

Naming conventions

project-1-s3-glue/

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 ▾

Name Type Last modified Size Storage class

Name	Type	Last modified	Size	Storage class
datasets/	Folder	-	-	-
gluecrawlerdataset/	Folder	-	-	-

Amazon S3 > Buckets > kpmg-lighthouse-projects-aws > project-1-s3-glue/ > datasets/

datasets/

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 ▾

Name Type Last modified Size Storage class

Name	Type	Last modified	Size	Storage class
customers.csv	csv	April 14, 2022, 12:17:20 (UTC+05:30)	7.8 KB	Standard
sales.csv	csv	April 14, 2022, 12:17:20 (UTC+05:30)	256.3 KB	Standard

Identity and Access Management (IAM)

Introducing the new IAM roles experience
We've redesigned the IAM roles experience to make it easier to use. Let us know what you think.

IAM > Roles

Roles (22) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name Trusted entities Last activity

Role name	Trusted entities	Last activity
AWSGlueServiceSageMakerNotebookRole-tejkpmgsagmakerole	AWS Service: sagemaker	19 minutes ago
AWSGlueServiceSageMakerNotebookRole-tejsagemakerrole	AWS Service: sagemaker	15 days ago
KinesisFirehoseServiceRole-tej-aws-case-ap-south-1-1648321071266	AWS Service: firehose	18 days ago
tej-kpmg-lighthouse-project1-gluerole	AWS Service: glue	21 minutes ago
tejemgluerole	AWS Service: glue	6 hours ago
tejemrgluerole	AWS Service: glue	6 hours ago
tejgluerole	AWS Service: glue	15 days ago

Screenshot of the AWS IAM Permissions page.

The left sidebar shows navigation links for Identity and Access Management (IAM), including Access management, Roles, Access reports, and Glue database.

The main content area displays a table of permissions policies:

Policy name	Type	Description
AmazonS3FullAccess	AWS managed	Provides full access to all buckets
AmazonRedshiftFullAccess	AWS managed	Provides full access to Amazon Redshift
PowerUserAccess	AWS managed - job function	Provides full access to AWS services
AWSLakeFormationDataAdmin	AWS managed	Grants administrative access

Below the table, there is a section titled "Permissions boundary - (not set)" with a note: "Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting but can be used to delegate permission management to others." A "Set permissions boundary" button is present.

Footer: © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Glue database

Screenshot of the AWS Glue Databases page.

The left sidebar shows navigation links for Data catalog, ETL, and AWS Glue Studio.

The main content area displays a table of databases:

Name	Description
projectemrsparkdb	
teigluedb	
tejkpmglighthouseproject1db	

The database "tejkpmglighthouseproject1db" is highlighted with a red box.

Footer: © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

AWS Lake Formation > Databases

Databases (0/3)

Name	Owner account ID	Shared resource
projectemrsparkdb	133837890951	-
tejgluedb	133837890951	-
tejkpmglighthouseproject1db	133837890951	-

Actions ▲ View tables Create database

Database Delete Edit Edit LF-tags Create resource link Permissions s3://tej-glue-data-lak... Grant (highlighted) Revoke Verify permissions View permissions

Feedback © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

AWS Lake Formation > How it works

1 Set administrative roles Decide who should be the administrators for your data lake, and optionally who can create new databases. Choose administrators (highlighted)

2 Define LF-tag ontology In order to create and manage catalog and data access permissions, define a set of LF-Tags that will help you quickly decide all types of access needs. Manage LF-Tags

3 Delegate LF-tag permissions - optional Lastly, you can decide who should see the LF-tag ontology and LF-tag catalog resources (databases, tables, columns) in order to control data access. Manage LF-tag permissions

Data lake administrators (0/2) Administrators can view all metadata in the AWS Glue Data Catalog. They can also grant and revoke permissions on data resources to principals, including themselves. Choose administrators

Name	Type
tej-kpmg-lighthouse-project1-gluerole	IAM role
tejgluerole	IAM role

Database creators Choose IAM principals permitted to create databases in your AWS Glue Data Catalog. Find database creators

Feedback © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

AWS Glue

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.

Data catalog

- Databases
- Tables
- Connections
- Crawlers**
- Classifiers
- Schema registries
- Schemas
- Settings

ETL

- AWS Glue Studio
- Jobs - New
- Jobs (legacy)
- ML Transforms
- Blueprints
- Workflows
- Triggers
- Dev endpoints
- Notebooks

Add crawler Run crawler Action Filter by tags and attributes Showing: 1 - 2 User preferences

Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
tejmrsparkcrawler		Ready	Logs	47 secs	47 secs	0	1
tejkpmglighthouseproject1crawler1sales		Ready	Logs	47 secs	47 secs	0	1

Feedback © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

AWS Glue

Dev endpoints A development endpoint is an environment used to develop and test your AWS Glue scripts.

ETL

- AWS Glue Studio
- Jobs - New
- Jobs (legacy)
- ML Transforms
- Blueprints
- Workflows
- Triggers
- Dev endpoints**
- Notebooks

Security

- Security configurations

Tutorials

- Add crawler
- Explore table
- Add job
- Resources
- What's new 10+

Add endpoint Action Filter by tags Showing: 1 - 1 User preferences

Endpoint name	Provisioning status	Running for
tejkpmglighthouseproject1endpoint	READY	20m

Feedback © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the AWS Glue Studio interface. On the left sidebar, under the 'Notebooks' section, there are several options: AWS Glue Studio, Jobs (legacy), ML Transforms, Blueprints, Workflows, Triggers, Dev endpoints, and Notebooks. The 'Notebooks' option is selected. The main content area displays a table of notebooks. The table has columns: Notebook name, Development endpoint, Status, Creation time, and Last modified time. There are two entries:

Notebook name	Development endpoint	Status	Creation time	Last modified time
aws-glue-tejkpmglighthouseproject1endpoint	tejkpmglighthouseproject1endpoint	Ready	14 April 2022 3:20 PM UTC+5:30	14 April 2022 3:25 PM UTC+5:30
aws-glue-tejnotebook	tejendpoint	Stopped	30 March 2022 3:34 AM UTC+5:30	30 March 2022 4:48 AM UTC+5:30

1)The Data for the Workshop

In the workshop, you learn about using PySpark to create Glue Job. PySpark is used to process and transform the data. Before you start the workshop, let's understand the data to be configured in the data lake for the workshop.

You will configure data lake with two data sets - **sales** and **customers**. The data sets are stored in Amazon S3. The AWS Glue and AWS Lake Formation services are used to create the data lake.

The following are the schema of the data sets:

Uber eats customers data set fields: {CUSTOMERID, CUSTOMERNAME, EMAIL, CITY, COUNTRY, TERRITORY, CONTACTFIRSTNAME, CONTACTLASTNAME}

Uber eats sales data set fields: {ORDERNUMBER, QUANTITYORDERED, PRICEEACH, ORDERLINENUMBER, SALES, ORDERDATE, STATUS, QTR_ID, MONTH_ID, YEAR_ID, PRODUCTLINE, MSRP, PRODUCTCODE, DEALSIZE, CUSTOMERID}

Let's start building now.

2)

Create IAM Role

You now create IAM Role which is used by the AWS Glue crawler, job and developer endpoint to perform tasks like data catalog creation, data transformation / processing.

1. Login to your AWS account and go to IAM Management Console. Click on the **Roles** menu in the left side and then click on the **Create role** button.

IAM dashboard

Security recommendations

- Add MFA for root user
- Root user has no active access keys

IAM resources

User groups	Users	Roles	Policies	Identity providers
0	0	10	1	0

What's new

- Right-size permissions for more roles in your account using IAM Access Analyzer to generate 50 fine-grained IAM policies per day. 4 months ago
- Amazon S3 Object Ownership can now disable access control lists to simplify access management for data in S3. 4 months ago
- Amazon Redshift simplifies the use of other AWS services by introducing the default IAM role. 4 months ago
- IAM Access Analyzer helps you generate fine-grained policies that specify the required actions for more than 50 services. 7 months ago

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

Identity and Access Management (IAM)

IAM > Roles

Roles (10) info

We've redesigned the IAM roles experience to make it easier to use. Let us know what you think.

Role name	Trusted entities	Last activity
AWSServiceRoleForEC2Spot	AWS Service: spot (Service-Linked Role)	-
AWSServiceRoleForRDS	AWS Service: rds (Service-Linked Role)	39 days ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-
databricks-workspace-stack-CopyZipsRole-DTYQKJVNSVL	AWS Service: lambda	28 days ago
databricks-workspace-stack-functionRole-V2PG87S54Q5F	AWS Service: lambda	28 days ago
db-7aca31d819a316fd3680f43fe179157-lam-role	Account: 414351767826	20 minutes ago
EC2Access	AWS Service: ec2	20 minutes ago

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

On the next screen, select **Glue** as the AWS Service. Click on the **Next: Permission** button.

