



# **Amazon Web Services Data Engineering Immersion Day**

---

Lab 4. AWS Lake Formation

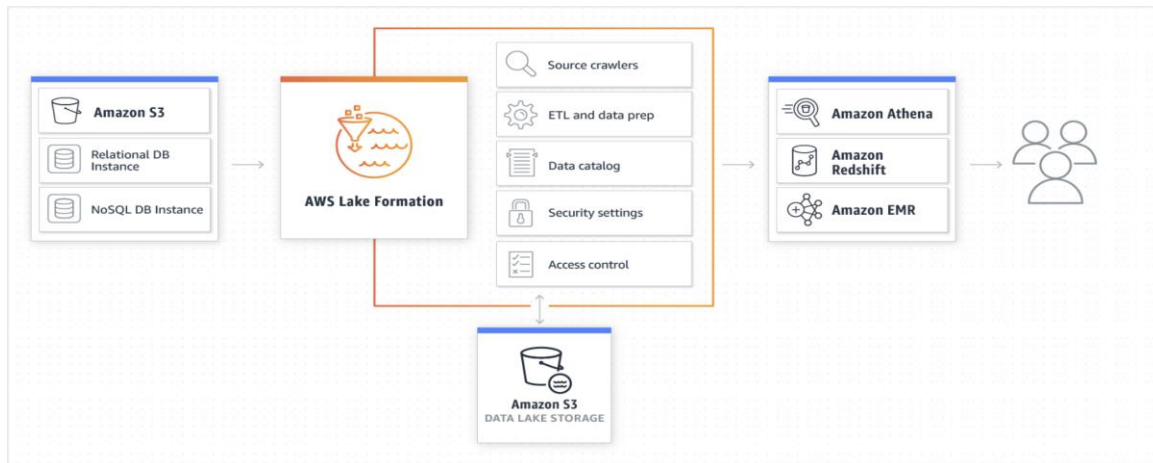
August 2020

## 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>15</i>
<i>Explore workflow results in Athena.....</i>	<i>15</i>
<i>[Optional] Grant fine grain access controls to Data Lake user .....</i>	<i>18</i>
<i>[Optional] Verify data permissions using Athena .....</i>	<i>22</i>

### Introduction

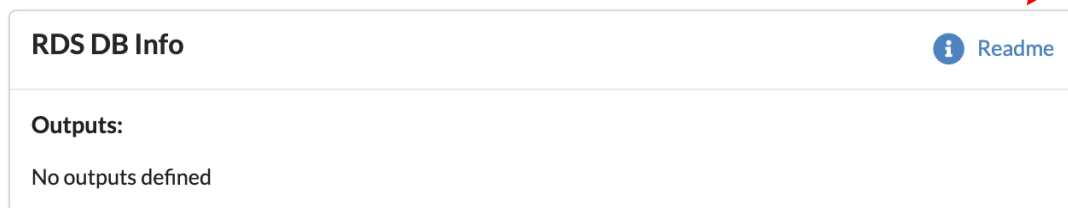
This lab will give you an understanding of the AWS Lake Formation – a service that makes it easy to set up a secure data lake, as well as Athena for querying the data you import into your data lake.



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.



