

Jaya Jagannatha Kalia Santa

Batch Time Analysis of Transactional Data

DESCRIPTION

Lenodo is a multinational e-commerce organization that sells products directly to consumers. The database administrator exports the data every night in a CSV file, but this export functionality is unused. Lenodo wants to use this data to uncover insights about the most-sold item and the countries where customers have bought this item.

You are a data analytics consultant, and you're asked to provide valuable insights and statistics across products, brands, categories, segments to the marketing, product, sales, and procurement teams and inform them about which product has the highest amount of sales and which product and its marketing needs the most improvement. These statistics will help to run effective digital marketing campaigns. The scope of this project is limited to data engineering and analysis.

Objective:

To use AWS Big Data stack for data engineering to analyze transactions, uncover patterns, and share actionable insights

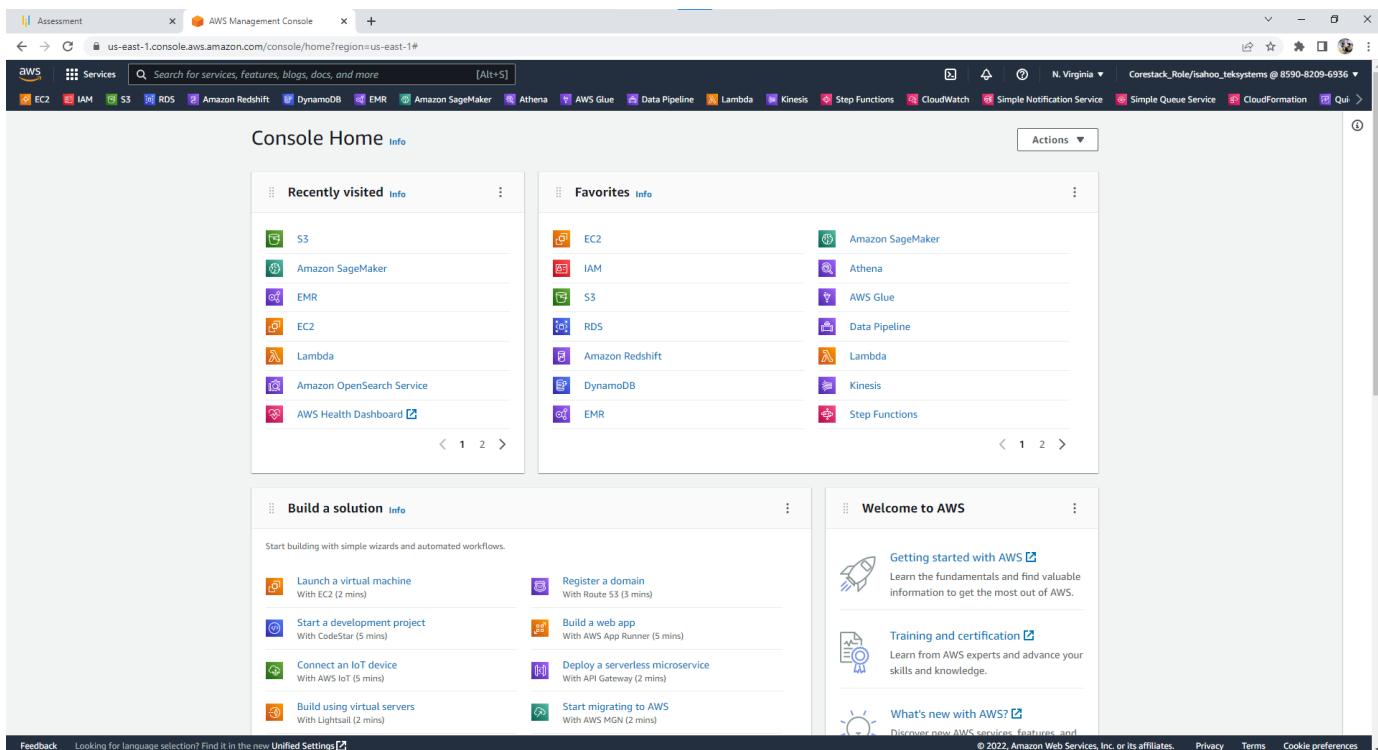
Steps to perform:

1. Create an S3 bucket with a unique name and upload the CSV file to the S3 bucket (ensure that the file is in UTF-8 format only)
2. Create a crawler to crawl the CSV data and generate a metadata catalog
3. Create a Glue job to transform the data into the Parquet format as CSV is not optimal for data warehouse queries
4. Add another crawler to crawl the Parquet data files to generate the metadata catalog of the Parquet file in order to query it with Athena
5. Query the data to identify the best-selling item and countries where customers have bought the most-sold item using Athena

Submitted By: Ipsita Sahoo

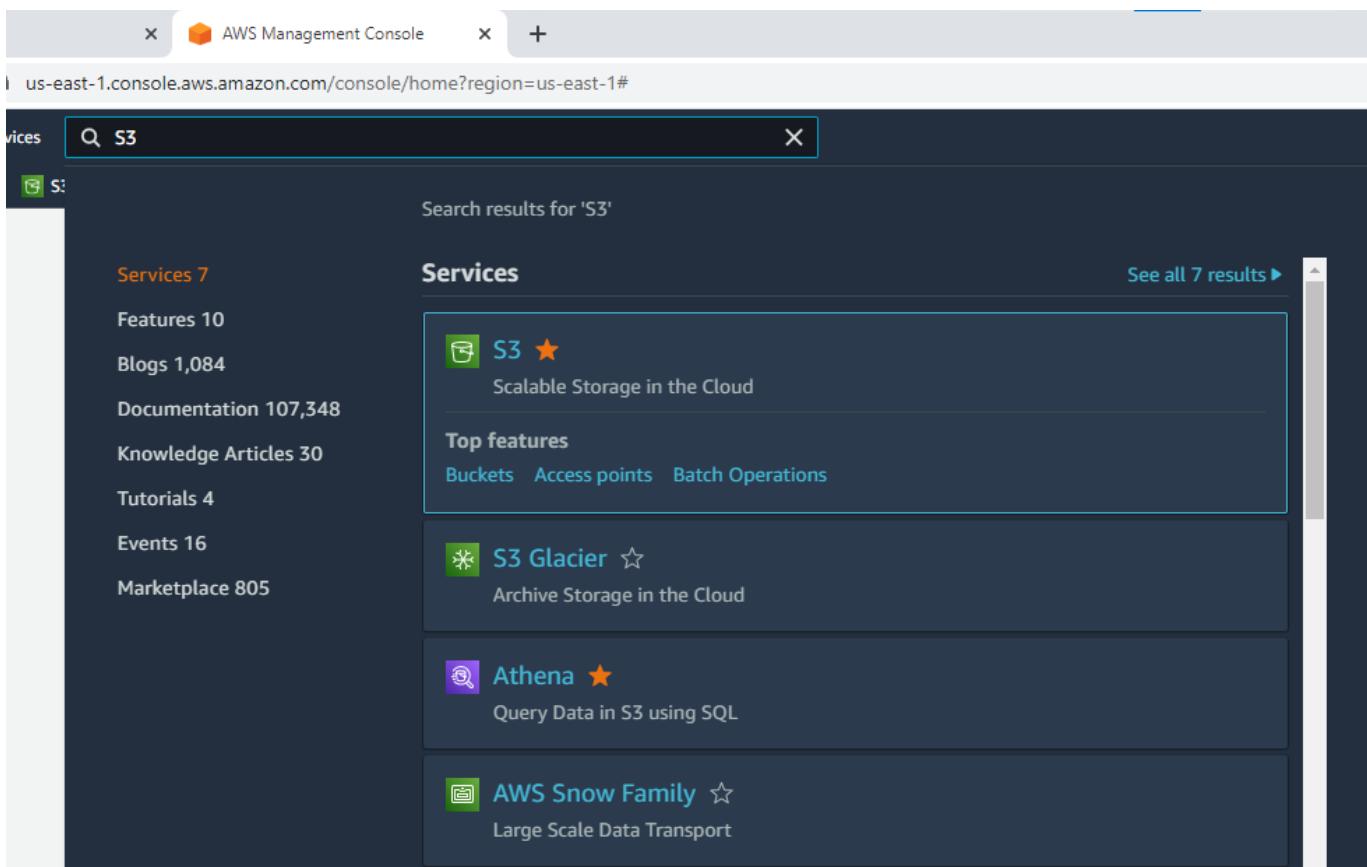
Step 1: Create an S3 bucket and upload the csv file

i. Open your AWS Management console home



The screenshot shows the AWS Management Console Home page. In the top navigation bar, there are tabs for Assessment, AWS Management Console, and a search bar. Below the navigation bar, there's a horizontal menu with icons for various services: EC2, IAM, RDS, Amazon Redshift, DynamoDB, EMR, Amazon SageMaker, Athena, AWS Glue, Data Pipeline, Lambda, Kinesis, Step Functions, CloudWatch, Simple Notification Service, Simple Queue Service, CloudFormation, and Quicksight. A dropdown menu for N. Virginia is open. On the right side, there's a user profile with the name 'Corestack_Role/sahoo_teksystems' and a session ID. The main content area has sections for 'Recently visited' (S3, Amazon SageMaker, EMR, EC2, Lambda, Amazon OpenSearch Service, AWS Health Dashboard) and 'Favorites' (EC2, IAM, S3, RDS, Amazon Redshift, DynamoDB, EMR, Amazon SageMaker, Athena, AWS Glue, Data Pipeline, Lambda, Kinesis, Step Functions). There are also sections for 'Build a solution' (Launch a virtual machine, Start a development project, Connect an IoT device, Build using virtual servers, Register a domain, Build a web app, Deploy a serverless microservice, Start migrating to AWS) and 'Welcome to AWS' (Getting started with AWS, Training and certification, What's new with AWS?). At the bottom, there are links for Feedback, Unified Settings, Privacy, Terms, and Cookie preferences.

ii. Search for S3 bucket



The screenshot shows the AWS Management Console Services search results for 'S3'. The search bar at the top contains 'S3'. The left sidebar has a 'Services' section with links to Features (10), Blogs (1,084), Documentation (107,348), Knowledge Articles (30), Tutorials (4), Events (16), and Marketplace (805). The main content area has a 'Services' heading and a 'See all 7 results' link. It lists three services: S3 (Scalable Storage in the Cloud, Top features: Buckets, Access points, Batch Operations), S3 Glacier (Archive Storage in the Cloud), and Athena (Query Data in S3 using SQL). Below these, there's a section for the AWS Snow Family (Large Scale Data Transport).

iii. Create a bucket

The screenshot shows the AWS S3 Management Console. At the top, there's a navigation bar with links like EC2, IAM, RDS, and various AWS services. Below the navigation bar, the main heading is "Amazon S3" with the subtext "Store and retrieve any amount of data from anywhere". A note below states: "Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance." To the right, there's a "Create a bucket" button. Below the main heading, there's a section titled "How it works" featuring a video thumbnail titled "Introduction to Amazon S3". To the right of the video, there's a "Pricing" section stating "With S3, there are no minimum fees. You only pay for what you use. Prices are based on the location of your S3 bucket." It includes a link to "Estimate your monthly bill using the AWS Simple Monthly Calculator" and a "View pricing details" link. Further down, there's a "Resources" section with links to "User guide", "API reference", "FAQs", and "Discussion forums". At the bottom of the page, there's a large orange "Create bucket" button.

iv. Give the name to the bucket and if u want u can change the Region

The screenshot shows the "Create bucket" wizard. The first step, "General configuration", is active. It has fields for "Bucket name" (containing "kalia-lenodo-bucket") and "AWS Region" (set to "Asia Pacific (Mumbai) ap-south-1"). There's also a section for "Copy settings from existing bucket - optional" with a "Choose bucket" button. The second step, "Object Ownership", is shown below. It has two options: "ACLs disabled (recommended)" (selected) and "ACLs enabled". The "ACLs disabled" option states: "All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies." The "ACLs enabled" option states: "Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs." At the bottom, it says "Object Ownership" and "Bucket owner enforced".

