



Amazon Web Services Data Engineering Immersion Day

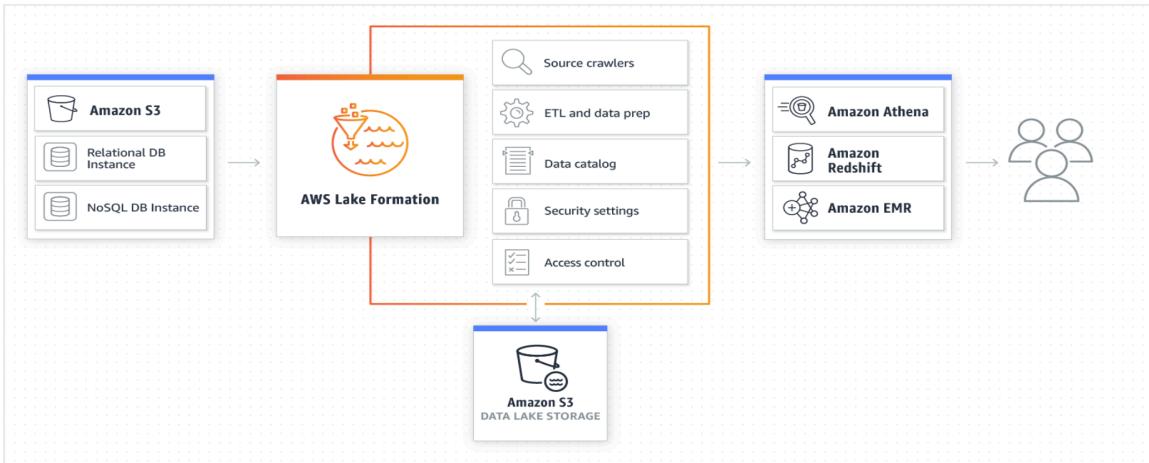
Lab 4. AWS Lake Formation
August 2020

Table of Contents

<i>Introduction</i>	3
<i>Prerequisites</i>	3
<i>Get Started Using the Lab Environment</i>	4
<i>Setup Network Configuration for AWS Glue</i>	6
<i>Create an IAM role to use with Lake Formation:</i>	6
<i>Create Glue JDBC connection for RDS</i>	6
<i>Lake Formation – Add Administrator and start workflows using Blueprints</i>	8
<i>Explore the Underlying Components of a Blueprint</i>	14
<i>Explore workflow results in Athena</i>	14
<i>[Optional] Grant fine grain access controls to Data Lake user</i>	17
<i>[Optional] Verify data permissions using Athena</i>	21

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.

A screenshot of the AWS CloudFormation Outputs tab. The tab has a header "RDS DB Info" and a "Readme" button with a red arrow pointing to it. Below the header, there is a section titled "Outputs:" with the sub-section "No outputs defined".

