# Free Trial Account

<https://signin.aws.amazon.com/signup?request_type=register>

After signup, you will be redirected to **AWS management Console**

## Login Credentials

<https://aws.amazon.com/console/>

root email: [khalid0142759@gmail.com](mailto:khalid0142759@gmail.com)

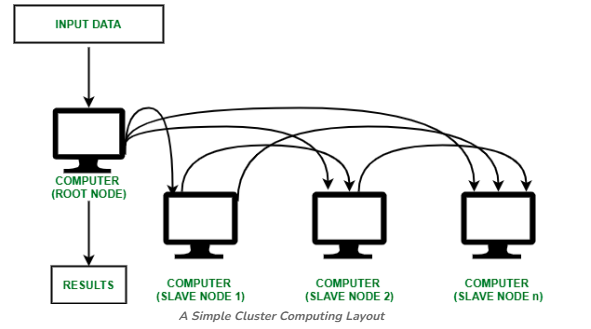
password: Shahela@24oct

Account ID: 442042545001

# What is cluster

A cluster in cloud computing is a group of servers or applications that work together to achieve a common goal. It is a collection of tightly or loosely connected computers that work together so that they act as a single entity.

### Architecture



### How do clusters work?

* **Nodes**: Each computer in a cluster is called a node.
* **Network**: Nodes are connected over a **fast** local area network (LAN) or wide area network (WAN).
* **Task distribution**: Clusters can distribute traffic across nodes.
* **Failover**: If one node fails, the application can continue running on the other nodes.

### ****Types of Cluster****

1. High performance (HP) clusters   
HP clusters use computer clusters and supercomputers to solve advance computational problems. They are used to performing functions that need nodes to communicate as they perform their jobs. They are designed to take benefit of the parallel processing power of several nodes.

#### ****2. Load-balancing clusters****

Incoming requests are distributed for resources among several nodes running similar programs or having similar content. This prevents any single node from receiving a disproportionate amount of task. This type of distribution is generally used in a web-hosting environment.

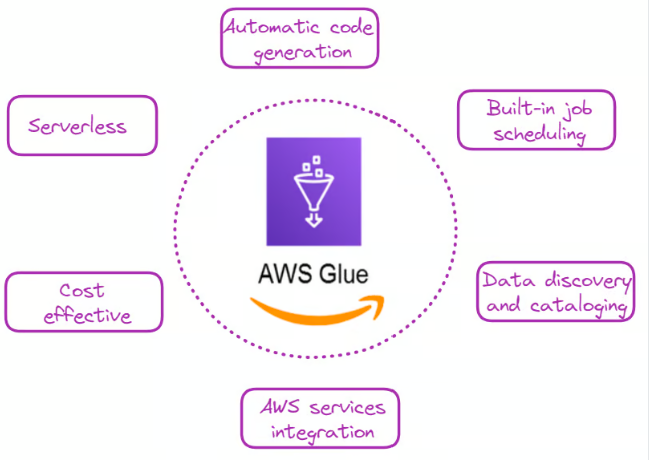
#### ****3. High Availability (HA) Clusters****

HA clusters are designed to maintain redundant nodes that can act as backup systems in case any failure occurs. Consistent computing services like business activities, complicated databases, customer services like e-websites and network file distribution are provided. They are designed to give uninterrupted data availability to the customers.

# AWS Glue

**AWS Glue is a fully managed ETL service** that makes it simple and cost-effective to categorize our data, clean it, enrich it, and move it reliably between various data stores. AWS Glue is used to prepare data from different sources and prepare that data for analytics, machine learning, and application development. It will reduce the manual effort by performing the automation of the jobs like data integration, data transformation, and data loading.

* AWS Glue is **a serverless data integration service** that makes it easy for analytics users to discover, prepare, move, and integrate data from multiple sources.
* **Automatic code generation:**AWS Glue can automatically generate Python or Scala code for ETL jobs.
* **Built-in job scheduling and monitoring:** It also includes authoring, running jobs, and implementing business workflows.
* **Data discovery and cataloging**: AWS Glue crawlers can automatically discover, catalog, and prepare data for analysis.
* connect to more than 70 diverse data sources and manage your data in a centralized data catalog.
* You can visually create, run, and monitor extract, transform, and load (ETL) pipelines to load data into your data lakes.
* **Integration with other AWS services:**you can immediately search and query cataloged data using Amazon Athena, Amazon EMR, and Amazon Redshift Spectrum.
* It integrates with AWS analytics services and Amazon S3 data lakes.
* **Apache spark** comes as inbuilt with glue



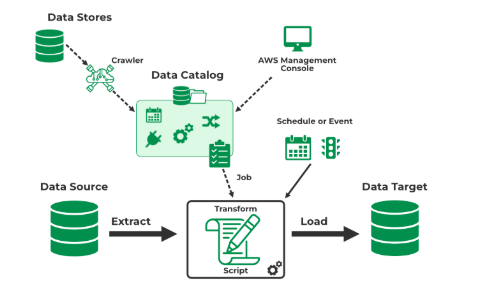
### AWS Glue Studio

AWS Glue Studio is a graphical interface that makes it easy to create, run, and monitor data integration jobs in AWS Glue. You can visually compose data transformation workflows and seamlessly run them on the Apache Spark–based serverless ETL engine in AWS Glue.

With AWS Glue Studio, you can create and manage jobs that gather, transform, and clean data. You can also use AWS Glue Studio to troubleshoot and edit job scripts.

