# Introduction

This workshop is designed to provide a **hands-on** learning experience for those looking to extract, transform and load data between data stores. The workshop uses AWS Glue components designed to **simplify** building ETL jobs at scale. Following diagram illustrates the high level architecture used in the workshop.

### [Workshop Data Set](https://catalog.us-east-1.prod.workshops.aws/workshops/0cba1e21-10d6-4e8e-b35f-a09338ee68d9/en-US/introduction" \l "workshop-data-set)

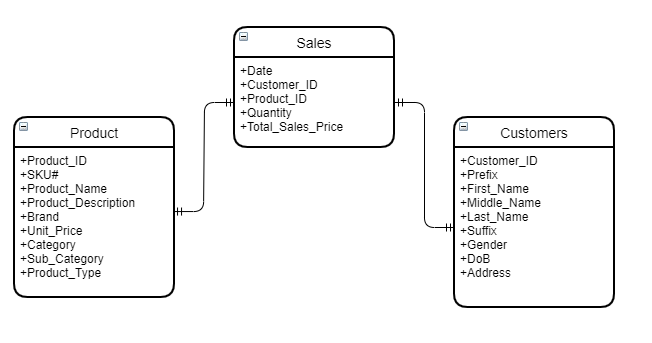
Company ABC is an online pet product retailer. They have recently launched a new product line for pets. They were expecting it to be a great success but sales from some regions were below their expectation. Marketing research suggests that limited awareness about the product in those regions is the main reason for inadequate sales. To address this problem, they want to start a targeted mail campaign in those regions that shows potential for increased sales. The team needs a fast turnaround, they don’t have much time to write code and manage infrastructure. At the same time, to avoid using any Personal Identifiable Information (PII) in their mail campaign, any customer PII data need to be redacted from their mailing list.

**Actions**

As a member of the ABC data engineering team, you are tasked to extract customer, product, and sales data. Following the extract you transform that data and create a dataset that has sales performance data for pet product categories by zip codes where specific product types need to be marketed. Sensitive data also needs to be redacted before allowing downstream analytics.

**Data Model**

The data model provided from the sales MySQL database is provided here:



### [Workshop Architecture](https://catalog.us-east-1.prod.workshops.aws/workshops/0cba1e21-10d6-4e8e-b35f-a09338ee68d9/en-US/introduction" \l "workshop-architecture)

Diagram

Description automatically generated ***Workshop Architecture Diagram***

This workshop takes about 1-2 hours to complete. After completing this workshop, you will learn how to use AWS Glue to accelerate time to authoring jobs to extract, transform and load data.

1. A MySql database instance is hosted on Amazon RDS pre-created for this lab. This database instance contains Product Sales data for your pet store company.
2. Glue Crawler job leverages a **pre-created connection** to RDS Mysql Database instance and crawls all the tables in the product sales DB.
3. Glue crawler will create product sales DB table entries in the AWS Glue Data Catalog which act as a central technical data catalog.
4. Using AWS Glue Studio visual interface, create a dataset for **aggregate** sales data reporting and **anonymize** the dataset for a mail marketing campaign.
5. Output dataset from the ETL jobs will be written into Amazon S3 backed data lake for further downstream analytics for mail marketing campagin management.

# How to Start?

If you are at an AWS hosted event, you will be provided with a temporary Event Engine AWS account for the workshop. Please follow the AWS Event instructions to access your Event Engine AWS account.

If you are running this workshop by yourself, your account must have the ability to create new IAM roles and scope other IAM permissions. Please follow Self Paced Labs instructions to prepare your AWS account.

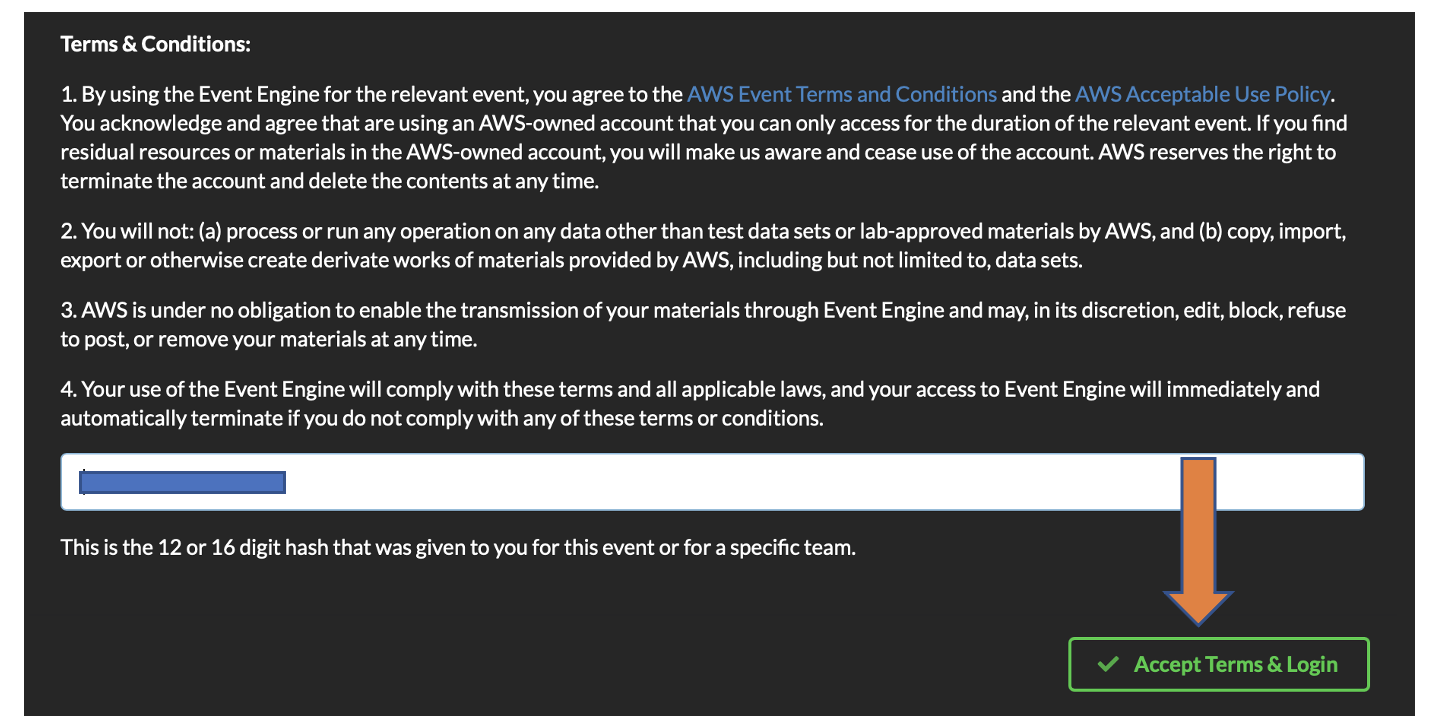
Step 1:

**AWS Event**

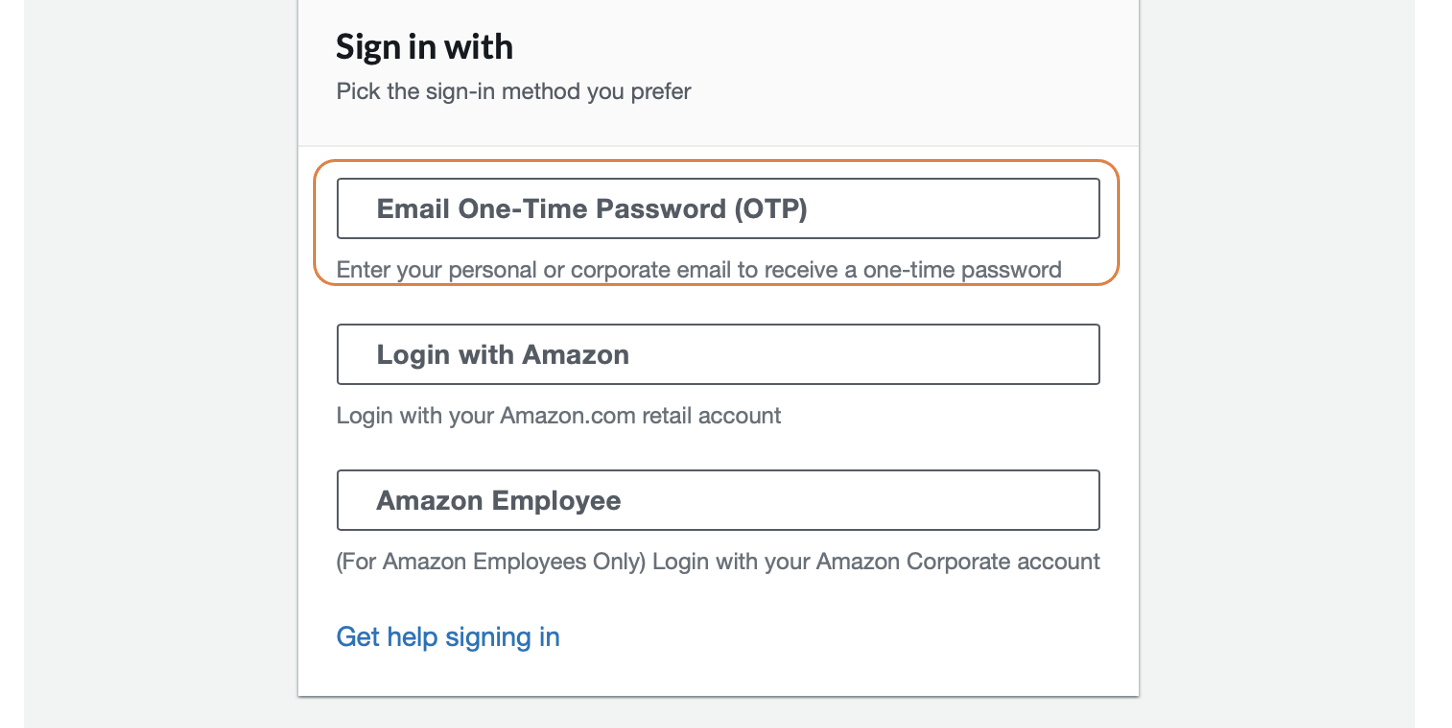
You will be provided with an AWS account to run this workshop. The temporary account is being created using Event Engine. You will be provided a participant hash key to login to your temporary account.