V. Keep all other options as default and create the bucket

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through new access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

Disable
 Enable

Tags (0) - optional

Track storage cost or other criteria by tagging your bucket. [Learn more](#)

No tags associated with this bucket.

[Add tag](#)

Default encryption

Automatically encrypt new objects stored in this bucket. [Learn more](#)

Server-side encryption

Disable
 Enable

► Advanced settings

ⓘ After creating the bucket you can upload files and folders to the bucket, and configure additional bucket settings.

[Cancel](#) [Create bucket](#)

Feedback Looking for language selection? Find it in the new [Unified Settings](#)

[Create bucket](#)

vi. Upload the file u want to work on in the bucket

The screenshot shows the AWS S3 Management Console. A green banner at the top indicates that the bucket 'kalia-lenodo-bucket' has been successfully created. Below the banner, the 'Account snapshot' section provides metrics: Total storage (27.0 MB), Object count (16), and Avg. object size (1.7 MB). A link to 'View Storage Lens dashboard' is available. The main area displays a table of buckets, with one entry for 'kalia-lenodo-bucket' which was created on June 9, 2022, at 15:53:27 (UTC+05:30) in the Asia Pacific (Mumbai) region.

The screenshot shows the 'kalia-lenodo-bucket' details page. The 'Objects' tab is selected, showing a table with zero entries. A large message 'No objects' is displayed, along with a note: 'You don't have any objects in this bucket.' An 'Upload' button is visible at the bottom of the table.

No objects

You don't have any objects in this bucket.

 Upload

Screenshot of the AWS S3 Management Console showing the upload process for a CSV file.

Upload Step:

- Upload:** Info
- Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more [Link]
- Drag and drop files and folders you want to upload here, or choose Add files, or Add folders.
- Files and folders (1 Total, 43.5 MB)**
 - All files and folders in this table will be uploaded.
- Destination:** Destination s3://kalia-lenodo-bucket
- Destination details:** Bucket settings that impact new objects stored in the specified destination.
- Permissions:** Grant public access and access to other AWS accounts.
- Properties:** Specify storage class, encryption settings, tags, and more.

Upload [Upload]

Feedback: Looking for language selection? Find it in the new Unified Settings [Link]

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Upload:

Upload succeeded
View details below.

Upload: status [Close]

The information below will no longer be available after you navigate away from this page.

Summary

Destination	Succeeded	Failed
s3://kalia-lenodo-bucket	✓ 1 file, 43.5 MB (100.00%)	✗ 0 files, 0 B (0%)

Files and folders (1 Total, 43.5 MB)

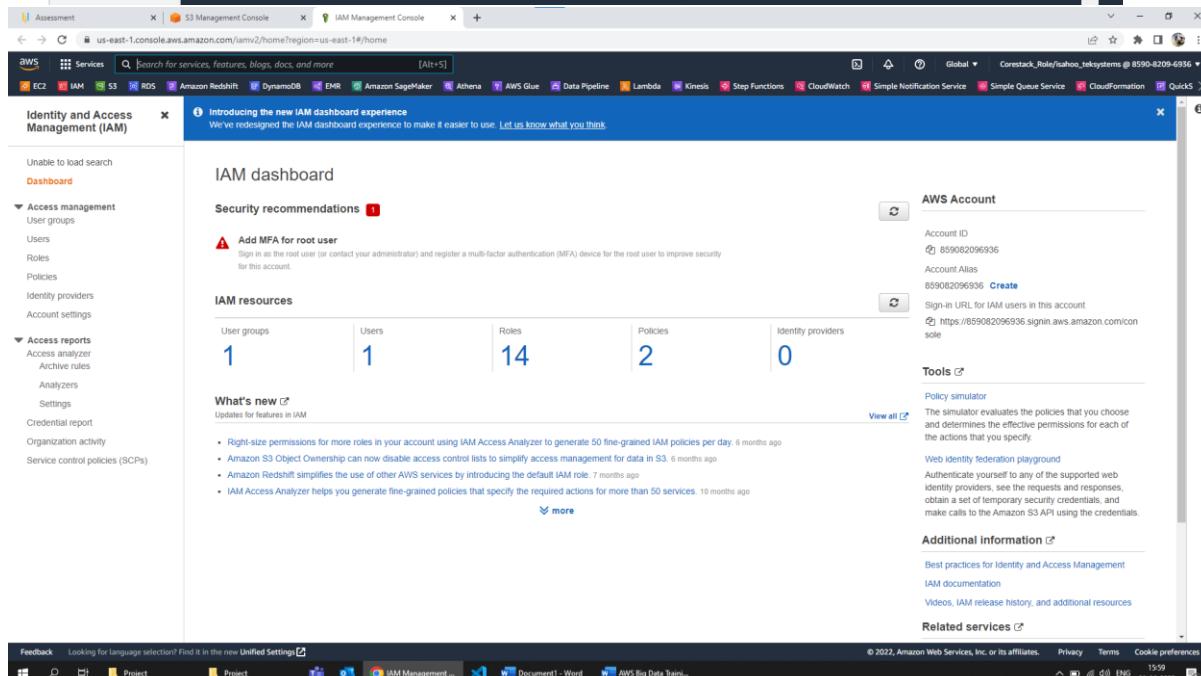
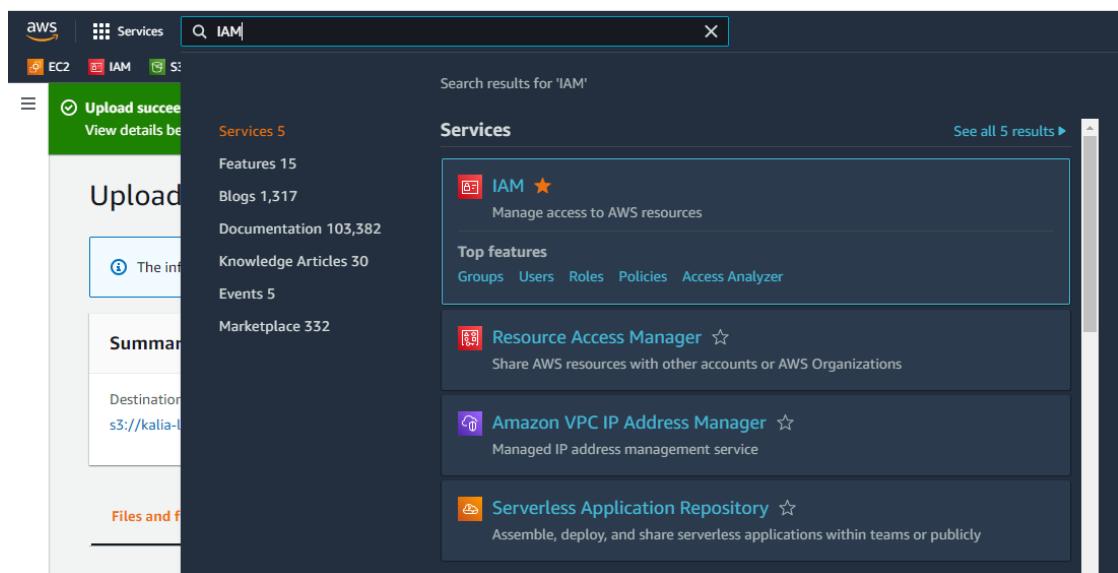
Name	Folder	Type	Size	Status	Error
data_utf8.csv	-	text/csv	43.5 MB	✓ Succeeded	-

Feedback: Looking for language selection? Find it in the new Unified Settings [Link]

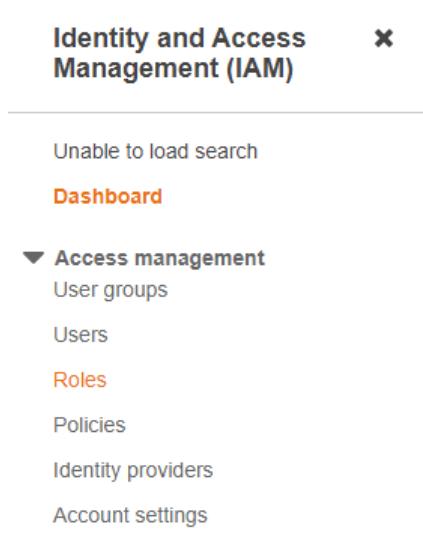
© 2022, Amazon Web Services, Inc. or its affiliates.

Step 2: Create an IAM role for Glue to access data from S3

i. Search for IAM on your Search bar



ii. Select Roles on the left panel



iii. Click create role

The screenshot shows the AWS IAM Management Console with the 'Roles' page open. The left sidebar shows navigation options like 'Identity and Access Management (IAM)', 'Access management', and 'Access reports'. The main area displays a table of existing roles, each with a checkbox, 'Role name', 'Trusted entities', and 'Last activity'. At the bottom left of the main area, there is a prominent blue button labeled 'Create role'.

iv. Select AWS service and search for Glue and select it along with the radio button to allow Glue to call AWS services on your behalf and click next.

The screenshot shows the 'Select trusted entity' step of the IAM Role creation wizard. On the left, a sidebar lists steps: Step 1 (Select trusted entity), Step 2 (Add permissions), and Step 3 (Name, review, and create). The main area has a title 'Select trusted entity' and a sub-section 'Trusted entity type'. It shows five options: 'AWS service' (selected), 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy'. Below this is a 'Use case' section with a note 'Allow an AWS service like EC2, Lambda, or others to perform actions in this account.' It lists 'Common use cases': 'EC2' and 'Lambda'. Under 'Use cases for other AWS services:', 'Glue' is selected. At the bottom right are 'Cancel' and 'Next' buttons.