### Architecture

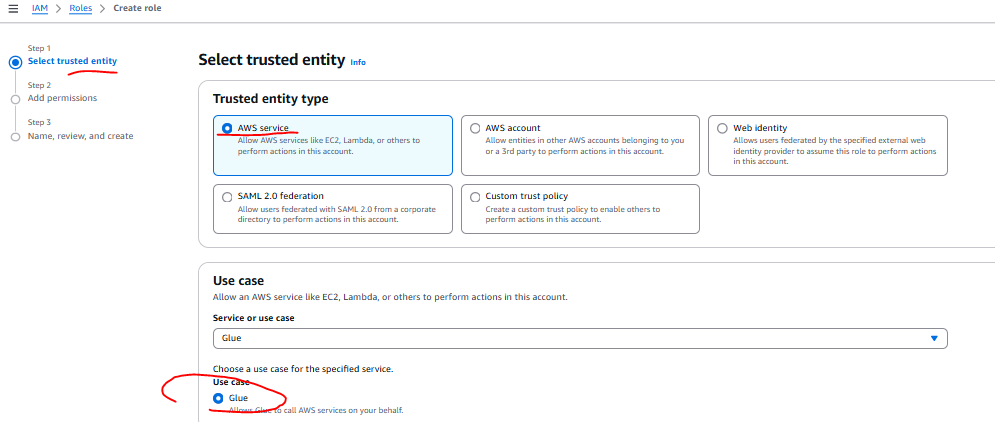
It is ETL Tool



### Set up AWS glue

<https://www.datacamp.com/tutorial/aws-glue>

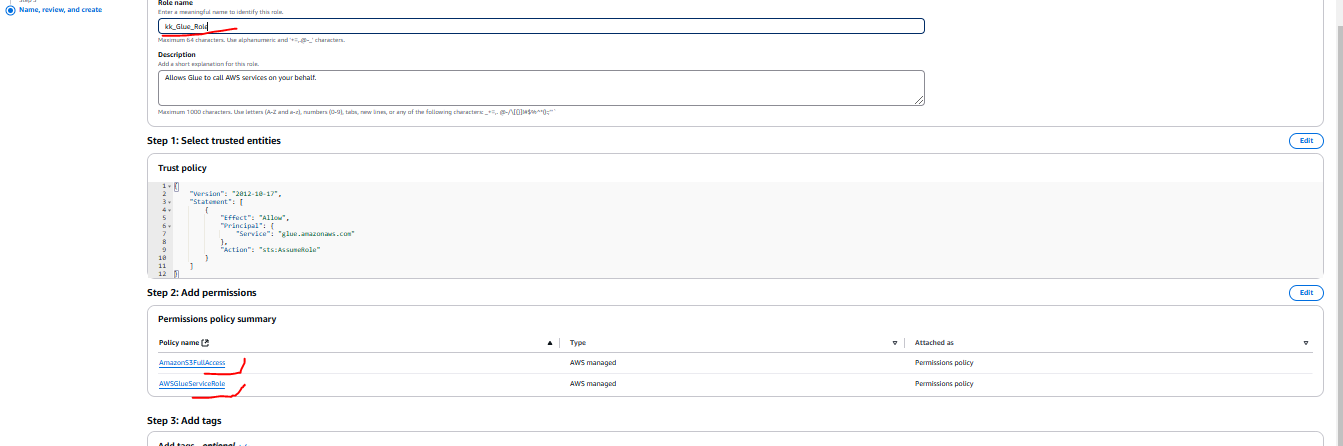
**Step1: set up IAM role**



**Step2: Select below policies**

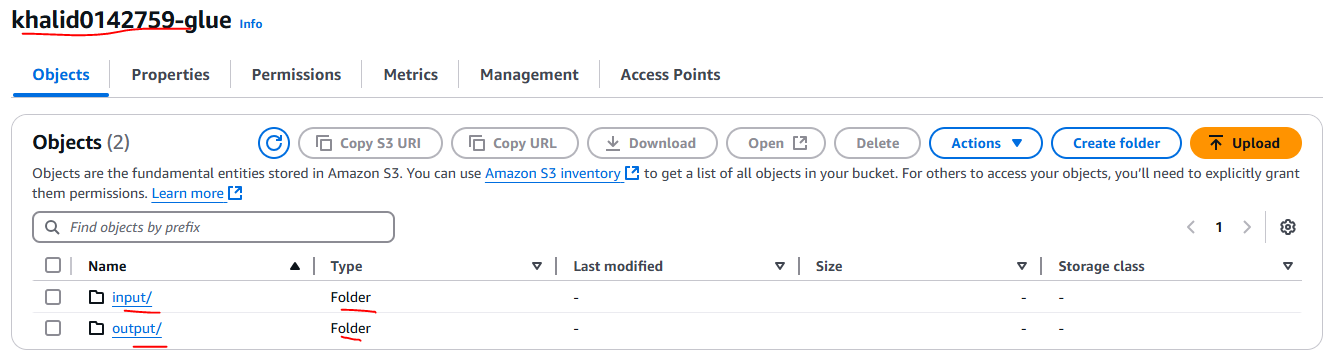
AWSGlueServiceRole

AmazonS3FullAccess-- you should create a more restrictive policy



**Step3: S3 bucket setup**

We'll store both input CSV files and output Parquet files on Amazon S3.



**Step4: Creating a Glue crawler**

A Glue crawler is used to discover and catalog our data automatically. This section explains how to create a crawler that scans all the input CSV files.

1. Go to the AWS Glue service in the AWS Management Console.
2. In the left sidebar, under "Data catalog," click on "Crawlers."
3. Click "Create crawler".
4. Name your crawler (e.g., "CSV-to-Parquet-Crawler") and leave the description field empty, then click "Next."
5. We need to specify “S3” as the data source, which is done by choosing "Add data source " as the crawler source type and clicking "Next."
6. Choose "S3" as the data store and specify your S3 bucket path, which you can do by selecting “Browse S3.” Then click "Next."
7. Select “Crawl all sub-folders”
8. Select “Add an S3 data source”, then click "Next."
9. Before moving further, check the “S3” data type, then “Next.”

