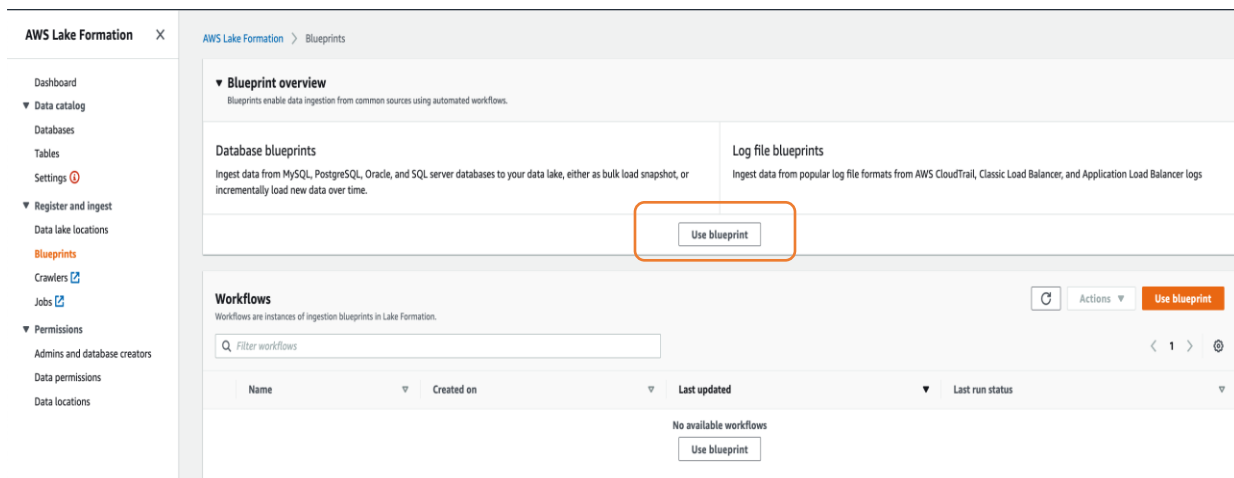
Lab 4. AWS Lake Formation



7. 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.



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



Lab 4. AWS Lake Formation

- a. For **Blueprint Type**, select **Database snapshot**
- b. Under **Import Source**
 - i. For **Database Connection** choose **glue-rds-connection**
 - ii. For **Source Data Path** enter **sportstickets/dms_sample/player**

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. Substitute the percent (%) wildcard for schema or table.

sportstickets/dms_sample/player

- c. 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

Lab 4. AWS Lake Formation

- iii. Add **/lakeformation** at the end of the bucket url path, e.g.
s3://mod-08b80667356c4f8a-dmslabs3bucket-nh54wqg771lk/**lakeformation**
- iv. For **Data Format** choose **Parquet**

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

d. For **Import Frequency**, Select **Run On Demand**

e. 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

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

IAM role

LakeFormationWorkflowRole

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

9. Leave other options as default, click **Create**, and wait for the console to report that the workflow was successfully created.

10. 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.
11. Once the workflow starts executing, you will see the status changes from running → discovering → Completed

Name	Created on	Last updated	Last run status
RDS-S3-Glue-Workflow	Mon, Mar 23, 2020, 10:50 PM UTC	Mon, Mar 23, 2020, 10:50 PM UTC	Discovering