Follow these steps to start using your account:

1. Go to [AWS Event Engine Portal](https://dashboard.eventengine.run/)



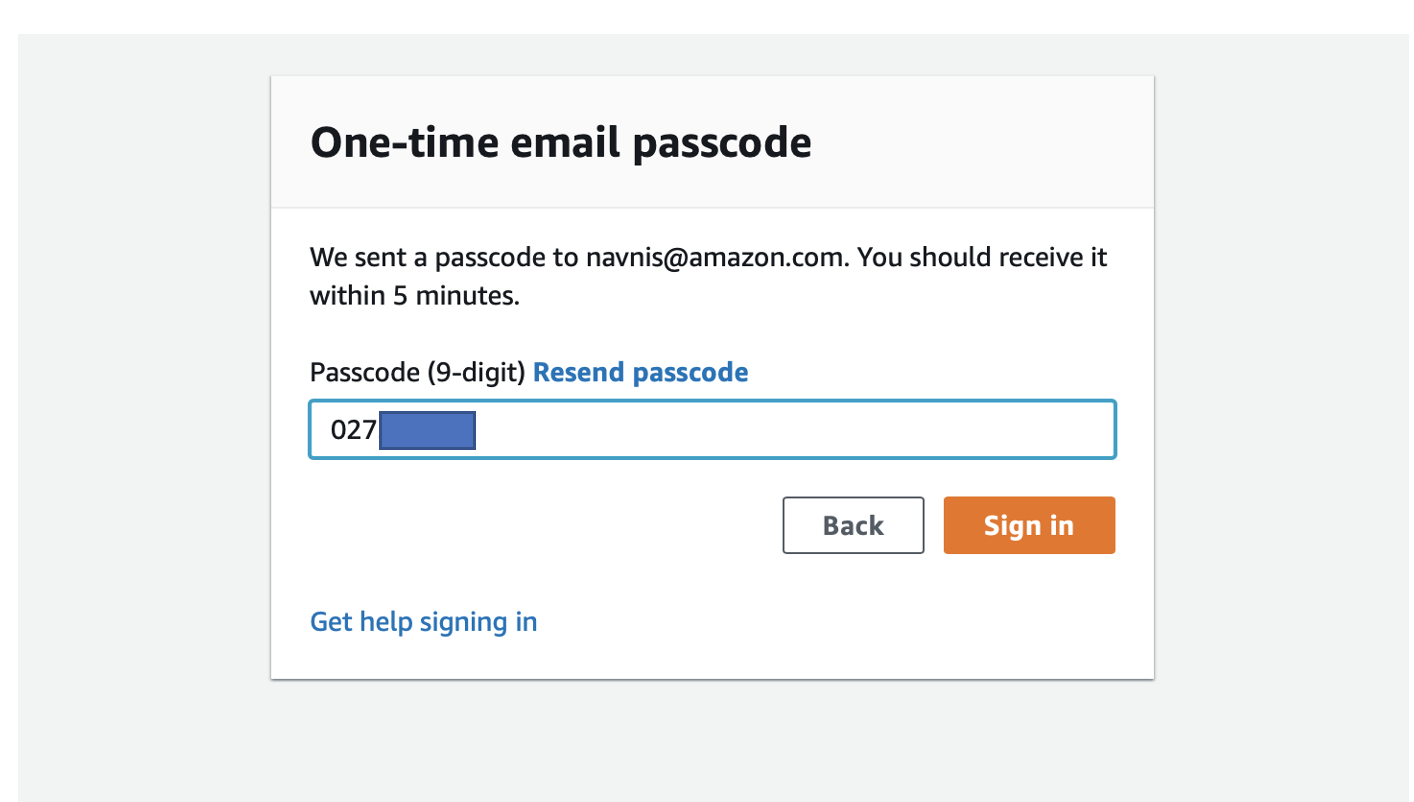
1. Enter the provided hash in the text box. The button on the bottom right corner changes to **Accept Terms & Login**. Click on that button to continue.
2. Click on **Email One-Time Password (OTP)**; You also have option to use your personal Amazon.com uid and password (Note - Not your AWS account credentials)



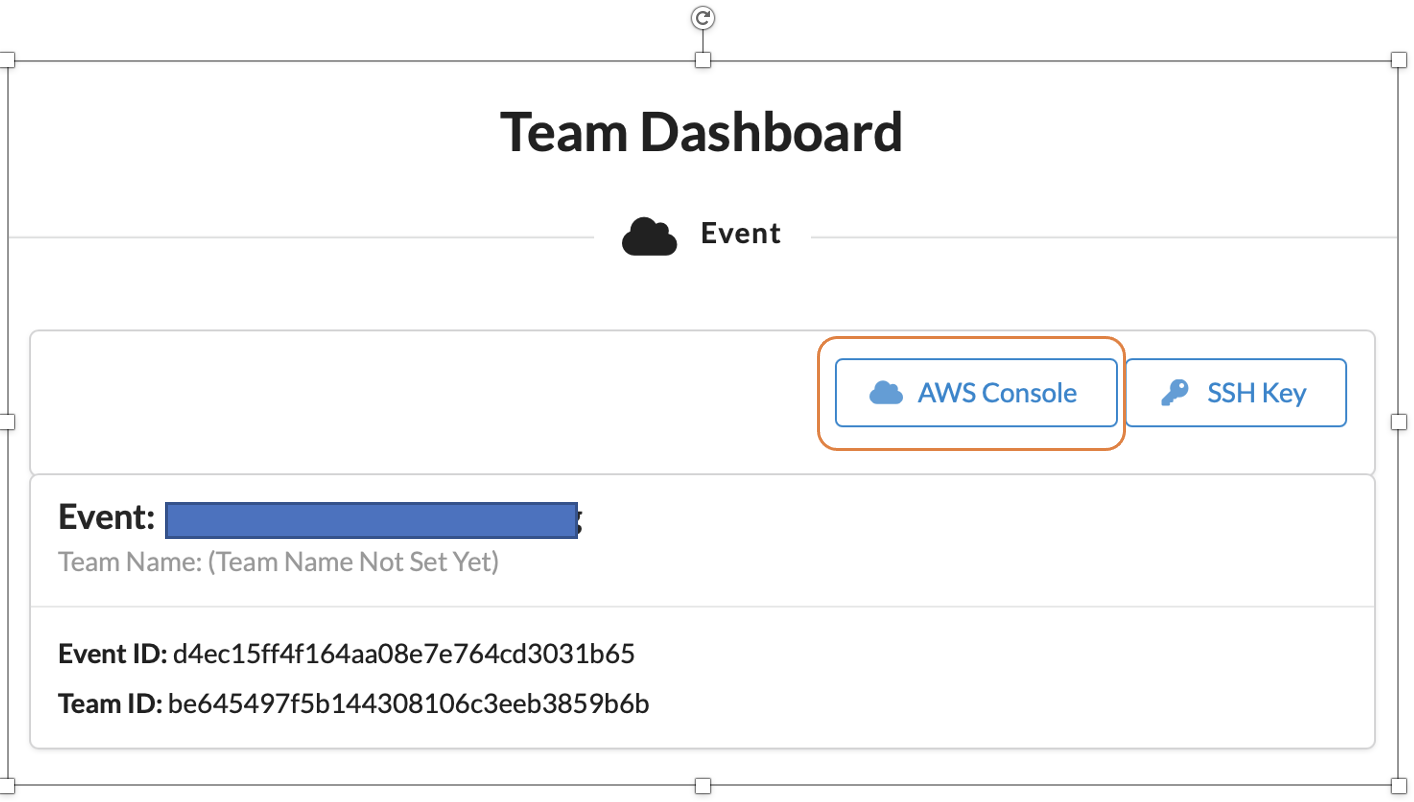
1. Provide your email address.