### Creating and running a Glue job

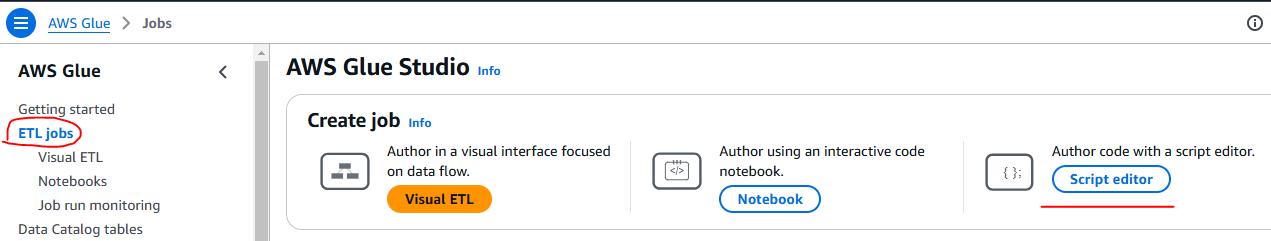
There are three way of creating glue job

1. Visual ETL
2. Python script
3. Interactive Notebook

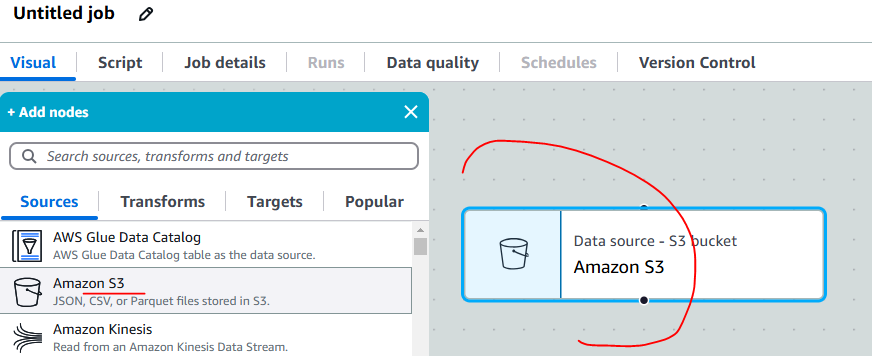
* Go to "ETL Jobs" in the AWS Glue console in the left sidebar.
* Click on Script Editor

[**https://www.youtube.com/watch?v=xPQYE865NmY**](https://www.youtube.com/watch?v=xPQYE865NmY)