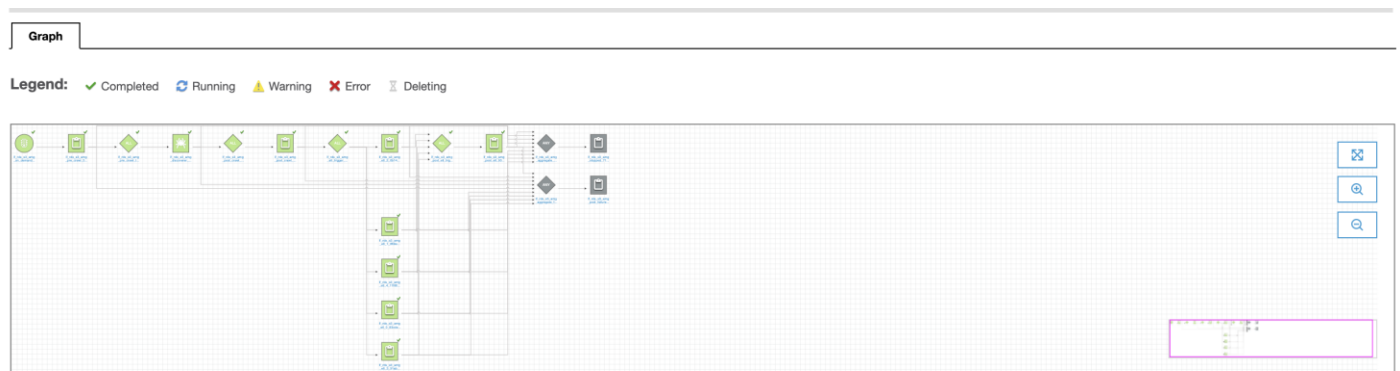
Explore the Underlying Components of a Blueprint

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**.

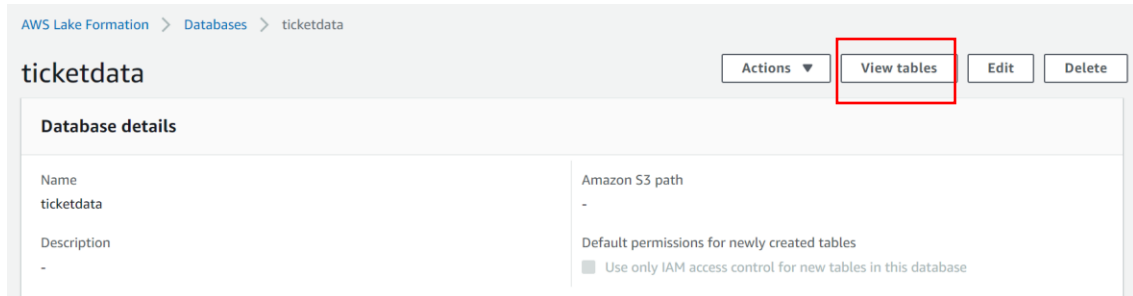
Name	Started on	Run ID
lf_rds_s3_amg	Mon, Dec 16, 2019, 6:07 AM UTC	wr_bcf4efeb081293bb613e61c2a586ee37f9ccbdd1ca97dd953d28514a0951908f

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.