2. Completed Lab 1. Hydrating the Data Lake with DMS
3. Completed 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.

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-YLTJA13WW6WT 


**S3BucketWorkgroupA**  
mod-3fccddd609114925-s3bucketworkgroupa-tbon3m1mkunh 



**S3BucketWorkgroupB**  
mod-3fccddd609114925-s3bucketworkgroupb-18ygl8nfp8ead 

**WorkgroupManagerUser**  
mod-3fccddd609114925-WorkgroupManagerUser-5IVE0UQNIBG4 

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

**Team Dashboard**

 Event

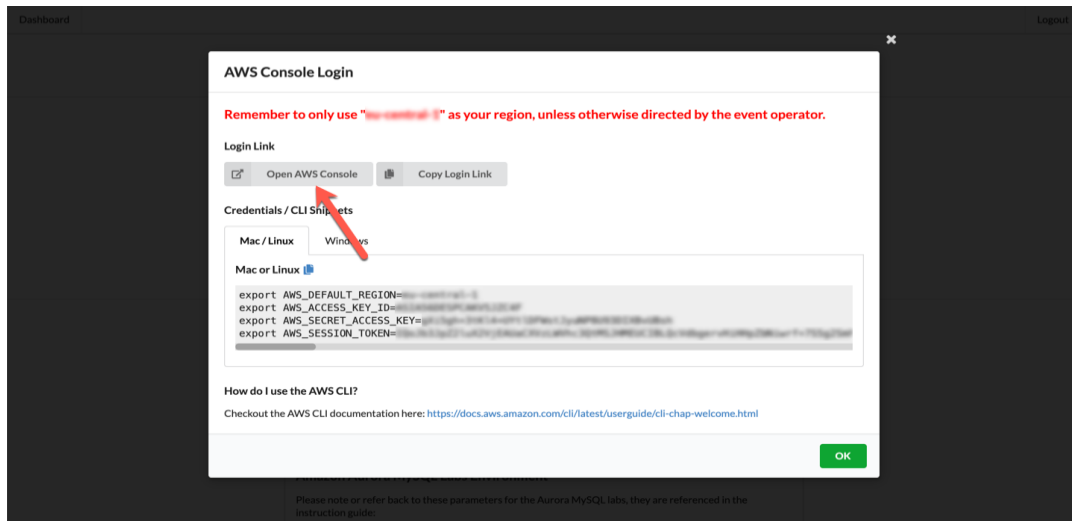
 

**Event:** Data Engineering Immersion Day - Test  
Team Name:

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 SKIP THIS SECTION

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.

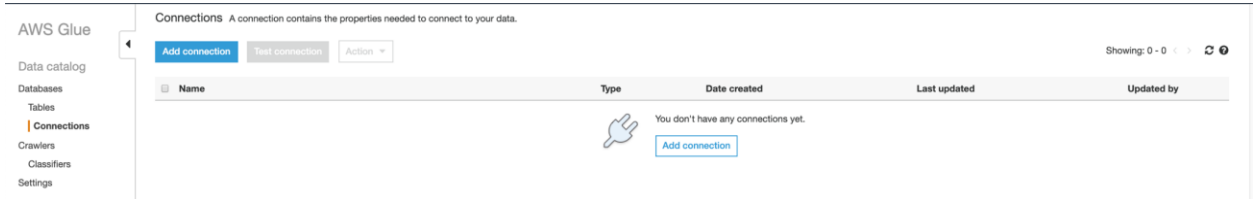
## Create an IAM role to use with Lake Formation: SKIP THIS SECTION

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 SKIP THIS SECTION

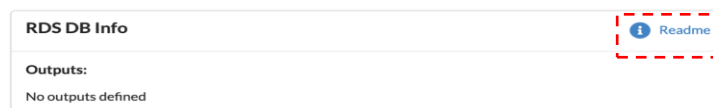
1. Navigate to the AWS Glue console: <https://console.aws.amazon.com/glue/home?region=us-east-1>
2. On the AWS Glue menu, select **Connections**.



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 called **<random>-LakeFormationWorkflowRole-<random>** and then click **Test Connection**.

×

## Test connection

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

**IAM role** ⓘ

↻

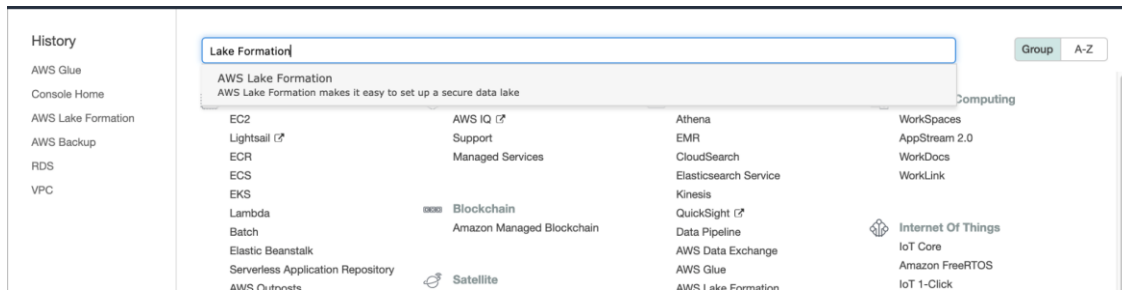
Ensure that this role has permission to access your data store.  
[Create IAM role.](#)