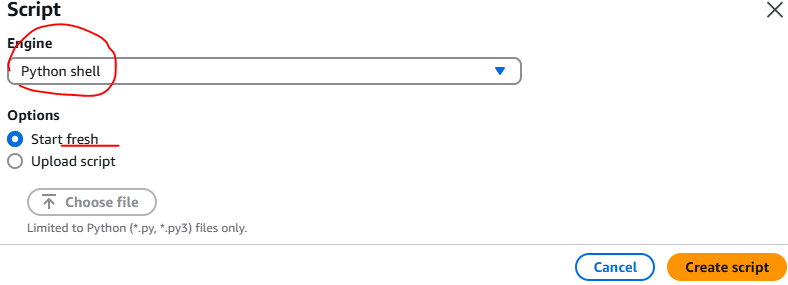
**Step1: Click on “Visual ETL”**

****

**Step2: Click on “Amazon S3” from sources**

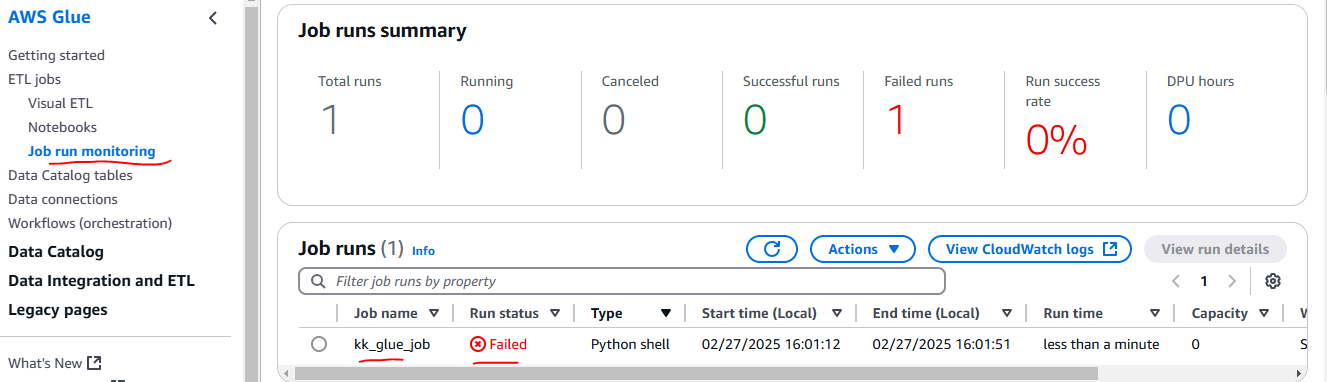
****

Select python shell as engine as below

****

**This is incomplete as of now. Explore further**

**We can monitor job as below**

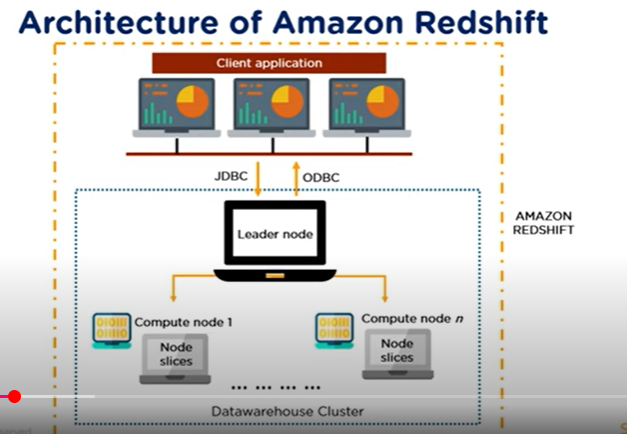
****

# AWS Redshift

In AWS, the primary data warehousing solution is **Amazon Redshift**. It is a **fully managed, cloud-based data warehouse** designed for **high-performance analytics** on large datasets. Amazon Redshift has a commercial license and is a part of Amazon’s web services. It handles large-scale of data and is known for its scalability. It does parallel processing of multiple data.

<https://www.youtube.com/watch?v=7bfOllAyxlg&t=639s>

### Architecture

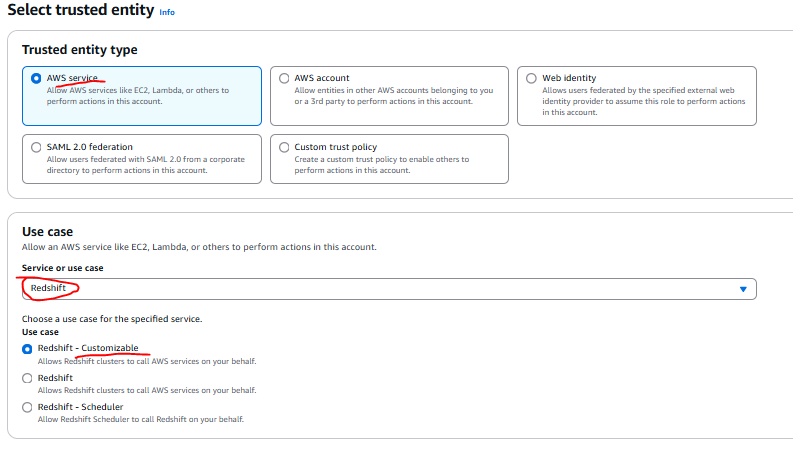


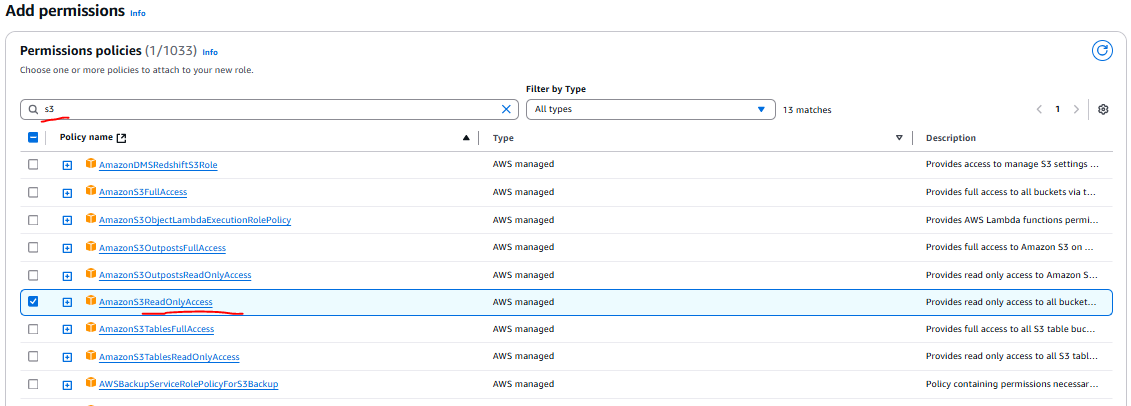
### Key Feature

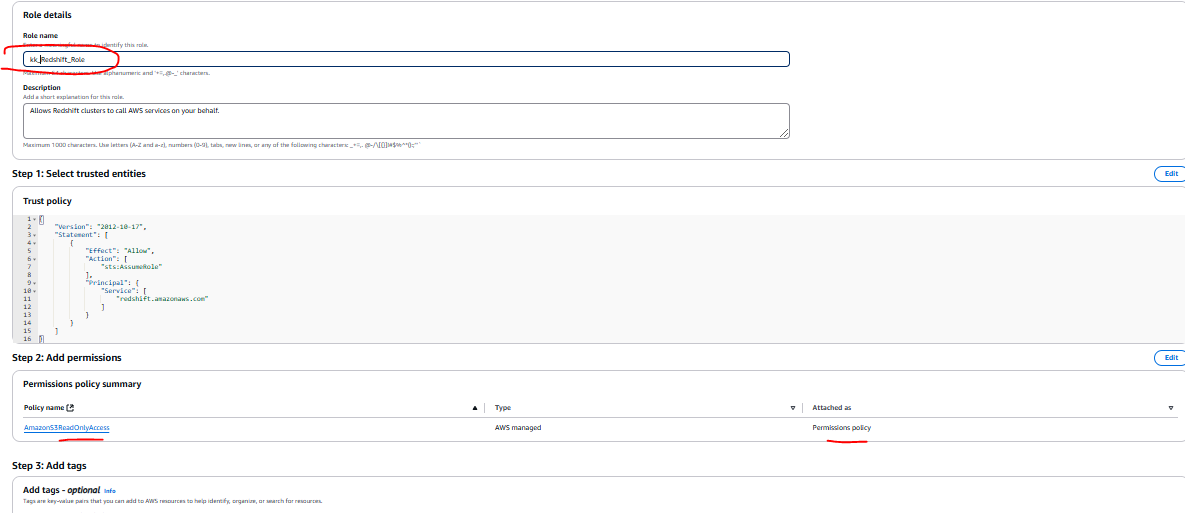
* Column storage—data storage in form of column, helps in optimizing query performance and
* Compression—column level operation which compress storage and decrease size