- v. Search and select services as shown below you need to add:
- S3 full access (To store and access data from S3 bucket)
 - Administrator access (Not really required but so as we do not face any issues further)
 - Glue Service role (Glue role)
 - Glue console full access (Glue role)

After done click next

The screenshot shows the AWS IAM Management Console interface. The user is in the 'Create role' wizard, specifically on the 'Add permissions' step. They are filtering the list of available policies by the keyword 'glue'. Two specific policies are highlighted with blue selection boxes: 'AWSGlueServiceRole' and 'AWSGlueConsoleFullAccess'. Other policies listed include 'AWSGlueServiceNotebookRole', 'AWSGlueConsoleSageMakerNotebookFullAccess', 'AWSGlueDataBrewFullAccessPolicy', and several AWS managed policies like 'AWSGlueSchemaRegistryReadonlyAccess' and 'AWSGlueSessionUserRestrictedNotebookPolicy'.

Permissions policies (Selected 4/756)
Choose one or more policies to attach to your new role.

Filter policies by property or policy name and press enter			
"admin" X	Clear filters	35 matches	< 1 2 > ⌂
Policy name ↗	Type	Description	
<input type="checkbox"/> + AWSSSOAdministrator	AWS managed	Administrator access for SSO Directory	
<input type="checkbox"/> + CloudWatchAgentAdminPolicy	AWS managed	Full permissions required to use AmazonCloudWatchAgent.	
<input type="checkbox"/> + DatabaseAdministrator	AWS manag...	Grants full access permissions to AWS services and actions required to set up and configure AWS database services.	
<input type="checkbox"/> + AWSSSOMasterAccountAdministrator	AWS managed	Provides access within AWS SSO to manage AWS Organizations master and member accounts and cloud application	
<input type="checkbox"/> + AWSCloud9Administrator	AWS managed	Provides administrator access to AWS Cloud9.	
<input type="checkbox"/> + AWSSSOMemberAccountAdministrator	AWS managed	Provides access within AWS SSO to manage AWS Organizations member accounts and cloud application	
<input type="checkbox"/> + SystemAdministrator	AWS manag...	Grants full access permissions necessary for resources required for application and development operations.	
<input checked="" type="checkbox"/> + AdministratorAccess	AWS manag...	Provides full access to AWS services and resources.	

Permissions policies (Selected 4/756)
Choose one or more policies to attach to your new role.

Filter policies by property or policy name and press enter			
"s3" X	Clear filters	9 matches	< 1 > ⌂
Policy name ↗	Type	Description	
<input type="checkbox"/> + AmazonDMSRedshiftS3Role	AWS managed	Provides access to manage S3 settings for Redshift endpoints for DMS.	
<input checked="" type="checkbox"/> + AmazonS3FullAccess	AWS managed	Provides full access to all buckets via the AWS Management Console.	

us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-1#roles/create?selectedService=Glue&selectedUseCase=AWSGlueServiceRole&step=addPermission&trustedEntityType=AWS_SERVICE

Add permissions

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Permissions policies (Selected 4/756)
Choose one or more policies to attach to your new role.

Filter policies by property or policy name and press enter			
"s3" X	Clear filters	9 matches	< 1 > ⌂
Policy name ↗	Type	Description	
<input type="checkbox"/> + AmazonDMSRedshiftS3Role	AWS managed	Provides access to manage S3 settings for Redshift endpoints for DMS.	
<input checked="" type="checkbox"/> + AmazonS3FullAccess	AWS managed	Provides full access to all buckets via the AWS Management Console.	
<input type="checkbox"/> + QuickSightAccessForS3StorageManagementA...	AWS managed	Policy used by QuickSight team to access customer data produced by S3 Storage Management Analytics.	
<input type="checkbox"/> + AmazonS3ReadOnlyAccess	AWS managed	Provides read only access to all buckets via the AWS Management Console.	
<input type="checkbox"/> + AmazonS3OutpostsFullAccess	AWS managed	Provides full access to Amazon S3 on Outposts via the AWS Management Console.	
<input type="checkbox"/> + AWSBackupServiceRolePolicyForS3Backup	AWS managed	Policy containing permissions necessary for AWS Backup to backup data in any S3 bucket. This includes read access to all S3 obj...	
<input type="checkbox"/> + AWSBackupServiceRolePolicyForS3Restore	AWS managed	Policy containing permissions necessary for AWS Backup to restore a S3 backup to a bucket. This includes read/write permissions t...	
<input type="checkbox"/> + AmazonS3ObjectLambdaExecutionRolePolicy	AWS managed	Provides AWS Lambda functions permissions to interact with Amazon S3 Object Lambda. Also grants Lambda permissions to write t...	
<input type="checkbox"/> + AmazonS3OutpostsReadOnlyAccess	AWS managed	Provides read only access to Amazon S3 on Outposts via the AWS Management Console.	

▶ Set permissions boundary - optional
Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

vi. Give a name a to the role u created and click create role

The screenshot shows the AWS IAM Management Console with the 'Create role' wizard open. The browser tabs include 'Assessment', 'S3 Management Console', and 'IAM Management Console'. The navigation bar shows 'Services' selected, with links for EC2, IAM, S3, RDS, Amazon Redshift, DynamoDB, EMR, Amazon SageMaker, Athena, AWS Glue, and Data Pipelines.

Role details

Role name: glue-service,console_s3_admin

Description: Allows Glue to call AWS services on your behalf.

Step 1: Select trusted entities

```
1 [ { "Version": "2012-10-17", "Statement": [ 2 { "Effect": "Allow", "Principal": { "Service": "glue.amazonaws.com" }, "Action": "sts:AssumeRole" } ] }
```

Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
AdministratorAccess	AWS managed - job function	Permissions policy
AWSGlueConsoleFullAccess	AWS managed	Permissions policy
AWSGlueServiceRole	AWS managed	Permissions policy
AmazonS3FullAccess	AWS managed	Permissions policy

Tags

Add tags (Optional)

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Create role

Feedback: Looking for language selection? Find it in the new Unified Settings. Privacy Terms Cookie preferences

© 2022, Amazon Web Services, Inc. or its affiliates.

Create role

Screenshot of the AWS IAM Management Console showing the list of roles. A search bar at the top right contains the text "glue-service.console_s3_admin". The results table shows 15 entries, with one row highlighted.

Role name	Trusted entities	Last activity
AWSServiceRoleForAmazonElasticFileSystem	AWS Service: elasticfilesystem (Service-Linked Role)	7 days ago
AWSServiceRoleForAmazonSageMakerNotebooks	AWS Service: sagemaker (Service-Linked Role)	7 days ago
AWSServiceRoleForAmazonSSM	AWS Service: ssm (Service-Linked Role)	4 hours ago
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable	AWS Service: dynamodb.application-autoscaling (Service-Linked Role)	4 days ago
AWSServiceRoleForCloudTrail	AWS Service: cloudtrail (Service-Linked Role)	-
AWSServiceRoleForElasticLoadBalancing	AWS Service: elasticloadbalancing (Service-Linked Role)	-
AWSServiceRoleForEMRCleanup	AWS Service: elasticmapreduce (Service-Linked Role)	2 hours ago
AWSServiceRoleForOrganizations	AWS Service: organizations (Service-Linked Role)	-
AWSServiceRoleForRDS	AWS Service: rds (Service-Linked Role)	1 hour ago
AWSServiceRoleForRedshift	AWS Service: redshift (Service-Linked Role)	24 hours ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-
Corestack_Role	Account: 859082096936	21 minutes ago
CS_Admin	Account: 905236315842	2 hours ago

IAM > Roles > glue-service.console_s3_admin

glue-service.console_s3_admin

Allows Glue to call AWS services on your behalf.

Summary

Creation date June 09, 2022, 16:06 (UTC+05:30)	ARN arn:aws:iam::859082096936:role/glue-service.console_s3_admin
Last activity None	Maximum session duration 1 hour

Permissions **Trust relationships** **Tags** **Access Advisor** **Revoke sessions**

Permissions policies (4)
You can attach up to 10 managed policies.

Policy name	Type	Description
AmazonS3FullAccess	AWS managed	Provides full access to all buckets via the AWS Management Console.
AWSGlueServiceRole	AWS managed	Policy for AWS Glue service role which allows access to related services including EC2, S3, and Cloudwatch Logs
AdministratorAccess	AWS managed - job function	Provides full access to AWS services and resources.
AWSGlueConsoleFullAccess	AWS managed	Provides full access to AWS Glue via the AWS Management Console

Step 3: Create a catalogue of the csv data

i. Search for Glue in the search bar

The screenshot shows the AWS Management Console search results for the term 'glue'. The search bar at the top contains 'glue'. The results are categorized under 'Services' with a count of 4. The first result is 'AWS Glue' with a star icon, followed by 'AWS Glue DataBrew', 'AWS Lake Formation', and 'Lambda'. On the left sidebar, there are sections for Identity and Access Management, Access management, Roles, Access reports, and Analytics.

The screenshot shows the AWS Glue console. The left sidebar is titled 'AWS Glue' and includes sections for Data Catalog, Databases, Tables, Connections, Crawlers, Classifiers, Schema registries, Schemas, Settings, ETL, AWS Glue Studio, Jobs, Security, and Security configuration. The main content area is titled 'AWS Glue' and 'Serverless data integration'. It features a callout for 'Use AWS Glue to move and prepare data for analytics and machine learning' with a 'Get started' button. Below this, there's a 'Benefits and features' section with cards for AWS Glue Data Catalog, Crawlers for data discovery, Job management, Visual job authoring, Code-based job authoring, Job run and resource monitoring, Connectors, and AWS Glue DataBrew. To the right, there's a 'Getting started' sidebar with links to 'What is AWS Glue?', 'AWS Training: Getting Started with AWS Glue', 'Documentation: Getting started with AWS Glue', 'What is AWS Glue Studio', 'Getting started with AWS Glue Studio', and 'Getting started with AWS Glue DataBrew'. At the bottom, there's a 'Pricing (US)' section with rates for Jobs, Crawlers, Development Endpoints, and Catalog storage. The browser address bar shows the URL: us-east-1.console.aws.amazon.com/glue/home?region=us-east-1#v2/home.

ii. Select databases in the left panel

The screenshot shows the 'Data Catalog' section of the AWS Glue console. The left sidebar has collapsed the 'Data Catalog' section. Under 'Databases', there are links for Tables, Connections, Crawlers, Classifiers, Schema registries, Schemas, and Settings. The main content area is titled 'AWS Glue' and shows the 'Data Catalog' interface with various database entries listed.

iii. Click add database give a name and click create

Databases A database is a set of associated table definitions, organized into a logical group.