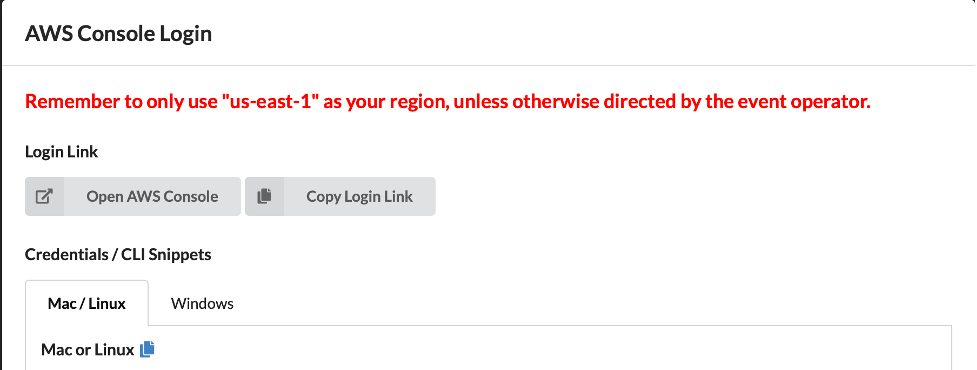
1. Provide the OTP you have received in your email.



1. Click **AWS Console** on the dashboard.



1. Take the defaults and click on **Open AWS Console**. This will open AWS Console in a new browser tab.



1. Once you have completed the steps above, let's head to [Lab 1](https://catalog.us-east-1.prod.workshops.aws/workshops/0cba1e21-10d6-4e8e-b35f-a09338ee68d9/en-US/lab01).

# Self Paced Labs

If you are participating in an event and accounts are provided by AWS, the resources for this workshop have been created for you already. You can skip this section

You need to be signed in an AWS Console with an administrator access to complete the workshop.

We will use CloudFormation to set up the AWS environment for the labs. AWS CloudFormation is a service that gives developers and businesses an easy way to create a collection of related AWS and third-party resources, and provision and manage them in an orderly and predictable fashion.

**Note : First you need to create the key pair in the region that you will be working on this workshop..**

CloudFormation template requires the private key pair to create Linux EC2 instance and setup the prerequisites for the workshop by securely SSH to the EC2 instance. If you already have an EC2 key pair in the region, proceed to the next session to launch the CloudFormation template.

### [Create your key pair using the Amazon EC2 console](https://catalog.us-east-1.prod.workshops.aws/event/dashboard/en-US/workshop/howtostart/self-paced-labs" \l "create-your-key-pair-using-the-amazon-ec2-console)

1. Navigate to the EC2 console by searching EC2 in the top search bar. Check the AWS region and ensure that you select the right region.
2. In the navigation pane, under **NETWORK & SECURITY**, choose Key Pairs.

Graphical user interface, text, application, email

Description automatically generated

1. On the right top corner click on **Create Key Pair**.
2. For **Name**, enter a descriptive name for the key pair. Amazon EC2 associates the public key with the name that you specify as the key name. A key name can include up to 255 ASCII characters. It can’t include leading or trailing spaces.
3. For **Private key file format**, choose .pem.
4. Click on **Create key pair**.
5. The private key file is automatically downloaded by your browser.

### [Launch the CloudFormation Template](https://catalog.us-east-1.prod.workshops.aws/event/dashboard/en-US/workshop/howtostart/self-paced-labs" \l "launch-the-cloudformation-template)

Before starting this workshop, you need to create the required AWS resources. To do this, we provide AWS CloudFormation template to create a stack that contains the resources.

1. Click the Launch Stack button below to create the resources required for this workshop.

[Launch Stack](https://console.aws.amazon.com/cloudformation/home?#/stacks/new?stackName=glue-workshop&templateURL=https://aws-data-analytics-workshops.s3.amazonaws.com/aws_glue/workshop2022/glue-workshop-cfl.yaml)

1. In the window that opens, leave all the options default and and click on **Next**.
2. On **Specify Stack details** choose the following
   * For Stack name enter **glue-workshop**
   * For EC2 Key Pair, select the Private Key (.pem file) you just created, from the dropdown menu

Leave all the other options as default and click on **Next**.