### How to create Redshift cluster

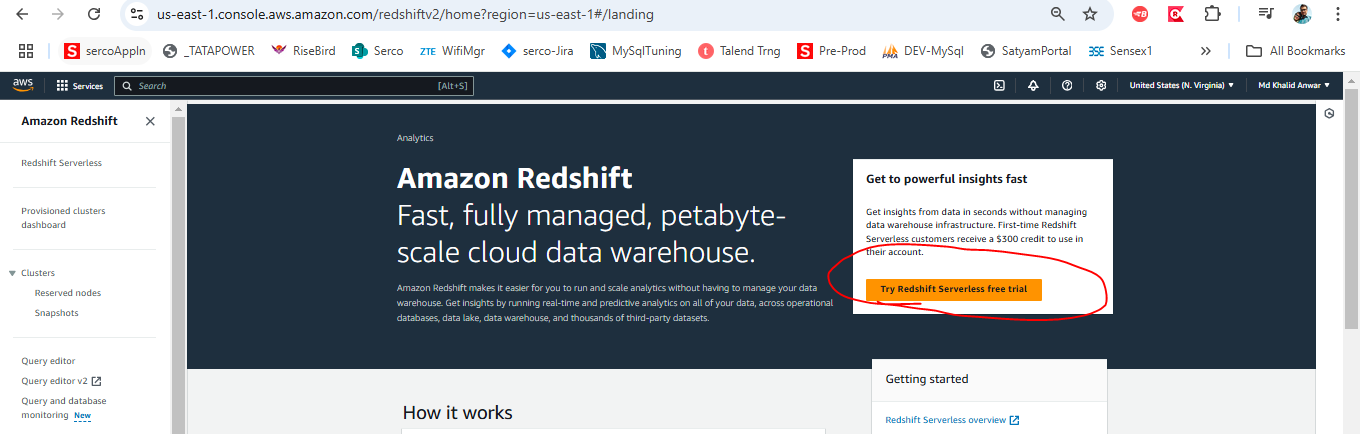
### Step1: Create role for redshift





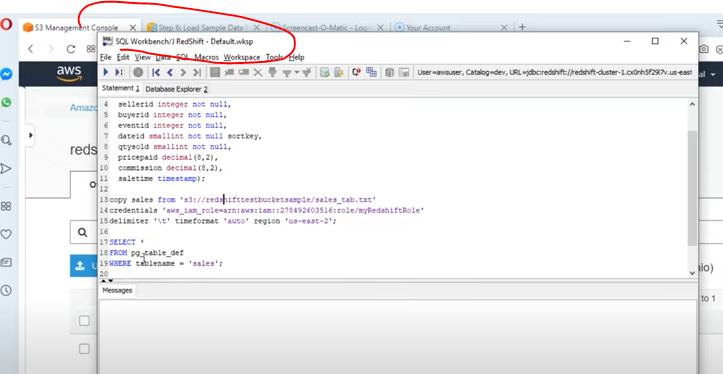


### Step2: Create Redshift cluster



### Access Redshift database

Install SQL workbench/Redshift as below



# AWS Athena

AWS Athena is a powerful, serverless query service that enables you to analyze data directly in Amazon S3 using standard SQL without complex ETL processes or infrastructure management.

* AWS Athena is its serverless architecture, which eliminates the need to manage underlying infrastructure
* no provisioning, scaling, or server management is required.
* Athena automatically scales to accommodate any data size, from gigabytes to petabytes, without manual intervention.
* Athena’s serverless architecture is also cost-effective, with no upfront costs—users only pay for the queries they run.

## Support for standard SQL

Athena supports standard SQL, allowing you to query data in S3. Athena's SQL engine is based on **Presto**, an open-source distributed SQL query engine that provides powerful and flexible querying capabilities, including support for complex joins, window functions, and array and map data types.

## Support for various data formats

Athena can query data stored in various formats, including **CSV**, **JSON**, **Avro**, **Parquet**, and **ORC**. By supporting columnar formats like Parquet and ORC, Athena optimizes query performance and cost by scanning only the necessary columns, reducing the amount of data processed.

## Data Partitioning

It runs queries in parallel by default, enabling it to process large-scale queries efficiently. Athena allows you to partition your data in S3, dramatically improving query performance and reducing costs. By organizing your data into partitions (e.g., by date or region), Athena can scan only the relevant portions of your dataset, minimizing the amount of data processed.

## Business intelligence reporting

Athena is also commonly used as part of a business intelligence (BI) stack, where it integrates with BI tools like **Amazon QuickSight** to enable data visualization and reporting.