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

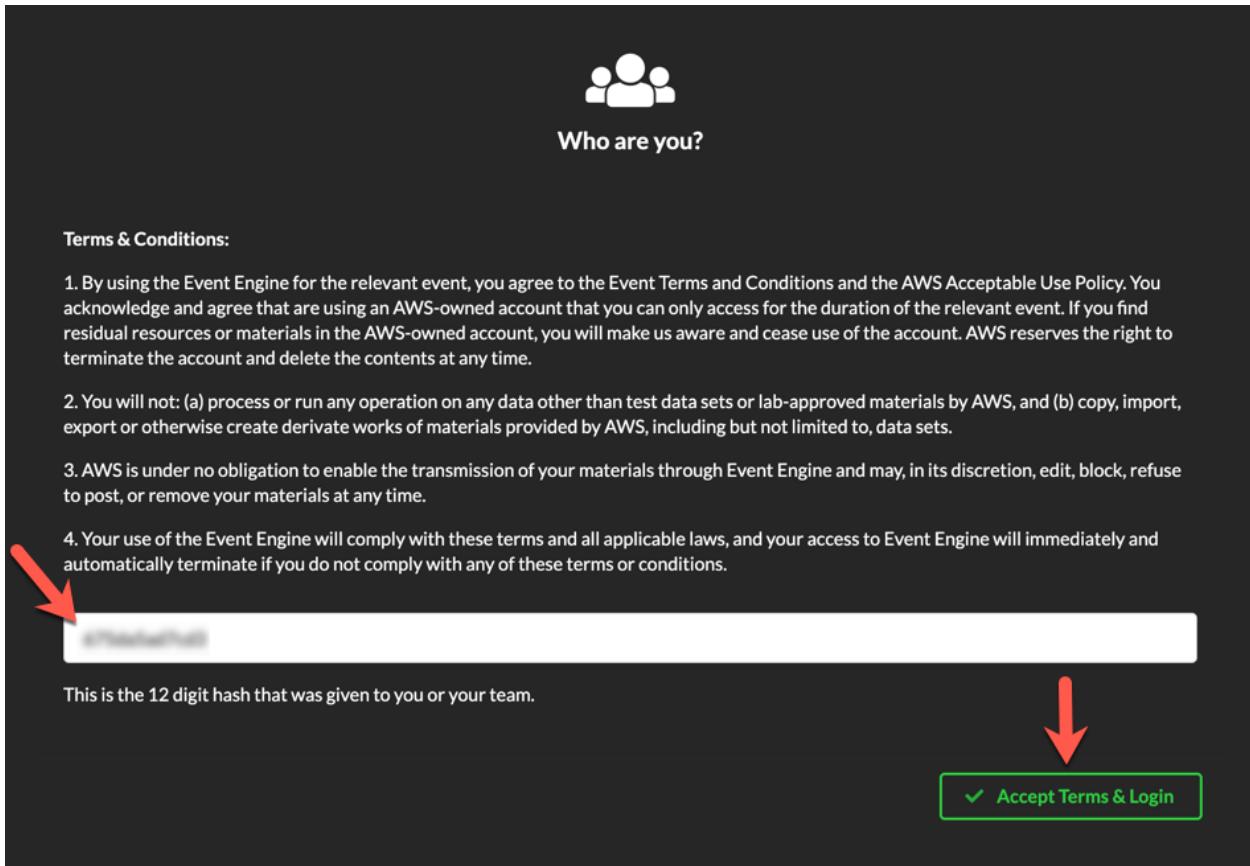
Get Started Using the Lab Environment

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:



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:

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

Lab 4. AWS Lake Formation

The screenshot shows the 'Environment Setup' module within a larger interface. At the top, there's a 'Modules' header with a gear icon. Below it, a 'Readme' link is visible. The main content area is titled 'Outputs:' and lists several AWS resources with their corresponding ARNs:

- S3 Bucket name: mod-3fcddd609114925-dmslabs3bucket-1ngczznd15u
- BusinessAnalystUser: mod-3fcddd609114925-BusinessAnalystUser-MB0XFZLQLOXX
- DMSLabRoleS3 ARN: arn:aws:iam::377243295828:role/mod-3fcddd609114925-DMSLabRoleS3-O2VT1RSN43SG
- Glue Lab Role: mod-3fcddd609114925-GlueLabRole-YLTJA13WW6WT
- S3BucketWorkgroupA: mod-3fcddd609114925-s3bucketworkgroupa-tbon3m1mkunh
- S3BucketWorkgroupB: mod-3fcddd609114925-s3bucketworkgroupb-18ygl8nfp8ead
- WorkgroupManagerUser: mod-3fcddd609114925-WorkgroupManagerUser-5IVE0UQNIBG4

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

The screenshot shows the 'Team Dashboard' interface. At the top, there's a 'Event' section with a cloud icon. Below it, two buttons are shown: 'AWS Console' and 'SSH Key'. An arrow points down to the 'AWS Console' button. The main content area displays event details:

Event: Data Engineering Immersion Day - Test
Team Name:
Event ID: d2302d4ae9ff4ea2857846b74f7de7e2
Team ID: 1c2f7ad7ec044b0b8276f917c5983133

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

The screenshot shows the 'AWS Console Login' dialog box. It includes a note about using the correct region, a 'Login Link' section with 'Open AWS Console' and 'Copy Login Link' buttons, and a 'Credentials / CLI Snippets' section for Mac/Linux and Windows. A red arrow points to the 'Open AWS Console' button. Below the dialog, a note says: 'Please note or refer back to these parameters for the Aurora MySQL labs, they are referenced in the instruction guide.'