Select trusted entity

Step 2 Add permissions

Step 3 Name, review, and create

Trusted entity type

- AWS service
- AWS account
- Web identity

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Common use cases

- EC2
- Lambda

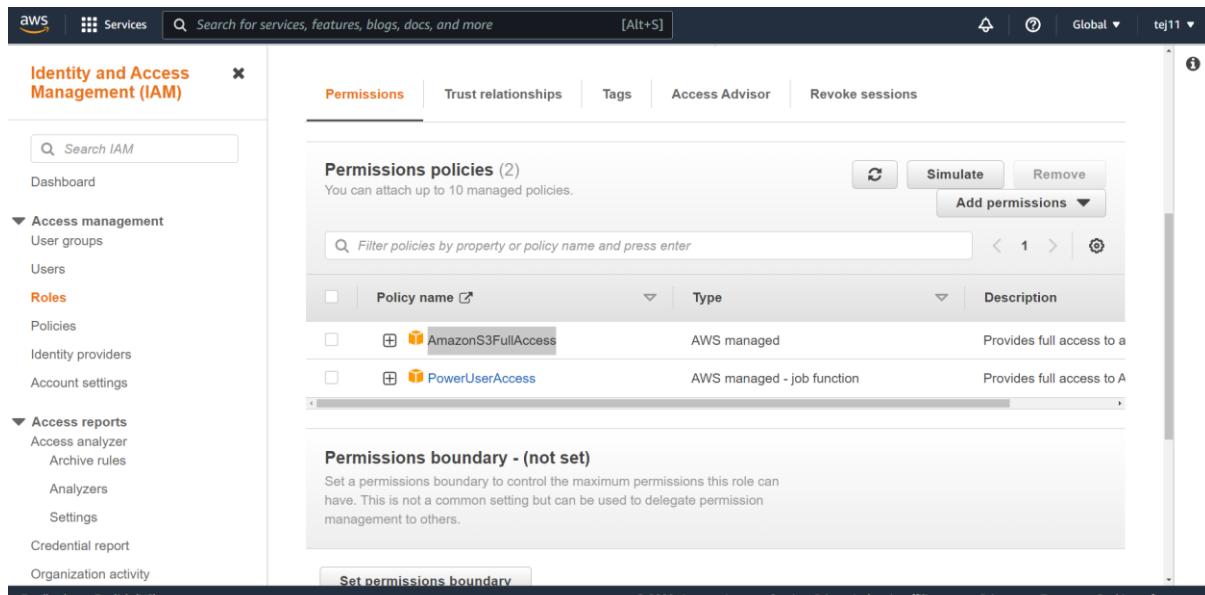
Search for service to view use case

glue

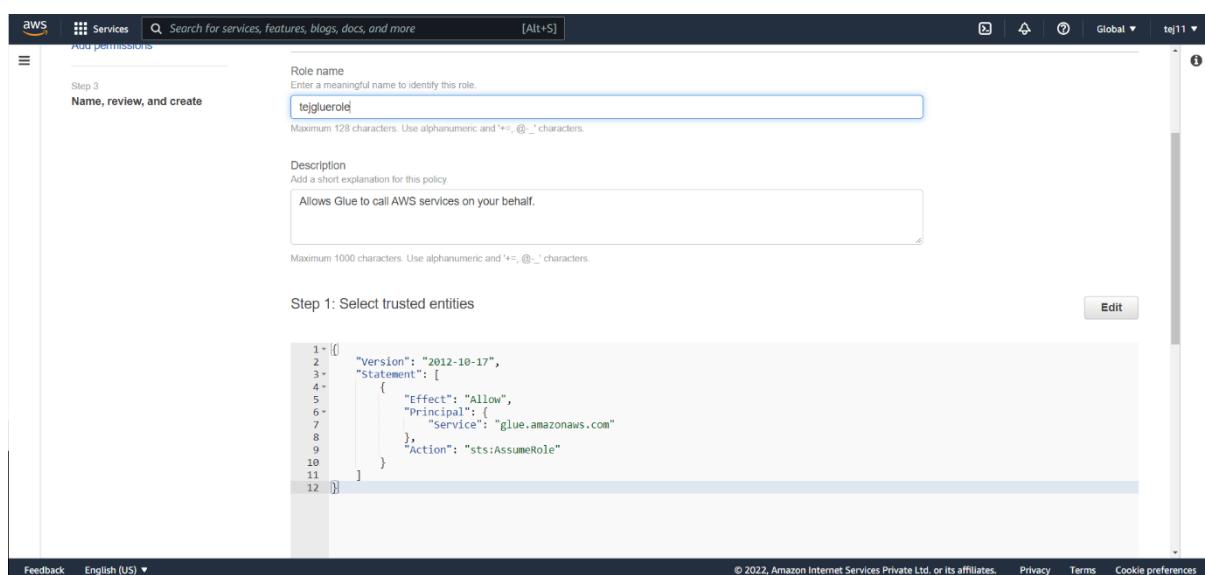
Cancel **Next**

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

On the next screen, select **PowerUserAccess**/**AmazonS3FullAccess** as the policies. The workshop is using this policy to simplify the authorization. However, in the production implementation, you select minimum required permission for the role. Click on the **Next: Tags** button.



The screenshot shows the AWS Identity and Access Management (IAM) console. On the left, the navigation pane includes sections for Access management, Roles, and Access reports. The main area displays the 'Permissions' tab of a role configuration page. Under 'Permissions policies (2)', two policies are listed: 'AmazonS3FullAccess' (AWS managed) and 'PowerUserAccess' (AWS managed - job function). Both policies provide full access to their respective services. Below this, a 'Permissions boundary - (not set)' section is present, which allows setting a boundary to control maximum permissions. The bottom of the page includes standard AWS footer links for feedback, language selection, and cookie preferences.



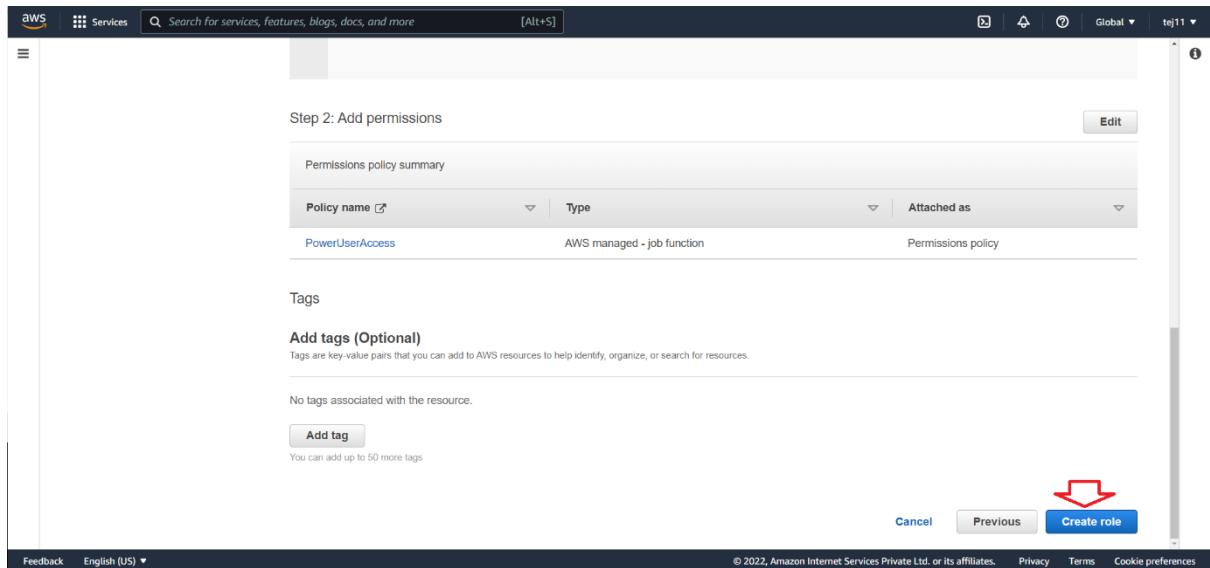
The screenshot shows the 'Name, review, and create' step of a new role creation process. The 'Role name' field contains 'tejgluerole'. The 'Description' field contains the text 'Allows Glue to call AWS services on your behalf.'. Below these fields, a large text area displays a JSON policy document:

```

1 - [{
2 -   "Version": "2012-10-17",
3 -   "Statement": [
4 -     {
5 -       "Effect": "Allow",
6 -       "Principal": "*",
7 -       "Service": "glue.amazonaws.com"
8 -     },
9 -     {
10 -       "Action": "sts:AssumeRole"
11 -     }
12 -   ]
}

```

The bottom of the page includes standard AWS footer links for feedback, language selection, and cookie preferences.



1. The role is created in no time. The next task is to create the S3 bucket for the data lake.

3

Name	AWS Region	Access	Creation date
aws-case-study-2-uber	Asia Pacific (Mumbai) ap-south-1	Objects can be public	March 26, 2022, 09:52:11 (UTC+05:30)
databricks-workspace-stack-lambdadipsbucket-r6pez9ps63p0	Asia Pacific (Mumbai) ap-south-1	Objects can be public	March 1, 2022, 03:09:38 (UTC+05:30)
db-7aca51d819a316fd3680f43feb179157-s-root-bucket	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	March 1, 2022, 03:09:38 (UTC+05:30)
full-load-auto-replication-pyspark	US East (N. Virginia) us-east-1	Objects can be public	February 18, 2022, 08:51:45 (UTC+05:30)
tej-aws-casesudy2-kinesis-firebase-stream-output	Asia Pacific (Mumbai) ap-south-1	Objects can be public	March 27, 2022, 00:29:15 (UTC+05:30)

Create bucket button to create a new bucket with name **tej-glue-data-lake**. If the bucket name is not available, then use a different name which is available.

General configuration

Bucket name: **tej-glue-data-lake** (highlighted with a red arrow)

AWS Region: **Asia Pacific (Mumbai) ap-south-1** (highlighted with a red arrow)

Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.
Choose bucket

Object Ownership Info
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership
 Bucket owner preferred
If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

Feedback English (US) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or S3. 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 permission 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.

⚠️ Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

I acknowledge that the current settings might result in this bucket and the objects within becoming public.

Feedback English (US) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

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

Server-side encryption
 Disable
 Enable

Advanced settings

Object Lock
Store objects using a write-once-read-many (WORM) model to help you prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely. [Learn more](#)

Disable
 Enable
Permanently allows objects in this bucket to be locked. Additional Object Lock configuration is required in bucket details after bucket creation to protect objects in this bucket from being deleted or overwritten.

Object Lock works only in versioned buckets. Enabling Object Lock automatically enables Bucket Versioning.

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

Cancel **Create bucket**

Feedback English (US) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the AWS S3 console with a success message at the top: "Successfully created bucket 'tej-glue-data-lake'". Below it, a tooltip says "To upload files and folders, or to configure additional bucket settings choose View details." A red arrow points to the newly created bucket "tej-glue-data-lake" in the list of buckets.

Name	AWS Region	Access	Creation date
aws-case-study-2-uber	Asia Pacific (Mumbai) ap-south-1	Objects can be public	March 26, 2022, 09:52:11 (UTC+05:30)
databricks-workspace-stack-lambda-zipsbucket-r6pe29ps63p0	Asia Pacific (Mumbai) ap-south-1	Objects can be public	March 1, 2022, 03:09:38 (UTC+05:30)
db-7aca51d819a316fd3680f43feb179157-s3-root-bucket	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	March 1, 2022, 03:09:38 (UTC+05:30)
full-load-auto-replication-pyspark	US East (N. Virginia) us-east-1	Objects can be public	February 18, 2022, 08:51:45 (UTC+05:30)
tej-aws-casesudy2-kinesis-firehose-stream-output	Asia Pacific (Mumbai) ap-south-1	Objects can be public	March 27, 2022, 00:29:15 (UTC+05:30)
tej-glue-data-lake	Asia Pacific (Mumbai) ap-south-1	Objects can be public	March 30, 2022, 00:52:41 (UTC+05:30)

Click on the **tej-glue-data-lake** bucket to open it. Within the **tej-glue-data-lake**, create two folders **data** and **script** using the **+ Create folder** button. The **data** folder is used to keep the data lake data while the **script** folder is used by Glue job to store the script.