The screenshot shows a user interface for managing databases. At the top, there are three buttons: 'Add database' (highlighted in blue), 'View tables', and 'Action'. On the right, it says 'Showing: 0 - 0' with navigation icons. Below this is a table with two columns: 'Name' and 'Description'. A large blue cylinder icon is shown with the text 'You don't have any databases defined in your data catalog.' and a blue 'Add database' button.

Databases A database is a set of associated table definitions, organized into a logical group.

This screenshot is identical to the one above, showing an empty database catalog with no databases listed.

Add database

Database name

► Description and location (optional)

Databases A database is a set of associated table definitions, organized into a logical group.

The screenshot shows the 'Databases' page after creating a new database. A green banner at the top states 'Database "lenodo_data" has been successfully created.' Below this, the database catalog lists 'lenodo_data' with its name and a small blue cylinder icon. The top navigation and filtering options are visible.

iv. Click on crawler on the left panel

AWS Glue

Data catalog

Databases

Tables

Connections

Crawlers

Classifiers

Schema registries

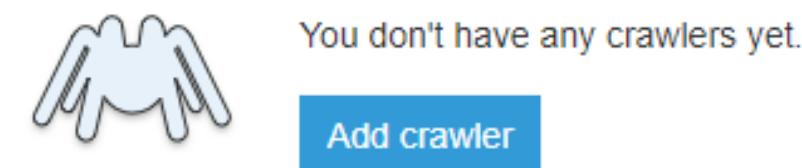
Schemas

Settings

v. Select Add crawler

Crawlers A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

The screenshot shows the AWS Glue Data Catalog interface. At the top, there are tabs for 'Add crawler' (which is highlighted in blue), 'Run crawler', and 'Action'. There is also a search bar labeled 'Filter by tags and attributes' and a user preferences icon. Below the tabs, a table header is shown with columns: Name, Schedule, Status, Logs, Last runtime, Median runtime, Tables updated, and Tables added. Under the 'Name' column, there is a small icon of two hands and a message: 'You don't have any crawlers yet.' followed by a blue 'Add crawler' button.



vi. Give the name and click next

The screenshot shows the 'Add crawler' wizard. The title is 'Add information about your crawler'. It has a section for 'Crawler name' where 'lenodo_crawler' is typed into a text input field. Below this, there is a note: '► Tags, description, security configuration, and classifiers (optional)'. At the bottom right is a blue 'Next' button.

- vii. As we must crawl all files in the bucket (since we only have 1 csv file and nothing else in the folder currently) so go with Data Source option and to Crawl all folders which are basically the default selections and click next

Specify crawler source type

Choose Existing catalog tables to specify catalog tables as the crawler source. The selected tables specify the data stores to crawl. This option doesn't support JDBC data stores.

Crawler source type

Data stores
 Existing catalog tables

Repeat crawls of S3 data stores

Crawl all folders
 Crawl all folders again with every subsequent crawl.
 Crawl new folders only
 Only Amazon S3 folders that were added since the last crawl will be crawled. If the schemas are compatible, new partitions will be added to existing tables.
 Crawl changed folders identified by Amazon S3 Event Notifications
 Rely on Amazon S3 events to control what folders to crawl.

[Back](#) [Next](#)

- viii. As our data source is in S3 so let it be S3, we must include the path to S3 so click on the folder to the right of Include Path choose the bucket (folder if it's in folder for my case it's in a bucket so I am selecting the bucket) and click select and click next to proceed.

Add a data store

Choose a data store

S3 [▼](#)

Connection

Select a connection [▼](#)

Optional: include a Network connection to use with this S3 target. Note that each crawler is limited to one Network connection so any future S3 targets will also use the same connection (or none, if left blank).

[Add connection](#)

Crawl data in

Specified path in my account
 Specified path in another account

Include path

s3://bucket/prefix/object [▼](#)

All folders and files contained in the include path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

Sample size (optional)

Enter a number between 1 and 249

This field sets the number of files in each leaf folder to be crawled. If not set, all the files are crawled.

► Exclude patterns (optional)

[Back](#) [Next](#)

Include path 

All folders and files contained in the include path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

Choose S3 path

- S3
- Okalia-lenodo-bucket
 - data_utf8.csv



Add a data store

Choose a data store **Connection** 

Optionally include a Network connection to use with this S3 target. Note that each crawler is limited to one Network connection so any future S3 targets will also use the same connection (or none, if left blank).

**Crawl data in**

- Specified path

Include path 

All folders and files contained in the include path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

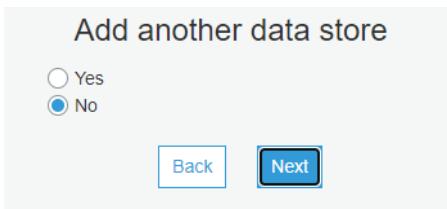
Sample size (optional)

This field sets the number of files in each leaf folder to be crawled. If not set, all the files are crawled.

- ▶ Exclude patterns (optional)



- ix. As I don't need any other additional data source I am selecting no and click next to proceed



- x. Select choose an existing IAM role as we have already created it, select the role and click next

Choose an IAM role

The IAM role allows the crawler to run and access your Amazon S3 data stores. [Learn more](#)

Update a policy in an IAM role
 Choose an existing IAM role
 Create an IAM role

IAM role [?](#)

AWSGlueServiceRole-

To create an IAM role, you must have **CreateRole**, **CreatePolicy**, and **AttachRolePolicy** permissions.

Create an IAM role named "**AWSGlueServiceRole**-rolename" and attach the AWS managed policy, **AWSGlueServiceRole**, plus an inline policy that allows read access to:

- s3://kalia-lenodo-bucket

You can also create an IAM role on the [IAM console](#).

Back Next

Choose an IAM role

The IAM role allows the crawler to run and access your Amazon S3 data stores. [Learn more](#)

Update a policy in an IAM role
 Choose an existing IAM role
 Create an IAM role

IAM role [?](#)

glue-service,console_s3_admin [▼](#) [↻](#)

This role must provide permissions similar to the AWS managed policy, **AWSGlueServiceRole**, plus access to your data stores.

- s3://kalia-lenodo-bucket

You can also create an IAM role on the [IAM console](#).

Back Next

- xi. We select Run on demand as frequency and click next

Create a schedule for this crawler

Frequency

Run on demand [▼](#)

Back Next

xii. Choose the database that we created earlier for crawler output and click next

Configure the crawler's output

Database ⓘ

Choose a database to contain tables

[Add database](#)

Prefix added to tables (optional) ⓘ

Type a prefix added to table names

► Grouping behavior for S3 data (optional)

► Configuration options (optional)

[Back](#) [Next](#)

Configure the crawler's output

Database ⓘ

Choose a database to contain tables

lenodo_data

[Add database](#)

Configure the crawler's output

Database ⓘ

lenodo_data

[Add database](#)

Prefix added to tables (optional) ⓘ

Type a prefix added to table names

► Grouping behavior for S3 data (optional)

► Configuration options (optional)

[Back](#) [Next](#)

xiii. Click finish in the review step and it will redirect u to crawler dashboard page

xiv. In the crawler section When the crawler gets created run it

The screenshot shows the AWS Glue service interface. On the left, there's a navigation sidebar with options like AWS Glue, Data catalog, Databases, Tables, Connections, Crawlers (which is highlighted in orange), Classifiers, Schema registries, Schemas, and Settings. The main content area has a header 'Crawlers' with a sub-instruction: 'A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.' Below this is a message: 'Crawler "lenodo_crawler" was created to run on demand. Run it now?' A button labeled 'Run crawler' is visible. The main table lists one crawler:

Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
lenodo_crawler		Ready		0 secs	0 secs	0	0

At the bottom, there are buttons for 'Add crawler', 'Run crawler', 'Action', and a search bar.

xv. When u see the table has got created u can move to tables section using the left panel and view your table

This screenshot shows the AWS Glue service interface again, focusing on the 'Tables' section. The left sidebar includes 'Tables' under the Crawlers category. The main table lists one table:

Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
lenodo_crawler		Starting		0 secs	0 secs	0	0

AWS Glue

Data catalog

Databases

Tables

Connections

Crawlers

Classifiers

Schema registries

Schemas

Settings

This screenshot shows the AWS Glue service interface again, focusing on the 'Tables' section. The left sidebar includes 'Tables' under the Crawlers category. The main table lists one table:

Name	Database	Location	Classification	Last updated	Deprecated
kalla_lenodo_bucket	lenodo_data	s3://kalla-lenodo-bucket/	csv	9 June 2022 4:18 AM UTC+5:30	

xvi. U can see the table preview using Athena open the search bar and search for Athena and select it choose Query Editor

The screenshot shows two views of the AWS console. The top view is the 'Services' search results page with 'athena' selected. It lists various services like Features, Blogs, Documentation, Knowledge Articles, Tutorials, Events, and Marketplace. The bottom view is the 'Amazon Athena' query editor interface, showing the 'How it works' diagram and various navigation links.

AWS Services Search Results:

- Search results for 'athena'
- Services 1
- Athena** ★
Query Data in S3 using SQL
- Features 3
- Blogs 774
- Documentation 38,270
- Knowledge Articles 30
- Tutorials 9
- Events 15
- Marketplace 85

Amazon Athena Query Editor:

- How it works diagram:
 - Point to your data source
 - Amazon Athena
 - Query results
- Navigation links:
 - Query editor
 - Workgroups
 - Data sources
 - Jobs
 - Workflows New
 - Powered by Step Functions
 - Enable compact mode
- Information boxes:
 - Begin querying your data
 - Pricing
 - Getting started
 - More resources

This is a detailed view of the Amazon Athena query editor interface. It includes the main navigation menu, a sidebar with 'Jobs' expanded, and a central area for running queries.