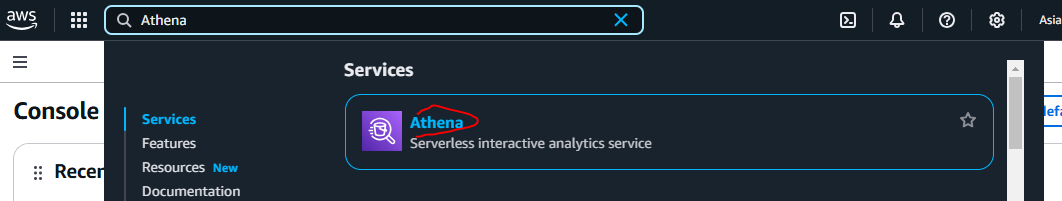
## Big Picture



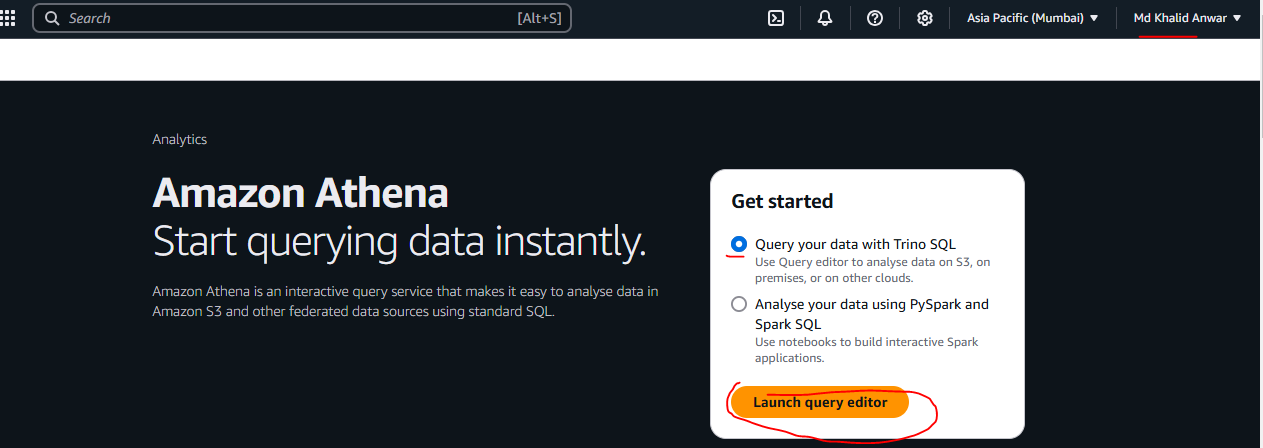
## Setting Up AWS Athena

<https://www.youtube.com/watch?v=fXO1YUOZ7vI>

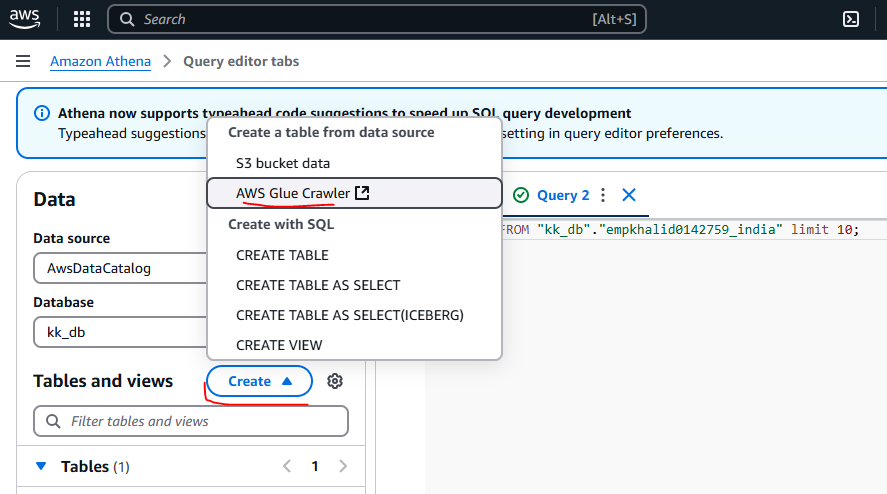
Step1: Search **Athena** as below



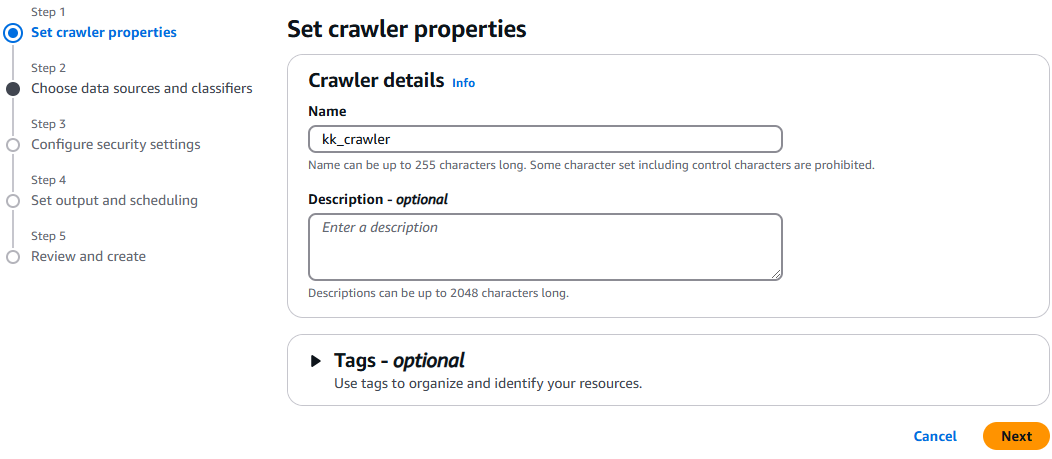
Step2: Click on “Launch Query Editor”



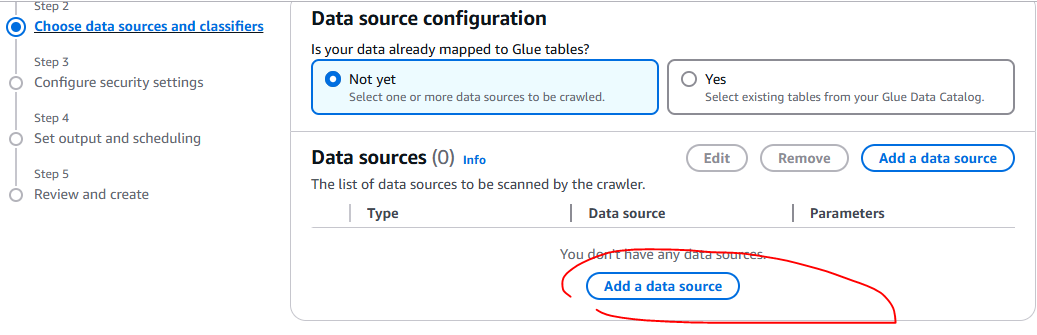
Click on “Create” under table and view section