The screenshot shows the AWS S3 console with the "tej-glue-data-lake" bucket selected. The top navigation bar has a "Create folder" button highlighted with a red arrow. The "Objects" tab is selected, showing a list of objects with a single entry: "No objects".

aws Services Search for services, features, blogs, docs, and more [Alt+S] Global tej11

Amazon S3

- Buckets
- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- Access analyzer for S3
- Block Public Access settings for this account
- Storage Lens
- Dashboards
- AWS Organizations settings
- Feature spotlight
- AWS Marketplace for S3

How to optimize your costs on S3.

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

Create folder

Feedback English (US) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

aws Services Search for services, features, blogs, docs, and more [Alt+S] Global tej11

Amazon S3

- Buckets
- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- Access analyzer for S3
- Block Public Access settings for this account
- Storage Lens
- Dashboards
- AWS Organizations settings
- Feature spotlight
- AWS Marketplace for S3

How to optimize your costs on S3.

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

Create folder

Feedback English (US) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

aws Services Search for services, features, blogs, docs, and more [Alt+S] Global tej11

Amazon S3

- Buckets
- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- Access analyzer for S3
- Block Public Access settings for this account
- Storage Lens
- Dashboards
- AWS Organizations settings
- Feature spotlight
- AWS Marketplace for S3

Successfully created folder "script". Operation successfully completed.

How to optimize your costs on S3.

Amazon S3 > Buckets > tej-glue-data-lake

tej-glue-data-lake Info

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](#)

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	<input type="checkbox"/> data/	Folder	-	-	-
<input type="checkbox"/>	<input type="checkbox"/> script/	Folder	-	-	-

Copy S3 URI Copy URL Download Open Actions Create folder Upload Find objects by prefix < 1 > ⌂

Feedback English (US) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Click on the **data** folder to open it. Within the **data** folder, create two folders **customers** and **sales** using the **+ Create folder** button.

The screenshot shows the AWS S3 console interface. On the left, the navigation pane includes options like Buckets, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, and Access analyzer for S3. The main area shows a success message: "Successfully created folder 'sales' Operation successfully completed." Below this, a blue banner provides tips on optimizing costs. The current path is Amazon S3 > Buckets > tej-glue-data-lake > data/. The "Objects" tab is selected, showing a list of objects with a count of 2. The objects are "customers/" and "sales/", both listed as Folder type. A red arrow points to the "data/" folder in the breadcrumb path, and another red arrow points to the "customers/" and "sales/" entries in the object list table.

Click to open **customers** and **sales** folders one by one and upload **customers.csv** and **sales.csv** files in the **customers** and **sales** folders respectively. Use **Upload** button to upload the files. The **customers.csv** and **sales.csv** files are available for download using the following links –

<https://filepost.io/d/XuX9yGpRfP>

upload steps

The screenshot shows the AWS S3 console interface for uploading files. The path is Amazon S3 > Buckets > tej-glue-data-lake > data/ > customers/ > Upload. The "Upload" tab is selected. A red arrow points to the "customers.csv" file in the "Files and folders" list, which shows a total size of 7.8 KB and a type of text/csv. The "Destination" section shows the target location as s3://tej-glue-data-lake/data/customers/. A red arrow also points to the "customers.csv" entry in the destination list.

The screenshot shows the 'Bucket settings' page for a specific S3 bucket. The 'Permissions' section is expanded, showing the 'Access control list (ACL)' configuration. A red arrow points to the radio button for 'Choose from predefined ACLs'. Another red arrow points to the radio button for 'Grant public-read access', which is selected. A third red arrow points to the checkbox 'I understand the risk of granting public-read access to the specified objects', which is checked.

The screenshot shows the 'Upload status' page after a file has been uploaded. The summary table indicates 1 file was uploaded successfully (7.8 KB) and 0 files failed (0 B). The 'Files and folders' tab is selected, showing a table with one item: 'sales.csv' (text/csv, 256.3 KB). A red box highlights the 'Close' button in the top right corner. A red annotation 'after pressing upload button' is placed above the file list table.

Next uploading sales data in sales folder.

The screenshot shows the 'Upload' dialog for a new file. In the 'Files and folders' section, 'sales.csv' is selected for upload. A red arrow points to the checkbox next to 'sales.csv'. The 'Destination' section shows the target location as 's3://tej-glue-data-lake/data/sales/'. The 'Permissions' and 'Properties' sections are also visible at the bottom. A red annotation 'Cancel' is placed near the 'Upload' button.

The screenshot shows the AWS S3 Bucket Permissions configuration page. In the 'Access control list (ACL)' section, the 'Grant public-read access' option is selected. A warning message at the bottom of this section states: 'Granting public-read access is not recommended. Anyone in the world will be able to access the specified objects.' There is also a checkbox for accepting the risk of granting public-read access.

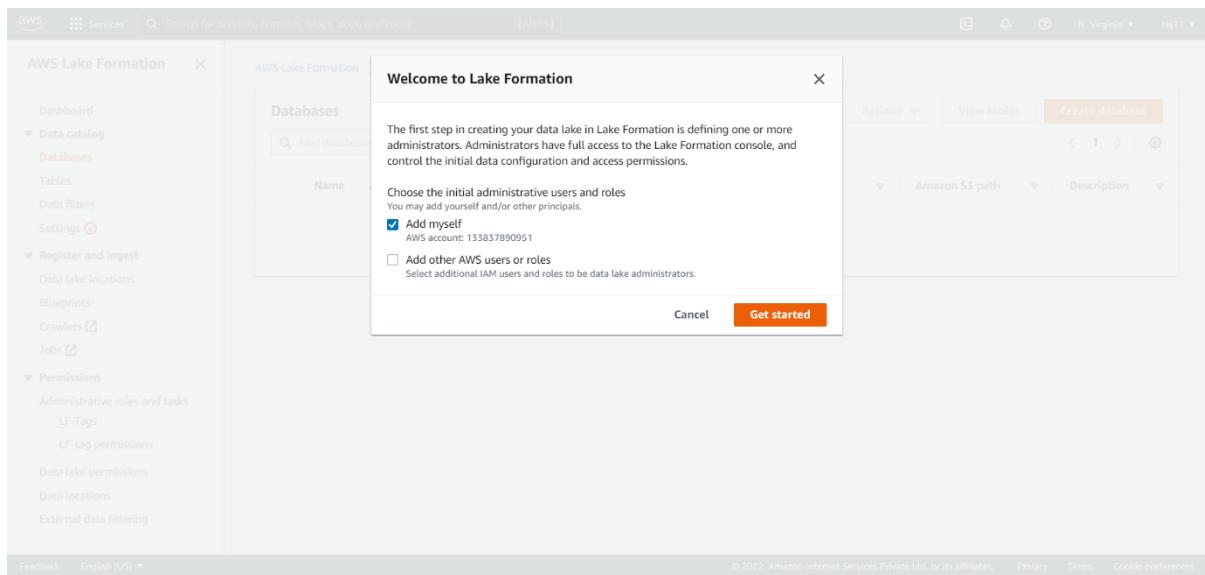
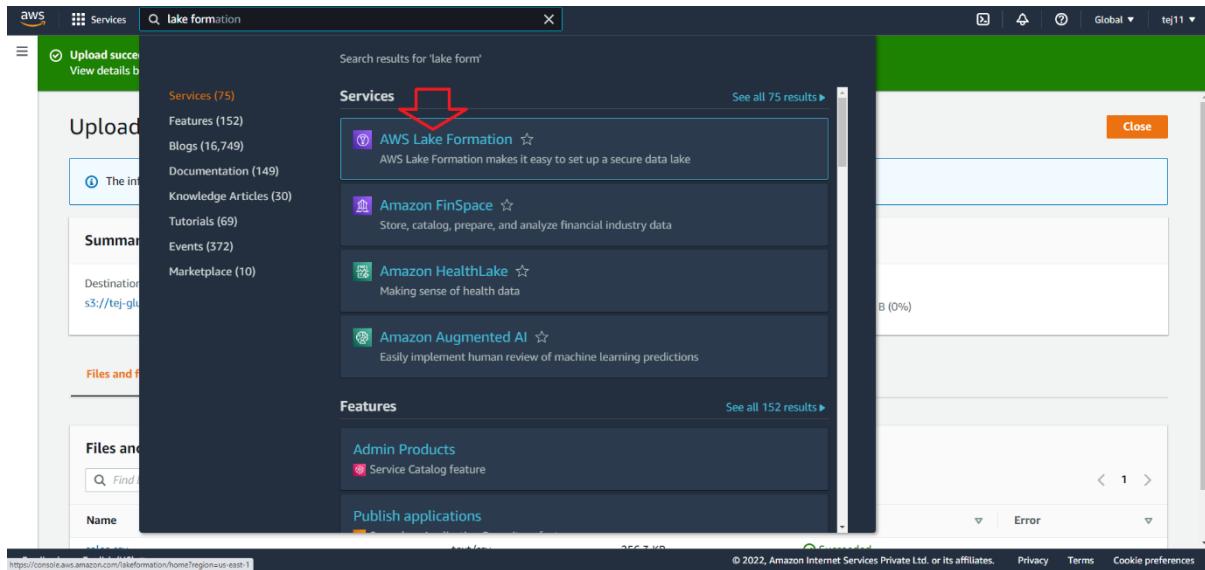
The screenshot shows the AWS S3 Bucket Properties configuration page. At the bottom right, there is a large orange 'Upload' button with a red arrow pointing towards it. To the left of the upload button, there is a 'Cancel' button.

1. *The data is ready. Let's start with the data lake configuration.