Amazon Athena:

- Query editor
- Workgroups
- Data sources
- Jobs**
- Workflows New
- Powered by Step Functions

Query Editor Area:

```
SELECT * FROM my_table;
```

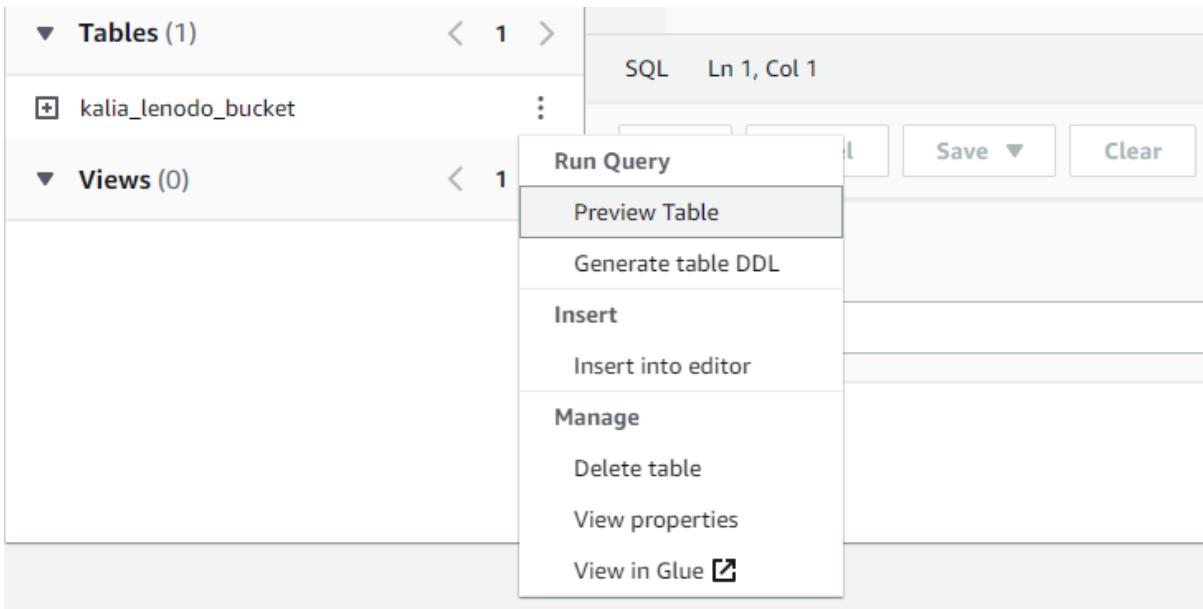
Bottom Navigation:

- Enable compact mode

- xvii. When the Athena editor will open up u can see that in the left side there is a Data Section we have a Data Source which is opened at AwsDataCatalog and the Database name is the same as the one we created before where our csv data catalogue table was created We can see that our table is present in the Tables list. Click on the 3 dots to the right of the table name and select preview table

The screenshot shows the AWS Athena Query Editor interface. On the left, under the 'Data' section, the 'Data source' is set to 'AwsDataCatalog' and the 'Database' is 'lenodo_data'. In the 'Tables and views' section, there is a table named 'kalia_lenodo_bucket'. The main area shows a single row with the value '1'. The SQL pane shows the query 'SELECT * FROM kalia_lenodo_bucket'. The Results pane indicates 'No results'.

The screenshot shows the AWS Athena Data section. The 'Data source' is set to 'AwsDataCatalog' and the 'Database' is 'lenodo_data'. Under 'Tables and views', there is one table named 'kalia_lenodo_bucket'. The interface includes a 'Create' button and a search bar for filtering tables and views.



- xviii. If u haven't used Athena before the query will load up on the screen but Run button will not be activated. To fix the issue create an Empty S3 bucket for Athena then return to Athena tab and click on settings in the same query editor and select Manage

- xix. In the query result and location click browse S3 and select the S3 bucket u created for Athena and click save return to the editor tab and now Run button must be activated

Manage settings

Query result location and encryption

Location of query result
Enter an S3 prefix in the current region where the query result will be saved as an object.

Expected bucket owner
Specify the AWS account ID that you expect to be the owner of your query results output location bucket.

Encrypt query results

Enable

Assign bucket owner full control over query results
Enabling this option grants the owner of the S3 query results bucket full control over the query results. This means that if your query result location is owned by another account, you grant full control over your query results to the other account.

Save

Settings successfully updated.

Amazon Athena > Query editor

Editor | Recent queries | Saved queries | **Settings**

Query result and encryption settings

Query result location and encryption

Query result location:

Encrypt query results:

Expected bucket owner:

Assign bucket owner full control over query results:

xx. Now u can select the query you want to run and select Run button to run the query

AWS Services Search for services, features, blogs, docs, and more [Alt+S] N. Virginia Corestack_Role@kalia_amazon-athena @ 8590-8209-6936 ▾

EC2 IAM S3 RDS Amazon Redshift DynamoDB DMR Amazon SageMaker Athena AWS Glue Data Pipeline Lambda Kinesis Step Functions CloudWatch Simple Notification Service Simple Queue Service CloudFormation Quicks ▾

Amazon Athena > Query editor

Data source: AwsDataCatalog Database: lenodo_data

Tables and views: kalia_lenodo_bucket

Completed

Results (10)

#	invoiceno	stockcode	description	quantity	invoicedate	unitprice	customerid	country
1	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850	United Kingdom
2	536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850	United Kingdom
3	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850	United Kingdom
4	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850	United Kingdom
5	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850	United Kingdom
6	536365	22752	SET 7 BABUSHKA NESTING BOXES	2	12/1/2010 8:26	7.65	17850	United Kingdom
7	536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	12/1/2010 8:26	4.25	17850	United Kingdom
8	536366	22633	HAND WARMER UNION JACK	6	12/1/2010 8:28	1.85	17850	United Kingdom
9	536366	22632	HAND WARMER RED POLKA DOT	6	12/1/2010 8:28	1.85	17850	United Kingdom
10	536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	12/1/2010 8:34	1.69	13047	United Kingdom

Step 4: Create a Glue job to transform the data into the Parquet format as CSV is not optimal for data warehouse queries

- i. Come to the AWS Glue and under the left panel u can see an Option Jobs (legacy) under ETL(Extract Transform Load) subheading select it and click add jobs

AWS Glue

Data catalog

Databases

Tables

Connections

Crawlers

Classifiers

Schema registries

Schemas

Settings

ETL

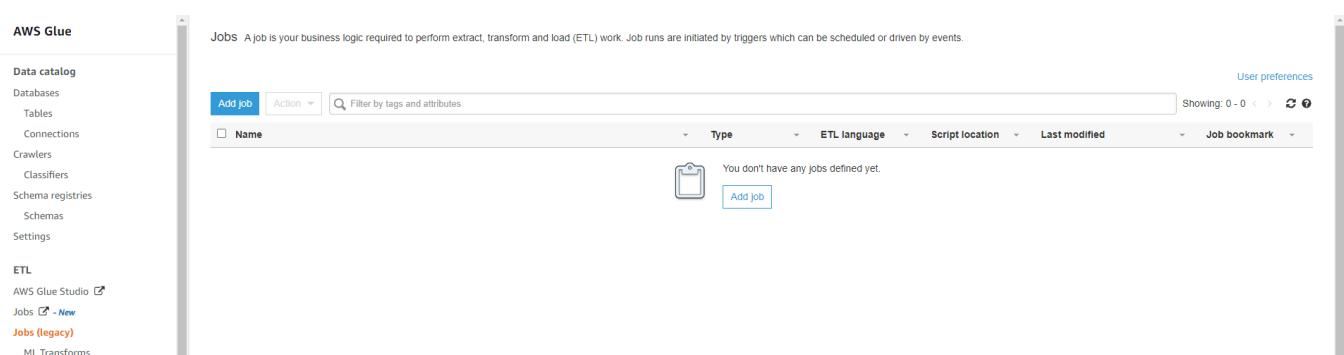
AWS Glue Studio 

Jobs  - New

Jobs (legacy)

ML Transforms

Blueprints



The screenshot shows the AWS Glue console interface. On the left, there's a navigation sidebar with sections for Data catalog, ETL, and AWS Glue Studio. Under ETL, the 'Jobs (legacy)' option is selected. The main content area has a heading 'Jobs' with a sub-instruction: 'A job is your business logic required to perform extract, transform and load (ETL) work. Job runs are initiated by triggers which can be scheduled or driven by events.' Below this, there's a search bar with 'Add job' and 'Action' buttons, and a filter bar with 'Filter by tags and attributes'. A table header includes columns for 'Name', 'Type', 'ETL language', 'Script location', 'Last modified', and 'Job bookmark'. A message at the bottom states 'You don't have any jobs defined yet.' with a blue 'Add job' button.



You don't have any jobs defined yet.

[Add job](#)

- ii. Give name to the job, select the IAM role we created before, select the execution environment u want here we are going to go with type Spark and glue version Spark 3.1, Python 3 (Glue Version 3.0)

New in AWS Glue X
Author jobs visually in [AWS Glue Studio](#).
With AWS Glue version 2.0, jobs start 10x faster and get 1-minute minimum billing. Test your existing jobs on the new version and make the switch. [Learn more](#)

Configure the job properties

Name
format_csv_parquet_job

IAM role i
glue-service,console_s3_admin ↻
Ensure that this role has permission to your Amazon S3 sources, targets, temporary directory, scripts, and any libraries used by the job. [Create IAM role](#).

Type
Spark

Glue version
Spark 3.1, Python 3 (Glue Version 3.0)

This job runs
 A proposed script generated by AWS Glue i
 An existing script that you provide
 A new script to be authored by you

Script file name
format_csv_parquet_job

S3 path where the script is stored
s3://aws-glue-scripts-859082096936-us-east-1/admin 📁

- iii. Under monitoring subsection select job metrics and continuous logging and select standard filter and click next

Temporary directory ⓘ

s3://aws-glue-temporary-859082096936-us-east-1/admin

▶ Advanced properties

▼ Monitoring options

Job metrics ⓘ

Continuous logging

Log filtering ⓘ

Standard filter No filter

Spark UI ⓘ

▶ Tags (optional)

▶ Security configuration, script libraries, and job parameters (optional)

▶ Catalog options (optional)

Next

- iv. In the data source select the one with the table and database that we want to convert click next

Choose a data source

Filter by attributes or search by keyword

Name	Database	Location	Classification
<input checked="" type="radio"/> kalia_lenodo_bucket	lenodo_data	s3://kalia-lenodo-bucket/	csv

Showing: 1 - 1 < >

- v. In transform type select change schema option and click next

Choose a transform type

Machine learning transforms are currently not supported for Glue 2.0.

Change schema
Change schema of your source data and create a new target dataset

Find matching records
Use machine learning to find matching records within your source data

Back **Next**

vi. In data target choose Data Source as S3 and format as Parquet

Choose a data target

Create tables in your data target
 Use tables in the data catalog and update your data target

Data store
Amazon S3

Format
Parquet

Connection
- Select one -
[Add connection](#)

Target path
s3://bucket/prefix/object 

[Back](#) [Next](#)

vii. For target path go to your S3 bucket and create a folder come back to glue tab here and update the target path to point to the new folder u created and click next

Amazon S3 > Buckets > kalia-lenodo-bucket

kalia-lenodo-bucket [Info](#)

[Objects](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

[!\[\]\(c751cb6741e1e0318ba9852832e07ff9_img.jpg\) Copy S3 URI](#) [!\[\]\(a38c897b15b546b1da0605aab96c93f1_img.jpg\) Copy URL](#) [!\[\]\(909b96b7db47b2714cb8619778c59a85_img.jpg\) Download](#) [!\[\]\(a9d47c3be488154830b913c625b957a4_img.jpg\) Open](#) [!\[\]\(ebb9e5d70f8af2adceacc5104b529ca7_img.jpg\) Delete](#)

[Actions ▾](#) [Create folder](#) [!\[\]\(23085005c59ebe7f8ffd25b8d40f6846_img.jpg\) Upload](#)

 Find objects by prefix < 1 > 

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	 data_utf8.csv	csv	June 9, 2022, 15:55:53 (UTC+05:30)	43.5 MB	Standard

[Create folder](#)

Create folder Info

Use folders to group objects in buckets. When you create a folder, S3 creates an object using the name that you specify followed by a slash (/). This object then appears as folder on the console. [Learn more](#)



Your bucket policy might block folder creation

If your bucket policy prevents uploading objects without specific tags, metadata, or access control list (ACL) grantees, you will not be able to create a folder using this configuration. Instead, you can use the [upload configuration](#) to upload an empty folder and specify the appropriate settings.