1. Leave all the options as default in the **Configure stack options** window and click on **Next**.
2. Under Capabilities (at the bottom of the page), choose:
   * **I acknowledge that AWS CloudFormation might create IAM resources with custom names**
3. Then click on **Create stack**.

The CloudFormation stack will roughly take 15-20 minutes to complete.

In the CloudFormation console wait for the status **CREATE\_COMPLETE** as shown below:

Graphical user interface

Description automatically generated

The CloudFormation stack will create following resources in your account:

* Setup the VPC network, deploying the following resources:
  + A VPC with a pair of public and one private subnets.
  + An internet gateway, with a default route on the public subnets.
  + A NAT gateway with default route in the private subnets.
  + Amazon S3 gateway VPC endpoints in the private subnets in two Availability Zones.
* Setup a Aurora MySQL database and associate the necessary Subnets
  + Create a database with name "gluestudio" containing three tables - Customer, Products and Sales
  + Create a Security Group with necessary rules to provide access to Aurora MySQL database
* An Amazon Simple Storage Service (Amazon S3) bucket in the same AWS Region and Globally unique name
* Setup the Cloud9 environment within the Public Subnet
* Create IAM role for Glue Service with appropriate rights to Glue and S3 bucket
* Setup an Glue Connection for accessing Aurora MySQL using JDBC connection

1. Once you have completed the steps above, let's head to [Lab 1](https://catalog.us-east-1.prod.workshops.aws/event/dashboard/en-US/workshop/lab01).

# Lab 1: Glue Crawler

In this section you will use a Glue connection to RDS Mysql Database both pre-created for this lab. You will also use **AWS Glue crawler** to extract product, sales and customer tables from RDS Mysql Database named gluestudio and populate these tables in AWS Glue data Catalog

#### [Creating Database for Glue Data Catalog](https://catalog.us-east-1.prod.workshops.aws/event/dashboard/en-US/workshop/lab01" \l "creating-database-for-glue-data-catalog)

1. Navigate to the **AWS Glue console** by searching **Glue** in the top search bar. Check the AWS region and ensure that you are using the right AWS region for this lab as provided by your instructor.

The AWS Glue Data Catalog is **an index to the location, schema, and runtime metrics of your data**. You use the information in the Data Catalog to create and monitor your ETL jobs. Information in the Data Catalog is stored as metadata tables, where each table specifies a single data store.

1. In the left navigation pane, under **Data Catalog**, click **Databases**.
2. Click **Add database** in the window on the right hand side pane. Graphical user interface, text, application, Teams

   Description automatically generated
3. On Create a database tab, provide a name for the new database as **gluestudio**, and click **Create database**. Graphical user interface, text, application, email

   Description automatically generated

#### [Create the Glue Crawlers to populate Glue Data Catalog](https://catalog.us-east-1.prod.workshops.aws/event/dashboard/en-US/workshop/lab01" \l "create-the-glue-crawlers-to-populate-glue-data-catalog)

1. In the left navigation pane, under **Data Catalog**, and click **Crawlers**.
2. Click **Create crawler** in the window on the right hand side pane.

Graphical user interface, application, Teams

Description automatically generated

1. On Set crawler properties tab, provide a name for the new Crawler as **gluestudio\_crawl**, click **Next**.

Graphical user interface, text, application, email

Description automatically generated

1. On Choose data sources and classifiers tab, click on **Add a data source**

Graphical user interface, text, application, email

Description automatically generated

1. On Add data source popup window, choose following
   * For **Data source** choose **JDBC** from the dropdown
   * For **Connection** choose **WorkshopGlueConnector** from the dropdown
   * Type **gluestudio/%** for \*Include path

Please note that “gluestudio” is the name of the sales database in RDS mysql

Then click on **Add a JDBC data source**

Graphical user interface, text, application, email

Description automatically generated

1. On Choose data sources and classifiers tab, click on **Next**
2. On Configure security settings tab, choose **Workshop-GlueServiceRole** from the dropdown and then click **Next**.

Graphical user interface, text, application, email, Teams

Description automatically generated

1. On Set output and scheduling tab, for **Target database** choose **gluestudio** from the dropdown and then click **Next**.

Graphical user interface, text, application, email

Description automatically generated

1. Go to the bottom of the page and click on **Create crawler** Graphical user interface, application

   Description automatically generated
2. New Crawler is added. Select the crawler you created and click on **Run**.

Graphical user interface, text, application, email

Description automatically generated

1. It will take a few minutes (<5 min) to complete. Wait for the “State” of the crawler to change from Running to Ready. Then look at the column named “Table changes” and it should read as “3 created

Graphical user interface, text, application

Description automatically generated

1. In the left navigation pane, under **Data Catalog**, click **Tables**. You should see the details for the three tables created.

Graphical user interface, application

Description automatically generated

Proceed to the next lab.

**Lab 2: Glue Studio - ETL**

Please complete [How to Start](https://catalog.us-east-1.prod.workshops.aws/event/dashboard/en-US/workshop/howtostart) and [Lab 1](https://catalog.us-east-1.prod.workshops.aws/event/dashboard/en-US/workshop/lab01) sections before starting this lab

In this lab, you will author two jobs to accomplish following objectives respectively.

1. Visually transform and create a dataset with sum of total sales aggregated by product type, category and zip code.
2. Redact PII data using out-of-box PII transform functions and create a dataset for marketing mail campaign in top performing zip codes.

You will accomplish these objectives by using A­WS Glue Studio visual interface without writing any code.

* [Joins and Aggregations](https://catalog.us-east-1.prod.workshops.aws/event/dashboard/en-US/workshop/lab02/aggregation)

# Joins and Aggregations

In this lab, we join three tables and do an aggregation to identify which product type and in which zip code generated the highest sales. We start by creating a new Glue Studio job using AWS Glue Studio.