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.

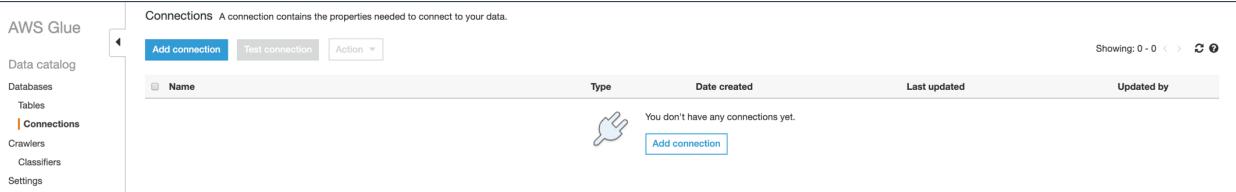
When you create a workflow, you must assign it an AWS Identity and Access Management (IAM) role that enables Lake Formation to set up the necessary resources on your behalf to ingest the data. In this lab, we've pre-created an IAM role for you, called **<random>-LakeFormationWorkflowRole-<random>**

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**.

Lab 4. AWS Lake Formation

Add connection

Set up your connection's properties.

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

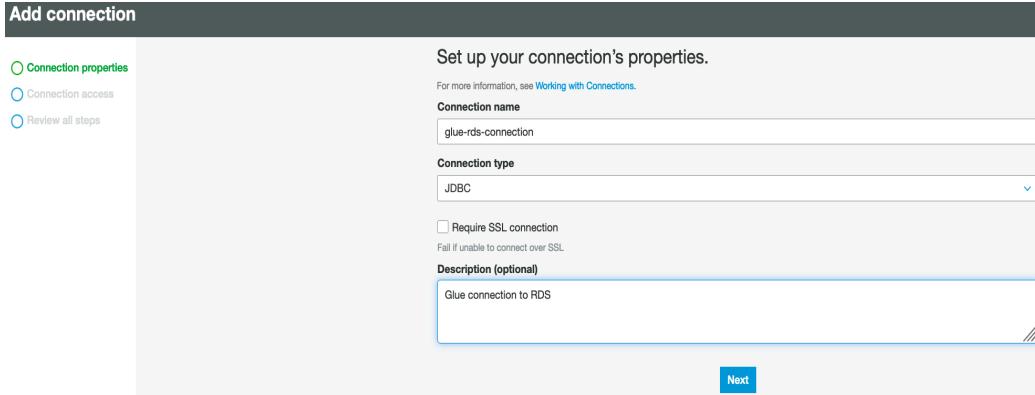
Connection name
glue-rds-connection

Connection type
JDBC

Require SSL connection
Fail if unable to connect over SSL

Description (optional)
Glue connection to RDS

Next

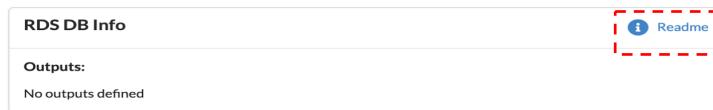


7. Input **JDBC URL** with the format of **jdbc:postgresql://[RDS_Server_Name] 5432/sportstickets**.

- a. Get the **RDS_Server_Name** from **RDS DB Info** dashboard.

RDS DB Info

Outputs:
No outputs defined



- 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.

Set up access to your data store.