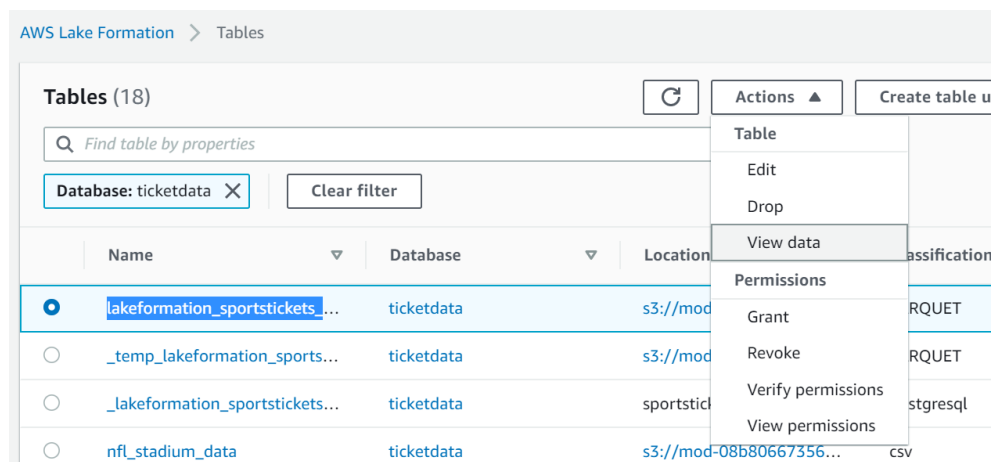
Explore workflow results in Athena

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



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

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



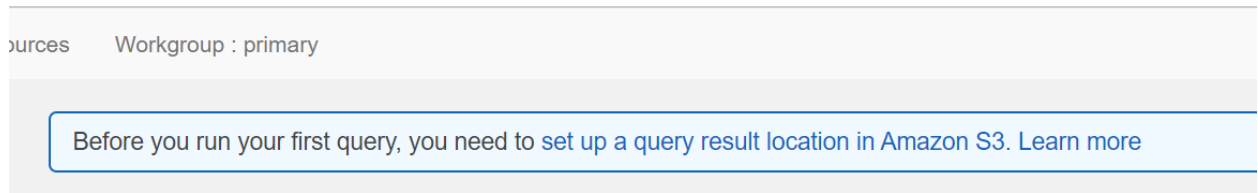
This will now take you to **Athena** console.

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**



Lab 4. AWS Lake Formation

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/** and click **Save**

A screenshot of a 'Settings' pop-up window. The title bar says 'Settings' with a close 'x' button. Below the title, it says 'Settings apply by default to all new queries. [Learn more](#)'. Underneath, it says 'Workgroup: primary'. There are two checkboxes: 'Encrypt query results' and 'Autocomplete', both with an information icon. The 'Query result location' field is a text input containing 's3://mod-08b80667356c4f8a-dmslabs3bucket-nh54wqg771lk/' with an information icon to its right. Below the input field, an example is shown: 'Example: s3://query-results-bucket/folder/'. At the bottom right, there are 'Cancel' and 'Save' buttons.

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:

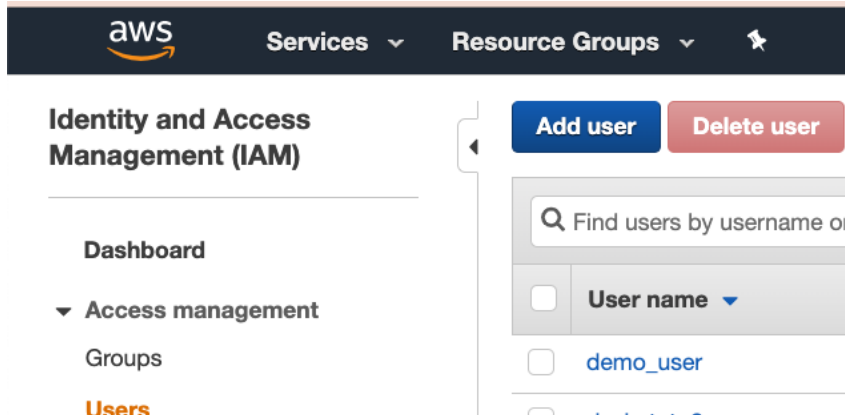
```
SELECT count(*) as recordcount FROM
"ticketdata"."lakeformation_sportstickets_dms_sample_player" limit 10;
```

Congratulation!!! 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**.

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*

[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**

Lab 4. AWS Lake Formation

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)

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

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

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

Add user Delete user

Find users by username or access key Showing 4 results

	User name	Groups	Access key age	Password age	Last activity	MFA
<input checked="" type="checkbox"/>	datalake_user	None	None	Today	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
AmazonAthenaFullAccess	AWS managed policy

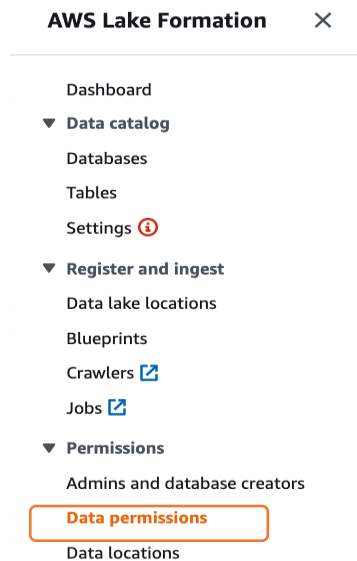
Attached directly

Permissions boundary (not set)