4

Configure Data Lake

The data is ready in Amazon S3 bucket. Let's start configuration of the data lake. In data lake, you create a database and configure it with Amazon S3 bucket location. A database is used to organize data catalog tables in the data lake. You will later configure crawler to automatically create the data catalog tables in the data lake.



Open the AWS Lake Formation console. If you are using Lake Formation for the first time in the region, it will ask you to create a data lake administrator. A data lake administrator is an IAM user or IAM role that performs administrative tasks on the data lake. For the first time user, it will pop-up a message to add administrators. Click on the **Add administrators** button to create administrators for your Data Lake.

Welcome to Lake Formation

X

The first step in creating your data lake in Lake Formation is to define one or more administrators. Administrators have full access to the Lake Formation system, and control the initial data configuration and access permissions.

Click Add administrators below to add your administrators.

Cancel

Add administrators

The screenshot shows the AWS Lake Formation 'How it works' section. On the left, there's a sidebar with navigation links like Dashboard, Data catalog, Tables, Data filters, Settings, Register and ingest, Crawlers, Jobs, Permissions, Administrative roles and tasks, LF-Tags, LF-tag permissions, Data lake permissions, Data locations, and External data filtering. The main area has a header 'How it works' with three numbered steps: 1. Set administrative roles (with a 'Choose administrators' button), 2. Define LF-tag ontology (with a 'Manage LF-Tags' button), and 3. Delegate LF-tag permissions - optional (with a 'Manage LF-tag permissions' button). Below these steps is a section titled 'Data lake administrators (0/2)' with a 'Find administrators' search bar and a table showing two IAM roles: 'db-7aca51db19a316fd5680f43feb179157-lam-role' and 'tejgluerole', both listed as 'IAM role'. At the bottom, there's a 'Database creators' section with a 'Find database creators' search bar and buttons for 'Revoke' and 'Grant'. The URL in the browser bar is <https://us-east-1.console.aws.amazon.com/lakeformation/home?region=us-east-1&default-permission-settings>.

Select your **AWS logged-in IAM user** from the drop down list. For the rest of workshop, the user will be considered as a data lake administrator and will have full access to the data lake. Click on the **Save** button.

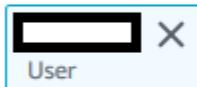
Manage data lake administrators

X

IAM users and roles

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

Choose IAM principals to add



User

Choose up to a maximum of 10 data lake administrators.

Active Directory users and groups (EMR beta only)

Enter one or more Active Directory users or groups.

Ex: arn:aws:iam::<AccountId>:saml-provider/<SamlProviderName>

Cancel

Save

Note: if you did not get the popup then that means the data lake already has an administrator. You can check that by clicking on the “Admins and database creators” menu in the left. If you see that your logged-in IAM username is listed as the “Data Lake Administrator” then you are ok to move to the next step. Otherwise, click on the “Grant” button to add “your AWS logged-in IAM user” as the administrator of the data lake.

After adding the administrator, you will create the database. On the AWS Lake Formation console, click on the **Databases** option on the left menu and then click on **Create database** button.

The screenshot shows the AWS Lake Formation interface. In the left sidebar, under 'Data catalog', the 'Databases' link is highlighted with a red arrow. The main content area is titled 'Databases' and shows a table with one row: 'No databases'. At the bottom right of the table is a 'Create database' button.

On the next screen, enter **tejgluedatabase** as the **Name**. On the **Location box**, select the S3 data lake path as **s3://tej-glue-data-lake/data**. If you created the bucket with different name, then you replace **tej-glue-data-lake** part with that name. Make sure you **Uncheck** the option - **Use only IAM access control for new tables in this database**. Leave rest of the options as default and click on **Create database** button.

The screenshot shows the 'Database details' creation form. It has two radio button options: 'Database' (selected) and 'Resource link'. The 'Name' field contains 'tejgluedatabase'. The 'Location - optional' field contains 's3://tej-glue-data-lake/data'. Under 'Default permissions for newly created tables', there is a checkbox for 'Use only IAM access control for new tables in this database', which is unchecked. At the bottom right is a 'Create database' button. A large red arrow points upwards from the bottom of the form towards the top of the page.

The database is added in no time. Now register the Amazon S3 bucket as your data lake storage. In the navigation pane, choose **Data lake locations**, and then choose **Register location**.

AWS Lake Formation

Data catalog

Tables

Data filters

Settings

Register and ingest

Data lake locations

Blueprints

Crawlers

Jobs

Permissions

Administrative roles and tasks

LF-Tags

LF-tag permissions

Data lake permissions

Data locations

External data filtering

AWS Lake Formation > Data lake locations

Find data lake storage

Amazon S3 path

IAM role

Last modified

No Data lake storage

Register location

Actions

Register location

Enter a path to the existing S3 bucket **s3://tej-glue-data-lake/data**. If you created the bucket with a different name, then you replace **tej-glue-data-lake** part with that name. For the IAM role, select the **AWSServiceRoleForLakeFormationDataAccess** role. Click **Register location** to save it.

Register location

Amazon S3 location

Register an Amazon S3 path as the storage location for your data lake.

Amazon S3 path

Choose an Amazon S3 path for your data lake.

s3://tej-glue-data-lake/data

Browse

Review location permissions - strongly recommended

Registering the selected location may result in your users gaining access to data already at that location. Before registering a location, we recommend that you review existing location permissions on resources in that location.

Review location permissions

IAM role

To add or update data, Lake Formation needs read/write access to the chosen Amazon S3 path. Choose a role that you know has permission to do this, or choose the **AWSServiceRoleForLakeFormationDataAccess** service-linked role. When you register the first Amazon S3 path, the service-linked role and a new inline policy are created on your behalf. Lake Formation adds the first path to the inline policy and attaches it to the service-linked role. When you register subsequent paths, Lake Formation adds the path to the existing policy.

AWSServiceRoleForLakeFormationDataAccess

Do not select the service linked role if you plan to use EMR.

Cancel

Register location

It is time to use crawler to catalog the S3 bucket data in the database.

Configure and Run Crawler

One of the fundamental principle of building the data lake is that every data in the data lake should be catalogued. The catalog is automated using crawlers in AWS Glue. The crawler uses role based authorization to create catalog in the data lake database. You created an IAM Role **tejgluerole** in the earlier task which the crawler will use to create data catalog in the database. You need to assign database permission for this role. After the permission configuration, you will create and run crawler to catalog the data.

Before granting , the permissions , please vie permission to already see , if permissions are assigned .

The screenshot shows the AWS Lake Formation interface. On the left, there's a sidebar with navigation links like Dashboard, Data catalog (with Databases selected), Tables, Data filters, Settings, Register and ingest (with Data lake locations, Blueprints, Crawlers, and Jobs), and Permissions (with Administrative roles and tasks, LF-Tags, LF-tag permissions, Data lake permissions, Data locations, and External data filtering). The main area is titled 'Databases (0/1)' and shows a single entry: Name: tejgluedb, Owner account ID: 133837890951, Shared resource: -, Shared resource owner: -. There's a 'Create database' button at the top right. A context menu is open over the 'tejgluedb' row, with options like Database, Delete, Edit, Edit LF-tags, Create resource link, Permissions (which is expanded to show Grant, Revoke, Verify permissions, and View permissions), and a 'View permissions' option that has a red arrow pointing to it.

The screenshot shows the AWS Lake Formation console. On the left, a navigation sidebar lists various options like Dashboard, Data catalog, Tables, Data filters, Settings, Register and ingest, and Permissions. Under Permissions, the 'Data lake permissions' section is selected. The main content area is titled 'Data permissions for database **tejgluedb** (2)'. It displays a table with two rows, each listing a table ('tejgluedb') and its catalog ID ('133837890951'). The 'Permissions' column shows 'All, Alter, Create table, Describe, Drop' for both entries. The 'Grantable' column shows 'All, Alter, Create table, Describe, Drop' for both entries. The 'RAM Resources' column shows '-' for both entries. At the top right of the table, there are 'C', 'Revoke', and 'Grant' buttons. Below the table, there are navigation arrows and a search bar. The bottom of the screen includes standard AWS footer links: Feedback, English (US), © 2022, Amazon Internet Services Private Ltd. or its affiliates., Privacy, Terms, and Cookie preferences.

Open the AWS Lake Formation console, click on the **Databases** option on the left. You will see **tejgluedatabase** database listed.

Select the **tejgluedatabase** database and click on the **Grant** menu option under the **Action** dropdown menu.

The screenshot shows the AWS Lake Formation console. The left sidebar is identical to the previous screenshot, with the 'Data catalog' section expanded and 'Databases' selected. The main content area is titled 'Databases (0/1)' and shows a single entry: 'tejgluedatabase' with catalog ID '133837890951'. A red arrow points to this row. To the right of the table, there is a context menu with the following options: Actions ▲, View tables, Create database, Database, Delete, Edit, Edit LF-tags, Create resource link, Permissions, Grant, Revoke, Verify permissions, and View permissions. A red arrow points to the 'Grant' option in this menu. The bottom of the screen includes the same footer links as the previous screenshot.

AWS Lake Formation > Grant permissions

Grant data permissions

Principals

IAM users and roles
Users or roles from this AWS account.

SAML users and groups
SAML users and group or QuickSight ARNs.

External accounts
AWS accounts or AWS organizations outside of this account.