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

JDBC URL 

jdbc:postgresql://dmslabinstance.crrprbscd9rq.us-east-1.rds.amazonaws.com:5432/sportstickets

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
master

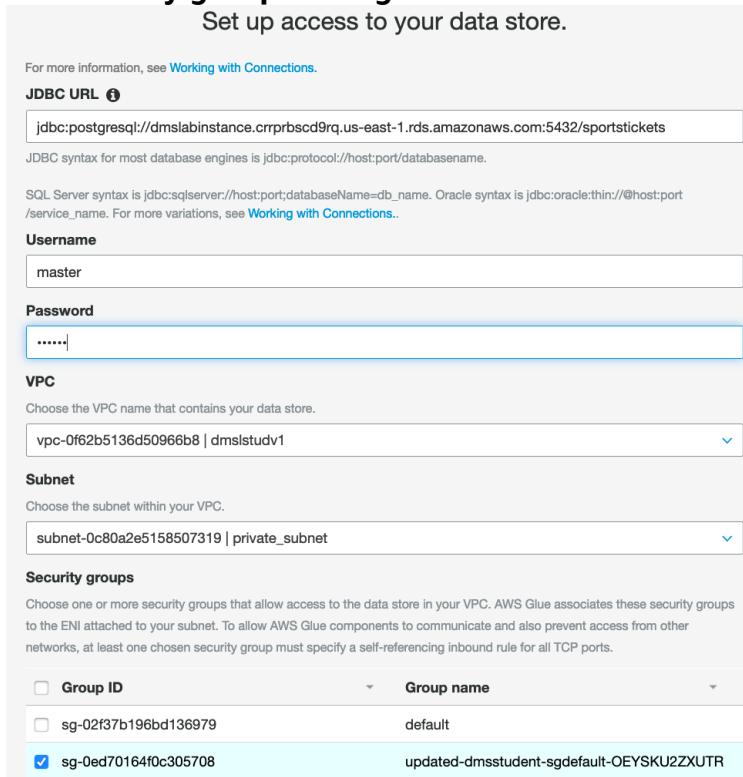
Password
.....

VPC
Choose the VPC name that contains your data store.
vpc-0f62b5136d50966b8 | dmslstudv1

Subnet
Choose the subnet within your VPC.
subnet-0c80a2e5158507319 | private_subnet

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

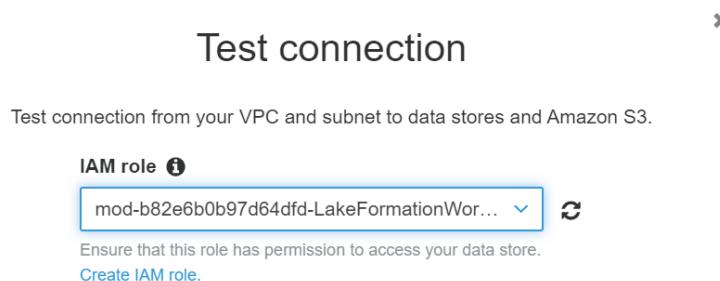


Lab 4. AWS Lake Formation

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

The screenshot shows the AWS Glue Connections page. On the left, there's a sidebar with options: AWS Glue, Data catalog, Databases, Tables, and Connections (which is highlighted). The main area has a title "Connections" with the subtitle "A connection contains the properties needed to connect to your data." Below this are three buttons: "Add connection", "Test connection" (which is highlighted in blue), and "Action". A table below shows a single entry: "Name" (checkbox) and "glue-rds-connection" (checkbox checked).

13. Choose the pre-created IAM role called <random>-
LakeFormationWorkflowRole-<random> and then click **Test Connection**.



Lake Formation – Add Administrator and start workflows using Blueprints.

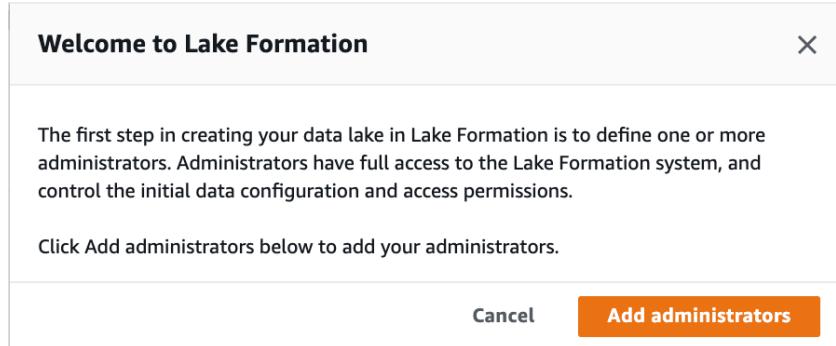
Navigate to the AWS Lake Formation service:

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

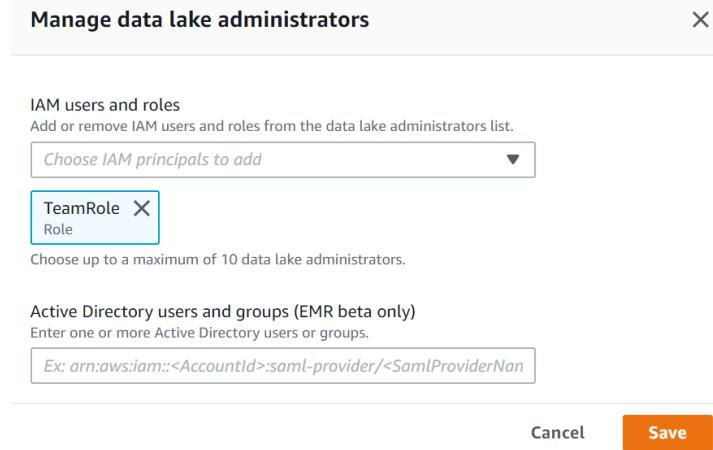
The screenshot shows the AWS Lake Formation service home page. On the left, there's a sidebar with links: History, AWS Glue, Console Home, AWS Lake Formation, AWS Backup, RDS, VPC, and AWS Outposts. The main area has a search bar with "Lake Formation" and a description: "AWS Lake Formation makes it easy to set up a secure data lake". Below this are several service tiles: EC2, Lightsail, ECR, ECS, EKS, Lambda, Batch, Elastic Beanstalk, Serverless Application Repository, AWS IQ, Support, Managed Services, Blockchain, Amazon Managed Blockchain, Satellite, Athena, EMR, CloudSearch, Elasticsearch Service, Kinesis, QuickSight, Data Pipeline, AWS Data Exchange, AWS Glue, AWS Lake Formation, WorkSpaces, AppStream 2.0, WorkDocs, WorkLink, Internet Of Things, IoT Core, Amazon FreeRTOS, and IoT 1-Click. There are also "Group" and "A-Z" buttons at the top right.

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**

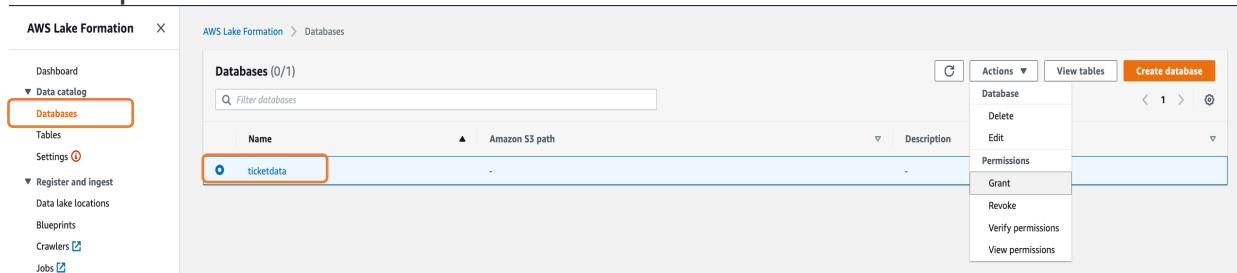
Lab 4. AWS Lake Formation



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



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.

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.
▼
mod-b82e6b0b97d64dfd-LakeFormationWorkflowRole-163KGWZCGXIZ X
Role
TeamRole X
Role

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

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.