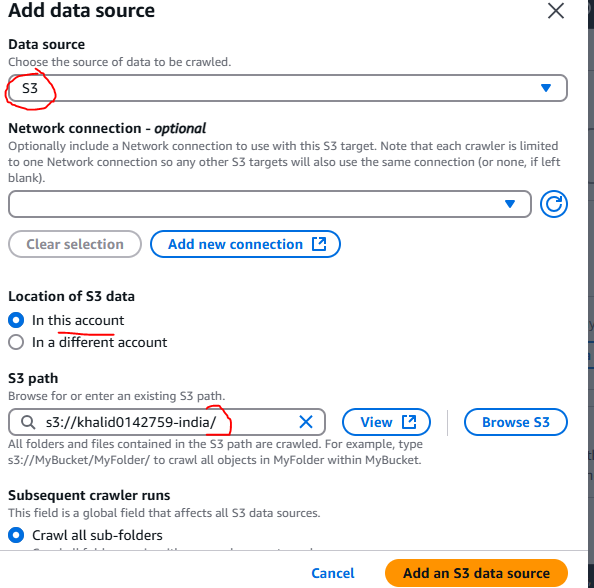
**Enter crawler Name**

****

**Click on data source**

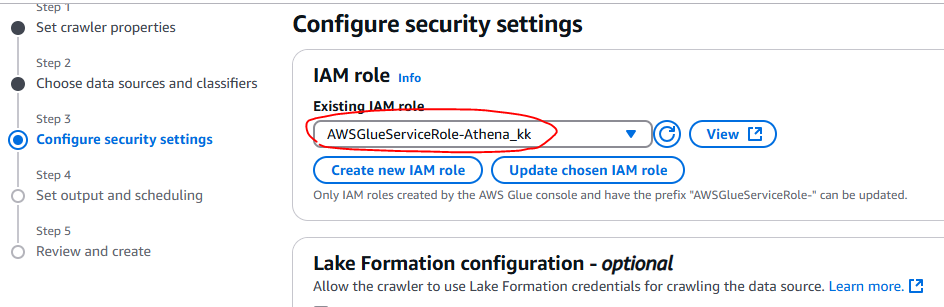
****

**Select Bucket name and ends with /**

****

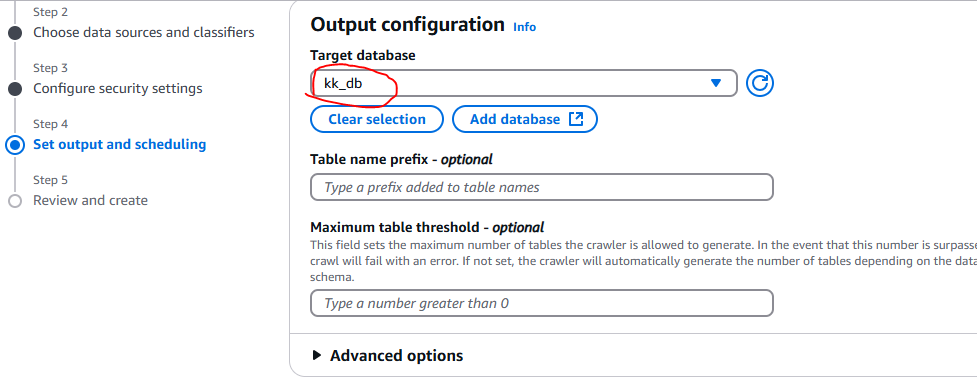
**Select IAM role which we already created**