IAM users and roles
Add one or more IAM users or roles.
Choose IAM principals to add

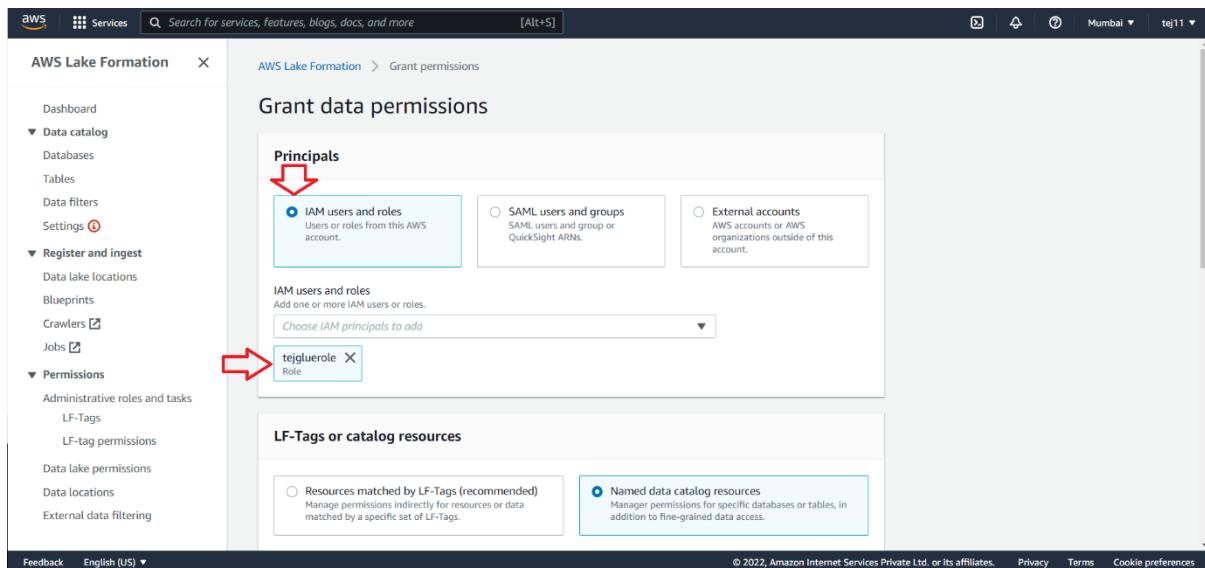
tejgluerole X
Role

LF-Tags or catalog resources

Resources matched by LF-Tags (recommended)
Manage permissions indirectly for resources or data matched by a specific set of LF-Tags.

Named data catalog resources
Manage permissions for specific databases or tables, in addition to fine-grained data access.

Feedback English (US) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences



AWS Lake Formation >

Database permissions

Tables - optional
Select one or more tables.
Choose tables Load more

Data filters - optional
Select one or more data filters.
Choose data filters Load more Create new

Database permissions
Choose specific access permissions to grant.

Create table Alter Drop
 Describe

Grantable permissions
Choose the permission that may be granted to others.

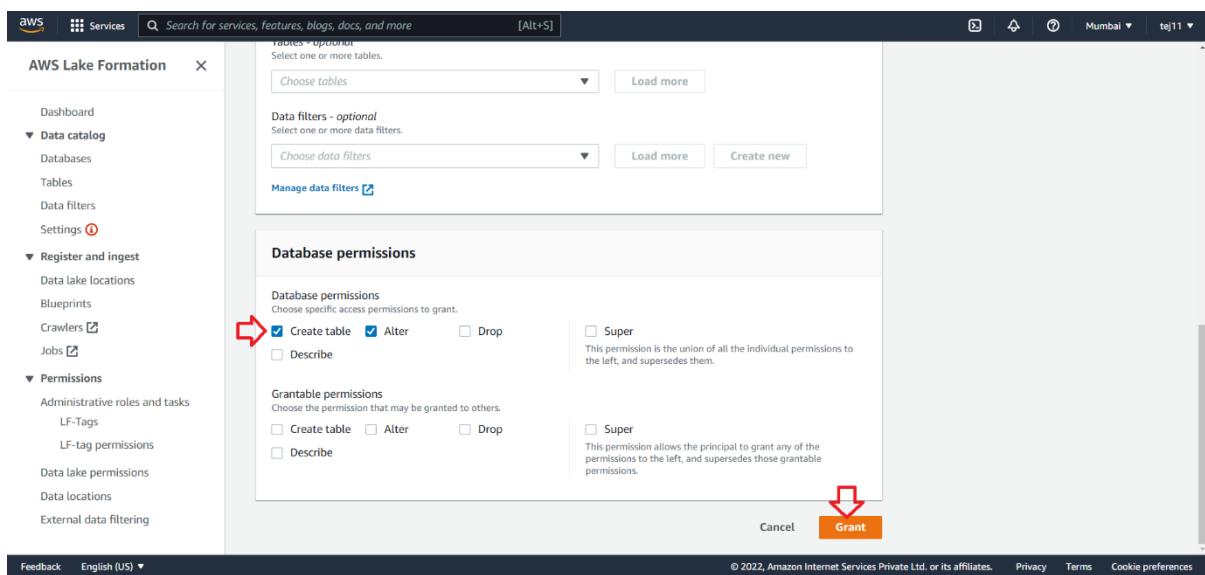
Create table Alter Drop
 Describe

Super
This permission is the union of all the individual permissions to the left, and supersedes them.

Super
This permission allows the principal to grant any of the permissions to the left, and supersedes those grantable permissions.

Cancel **Grant**

Feedback English (US) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences



Screenshot of the AWS Lake Formation Permissions page showing catalog permissions granted to IAM roles.

Grant permissions success
Successfully granted catalog permissions to ARNs: arn:aws:iam::133837890951:role/tejgluerole.

Data permissions (3)

Principal	Principal type	Resource type	Database	Table	Resource	Catalog	LF-tag expression
IAMAllowedPrincipals	Group	Database	tejgluedatabase	-	tejgluedatabase	133837890951	-
root	-	Database	tejgluedatabase	-	tejgluedatabase	133837890951	-
tejgluerole	IAM role	Database	tejgluedatabase	-	tejgluedatabase	133837890951	-

A red arrow points to the "tejgluerole" row in the table.

Screenshot of the AWS Lake Formation Permissions page showing catalog permissions granted to IAM roles.

Grant permissions success
Successfully granted catalog permissions to ARNs: arn:aws:iam::133837890951:role/tejgluerole.

Data permissions (3)

Resource	Table	Catalog	LF-tag expressions	Permissions	Grantable
tejgluedatabase	-	133837890951	-	All	-
tejgluedatabase	-	133837890951	-	All, Alter, Create table, Describe, Drop	All, Alter, Create table, Describe, Drop
tejgluedatabase	-	133837890951	-	Alter, Create table	All, Alter, Create table, Describe, Drop

A red arrow points to the "tejgluedatabase" row in the table.

After assigning permission, time to configure and run crawler. Open the AWS Glue console. Click on the **Crawlers** menu in the left and then click on the **Add crawler** button.

AWS Lake Formation

Grant permissions success
Successfully granted catalog permissions to ARNs: arn:aws:iam::133837890951:role/tejgluerole.

AWS Lake Formation > Permissions

Too many permissions? Filter by database or table. In the navigation page, choose Databases or Tables. Then choose a database or table, and on the Actions menu, choose View Permissions.

Data permissions (3)

Principal	Principal type	Resource type	Database	Table	Resource	Catalog	LF-tag expression
IAMAllowedPrincipals	Group	Database	tejgluedatabase	-	tejgluedatabase	133837890951	-
root	-	Database	tejgluedatabase	-	tejgluedatabase	133837890951	-
tejgluerole	IAM role	Database	tejgluedatabase	-	tejgluedatabase	133837890951	-

Feedback English (US) ▾

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

AWS Glue

AWS Glue

AWS Glue is a fully managed ETL (extract, transform, and load) service that makes it simple and cost-effective to categorize your data, clean it, enrich it, and move it reliably between various data stores.

Get started Getting started guide

Build your AWS Glue Data Catalog

AWS Glue automatically stores metadata in a central data catalog. It can create table definitions for many common data stores, including, S3 buckets, web logs, and AWS databases. AWS Glue recognizes, infers, organizes, and classifies your data.

Generate and edit transformations

PySpark transformation scripts are auto generated using source and target metadata. You can store customized versions to transform your data to meet your business needs. AWS Glue provides an environment to modify your jobs.

Schedule and run your jobs

AWS Glue runs your ETL jobs in a serverless environment. You don't need to set up the infrastructure. You just use Amazon's infrastructure and pay for the resources you use. You can define triggers to run jobs based on a schedule or event. AWS Glue enables you to monitor your jobs.

Feedback English (US) ▾

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

AWS Glue

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.

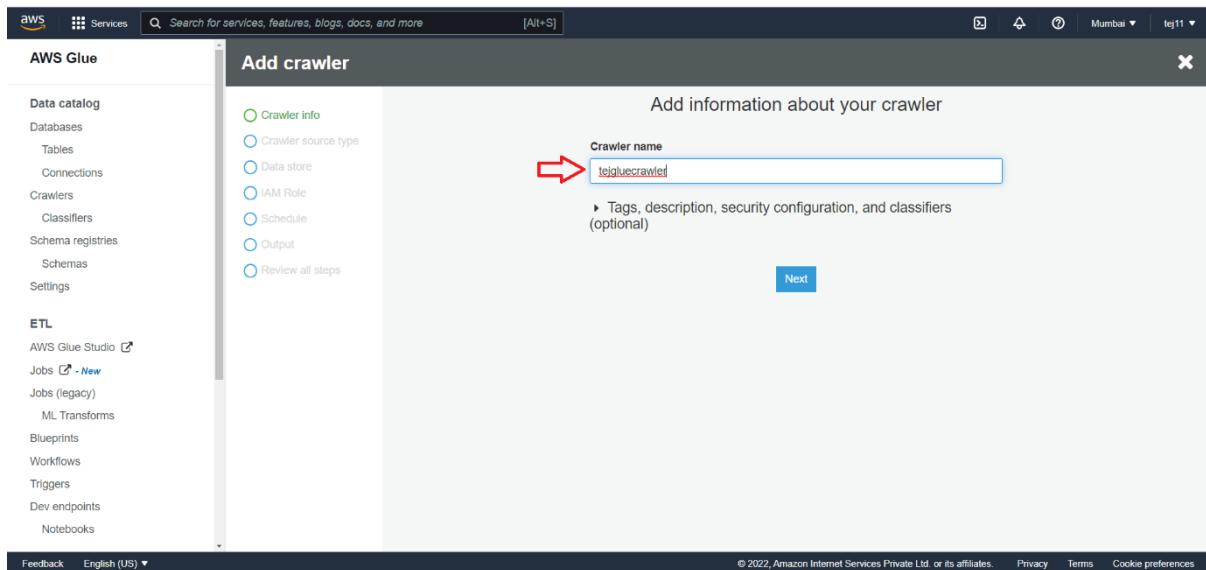
Add crawler Run crawler Action ▾ Filter by tags and attributes User preferences Showing: 0 - 0

Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
You don't have any crawlers yet.							