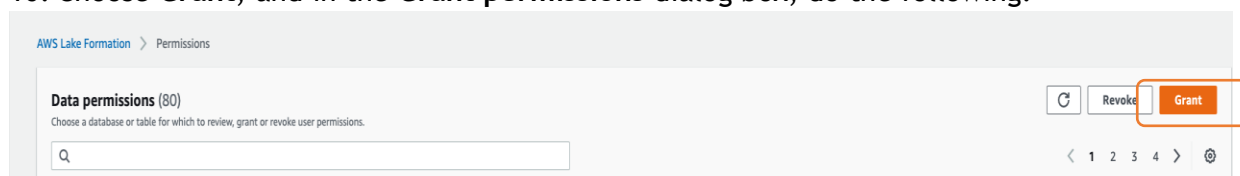
Use the following json snippet replacing `<your_dmslabs3bucket_unique_name>` with the name of your dmslabs3bucket, e.g. `mod-08b80667356c4f8a-dmslabs3bucket-nh54wqg771lk`

```
{
  "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**
9. Navigate to [Lake Formation console](#), under **Permissions** choose **Data permissions**.



10. Choose **Grant**, and in the **Grant permissions** dialog box, do the following:



Lab 4. AWS Lake Formation

- For **IAM user and roles**, choose **datalake_user**.
- For **Database**, choose **ticketdata**
- The **Table** list populates.
- For **Table**, choose **lakeformation_sportstickets_dms_sample_player**.
- For **Columns**, select **Include Columns** and choose **id**, **first_name**
- For **Table permissions**, untick **Super** and choose **Select**.

11. Choose Grant.

Grant permissions gov__sportstickets_dms_sample_player ✕
Choose the access permissions to grant. IAM permissions must also allow access.

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

Choose IAM principals to add

data_lake_user ✕
User

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>

Column - optional
Choose filter type

Include columns