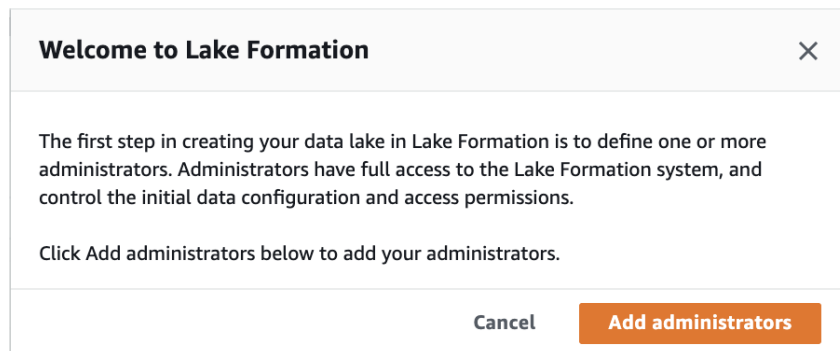
## Lake Formation – Add Administrator and start workflows using Blueprints. **GO TO STEP 4**

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

**Manage data lake administrators** ✕

**IAM users and roles**  
Add or remove IAM users and roles from the data lake administrators list.  

Choose IAM principals to add ▼

TeamRole ✕

Role

Choose up to a maximum of 10 data lake administrators.

**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>`

Cancel

Save

- 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**

The screenshot shows the AWS Lake Formation console. On the left, the 'Databases' link is highlighted in the navigation pane. The main area shows a table of databases with one entry, 'ticketdata'. The 'Actions' menu is open for the 'ticketdata' row, and the 'Grant' option is selected. The 'Permissions' sub-menu is also visible, showing 'Grant', 'Revoke', 'Verify permissions', and 'View permissions'.

- 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.

**No TeamRole for now**

## Lab 4. AWS Lake Formation

**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

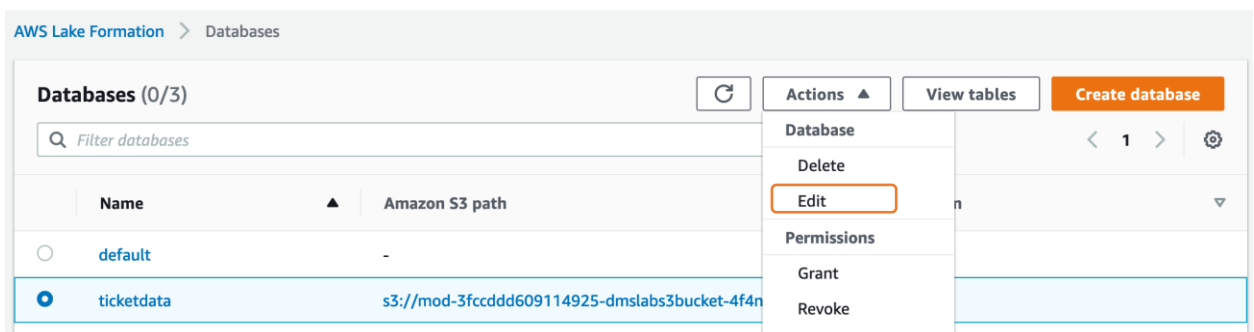
☒ **Super**  
This permission allows the principal to grant any of the above permissions and supersedes those grantable permissions.

Cancel

Grant

If you can't choose Super,  
select all of the above: Create, Alter, Drop

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



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.

## Lab 4. AWS Lake Formation

### 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

8. 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.

Actions Use blueprint

Name	Created on	Last updated	Last run status
No available workflows			

Use blueprint

- 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**

## Lab 4. AWS Lake Formation

The screenshot shows the 'Use a blueprint' wizard in AWS Lake Formation. The breadcrumb trail is 'AWS Lake Formation > Blueprints > Use a blueprint'. The main heading is 'Use a blueprint'. Below it, the 'Blueprint type' section is titled 'Configure a blueprint to create a workflow.' and contains five options: 'Database snapshot' (selected), 'Incremental database', 'AWS CloudTrail', 'Classic Load Balancer logs', and 'Application Load Balancer logs'. The 'Database snapshot' option is highlighted with a blue border and a blue radio button. Below this, the 'Import source' section is titled 'Configure the workflow source.' and contains a 'Database connection' dropdown menu with 'glue-rds-connection' selected, a 'Source data path' text input field with 'sportstickets/dms\_sample/player' entered, and a 'Create a connection in AWS Glue' link.

- 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.