6. Select Actions->Edit on the ticketdata database

AWS Lake Formation > Databases			
Databases (0/3)		Actions ▲	
		View tables	
<input type="button" value="C"/>		<input type="button" value="Delete"/>	<input type="button" value="Create database"/>
<input type="button" value="Filter databases"/>		<input type="button" value="Edit"/>	< 1 > <input type="button" value=""/>
Name	Amazon S3 path	Actions	Permissions
default	-	<input type="button" value="Delete"/>	<input type="button" value="Grant"/> <input type="button" value="Revoke"/>
ticketdata	s3://mod-3fcddd609114925-dmslabs3bucket-4f4n	<input type="button" value="Edit"/>	

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

Description - *optional*

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

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.

Filter workflows

Name Created on Last updated Last run status

No available workflows

Use blueprint

C Actions 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

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](#)

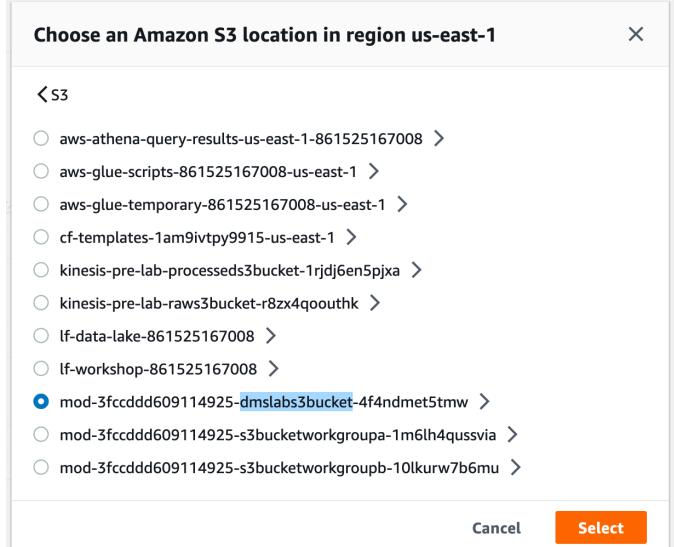
glue-rds-connection

Source data path
Enter the path from which to ingest data. For JDBC databases with schema support, enter database/schema/table. 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.



- 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**

Lab 4. AWS Lake Formation

Import target
Configure the target of the workflow.

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

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

Data format
Choose the output data format.

- 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

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

IAM role

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

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.

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.

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**.

Lab 4. AWS Lake Formation

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

The screenshot shows the 'Workflows (0/1)' section of the AWS Lake Formation console. It displays a table with columns: Name, Created on, Last updated, and Last run status. There is one entry: 'RDS-S3-Glue-Workflow' created on Mar 23, 2020, at 10:50 PM UTC, last updated on the same day at 10:50 PM UTC, and currently in the 'Discovering' state.

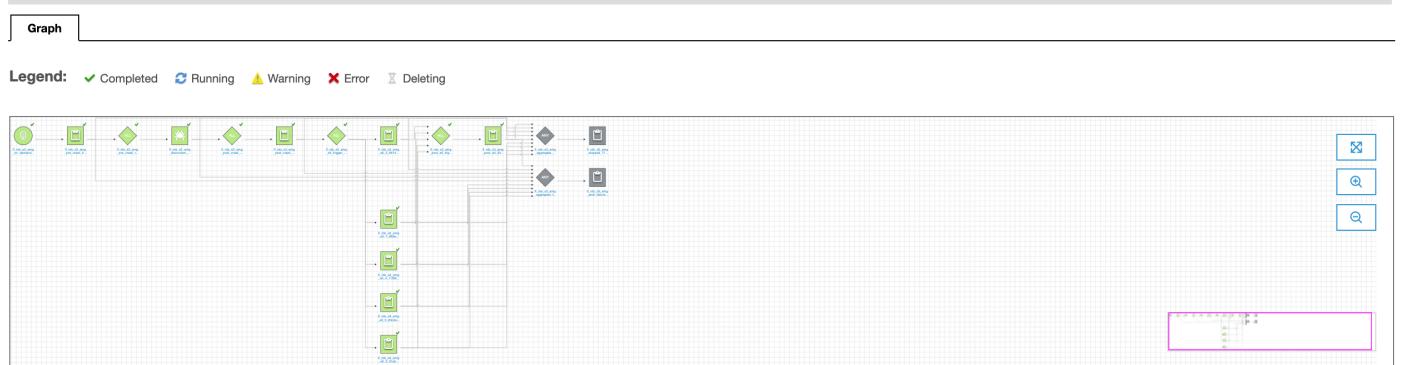
Explore the Underlying Components of a Blueprint

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**.