Include columns
Grant permissions to access the selected columns.

Choose columns

id ✕ double first_name ✕ string

Table permissions
Choose the specific access permissions to grant.

☐ Alter ☐ Insert ☐ Drop ☐ Delete ☒ Select

☐ **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.

☐ Alter ☐ Insert ☐ Drop ☐ Delete ☐ Select

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

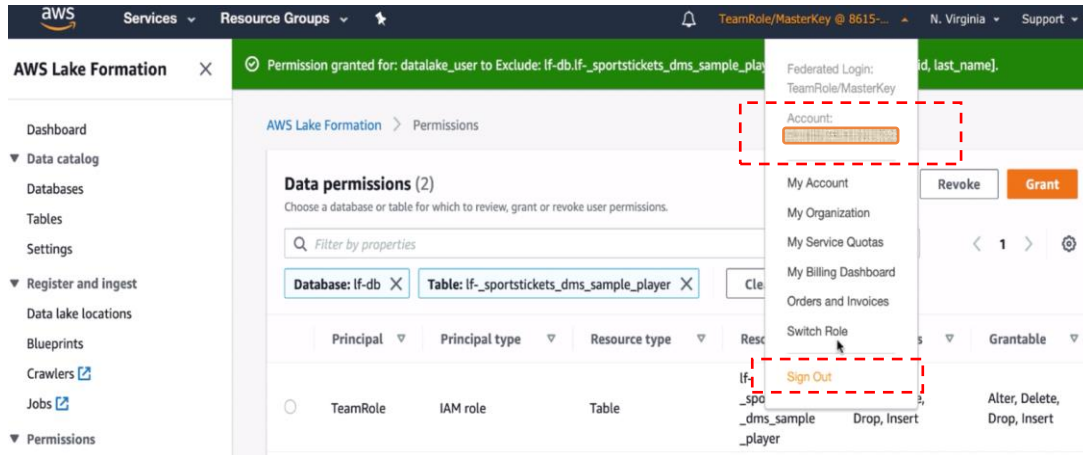
Cancel

Grant

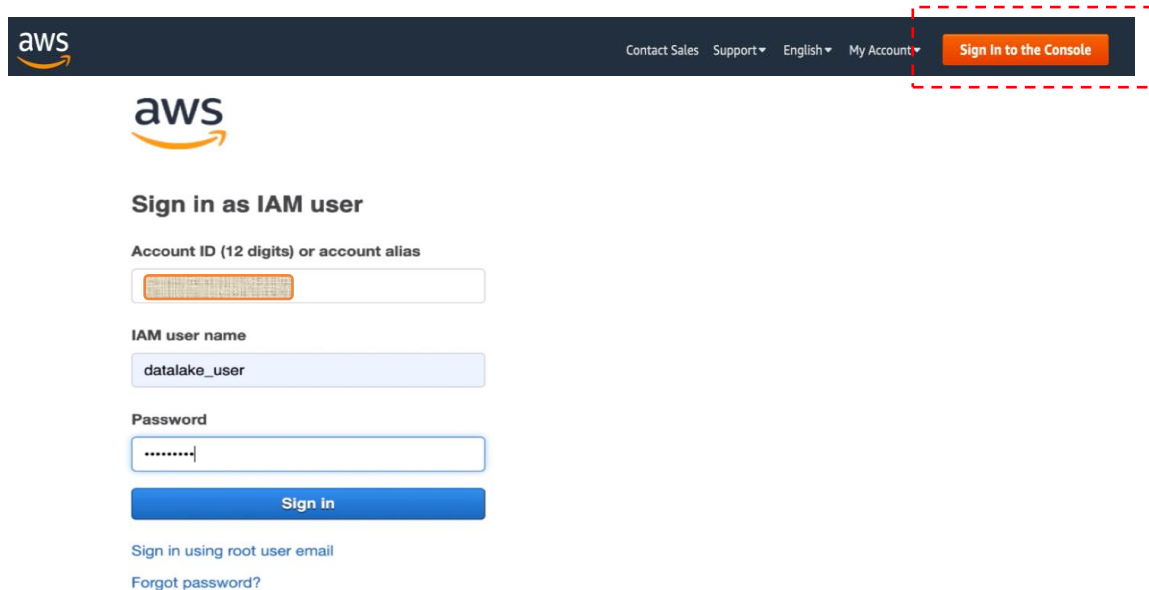
[Optional] Verify data permissions using Athena

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

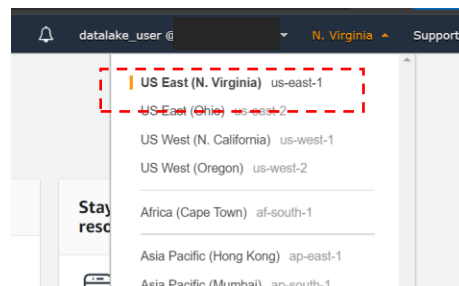
1. Sign out your AWS Account. Before doing that, write down your **Account ID**.



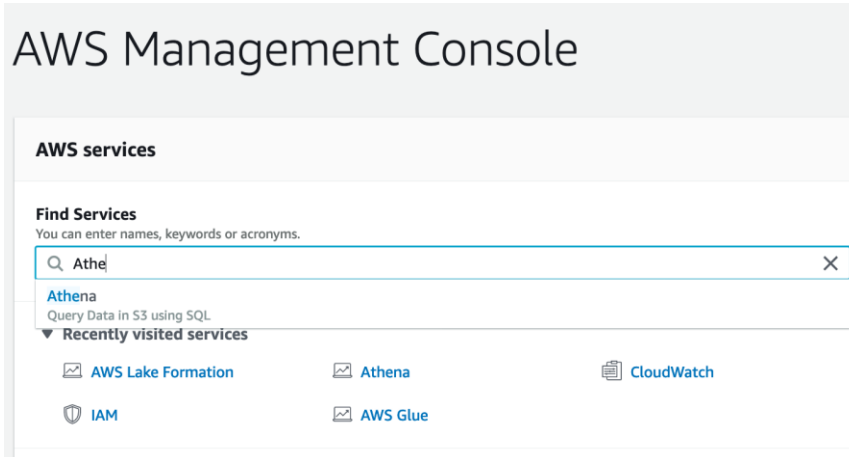
2. On the same web page, sign back in as the IAM user **datalake_user**, using **master123** as password. Note: remove *hyphens* '-' from your Account ID



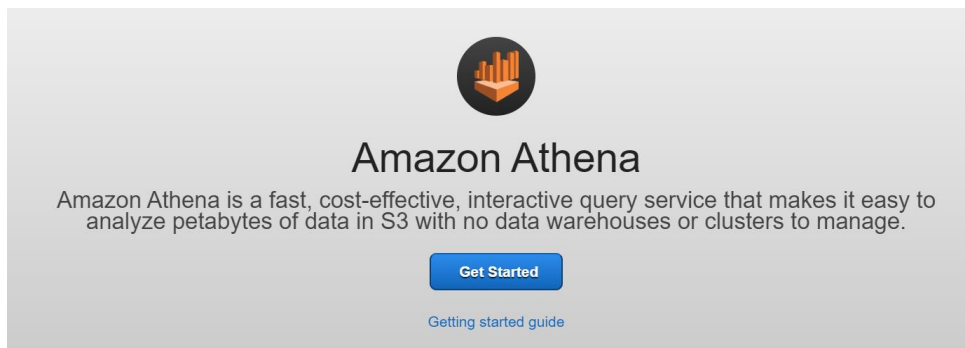
1. Make sure to change the region to **us-east-1** (N. Virginia):



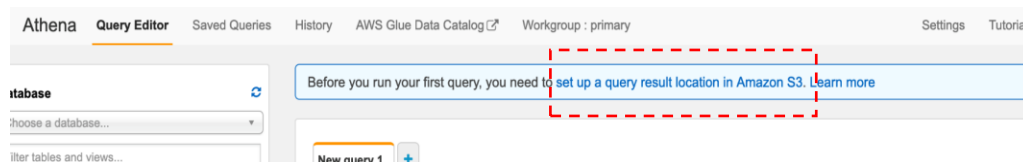
2. Navigate to **Athena console**



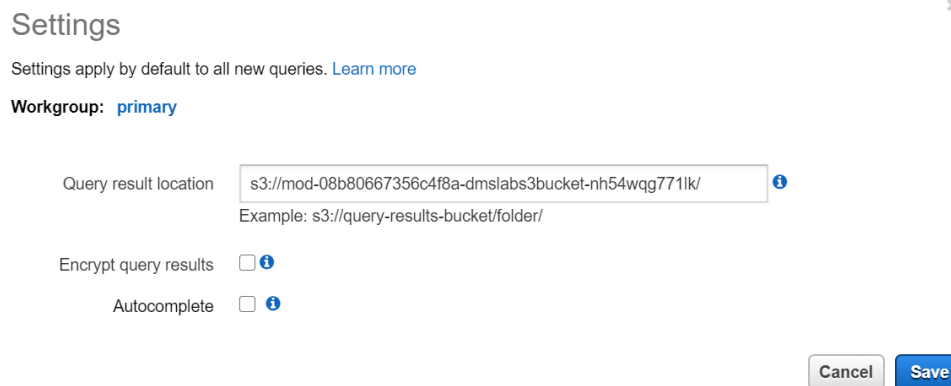
3. If you see a “Get Started” page, if it’s the first time to use Athena in this AWS Account. To proceed, click Get Started



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**



Lab 4. AWS Lake Formation

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

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

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

The screenshot shows the AWS Athena Query Editor interface. On the left, the 'Data source' is set to 'AwsDataCatalog' and the 'Database' is 'ticketdata'. A list of tables is shown, including 'lakeformation_sportstickets_dms_sample_player'. The main query editor displays the SQL query: `SELECT * FROM "ticketdata"."lakeformation_sportstickets_dms_sample_player" limit 10;`. Below the query editor, the 'Run query' button is visible, along with 'Save as' and 'Create' buttons. The execution status shows a run time of 0.62 seconds and 43.71 KB of data scanned. The 'Results' section displays a table with two columns: 'id' and 'first_name'. The table contains 10 rows of data, showing IDs from 1.0 to 10.0 and first names from Adam to Brandon. The 'last_name', 'sports_team_id', and 'full_name' columns are not visible in the results, demonstrating column-level access control.

	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	Brandon