The screenshot shows a dialog box titled 'Choose an Amazon S3 location in region us-east-1'. It contains a list of S3 buckets with radio buttons next to them. The bucket 'mod-3fccddd609114925-dmslabs3bucket-4f4ndmet5tmw' is selected. At the bottom, there are 'Cancel' and 'Select' buttons.

- iii. Add **/lakeformation** at the end of the bucket url path, e.g.  
`s3://mod-o8b8o667356c4f8a-dmslabs3bucket-nh54wqg771lk/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 Data Catalog. [Create database](#)

ticketdata

▼

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

s3://dmslab-student-dmslabs3bucket-4a27jjap6c5t/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** choose the precreated **...-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.

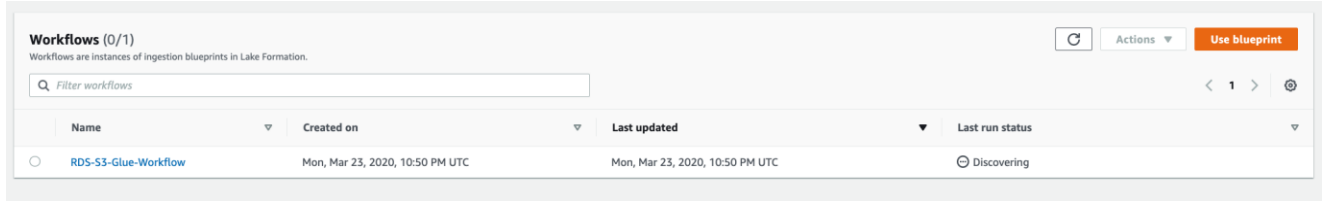
5

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 5-10s delay in the blueprint showing up. You may have to **hit refresh**.
11. Once the workflow starts executing, you will see the status changes from running → discovering → Completed

## Lab 4. AWS Lake Formation

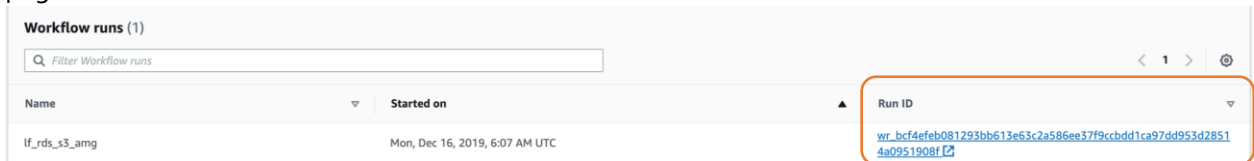


Workflows (0/1)			
Workflows are instances of ingestion blueprints in Lake Formation.			
Filter workflows			
Name	Created on	Last updated	Last run status
<a href="#">RDS-S3-Glue-Workflow</a>	Mon, Mar 23, 2020, 10:50 PM UTC	Mon, Mar 23, 2020, 10:50 PM UTC	Discovering

### 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 15-20 mins to finish its first execution. In the meantime, let us drill down to see what it creates for us;

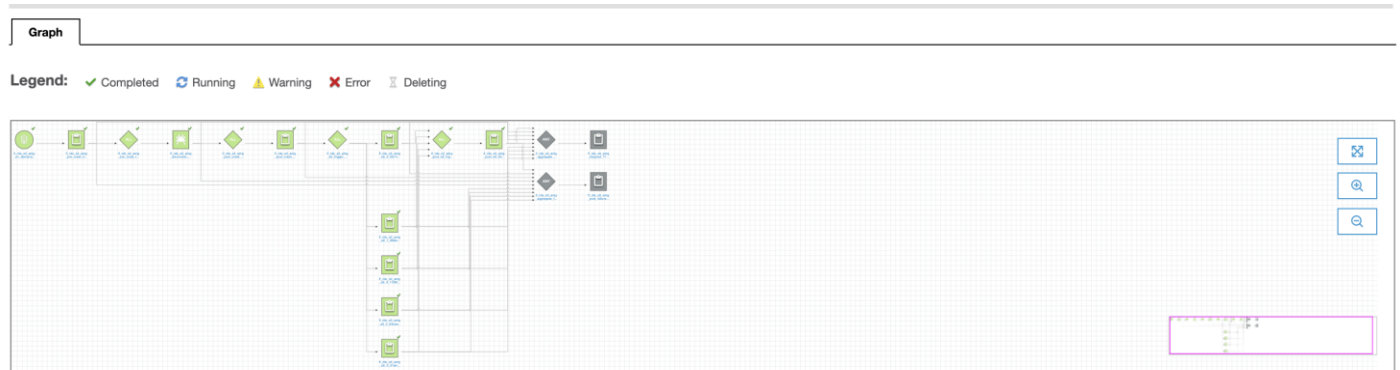
1. On the **Lake Formation console**, in the navigation pane, choose **Blueprints**
2. In the **Workflow section**, click on the **Workflow name**. This will direct you to the Workflow run page. Click on the **Run Id**.