To begin this lab, click on AWS Glue Studio link on the left menu pane on your browser under **Data Integration and ETL** menu option as seen in the screenshot below.

Graphical user interface, text, application

Description automatically generated

* Open the menu on the left and click **Jobs**.
* In the **Jobs** page, select **Visual with a blank canvas** and click **Create**.

Graphical user interface, application

Description automatically generated

1. Change the name of the job at the top left of the page from Untitled job to Job 1 - Aggregations.

Graphical user interface

Description automatically generated with low confidence

1.1 Click on the tab named **Job details** above the visual canvas and go to the field named **IAM Role**. Select Workshop-GlueServiceRole from the drop down.

Graphical user interface, text, application, email

Description automatically generated

1.2 Scroll down to the field **Requested number of workers** and lower it to 5. Click on the drop down menu for the field **Job bookmark** and select Disable. Finally, lower the value for the fieled **Number of retries** to 0.

Graphical user interface, text, application, email

Description automatically generated

1. Let's add three data sources to the Canvas using the Aurora MySQL tables that were crawled in Lab 1:

2.1 Click on the drop down tab **Source** right above the visual canvas and select **AWS Glue Data Catalog**. This should create a node on your canvas with the name AWS Glue Data Catalog.

Graphical user interface, text, application, table

Description automatically generated

2.2 On the right side of the page, select the tab **Data source properties - Data Catalog**. Now, select gluestudio from the drop down under **Database**. For **Table**, select gluestudio\_sales from the drop down.

Graphical user interface, text, application

Description automatically generated

2.3 Click on the tab **Node properties** and change **Name** to MySQL - Sales.

Graphical user interface, text, application, email

Description automatically generated

2.4 Repeat the same actions for Customers and Products tables that were created in lab 1. This should create two more nodes on your canvas as seen in screenshot below.

Graphical user interface, application

Description automatically generated with medium confidence

1. Let's join the Sales and Customer tables using nodes **MySQL - Sales** and **MySQL - Customers**. However before we join, let's rename columns that have same name in both the tables to avoid duplicates.

3.1 On your canvas, click on the node **MySQL - Customers**. Notice how the color of the node border changes to blue color when selected. Now click on the tab **Action** at the top of the canvas and select Apply Mapping from the drop down options.

On the right, under the **Transform** tab locate the source key **customer\_id**. Append customers\_ to the beginning of field **Target key**. Notice how you can also change the data types as well as drop unwanted column using this transform.

Graphical user interface

Description automatically generated

Click on the tab **Node Properties** and change the **Name** to Customers - Resolve Column Conflict.

Graphical user interface, text, application, email

Description automatically generated

3.2 We are now ready to join the tables. On your canvas, click on **MySQL - Sales** node. Now click on the **Action** drop down tab above the canvas and select Join.

Graphical user interface, text, application

Description automatically generated

3.3 On the right, click on the tab **Node properties**. Let's change **Name** to Join - Sales/Customers. Now, click on the drop down under **Node parents** to add another joining node and select the checkbox for Customers - Resolve Column Conflict under \*Transforms section.

Graphical user interface, text, application, email

Description automatically generated

3.4 Click on **Transform** tab to add the join conditions per the screenshot below.

Graphical user interface, text, application

Description automatically generated

1. Let's now join above to the products table.

4.1 On your canvas, click on **Action** tab and select Join from the drop down. On your right, navigate to the tab **Node properties** and change the **Name** to Join - Sales/Customers/Product. Click on the **Node Parents** drop down and select the checkbox for MySQL - Products.

Text, table

Description automatically generated

4.2 Click on the **Transform** tab and add the join condition as seen in the screenshot below.

Graphical user interface, text, application, Teams

Description automatically generated

4.3 At this point your canvas should look as shown in the screenshot below.

Diagram

Description automatically generated

1. We are only interested in actual sales, so Let's filter out records that show zero sales. On your canvas, click on **Action** drop down tab and select **Filter**.

On your right under the **Transform** tab, select the radio button Global AND. Change the filter condition as per the screenshot below.

Graphical user interface

Description automatically generated with medium confidence

5.1 Click on tab Node properties and change the Name to Filter out zero sales.

Graphical user interface, text, application, email

Description automatically generated

1. At this point, we are ready to do an aggregation transformation by grouping the data on zip, product\_type and category and doing a sum of the total\_sales.

6.1 From the **Action** tab above the canvas, select Aggregate from the drop down. Select the **Fields to group by** zip, product\_type and category.

Click on Aggregate another column button and then from the Field to aggregate drop down, select total\_sales and select sum from the Aggregate function drop down.

Graphical user interface, text, application, email

Description automatically generated

6.2 Click on the tab Output schema and notice how Glue defaults the aggregated column name to **sum(total\_sales)**

Graphical user interface

Description automatically generated with medium confidence

1. Let's rename the new column name sum(total\_sales) to sum\_total\_sales and datatype to double.

From the **Action** tab drop down above the canvas, select Apply mapping and change the values on the right as per the screenshot below.

Graphical user interface, application

Description automatically generated

1. Now we are ready to save the output to S3 and create a table in the Glue catalog to query it.

8.1 Click on the drop down tab **Target** above the canvas and select **Amazon S3**.

Graphical user interface, application, table

Description automatically generated

8.2 Now let's configure the target S3 properties by updating Format, Compression Type, S3 Target Location and table option as shown in the below screenshot.

For S3 Target Location field, click on the button Browse S3 and select the radio button for the S3 bucket that starts with **glue-workshop** and append /glue-output/ to the path.