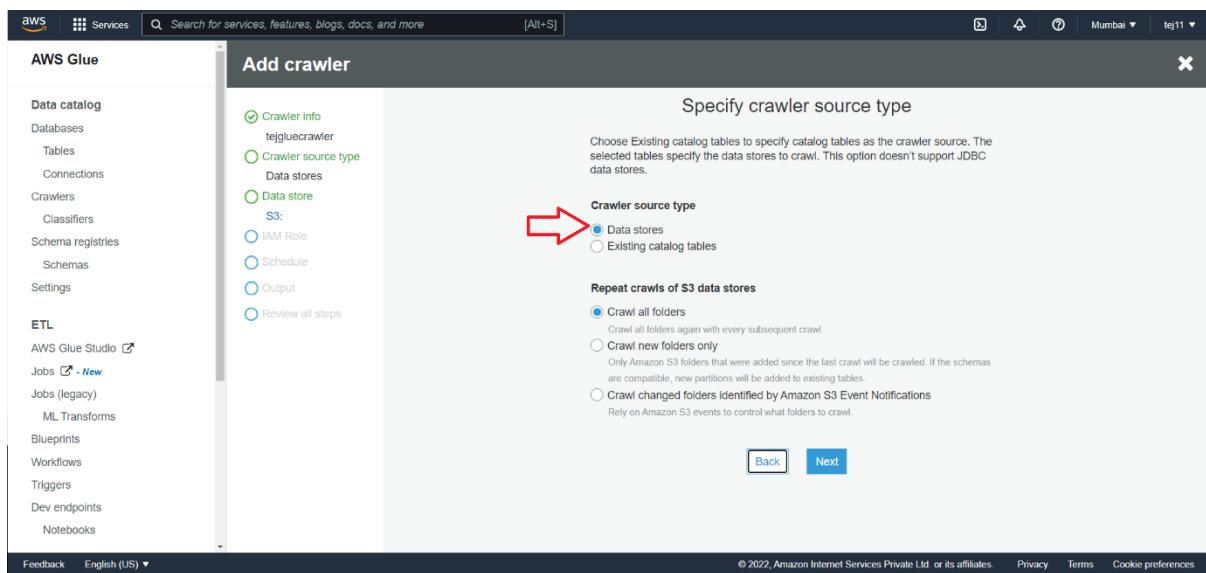
Add crawler

Feedback English (US) ▾

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences



On the next screen, select **Data stores** as the Crawler source type and click **Next**.



Screenshot of the AWS Glue 'Add crawler' wizard Step 1: Choose a data store

The left sidebar shows the AWS Glue navigation menu. The main panel is titled 'Add crawler' and has the sub-section 'Choose a data store'. It includes fields for 'Connection' (set to 'S3'), 'Crawl data in' (set to 's3://tej-glue-data-lake/data'), and 'Include path' (set to 's3://tej-glue-data-lake/data'). A red arrow points to the 'Include path' field.

Screenshot of the AWS Glue 'Add crawler' wizard Step 2: Add another data store

The left sidebar shows the AWS Glue navigation menu. The main panel is titled 'Add crawler' and has the sub-section 'Add another data store'. It includes a radio button group for 'Yes' or 'No' (set to 'No') and 'Back' and 'Next' buttons. A red arrow points to the 'Yes' radio button.

Screenshot of the AWS Glue 'Add crawler' wizard Step 3: Choose an IAM role

The left sidebar shows the AWS Glue navigation menu. The main panel is titled 'Add crawler' and has the sub-section 'Choose an IAM role'. It includes a description of what an IAM role does, a radio button group for 'Update a policy in an IAM role', 'Choose an existing IAM role' (selected), and 'Create an IAM role'. A dropdown menu shows 'tejgluerole' selected. A red arrow points to the 'Choose an existing IAM role' radio button, and another red arrow points to the 'tejgluerole' dropdown entry.

Screenshot of the AWS Glue 'Add crawler' wizard Step 1: Create a schedule for this crawler.

The 'Frequency' dropdown menu is open, showing the following options:

- Run on demand (selected)
- Hourly
- Daily
- Choose days
- Weekly
- Monthly

A red arrow points to the 'Run on demand' option in the dropdown menu.

Screenshot of the AWS Glue 'Add crawler' wizard Step 1: Create a schedule for this crawler.

The 'Frequency' dropdown menu is open, showing the following options:

- Run on demand (selected)
- Hourly
- Daily
- Choose days
- Weekly
- Monthly

Back and Next buttons are visible at the bottom of the dropdown menu.

Screenshot of the AWS Glue 'Add crawler' wizard Step 2: Configure the crawler's output.

The 'Database' dropdown menu is open, showing the following options:

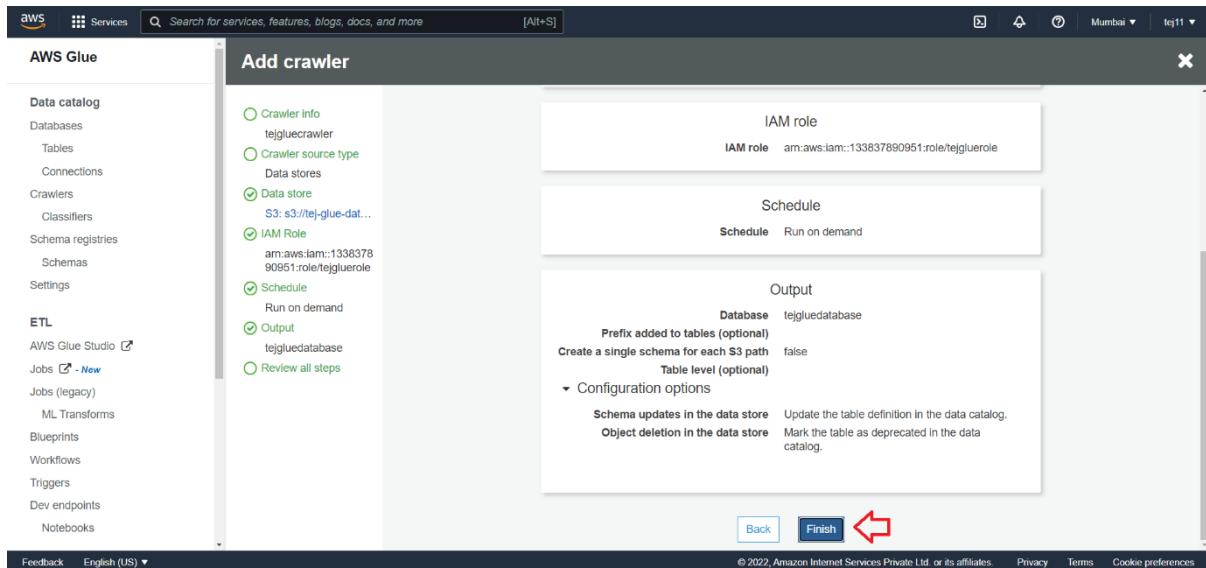
- Choose a database to contain tables
- tejgluedatabase (selected)
- Add database
- Mygluedatabase

A red arrow points to the 'tejgluedatabase' option in the dropdown menu.

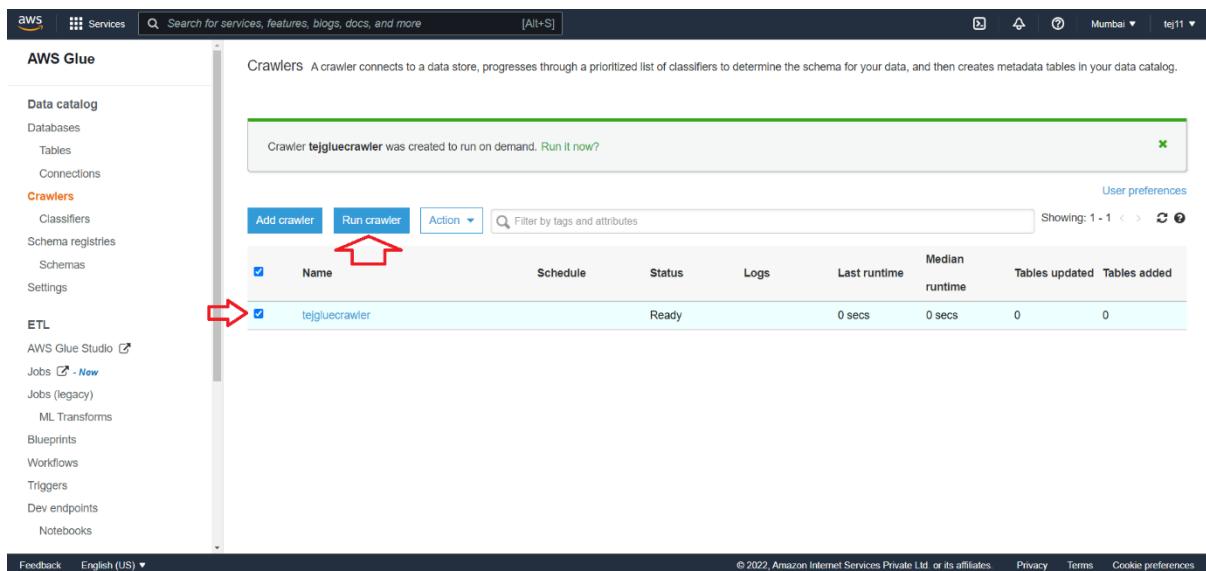
Below the dropdown, there are two sections:

- Prefix added to tables (optional):** A text input field with placeholder text "Type a prefix added to table names".
- Grouping behavior for S3 data (optional):** A collapsed section with a plus sign.
- Configuration options (optional):** A collapsed section with a plus sign.

Back and Next buttons are visible at the bottom of the configuration section.



The crawler is created in no time. Click on **Run it now?** link. Alternatively, you can select the crawler and run the crawler from the **Action** menu.



Screenshot of the AWS Glue Crawler list page. A message at the top says "Crawler 'tejgluecrawler' is now running." The crawler table shows one entry:

Name	Status	Last runtime	Median runtime	Tables updated	Tables added
tejgluecrawler	Starting	0 secs	0 secs	0	0

Red annotations: A red house icon is placed over the 'Starting' status cell, and another red house icon is placed over the 'Starting' status column header.

It may take up to a couple of minutes for the crawler to finish crawling the bucket. You should be able to see a success message that there are two tables created by the crawler in **tejgluedatabase** database. These are two tables for the **sales** and **customers** data in Amazon S3.