Workflow runs (1)	
Filter Workflow runs	
Name	Started on
lf_rds_s3_amg	Mon, Dec 16, 2019, 6:07 AM UTC

Run ID: [wr\\_bcf4efeb081293bb613e63c2a586ee37f9ccbdd1ca97dd953d28514a0951908f](#)

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

## Lab 4. AWS Lake Formation

2. Navigate to **Databases** on the left panel and select **ticketdata**
3. Click on **View tables**

The screenshot shows the AWS Lake Formation console with the breadcrumb 'AWS Lake Formation > Databases > ticketdata'. The main heading is 'ticketdata'. On the right, there are buttons for 'Actions', 'View tables', 'Edit', and 'Delete'. Below this is a 'Database details' section with two columns. The left column contains 'Name' (ticketdata) and 'Description' (-). The right column contains 'Amazon S3 path' (-) and 'Default permissions for newly created tables' (a checkbox for 'Use only IAM access control for new tables in this database' which is checked).

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**

The screenshot shows the 'Tables (18)' view in the AWS Lake Formation console. A search bar at the top says 'Find table by properties'. Below it, a filter is set to 'Database: ticketdata'. A table lists several tables. The first table, 'lakeformation\_sportstickets...', is selected. A context menu is open over this table, showing options: 'Table', 'Edit', 'Drop', 'View data' (highlighted), 'Permissions', 'Grant', 'Revoke', 'Verify permissions', and 'View permissions'. The table has columns for 'Name', 'Database', 'Location', and 'Classification'. Other tables listed include '\_temp\_lakeformation\_sports...', '\_lakeformation\_sportstickets...', and 'nfl\_stadium\_data'.

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

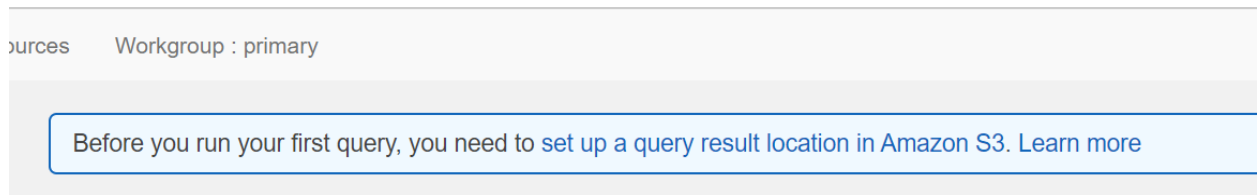
If you see a "Get Started" page, it's because it's the first time we're using Athena in this AWS Account. To proceed, click **Get Started**

The screenshot shows the Amazon Athena 'Get Started' page. At the top is the Athena logo (a stylized orange cube). Below it is the heading 'Amazon Athena'. The text reads: 'Amazon Athena is a fast, cost-effective, interactive query service that makes it easy to analyze petabytes of data in S3 with no data warehouses or clusters to manage.' At the bottom is a blue 'Get Started' button and a link to the 'Getting started guide'.