The screenshot shows the 'Workflow runs (1)' section of the AWS Lake Formation console. It displays a table with columns: Name and Started on. There is one entry: 'lf_rds_s3_amg' started on Dec 16, 2019, at 6:07 AM UTC. The 'Run ID' column shows 'wr_bcf4efeb081293bb613e63c2a586ee37f9ccbdd1ca97dd953d28514a0951908f'. The entire row for this run is highlighted with an orange border.

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 'ticketdata' database details page in the AWS Lake Formation console. At the top, there are navigation links: 'AWS Lake Formation > Databases > ticketdata'. Below the title 'ticketdata' are four buttons: 'Actions ▾', 'View tables', 'Edit', and 'Delete'. A 'Database details' section contains fields for 'Name' (set to 'ticketdata') and 'Amazon S3 path' (set to '-'). There is also a 'Description' field with a single dash. Underneath, there are sections for 'Default permissions for newly created tables' and a checkbox for 'Use only IAM access control for new tables in this database'.

- 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)' page in the AWS Lake Formation console. The table list includes columns for 'Name', 'Database', and 'Location'. One row is highlighted: 'lakeformation_sportstickets_dms_sample_player' under 'Database: ticketdata'. To the right of the table list is a context menu for the selected table, with 'View data' highlighted. Other options in the menu include 'Edit', 'Drop', 'Permissions', 'Grant', 'Revoke', 'Verify permissions', and 'View permissions'.

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. It features a large orange bar chart icon, the text 'Amazon Athena', and a brief description: '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.' Below this is a blue 'Get Started' button and a link to 'Getting started guide'.

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

Lab 4. AWS Lake Formation

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

sources Workgroup : primary

Query result location: s3://mod-08b80667356c4f8a-dmslabs3bucket-nh54wqq771lk/ ⓘ
Example: s3://query-results-bucket/folder/

Encrypt query results: ⓘ

Autocomplete: ⓘ

Cancel Save

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: s3://mod-08b80667356c4f8a-dmslabs3bucket-nh54wqq771lk/ ⓘ
Example: s3://query-results-bucket/folder/

Encrypt query results: ⓘ

Autocomplete: ⓘ

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

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. 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**.

Identity and Access Management (IAM)

Add user **Delete user**

Find users by username or

User name: **demo_user**

Groups

Users

Dashboard

Access management

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**

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		Q athena	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**

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	<input type="button"/> Send email

7. 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

Lab 4. AWS Lake Formation

The screenshot shows the 'Permissions' tab of an IAM user's configuration. It lists one attached policy: 'AmazonAthenaFullAccess', which is an AWS managed policy. There are buttons for 'Add permissions' and 'Add inline policy'.

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 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

The screenshot shows the AWS Lake Formation interface. In the left sidebar, under 'Permissions', the 'Data permissions' section is selected. On the main page, under 'Data permissions (80)', there is a search bar and a 'Grant' button highlighted with an orange border. The URL in the address bar is 'AWS Lake Formation > Permissions'.

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

The screenshot shows the 'Grant permissions' dialog box for the database 'gov_sportstickets_dms_sample_player'. It includes sections for IAM users and roles (with 'data_lake_user' selected), Active Directory users and groups, column selection ('Include columns' set to 'id, first_name'), table permissions (with 'Select' checked), grantable permissions (with 'Select' checked), and a 'Grant' button at the bottom.