Folder

Folder name

 /

Folder names can't contain "/". [See rules for naming](#)

Server-side encryption

The following settings apply only to the new folder object and not to the objects contained within it.

Server-side encryption

- Disable
- Enable

[Cancel](#)

[Create folder](#)

Create folder

Assessment X kalia-lenodo-bucket X AWS Glue Console X Athena X | +

Services Q IAM X EC2 IAM S3 RDS Amazon Redshift DynamoDB EMR Amazon SageMaker Athena AWS Glue Data Pipe >

Successfully created folder "parquet_format". Operation successfully completed.

Amazon S3 > Buckets > kalia-lenodo-bucket

kalia-lenodo-bucket Info

[Objects](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

[Actions](#) [Create folder](#) [Upload](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	data_utf8.csv	csv	June 9, 2022, 15:55:53 (UTC+05:30)	43.5 MB	Standard
<input type="checkbox"/>	parquet_format/	Folder	-	-	-

Feedback Looking for language selection? Find it in the new [Unified Settings](#) © 2022, Amazon Web Services, Inc. or its affiliates.

Privacy Terms Cookie preferences

Target path

s3://bucket/prefix/object



Choose S3 path

S3

- Oaws-glue-scripts-859082096936-us-east-1
- Oaws-glue-temporary-859082096936-us-east-1
- Okalia-amazon-athena
- Okalia-lenodo-bucket
 - parquet_format

Select

Choose a data target

Data store

Amazon S3

**Format**

Parquet

**Connection**

- Select one -

**Add connection****Target path**

s3://kalia-lenodo-bucket/parquet_format

**Back****Next**

viii. You can see the conversion here click Save Job and Edit Script which will open up the code

Code (Auto Generated) :

```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job

## @params: [JOB_NAME]
args = getResolvedOptions(sys.argv, ['JOB_NAME'])

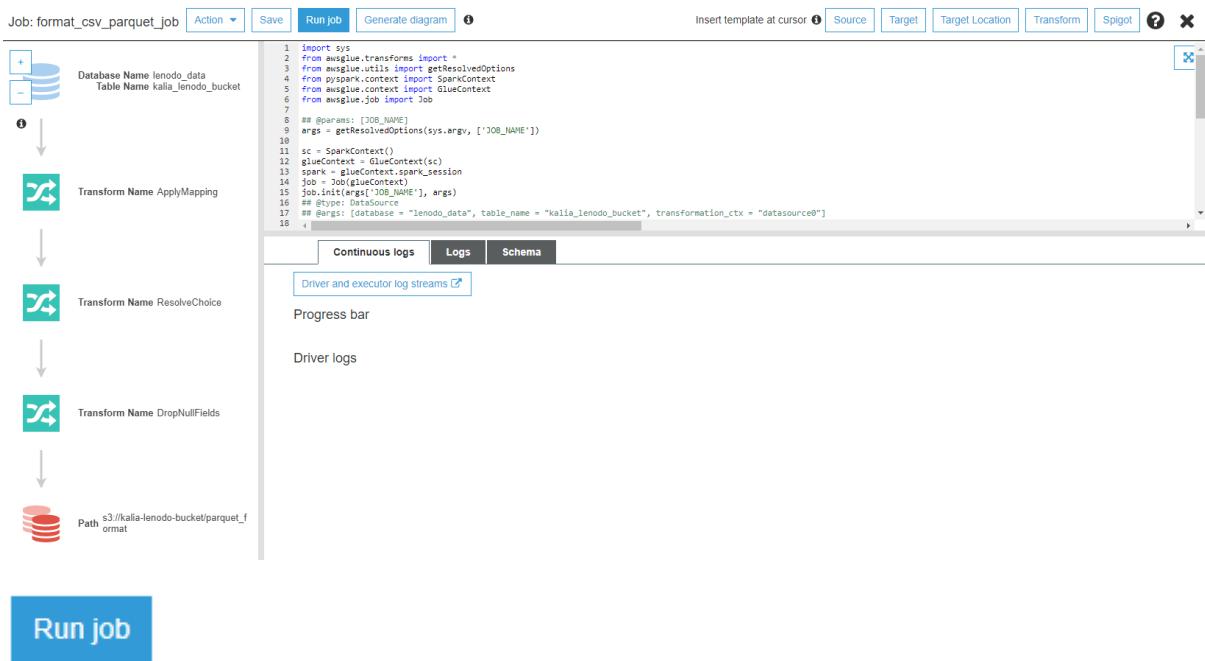
sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args['JOB_NAME'], args)
```

```

## @type: DataSource
## @args: [database = "lenodo_data", table_name = "kalia_lenodo_bucket",
transformation_ctx = "datasource0"]
## @return: datasource0
## @inputs: []
datasource0 = glueContext.create_dynamic_frame.from_catalog(database = "lenodo_data",
table_name = "kalia_lenodo_bucket", transformation_ctx = "datasource0")
## @type: ApplyMapping
## @args: [mapping = [("invoiceno", "string", "invoiceno", "string"), ("stockcode",
"string", "stockcode", "string"), ("description", "string", "description", "string"),
("quantity", "long", "quantity", "long"), ("invoicedate", "string", "invoicedate",
"string"), ("unitprice", "double", "unitprice", "double"), ("customerid", "long",
"customerid", "long"), ("country", "string", "country", "string")], transformation_ctx =
"applymapping1"]
## @return: applymapping1
## @inputs: [frame = datasource0]
applymapping1 = ApplyMapping.apply(frame = datasource0, mappings = [("invoiceno",
"string", "invoiceno", "string"), ("stockcode", "string", "stockcode", "string"),
("description", "string", "description", "string"), ("quantity", "long", "quantity",
"long"), ("invoicedate", "string", "invoicedate", "string"), ("unitprice", "double",
"unitprice", "double"), ("customerid", "long", "customerid", "long"), ("country",
"string", "country", "string")], transformation_ctx = "applymapping1")
## @type: ResolveChoice
## @args: [choice = "make_struct", transformation_ctx = "resolvechoice2"]
## @return: resolvechoice2
## @inputs: [frame = applymapping1]
resolvechoice2 = ResolveChoice.apply(frame = applymapping1, choice = "make_struct",
transformation_ctx = "resolvechoice2")
## @type: DropNullFields
## @args: [transformation_ctx = "dropnullfields3"]
## @return: dropnullfields3
## @inputs: [frame = resolvechoice2]
dropnullfields3 = DropNullFields.apply(frame = resolvechoice2, transformation_ctx =
"dropnullfields3")
## @type: DataSink
## @args: [connection_type = "s3", connection_options = {"path": "s3://kalia-lenodo-
bucket/parquet_format"}, format = "parquet", transformation_ctx = "datasink4"]
## @return: datasink4
## @inputs: [frame = dropnullfields3]
datasink4 = glueContext.write_dynamic_frame.from_options(frame = dropnullfields3,
connection_type = "s3", connection_options = {"path": "s3://kalia-lenodo-
bucket/parquet_format"}, format = "parquet", transformation_ctx = "datasink4")
job.commit()

```

ix. Click Run Job → Run Job to run

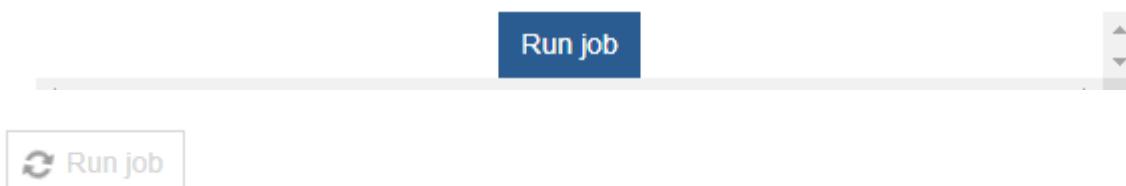


Parameters (optional)

Review and override parameter values, as needed, before running this job. Changes affect this run only. Edit a job to change default parameter values.

- ▶ Advanced properties
- ▶ Monitoring options
- ▶ Security configuration, script libraries, and job parameters

Only job **format_csv_parquet_job** is run. Jobs dependent on the completion of job **format_csv_parquet_job** will not be run. To run a job and trigger dependent jobs, define an on-demand trigger.



- x. You can keep track of the driver logs when job is finished u can check the S3 bucket folder to see if the Parquet format file was successfully generated

The screenshot shows the AWS Glue Job Editor interface. A workflow named "Job: format_csv_parquet_job" is displayed. The workflow starts with a "Database Name lenodo_data" node, followed by three transformation steps: "Transform Name ApplyMapping", "Transform Name ResolveChoice", and "Transform Name DropNullFields". The final step is a "Path s3://kalla-lenodo-bucket/parquet_format" sink. The "Logs" tab is selected, showing a log stream with several INFO-level entries. Below the logs is a "Driver and executor log streams" section. The left sidebar contains navigation links for Data catalog, ETL, Security, and Tutorials.

The screenshot shows the AWS S3 console. It lists a single object named "parquet_format" in the "kalla-lenodo-bucket" folder. The object is a Parquet file, as indicated by the ".parquet" extension in the "Name" column. The "Properties" tab is selected, showing details like Type: parquet, Last modified: June 9, 2022, 16:49:37 (UTC+05:50), Size: 3.4 MB, and Storage class: Standard.

As u can see the data is generated successfully here

Step 5: Add another crawler to crawl the Parquet data files to generate the metadata catalog of the Parquet file in order to query it with Athena

- i. Open the Crawler section and add a crawler in the exact same way we did before

The screenshot shows the AWS Glue Data Catalog interface. On the left, a sidebar lists 'Data catalog' sections: Databases, Tables, Connections, **Crawlers**, Classifiers, Schema registries, Schemas, and Settings. A blue 'Add crawler' button is highlighted. The main area displays a table of existing crawlers:

Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
lenodo_crawler		Ready	Logs	48 secs	48 secs	0	1

Below this, a modal window titled 'Add information about your crawler' is open. It contains a 'Crawler name' input field with 'transformed_data_crawler' typed in. A note below says: 'Tags, description, security configuration, and classifiers (optional)'. A large 'Next' button is at the bottom of the modal.