Screenshot of the AWS Glue Crawler list page. The crawler table shows the crawler has stopped:

Name	Status	Last runtime	Median runtime	Tables updated	Tables added
tejgluecrawler	Stopping	48 secs	48 secs	0	2

Red annotations: A red house icon is placed over the 'Stopping' status cell, and another red house icon is placed over the 'Stopping' status column header. Red text "after it finishes , it stops ." is overlaid on the page.

The screenshot shows the AWS Glue console. On the left, there's a navigation sidebar with sections like Data catalog, Crawlers (which is selected), ETL, and AWS Glue Studio. The main area displays a message: "Crawler 'tejgluecrawler' completed and made the following changes: 2 tables created, 0 tables updated. See the tables created in database **tejgluedatabase**." Below this is a table with one row:

	Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
	tejgluecrawler		Ready	Logs	48 secs	48 secs	0	2

At the bottom, there are links for Feedback, English (US), and a copyright notice: © 2022, Amazon Internet Services Private Ltd. or its affiliates.

It may take up to a couple of minutes for the crawler to finish crawling the bucket. You should be able to see a success message that there are two tables created by the crawler in **tejgluedatabase** database. These are two tables for the **sales** and **customers** data in Amazon S3.

Go back to the AWS Lake Formation console, click on the **Tables** menu in the left. You can see two tables **sales** and **customers** created.

The screenshot shows the AWS Lake Formation console with the "Data catalog settings" page open. The left sidebar has sections like Dashboard, Data catalog (which is selected), Databases, Tables, Data filters, Settings (with a red arrow pointing to it), Register and ingest, and Permissions. The main area shows "Default permissions for newly created databases and tables" with two checkboxes:
 Use only IAM access control for new databases
 Use only IAM access control for new tables in new databases

A red annotation with the text "uncheck those 2 if tables not created." is placed over these checkboxes. Below this is another section: "Default permissions for AWS CloudTrail". It includes a "Resource owners" field with a placeholder "Enter an AWS account ID" and a note: "Enter one or more AWS account IDs. Press Enter after each ID." At the bottom, there's an error message: "Your account is not a member of an organization." with a close button. There are "Cancel" and "Save" buttons at the bottom right.

Role Permission to the Catalog

, but datalake is experiencing a bug , which will be rectified soon . if you are not able to see the tables .

6

Create Developer Endpoint

Developer endpoint provides development environment to create Glue Job using languages and frameworks like PySpark. In this task, you create a developer endpoint which you will use to code with PySpark.

1. Goto the AWS Glue console, click on the **Dev endpoints** option in the left menu and then click on the **Add endpoint** button.

The screenshot shows the AWS Glue Dev endpoints page. On the left sidebar, under the ETL section, the 'Dev endpoints' option is selected and highlighted with a red arrow. The main content area displays a message about development endpoints and a table with one row. The 'Add endpoint' button at the bottom left of the table is also highlighted with a red arrow.

The screenshot shows the 'Add development endpoint' wizard, Step 1: Set up your development endpoint. The 'Development endpoint name' field contains 'tejendpoint' and the 'IAM role' dropdown contains 'tejgluerole'. Both of these fields are highlighted with red arrows. The sidebar on the left is identical to the previous screenshot.

On the next screen, select **Skip networking information** as the option and click on the **Next** button.

On the next **Add an SSH public key (Optional)** screen, click on the **Next** button.

On the next **Review** screen, click on the **Finish** button. The endpoint creation will start.

The screenshot shows the 'Add development endpoint' wizard in AWS Glue Studio. The left sidebar lists ETL, Security, Tutorials, and Dev endpoints. The main panel is titled 'Review' and contains three sections: 'Dev endpoint properties' (Name: tejendpoint, Tags: -), 'Networking' (VPC, Subnet, Security groups), and 'SSH public key' (Public key contents). At the bottom are 'Back' and 'Finish' buttons, with a red arrow pointing to 'Finish'.

It will take some 8-10 mins for the developer endpoint to be ready. Wait till the status changes to **READY**.

The screenshot shows the 'Dev endpoints' list in AWS Glue Studio. The left sidebar shows 'Dev endpoints' selected. The main area displays a message about billing and provisioning, followed by a table with one row for 'tejendpoint'. The table columns are 'Endpoint name' (tejendpoint), 'Provisioning status' (PROVISIONING), and 'Running for'. A red arrow points to the 'tejendpoint' row. The URL at the bottom is https://ap-south-1.console.aws.amazon.com/glue/home?region=ap-south-1#etltab=triggers.

Endpoint name	Provisioning status	Running for
tejendpoint	PROVISIONING	

Screenshot of the AWS Glue Dev endpoints page showing a single endpoint named "tejendpoint" in the READY state.

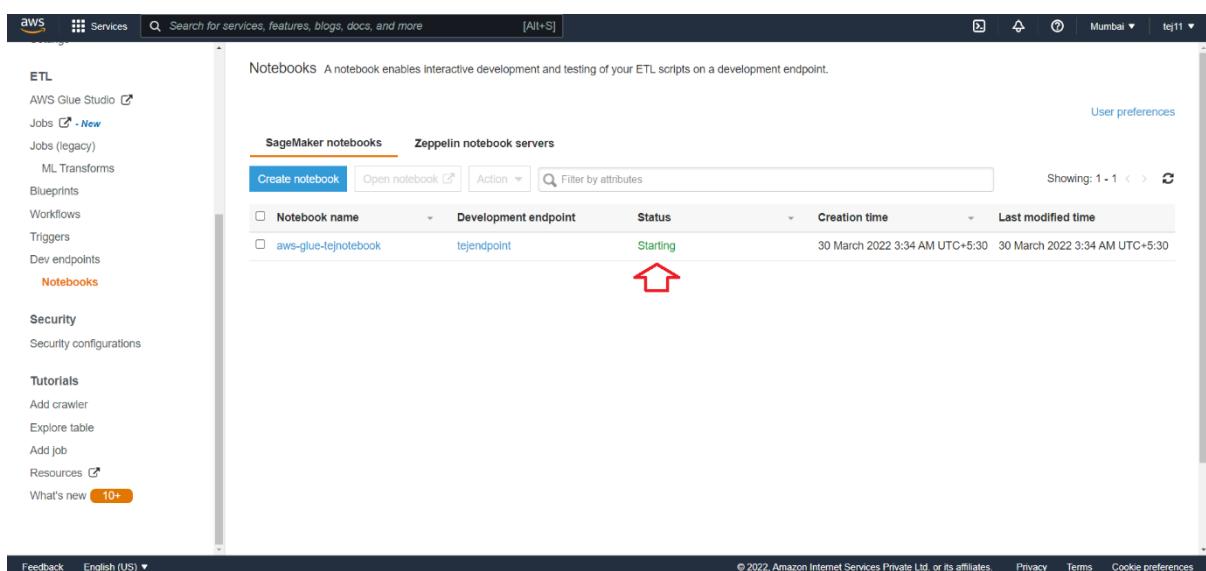
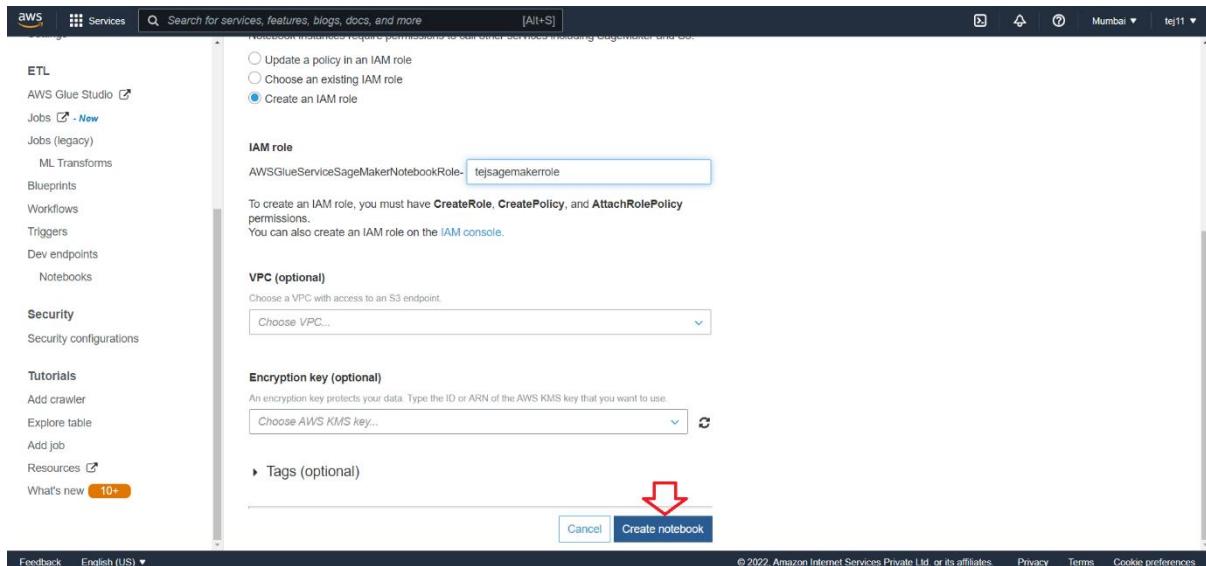
The screenshot shows the AWS Glue Studio interface with the "Dev endpoints" section selected. A red arrow points to the "READY" status of the "tejendpoint".

Screenshot of the AWS Glue Dev endpoints page showing the "Action" dropdown menu for the "tejendpoint" endpoint.

The "Action" dropdown menu is open, showing options: "Create Zeppelin notebook server", "Create SageMaker notebook", "Update ETL libraries", "Rotate SSH key", and "Delete". A red arrow points to the "Create SageMaker notebook" option.

Screenshot of the "Create and configure a notebook" wizard for creating a SageMaker notebook.

The "Notebook name" field contains "aws-glue-tejnotebook" (highlighted by a red arrow). The "Attach to development endpoint" dropdown is set to "tejendpoint" (highlighted by a red arrow). The "IAM role" dropdown is set to "AWSGlueServiceSageMakerNotebookRole-tejagemakerrole" (highlighted by a red arrow).



The development environment is ready. Let's do PySpark programming in notebook which then you use later to create a Glue job.