## 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 the 'Settings' pop-up window in the Amazon Lake Formation console. The window has a title bar with 'Settings' and a close button. Below the title, it says 'Settings apply by default to all new queries. [Learn more](#)'. The 'Workgroup' is set to 'primary'. There is a 'Query result location' field with a text input containing 's3://mod-08b80667356c4f8a-dmslabs3bucket-nh54wqg771lk/' and an information icon. Below the input, an example is shown: 'Example: s3://query-results-bucket/folder/'. There are two checkboxes: 'Encrypt query results' and 'Autocomplete', both currently unchecked, each with an information icon. 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;
```

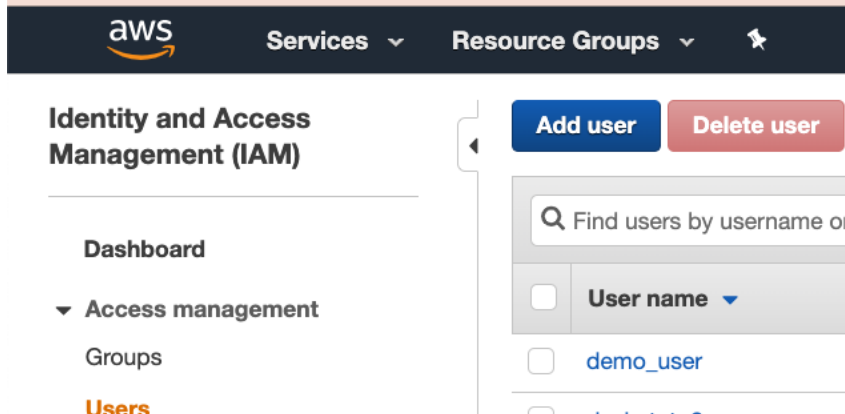
**Congratulation!!! You have completed lake formation lab. To explore more fine grain data lake security feature, continue to next section.**

— — — OPTIONAL — — —

## [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. Login as admin user to your account. Navigate to **IAM Console**: <https://console.aws.amazon.com/iam/home?region=us-east-1#/users> 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	<a href="#">Send email</a>


- 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
<input type="checkbox"/>	EC2Default	None	None	None	None	Not enabled

## Lab 4. AWS Lake Formation

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](#)

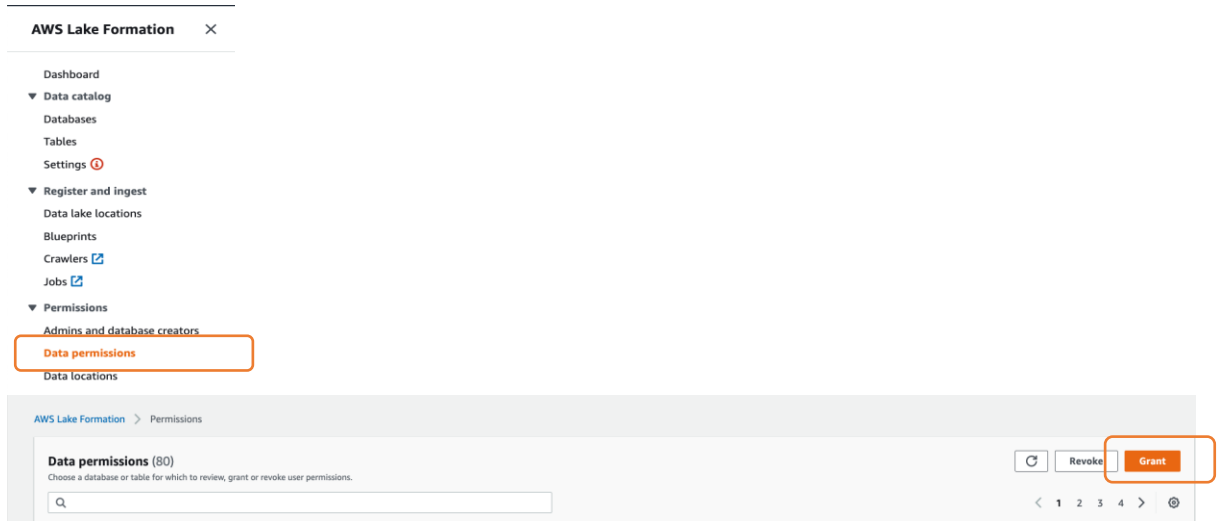
Policy name ▼	Policy type ▼
Attached directly	
▶  AmazonAthenaFullAccess	AWS managed policy ✕
▶ Permissions boundary (not set)	

Use the following json snippet replacing `<your_dmslabs3bucket_unique_name>` with the name of your dmslabs3bucket, e.g. `mod-o8b8o667356c4f8a-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 the **Lake Formation console**:  
<https://console.aws.amazon.com/lakeformation/home?region=us-east-1#dashboard>, in the navigation pane, under **Permissions**, choose **Data permissions**.