On the right, another modal window titled 'Specify crawler source type' is open. It says: 'Choose Existing catalog tables to specify catalog tables as the crawler source. The selected tables specify the data stores to crawl. This option doesn't support JDBC data stores.' It includes sections for 'Crawler source type' (radio buttons for 'Data stores' and 'Existing catalog tables', with 'Data stores' selected), 'Repeat crawls of S3 data stores' (radio buttons for 'Crawl all folders', 'Crawl new folders only', and 'Crawl changed folders identified by Amazon S3 Event Notifications', with 'Crawl all folders' selected), and 'Back' and 'Next' buttons at the bottom.

- ii. Make sure u select the correct path in the Data Source it must point to the folder where u added the Parquet file using the Job

Add a data store

Choose a data store

S3

Connection

Select a connection

Optional include a Network connection to use with this S3 target. Note that each crawler is limited to one Network connection so any future S3 targets will also use the same connection (or none, if left blank).

Add connection

Crawl data in

Specified path in my account
 Specified path in another account

Include path

s3://kalia-lenodo-bucket/parquet_format

All folders and files contained in the include path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

Sample size (optional)

Enter a number between 1 and 249

This field sets the number of files in each leaf folder to be crawled. If not set, all the files are crawled.

► Exclude patterns (optional)

[Back](#) [Next](#)

Add another data store

Yes
 No

[Back](#) [Next](#)

Choose an IAM role

The IAM role allows the crawler to run and access your Amazon S3 data stores. [Learn more](#)

Update a policy in an IAM role
 Choose an existing IAM role
 Create an IAM role

IAM role [i](#)

glue-service,console_s3_admin

This role must provide permissions similar to the AWS managed policy, **AWSGlueServiceRole**, plus access to your data stores.

- s3://kalia-lenodo-bucket/parquet_format

You can also create an IAM role on the [IAM console](#).

[Back](#) [Next](#)

Create a schedule for this crawler

Frequency

Run on demand

Back

Next

- iii. You can choose a different database if u need here I will go with the one I was using (which we created before)

Configure the crawler's output

Database i

lenodo_data

Add database

Prefix added to tables (optional) i

Type a prefix added to table names

- ▶ Grouping behavior for S3 data (optional)
- ▶ Configuration options (optional)

Back

Next

The screenshot shows the 'Add crawler' configuration interface in the AWS Glue console. The left sidebar lists various AWS services like EC2, IAM, RDS, etc. The main panel is titled 'Add crawler' and contains several configuration sections:

- Crawler info:** Name: transformed_data_crawler, Tags: -
- Data stores:** Data store: S3, Include path: s3://kalla-lenodo-bucket/parquet_format, Connection: -
- IAM role:** IAM role: arn:aws:iam:859082096936:role/glue-service.console_s3_admin
- Schedule:** Schedule: Run on demand
- Output:** Database: lenodo_data, Prefix added to tables (optional): lenodo_data, Create a single schema for each S3 path: false, Table level (optional): -

At the bottom right of the configuration panel are 'Back' and 'Finish' buttons.

iv. Run the crawler and generate the catalog table.

Crawlers A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

The screenshot shows two main sections of the AWS Glue console:

- Crawlers:** A table listing crawlers. One crawler, "transformed_data_crawler", is selected and has a green checkmark. It is currently "Ready".
- AWS Glue Catalog:** A table showing tables in the catalog. Two tables are listed: "parquet_format" and "kalia_lenodo_bucket". Both belong to the "lenodo_data" database.

v. U can go to Athena's query editor the refresh it to see the new table next u can preview the table

The screenshot shows the Amazon Athena Query Editor interface:

- Editor:** The active tab where a query is being run.
- Query 6:** The current query being executed: `SELECT * FROM "lenodo_data"."kalia_lenodo_bucket" limit 10;`
- Results (10):** The results of the query, displaying 10 rows of data from the "kalia_lenodo_bucket" table.

#	invoiceno	stockcode	description	quantity	invoicedate	unitprice	customerid	country
1	560773	90116	FRUIT SALAD BAG CHARM	1	7/20/2011 16:17	2.46		United Kingdom
2	560773	90122A	PINK CRYSTAL+GLASS BRACELET	1	7/20/2011 16:17	4.98		United Kingdom
3	560773	90122B	JADE CRYSTAL+GLASS BRACELET	1	7/20/2011 16:17	4.98		United Kingdom
4	560773	90130A	WHITE STONE/CRYSTAL EARRINGS	1	7/20/2011 16:17	2.9		United Kingdom
5	560773	90130C	GREEN STONE/CRYSTAL EARRINGS	1	7/20/2011 16:17	2.9		United Kingdom
6	560773	90160A	PURPLE BOUDICCA LARGE BRACELET	1	7/20/2011 16:17	7.07		United Kingdom
7	560773	90169	DAISY HAIR COMB	2	7/20/2011 16:17	2.48		United Kingdom
8	560773	90200A	PURPLE SWEETHEART BRACELET	1	7/20/2011 16:17	4.15		United Kingdom

Data

Data source

AwsDataCatalog

Database

lenodo_data

Tables and views

Create ▾

Filter tables and views

Tables (2)

- + kalia_lenodo_bucket
- parquet_format
 - invoiceno string
 - stockcode string
 - description string
 - quantity bigint
 - invoicedate string
 - unitprice double
 - customerid bigint
 - country string

Views (0)

Assessment | kalia-lenodo-bucket - S3 bucket | AWS Glue Console | AWS Console | CloudWatch Management Console | Athena

Services Search for services, features, blogs, docs, and more [Alt+S]

EC2 IAM S3 RDS Amazon Redshift DynamoDB EMR Amazon SageMaker Athena AWS Glue Data Pipeline Lambda Kinesis Step Functions CloudWatch Simple Notification Service Simple Queue Service CloudFormation Quicksight

Data source: AwsDataCatalog Database: lenodo_data

Tables and views Create ▾

Query 5 × | Query 6 × | Query 7 ×

```
1 SELECT * FROM "lenodo_data"."parquet_format" limit 10;
```

SQL Ln 1, Col 1

Run again Cancel Save Clear Create

Completed Time in queue: 109 ms Run time: 4.98 sec Data scanned: 3.39 MB

Results (10)

#	invoiceno	stockcode	description	quantity	invoicedate	unitprice	customerid	country
1	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850	United Kingdom
2	536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850	United Kingdom
3	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850	United Kingdom
4	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850	United Kingdom
5	536365	84029E	RED WOOLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850	United Kingdom
6	536365	22752	SET 7 BABUSHKA NESTING BOXES	2	12/1/2010 8:26	7.65	17850	United Kingdom
7	536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	12/1/2010 8:26	4.25	17850	United Kingdom
8	536366	22633	HAND WARMER UNION JACK	6	12/1/2010 8:28	1.85	17850	United Kingdom
9	536366	22652	HAND WARMER RED POLKA DOT	6	12/1/2010 8:28	1.85	17850	United Kingdom
10	536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	12/1/2010 8:34	1.69	13047	United Kingdom

Jaya Jagannatha

Step 6: Query the data to identify the best-selling item and countries where customers have bought the most-sold item using Athena

i. Best Selling Product

The screenshot shows the AWS Athena console interface. The query editor displays the following SQL code:

```
-- Jaya Jagannatha
-- best-selling product
SELECT
    stockcode,
    count(stockcode) as soled_stockcode_count
FROM "lenovo_data"."parquet_format"
group by stockcode
order by soled_stockcode_count desc
limit 5;
```

The results table shows the top 5 best-selling products:

stockcode	soled_stockcode_count
85123A	2313
22423	2203
85099B	2159
47566	1727
20725	1639

ii. In which countries the best selling product is sold

The screenshot shows the AWS Athena console interface. The query editor displays the following SQL code:

```
-- Jaya Jagannatha
-- Country where best-selling product is sold
select distinct country,
    stockcode
From "lenovo_data"."parquet_format"
where stockcode = (
    SELECT stockcode
    from (
        SELECT stockcode,
            count(stockcode) as soled_stockcode_count
        FROM "lenovo_data"."parquet_format"
        group by stockcode
        order by soled_stockcode_count desc
        limit 1
    )
);
```

The results table shows the countries where the best-selling product (85123A) is sold:

country	stockcode
France	85123A
Finland	85123A
Netherlands	85123A
Australia	85123A
Italy	85123A

The screenshot shows the AWS Athena Query Editor interface. The left sidebar displays the schema for the `lenovo_data` dataset, which includes columns for `stockcode`, `description`, `quantity`, `invokedate`, `unitprice`, `customerid`, and `country`. The main area shows the results of a query, titled "Results (16)". The table has two columns: "country" and "stockcode". The data consists of 16 rows, each mapping a country to a specific stock code. The countries listed are France, Finland, Netherlands, Australia, Italy, Singapore, Spain, Malta, Israel, United Kingdom, Channel Islands, Germany, Switzerland, Cyprus, EIRE, and Portugal. The stock codes are all 85123A.

country	stockcode
France	85123A
Finland	85123A
Netherlands	85123A
Australia	85123A
Italy	85123A
Singapore	85123A
Spain	85123A
Malta	85123A
Israel	85123A
United Kingdom	85123A
Channel Islands	85123A
Germany	85123A
Switzerland	85123A
Cyprus	85123A
EIRE	85123A
Portugal	85123A

iii. Which products and its marketing needs most improvement (Showing only 5 here)

The screenshot shows the AWS Athena Query Editor interface. The top navigation bar includes tabs for "Query 6", "Query 7", "Query 5", "Query 10", "Query 11", "Query 12", "Query 13", "Query 14", "Query 15", and "Query 16". The current query, "Query 12", is highlighted. The SQL code for this query is:

```

1 -- Jaya Jagannatha
2 -- which product and its marketing needs the most improvement
3 SELECT stockcode, count(stockcode) as soled_stockcode_count
4 FROM "lenovo_data"."parquet_format"
5 group by stockcode
6 order by soled_stockcode_count
7 limit 10;

```

The results table, titled "Results (10)", lists 10 stock codes, each with a count of 1. The stock codes are 21589, 72814, 37503, 728038, 84968f, 72803b, 22275, 35969, DCGS0057, and 85035b.

stockcode	soled_stockcode_count
21589	1
72814	1
37503	1
728038	1
84968f	1
72803b	1
22275	1
35969	1
DCGS0057	1
85035b	1

- iv. Detailed view of which country has sale of which product(stockcode) and with which quantity where it's ordered by descending order. As the query result is very long I have just taken screenshot of top 13 rows

```

1 -- Jaya Zaganatha
2 select country,stockcode,count(stockcode) as number_sold FROM "lenovo_data"."