Glue notebook job

```
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

glueContext = GlueContext(SparkContext.getOrCreate())

salesDF = glueContext.create_dynamic_frame.from_catalog(
    database="tejgluedb",
    table_name="sales")

customerDF = glueContext.create_dynamic_frame.from_catalog(
    database="tejgluedb",
    table_name="customers")

salesDF.printSchema()
customerDF.printSchema()

customersalesDF=Join.apply(salesDF, customerDF, 'customerid', 'customerid')

customersalesDF.printSchema()

customersalesDF = customersalesDF.drop_fields(['`customerid`'])

customersalesDF.printSchema()

glueContext.write_dynamic_frame.from_options(customersalesDF, connection_type =
"s3", connection_options = {"path": "s3://tej-glue-data-lake/data/customersales"}, format = "json")
```

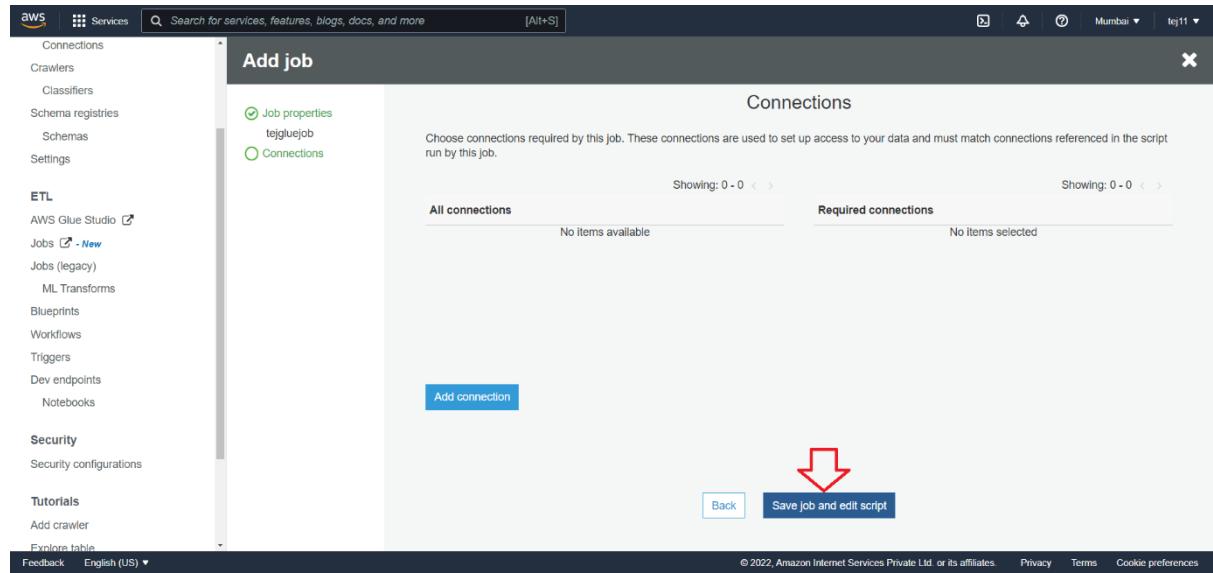
Glue job

The screenshot shows the AWS Glue Studio interface. On the left, a sidebar menu includes sections for Connections, Crawlers, Classifiers, Schema registries, Schemas, Settings, ETL (AWS Glue Studio, Jobs - New, Jobs (legacy), ML Transforms, Blueprints, Workflows, Triggers, Dev endpoints, Notebooks), Security (Security configurations), Tutorials (Add crawler), and Explore table. The main content area is titled "Jobs" and describes them as "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." It features a search bar, an "Add job" button, and a table header with columns: Name, Type, ETL language, Script location, Last modified, Job bookmark. A message at the bottom states "You don't have any jobs defined yet." and includes an "Add job" button.

The screenshot shows the "Add job" configuration dialog. The left sidebar is identical to the previous screenshot. The main dialog is titled "Configure the job properties". It has two tabs: "Job properties" (selected) and "Connections". Under "Job properties", fields include: "Name" (tejgluejob), "IAM role" (tejgluerole), "Type" (Spark), "Glue version" (Spark 2.4, Python 3 (Glue Version 2.0)), "This job runs" (radio button selected for "A new script to be authored by you"), and "Script file name" (tejgluejob). Red arrows point to the "Name", "IAM role", "Type", "Glue version", and "Script file name" fields.

The screenshot shows the "Add job" configuration dialog with the "Advanced properties" section expanded. The left sidebar is identical. The main dialog now includes sections for "Side version" (Spark 2.4, Python 3 (Glue Version 2.0)), "This job runs" (radio button selected for "A new script to be authored by you"), "Script file name" (tejgluejob), "S3 path where the script is stored" (s3://tej-glue-data-lake/script), and "Temporary directory" (s3://tej-glue-data-lake/script). Red arrows point to the "Script file name", "S3 path where the script is stored", and "Temporary directory" fields. Other collapsed sections include "Monitoring options", "Tags (optional)", and "Security configuration, script libraries, and job parameters".

On the next **Connections** page, click on the **Save job and edit script** button. You come to the page where you can write your code. It is the same code you wrote in the previous task in the notebook.



In the editor

```
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
```

```
glueContext = GlueContext(SparkContext.getOrCreate())
```

```
salesDF = glueContext.create_dynamic_frame.from_catalog(  
    database="tejgluedb",  
    table_name="sales")  
  
customerDF = glueContext.create_dynamic_frame.from_catalog(
```

```
database="tejgluedb",
table_name="customers")
```

```
customersalesDF=Join.apply(salesDF, customerDF, 'customerid', 'customerid')
```

```
customersalesDF = customersalesDF.drop_fields(['`customerid`'])
```

```
glueContext.write_dynamic_frame.from_options(customersalesDF, connection_type =
"s3", connection_options = {"path": "s3://tej-glue-data-
lake/data/customersalesjobsscript"}, format = "json")
```

```
3 from awsglue.utils import getResolvedOptions
4 from pyspark.context import SparkContext
5 from awsglue.context import GlueContext
6 from awsglue.job import Job
7
8 glueContext = GlueContext(SparkContext.getOrCreate())
9
10 salesDF = glueContext.create_dynamic_frame.from_catalog(
11     database="tejgluedb",
12     table_name="sales",
13     transformation_ctx="salesDF")
14 customerDF = glueContext.create_dynamic_frame.from_catalog(
15     database="tejgluedb",
16     table_name="customers")
17 customersalesDF=Join.apply(salesDF, customerDF, 'customerid', 'customerid')
18 customersalesDF = customersalesDF.drop_fields(['`customerid`'])
19
20 glueContext.write_dynamic_frame.from_options(customersalesDF, connection_type = "s3", connection_options = {"path": "s3://tej-glue-data-lake/data/customersalesjobsscript"}, format = "json")
```

Screenshot of AWS Glue Studio Job Editor:

The job is named "tejgluejob". A message states: "The diagram cannot be generated. Check the annotations in your script." The code editor shows the following Python script:

```
1 import sys
2 from awsglue.transforms import *
3 from awsglue.utils import getResolvedOptions
4 from pyspark.context import SparkContext
5 from awsglue.context import GlueContext
6 from awsglue.job import Job
7
8 glueContext = GlueContext(SparkContext.getOrCreate())
9
10 salesDF = glueContext.create_dynamic_frame.from_catalog(
11     database="tejgluedb",
12     table_name="sales")
13 customerDF = glueContext.create_dynamic_frame.from_catalog(
14     database="tejgluedb",
15     table_name="customers")
16
17 customersalesDF=join.apply(salesDF, customerDF, 'customerid', 'customerid')
18 customersalesDF = customersalesDF.drop_fields(['customerid'])
19
20 glueContext.write_dynamic_frame.from_options(customersalesDF, connection_type = "s3", connection_options = {"path": "s3://tej-glue-data"}, format = "json")
```

Logs and Schema tabs are visible at the bottom.

Screenshot of AWS Glue Studio Jobs Overview:

The table lists the job "tejglue" with the following details:

Name	Type	ETL language	Script location	Last modified	Job bookmark
tejglue	Spark	python	s3://tej-glue-data...	30 March 2022 3:57 AM ...	Disable

Action dropdown menu options include: Run job, Stop job run, Choose job triggers, Delete, Edit job, Edit script, Reset job bookmark, and Create development endpoint.

History, Details, Script, and Metrics tabs are present at the bottom.

Screenshot of the AWS Glue Studio Jobs (legacy) interface showing the parameters for the job "tejgluejob".

The "Parameters (optional)" dialog is open, listing:

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

A red arrow points to the "Run job" button.

The main interface shows a table of runs:

Run ID	Retry attempt status	Error	Output Logs	Error logs	Glue version	Maximum capacity	Triggered by	Start time	End time	Start-up time	Execution time	Timeout Delay	Job run input
No job runs found.													

Feedback English (US) ▾

Screenshot of the AWS Glue Studio Jobs (legacy) interface showing the details of the job "tejgluejob".

The "Details" tab is selected, displaying:

- Name: tejgluejob
- Type: Spark
- ETL language: python
- Script location: s3://tej-glue-da...
- Last modified: 30 March 2022 3:57 AM ...
- Job bookmark: Disable

Red arrows point to the "Run ID", "Logs", "Start-up time", "Execution time", and "Timeout Delay" columns in the run history table.

The run history table shows one entry:

Run ID	Retry attempt status	Error	Output Logs	Error logs	Glue version	Maximum capacity	Triggered by	Start time	End time	Start-up time	Execution time	Timeout Delay	Job run input
jr_2c11f7271f...	Running		Logs	Error logs	2.0	10		30 ...		0 secs	37 secs	2880 mins	s3://tej-glue-d...

Feedback English (US) ▾

S | Services | Search for services, features, blogs, docs, and more [Alt+S] | Global ▾ | tej11 ▾

Amazon S3 X

Buckets

- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- Access analyzer for S3

Block Public Access settings for this account

▼ Storage Lens

- Dashboards
- AWS Organizations settings

Feature spotlight ⓘ

► AWS Marketplace for S3

Objects / Properties

Objects (4)

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](#)

Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Find objects by prefix

Name	Type	Last modified	Size	Storage class
customers/	Folder	-	-	-
customersales/	Folder	-	-	-
customersalesjobscript/	Folder	-	-	-
sales/	Folder	-	-	-

Feedback English (US) ▾ © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