Graphical user interface, text, application, email

Description automatically generated

8.3 Click on the tab **Script** above the canvas to notice how AWS Glue generated Spark(PySpark) code for the job you just built.

Text

Description automatically generated

1. Your Canvas should look like in the screenshot below.

Diagram

Description automatically generated

9.1 Now we are ready to save the job. Click **Save** button on the top.

1. Now the job is ready for execution, click the **Run** button on the top right corner of your screen. You can check the status of the job run by clicking on the **Runs** tab.
2. Once the job run is completed and the **Run status** indicates **Succeeded**, you can check the Glue catalog for the new table that is created automatically. You should see the table created in this lab by navigating to the AWS Glue landing page (Search Glue in the Console search bar) and selecting Tables on the left hand pane .



Click on the table name to view the **Table details**.

Graphical user interface, text, application, email

Description automatically generated

1. Optional. To view the data, go the **Action** button on the top right of the screen and select **View data** from the drop down. This will take you to Amazon Athena service which is a serverless interactive services to query data on S3.

Graphical user interface, application

Description automatically generated

1. Optional. In Athena, follow instructions on the top banner and click on the button **Edit settings**. Select S3 bucket starting with glue-workshop and append /athena-output/ to the bucket path and click **Save**.

13.1 select **gluestudio** database and navigate to the **Tables** section below and click on the three vertical dots on the right for sales-aggregation-report table and select **Preview Table**.

Graphical user interface, application

Description automatically generated

Athena will run a preview and display the results.

Table

Description automatically generated

## [Summary](https://catalog.us-east-1.prod.workshops.aws/event/dashboard/en-US/workshop/lab02/aggregation" \l "summary)

In this lab, you have learned how to **extract** from a MySql DB and **transform** the data visually performing multiple actions of changing data types, selecting specific columns, joining tables, filtering data, and performing aggregations.

You finally created the sales performance dataset for ABC's product lines by Region, Category and Product Type where mail marketing campaign analysis can be done. However, before this dataset is handed over to downstream analytics for mail campaign, sensitive data needs to be anonimized. Move on to the next lab to learn how to handle sensitive data redaction.

**Lab 3: (Optional) Personal Identifiable Information (PII)**

In this lab, we will be creating an anonymized mailing list for the top performing products in a geographical area. As the customer data has personal data, we will be using Glue's PII detection feature to find the columns that hold PII data and redact the data in those columns. This redacted table is then joined to the aggregated sales report table that was generated in the [previous lab](https://catalog.us-east-1.prod.workshops.aws/event/dashboard/en-US/workshop/aggregation) to create the mailing list table.

* [Create Custom Regex Pattern](https://catalog.us-east-1.prod.workshops.aws/event/dashboard/en-US/workshop/lab03/customregex)
* [Job to handle PII data](https://catalog.us-east-1.prod.workshops.aws/event/dashboard/en-US/workshop/lab03/piijob)

# Create Custom Regex Pattern

In this lab, we will be creating custom regex pattern to identify sensitive column which is useful when your column has data formats not supported by Glue's auto PII detection feature.

1. Date format isn't currently supported by Glue's built-in PII detection feature. So, we will use the date of birth column **dob** and redact it.

1.1. From the left hand pane, click on the link Sensitive data detection as shown in the screenshot below

Graphical user interface, application

Description automatically generated

1.2 On the right hand pane, click on the button Create  
Graphical user interface, application

Description automatically generated

1.3 Enter **Pattern name** as DateOfBirth. Use the following REGEX expression and paste it into Expression field:

(?:19\d{2}|20[01][0-9]|2020)[-/.](?:0[1-9]|1[012])[-/.](?:0[1-9]|[12][0-9]|3[01])

1.4 To increase accuracy, add any context words. In this case, add the word dob to the context words. Click **Create pattern** on the bottom right.

Graphical user interface, text, application, email

Description automatically generated

Now, lets proceed to create a new Glue Studio job

# Job to handle PII data

Now, we are ready to create a new Glue Studio job to redact sensitive data from your dataset using Glue built-in PII feature without writing code.

1. First, open the menu on the left and click on **Jobs**. Then, in the **Jobs** page, select **Visual with a blank canvas** and click on **Create**.

Graphical user interface, application

Description automatically generated

1.1 Rename the job Job 2 - PII

Graphical user interface, application, Word

Description automatically generated

1. Add a new sources from the AWS Glue Data Catalog.

Graphical user interface, application, table

Description automatically generated

1. Select the MySQL customer table:

Database: gluestudio

Table: gluestudio\_customers

Graphical user interface, text, application, email

Description automatically generated

3.1 Rename the name of the node to MySQL - Customers:

Graphical user interface, text, application, email

Description automatically generated

1. Click on the **Action** tab drop down above the canvas and select Apply Mapping transform. On the right pane, change the **Data type** for column **dob** to string data type. This is to prepare it so that the PII can detect sensitive date information with Regex expressions.

Graphical user interface, table

Description automatically generated

1. Click on the **Action** tab drop down above the canvas and select Detect Sensitive Data

Graphical user interface, text, application

Description automatically generated

5.1 Rename the node to Detect PII - Names

Graphical user interface, text, application, email

Description automatically generated

1. From the **Transform** tab, configure to detect sensitive data by selecting **Find columns that contain sensitive data**, setting the **Sample portion** to 55% and **Detection threshold** to 85%. For each use case, you will need to do trial-and-error to get an optimum value to detect required PII data. Here, sampling about 55% of the data with a detection threshold of about 85% is meant toward higher recall than precision.