## Lab 4. AWS Lake Formation



10. Choose **Grant**, and in the **Grant permissions** dialog box, do the following:
  - a. For **IAM user and roles**, choose **datalake\_user**.
  - b. For **Database**, choose **ticketdata**
  - c. The **Table** list populates.
  - d. For **Table**, choose **lakeformation\_sportstickets\_dms\_sample\_player**.
  - e. For **Columns**, select **Include Columns** and choose **id, first\_name**
  - f. 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. In a new incognito browser window, navigate to the sign-in URL you wrote down earlier when you created an IAM User. Sign in as **datalake\_user** using **master123** as password

### Sign in as IAM user

Account ID (12 digits) or account alias

IAM user name

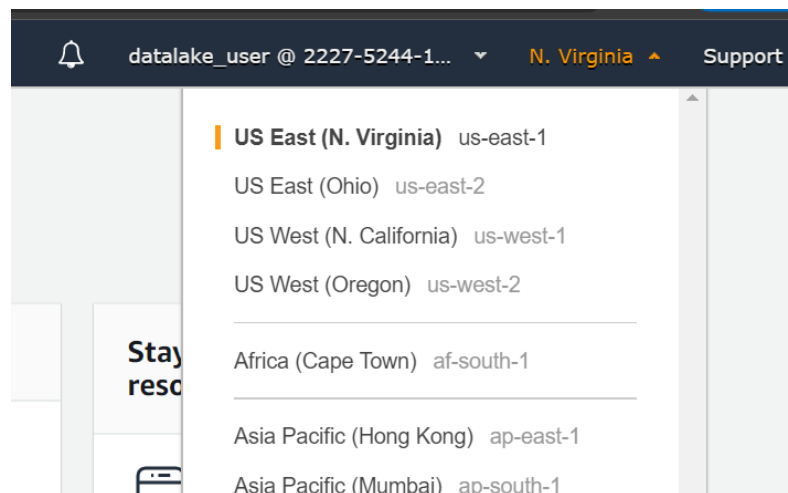
Password

Sign in

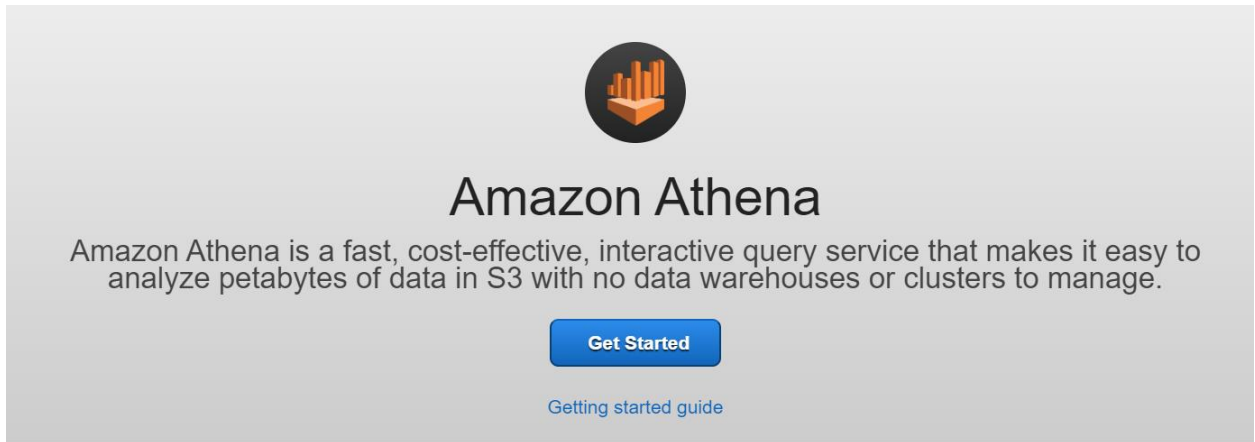
[Sign in using root user email](#)

[Forgot password?](#)

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



3. Navigate to the **Athena console (Services -> Athena)**. 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

sources Workgroup : primary

Before you run your first query, you need to [set up a query result location in Amazon S3](#). [Learn more](#)

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**

#### Settings

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

Workgroup: **primary**

Query result location  ⓘ  
Example: s3://query-results-bucket/folder/

Encrypt query results ☐ ⓘ

Autocomplete ☐ ⓘ

- Next, ensure database **ticketdata** is selected.
- Now run a **Select** query on the **lakeformation\_sportstickets\_dms\_sample\_player** table within the **ticketdata** database:

```
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 query result. The **datalake\_user** cannot see **last\_name**, **sports\_team\_id**, **full\_name** columns in the table.