[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

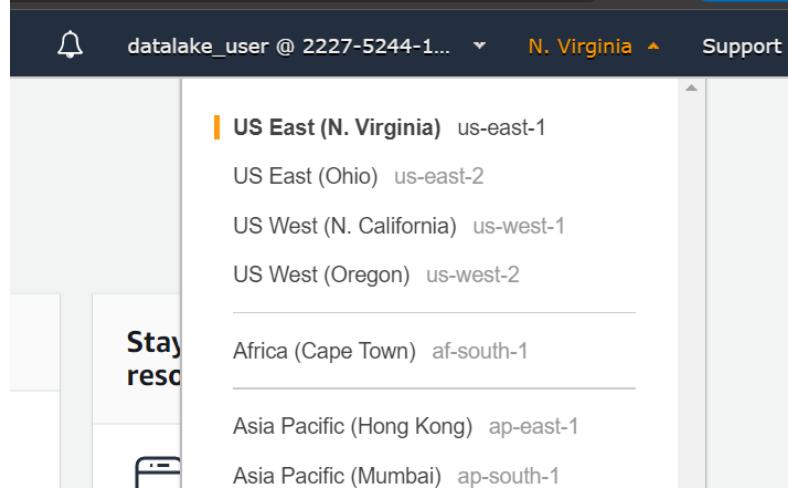
 

Sign in

[Sign in using root user email](#)

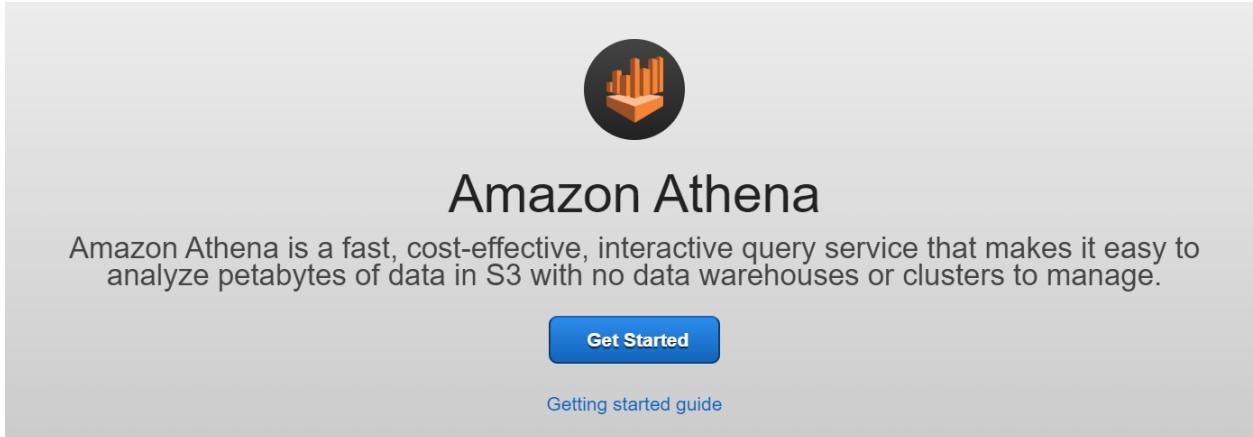
[Forgot password?](#)

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

Lab 4. AWS Lake Formation

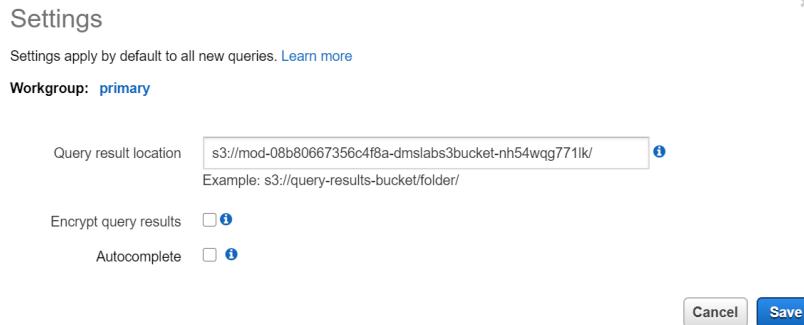


Then click **set up a query result location** in Amazon S3 at the top

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**



4. Next, ensure database **ticketdata** is selected.
5. 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;
```

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.