Graphical user interface, text, application, email

Description automatically generated

6.1 Choose **Select specific patterns** and browse for **Person's name** for the pattern to look for.

Graphical user interface, text, application, email

Description automatically generated

6.2 For **Actions**, select radio button **Redact detected text** and type Current resident for the **Replacement text** field.

Graphical user interface, text, application, email

Description automatically generated

In the above step, you leveraged Glue's built-in auto-detection feature to identify PII column and then redacted it. Now let's use the custom regex pattern created in earlier section to identify sensitive column for the data formats not supported by Glue's auto PII detection feature.

1. Select the last node in the canvas. Click on **Action** drob down tab above the canvas and select Detect Sensitive Data

7.1 In the **Transform** tab, select Find sensitive data in each row, choose Select specific patterns

Graphical user interface, text, application, email

Description automatically generated

7.2 Then browse for selected patterns and choose the DateOfBirth custom pattern that was just created above.

Graphical user interface, application

Description automatically generated

7.3 Under actions, select Redact detected text and leave the Replacement text blank so that detected dates are simply erased.

Graphical user interface, text, application, email

Description automatically generated

7.4 Click on **Node properties** tab and change **name** to Detect PII - Dates.

Graphical user interface, text, application, email

Description automatically generated

7.5 It should now look like in the screenshot below. If you see any additional attributes or text, delete it manually to match the screen below.

Graphical user interface, text, application, email

Description automatically generated

1. Now, we are ready join this to the previous job output but before join can be done, lets add a source node for that data set,

Click the **Source** drop down above the canvas and select AWS Glue Data Catalog. Select the **Database** and **Table** as per the screenshot below.

Graphical user interface, text, application, email

Description automatically generated

8.1 Change the **Name** of this node to Aggregate - Sales using the tab **Node properties**

1. Now we are only going to select a certain slice of the data. To accomplish that, click the **Action** tab drop down and select **SQL Query**, so that we can customize more specifically our result set.

On the right, we will select from the sales-aggregation-report only two fields and filter data where the product\_type is "Exercise Pen" and sum\_total\_sales >= 2000. Copy the SQL query below and paste it in the **SQL query** box.

select zip as customer\_zip, sum\_total\_sales from myDataSource

where product\_type like 'Exercise Pen%' and sum\_total\_sales >= 2000

Graphical user interface, text, application, email

Description automatically generated

9.1 To ensure that there will not be a column fieldname conflict later, we will change the Output schema so that zip becomes customer\_zip. Select the tab Output schema and click on the button Edit. Change zip to customer\_zip and click the **Apply** button:

Graphical user interface

Description automatically generated

1. **Join** between the PII redacted data node and the SQL transformed data by using a **Join** transform. Click **Action** drop down tab and select Join.

10.1 On the right, click **Node properties** tab to select the other **Node parents** by selecting the checkbox **Detect PII - Dates**.

Graphical user interface, text, application, email

Description automatically generated

11.2 Now from **Transform** tab, click the button Add condition and select zip from the **Detect PII - Dates** node and customer\_zip from the **SQL** node as the **Join condition**

Graphical user interface, text, application

Description automatically generated

1. Finally, add the S3 target by clicking **Target** dropdown tab and selecting **Amazon S3**.

For S3 Target Location field, click on the button Browse S3 and select the radio button for the S3 bucket that starts with glue-workshop and append **/glue-output/mailing-list/**

Configure the **Data target properties - S3** as per the two screenshots below

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

1. Now, lets set the Job Parameters

12.1 Click on the tab named **Job details** above the visual canvas and go to the field named **IAM Role**. Select Workshop-GlueServiceRole from the drop down.

Graphical user interface, text, application, email

Description automatically generated

12.2 Scroll down to the field **Requested number of workers** and lower it to 5. Click on the drop down menu for the field **Job bookmark** and select Disable. Finally, lower the value for the fieled **Number of retries** to 0.

Graphical user interface, text, application, email

Description automatically generated

12.3 Click **Save** button on top right. DO NOT miss this step as your workspace will be lost if you moved to next step without saving the job.

Icon

Description automatically generated with low confidence

Your final Glue job diagram should look like this:

Diagram

Description automatically generated

1. Click **Run** button on the top right of your browser.
2. Once the job is completed and shows **succeeded** in the Run status, you can view the new table by going to the **Tables** section of the **Data Catalog**, go to the row that shows the table name mailing-list and click on the link that says **Table data**

14.1 (Optional) You will be taken to Athena to view the data. In Athena, click on the three dots on the mailing-list table and select **Preview Table**

Graphical user interface, text, application

Description automatically generated

14.2 (Optional) It will run the preview query and display the results.

Graphical user interface, text, application, table

Description automatically generated

### [Summary](https://catalog.us-east-1.prod.workshops.aws/event/dashboard/en-US/workshop/lab03/piijob" \l "summary)

In this lab, you have learned how to redact sensitive data from your dataset using Glue built-in PII feature without writing code.

### [Learn more on how to manage Security in AWS Glue](https://catalog.us-east-1.prod.workshops.aws/event/dashboard/en-US/workshop/lab03/piijob" \l "learn-more-on-how-to-manage-security-in-aws-glue)

Cloud security at AWS is the highest priority. To learn how to configure AWS Glue to meet your security and compliance objectives, please refer to the documentations that can be found in [Security in AWS Glue](https://docs.aws.amazon.com/glue/latest/dg/security.html).

## [CONGRATULATIONS!](https://catalog.us-east-1.prod.workshops.aws/event/dashboard/en-US/workshop/lab03/piijob" \l "congratulations!)

You have now successfully completed this workshop and learned how you can extract, transform and load jobs using AWS Glue Studio.