```

- v. Creating a different table and storing this data for further analysis

```

1 -- Jaya Zaganatha
2 create table countrywise_data as
3 select country,
4       stockcode,
5       count(stockcode) as number_sold
6  FROM "lenovo_data"."

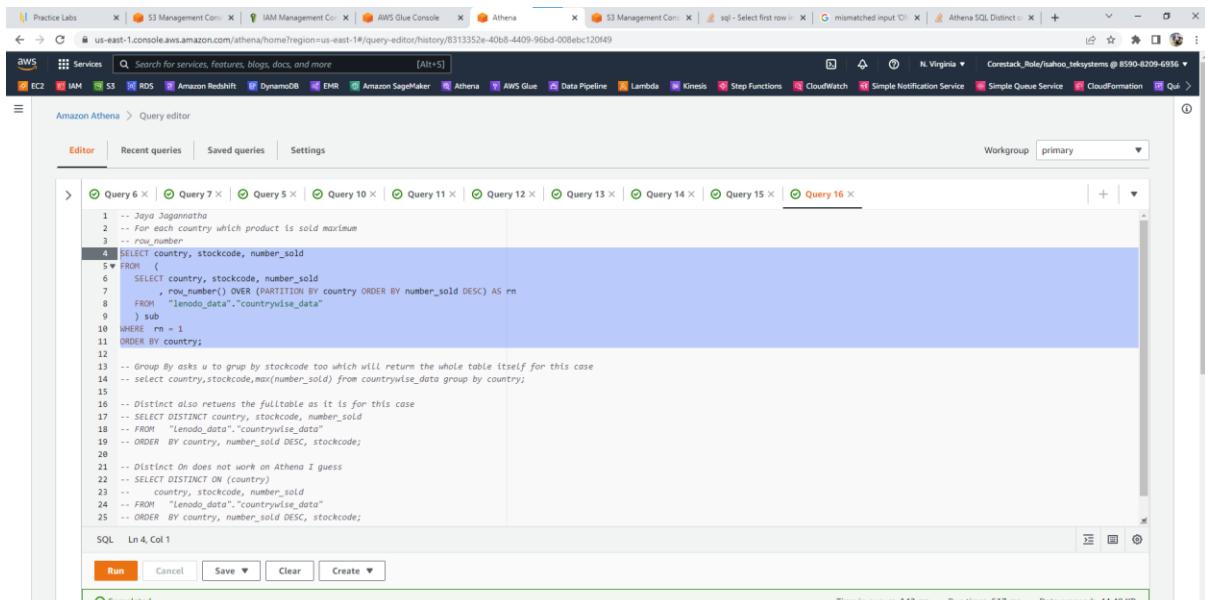
```

```

1 -- Jaya Zaganatha
2 select * from countrywise_data
SQL Ln 2, Col 1
Run again Cancel Save ▾ Clear Create ▾
Completed Time in queue: 230 ms Run time: 770 ms Data scanned: 44.48 KB
Results (100+)
Search rows
# country stockcode number_sold
6 Australia 22382 8
7 Australia 22699 8
8 Australia 84978 8
9 Australia 22138 8
10 Australia 47566 8
11 Australia 48138 7
12 Australia 23236 7
13 Australia 22619 7
14 Australia 21915 7
15 Australia 23206 7
16 Australia 23298 7
17 Australia 22492 7
18 Australia 475908 7
19 Australia 22698 7
20 Australia 23245 7
21 Australia 22960 6
22 Australia 22629 6
23 Australia 21936 6

```

- vi. For each country which is the maximum sold product with the count it is sold at and with its stockcode.



```

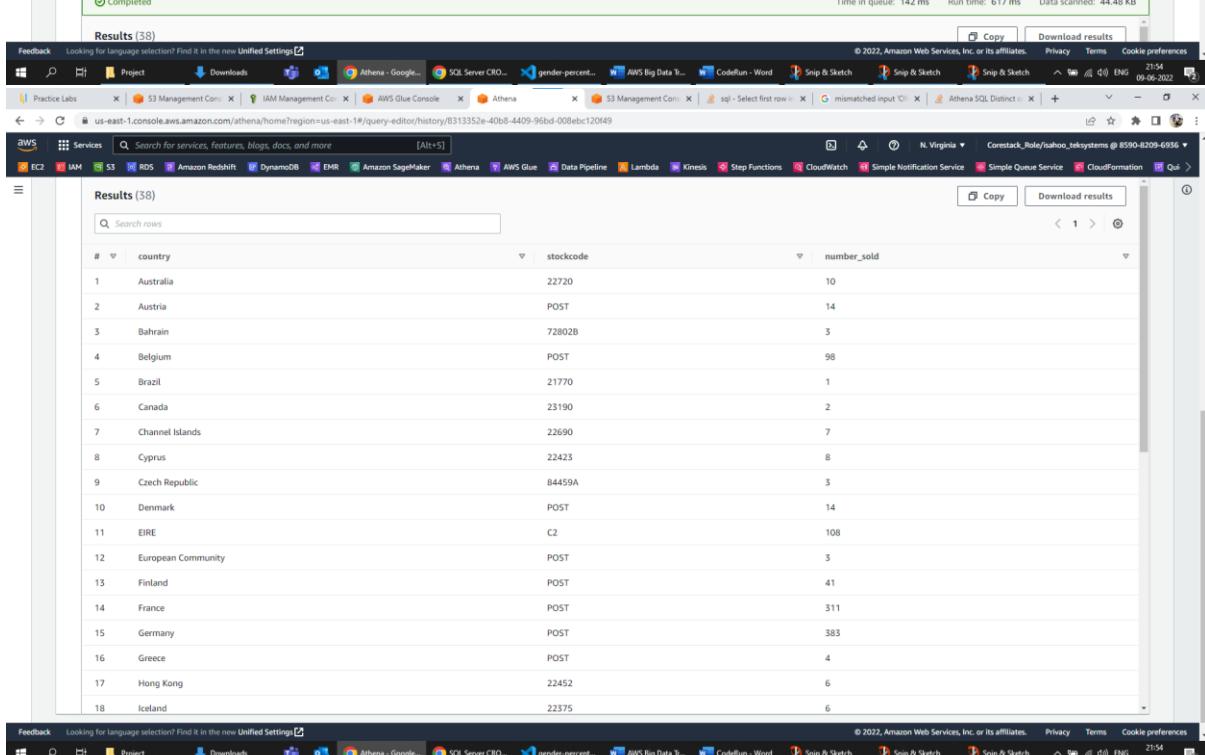
1 -- Japa Seppantha
2 -- For each country which product is sold maximum
3 -- row number
4 SELECT country, stockcode, number_sold
5 FROM (
6   SELECT country, stockcode, number_sold
7     , row_number() OVER (PARTITION BY country ORDER BY number_sold DESC) AS rn
8   FROM "lenovo_data"."countrywise_data"
9 ) sub
10 WHERE rn = 1
11 ORDER BY country;
12
13 -- Group By asks u to group by stockcode too which will return the whole table itself for this case
14 -- select country,stockcode,max(number_sold) from countrywise_data group by country;
15
16 -- Distinct also returns the fullTable as it is for this case
17 -- SELECT DISTINCT country, stockcode, number_sold
18 -- FROM "lenovo_data"."countrywise_data"
19 -- ORDER BY country, number_sold DESC, stockcode;
20
21 -- Distinct On does not work on Athena I guess
22 -- SELECT DISTINCT ON (country)
23 -- country, stockcode, number_sold
24 -- FROM "Lenovo_data"."countrywise_data"
25 -- ORDER BY country, number_sold DESC, stockcode;
SQL Ln 4, Col 1
Run Cancel Save ▾ Clear Create ▾
Completed Time in queue: 142 ms Run time: 617 ms Data scanned: 44.48 KB

```

Results (38)

#	country	stockcode	number_sold
1	Australia	22720	10
2	Austria	POST	14
3	Bahrain	72802B	3
4	Belgium	POST	98
5	Brazil	21770	1
6	Canada	23190	2
7	Channel Islands	22690	7
8	Cyprus	22423	8
9	Czech Republic	84459A	3
10	Denmark	POST	14
11	EIRE	C2	108
12	European Community	POST	3
13	Finland	POST	41
14	France	POST	311
15	Germany	POST	383
16	Greece	POST	4
17	Hong Kong	22452	6
18	Iceland	22375	6

Feedback Looking for language selection? Find it in the new Unified Settings [Feedback](#) © 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 21:54 09-06-2022



SQL Query:

```
-- Jaya Jagannatha Kalia Santa

-- csv data table
SELECT * FROM "lenodo_data"."kalia_lenodo_bucket" limit 10;

-- parquet data table
SELECT * FROM "lenodo_data"."parquet_format" limit 10;
```

```
-- best-selling product
SELECT
    stockcode,
    count(stockcode) as soled_stockcode_count
FROM "lenodo_data"."parquet_format"
group by stockcode
order by soled_stockcode_count desc
limit 5;
```

```
-- Country where best-selling product is sold
select distinct country,
    stockcode
From "lenodo_data"."parquet_format"
where stockcode = (
    SELECT stockcode
    from (
        SELECT stockcode,
            count(stockcode) as soled_stockcode_count
        FROM "lenodo_data"."parquet_format"
        group by stockcode
        order by soled_stockcode_count desc
        limit 1
    )
);
```

```
-- which product and its marketing needs the most improvement
SELECT stockcode, count(stockcode) as soled_stockcode_count
FROM "lenodo_data"."parquet_format"
group by stockcode
order by soled_stockcode_count;
```

```
-- For each country what product sold for which quantity (for each country quantity is in
-- decending order)
select country,
       stockcode,
       count(stockcode) as number_sold
FROM "lenodo_data"."parquet_format"
group by country,
       stockcode
order by country,
       number_sold desc;

-- Put the data generated in previous query to a table
create table countrywise_data as
select country,
       stockcode,
       count(stockcode) as number_sold
FROM "lenodo_data"."parquet_format"
group by country,
       stockcode
order by country,
       number_sold desc;
-- Print the data using select
select * from countrywise_data;
```

```
-- Jaya Jagannatha
-- For each country which product is sold maximum
-- row_number works correctly
select country,stockcode,number_sold
from (
    select country,stockcode,number_sold,
    row_number() over (partition by country order by number_sold desc) as rn
    from "lenodo_data"."countrywise_data"
) sub
where rn = 1
order by country;

-- Group By asks u to grpup by stockcode too which will return the whole table itself for
this case
-- select country,stockcode,max(number_sold) from countrywise_data group by country;

-- Distinct also retuens the fulltable as it is for this case
-- SELECT DISTINCT country, stockcode, number_sold
-- FROM "lenodo_data"."countrywise_data"
-- ORDER BY country, number_sold DESC, stockcode;

-- Distinct On does not work on Athena I guess
-- SELECT DISTINCT ON (country)
--     country, stockcode, number_sold
-- FROM "lenodo_data"."countrywise_data"
-- ORDER BY country, number_sold DESC, stockcode;
```