**Or Enter new RoleName which will be created by this wizard**

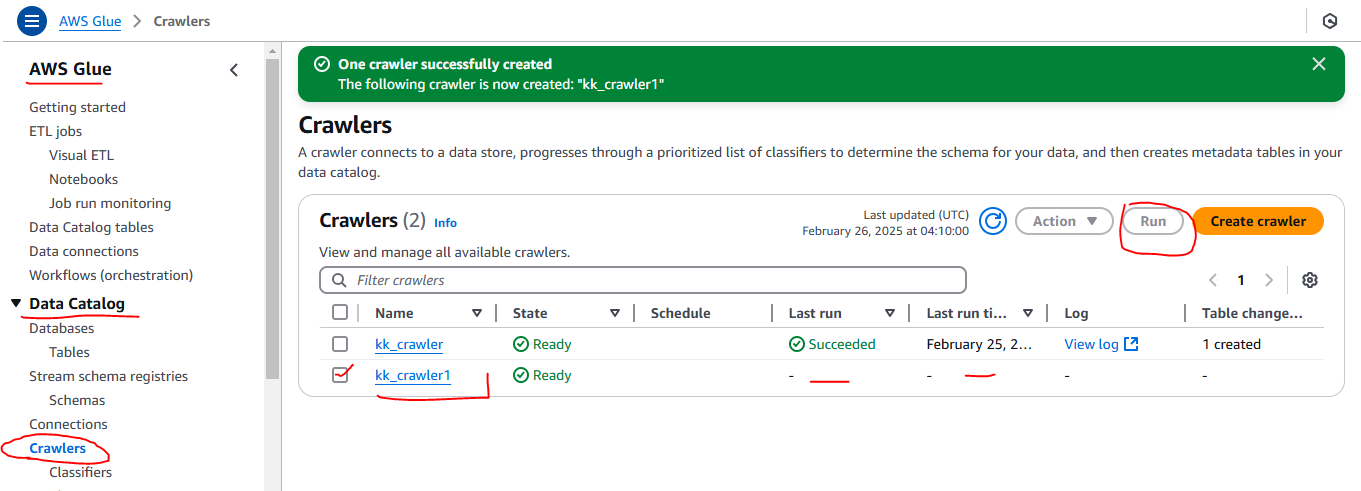
****

**Select database name which we already created**

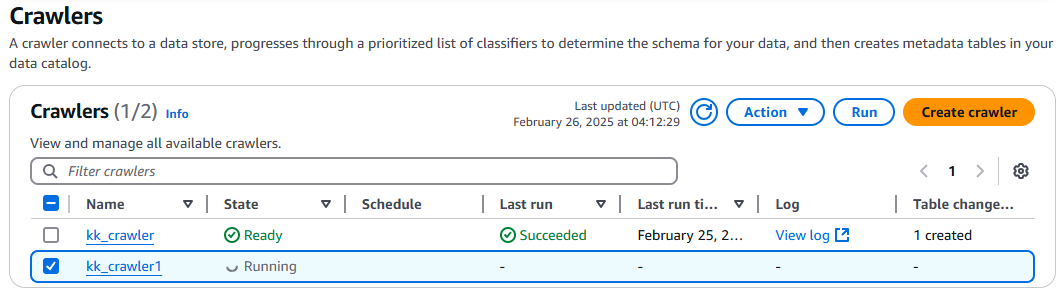
**Or Enter new database Name which will be created by this wizard**

****

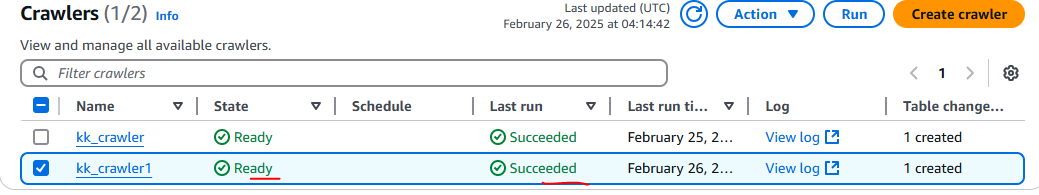
**After crawler is created. Click on crawler in left pan**

****

**Select crawler and click on Run button**

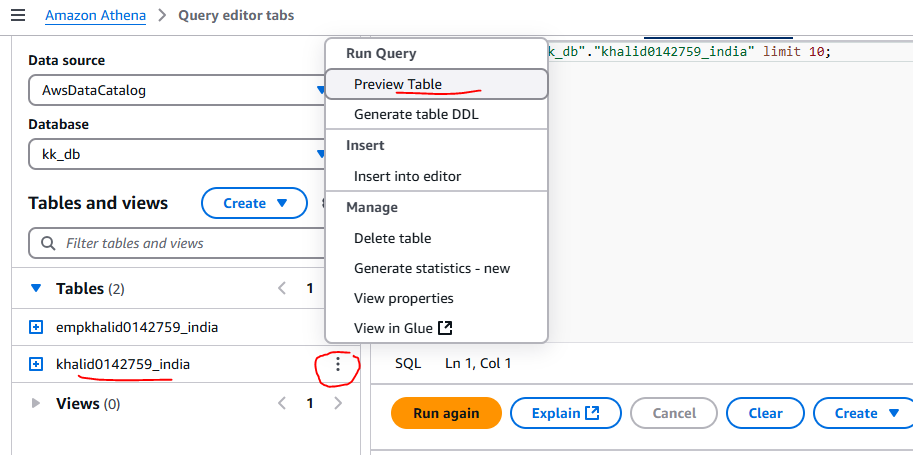
****

**After successful run**

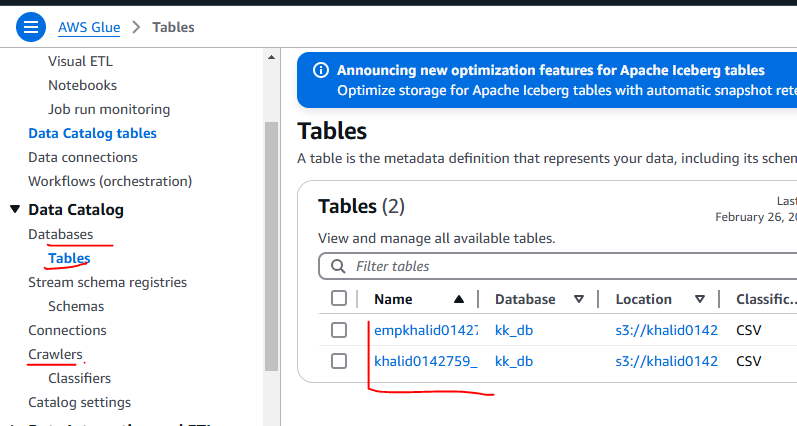
****

**Go to Athena to see table which has been created as below**

**Click on …(3 dots) and click on “preview table” to see table record.**

****

**In Addition, database and table list can also be seen in AWS Glue like below**

****

## Best practices

### Optimizing data formats (e.g., Parquet, ORC)

Raw data stored in CSV may be the most straightforward but inefficient. Storing our data in a compressed format like Parquet or ORC format will save on data read costs.

An additional benefit of Parquet and ORC is their columnar-based compression. Athena’s optimizer allows it to look only for particular data columns instead of working through the entire table to perform calculations.

### Partitioning data to improve query performance

Partitioning data means regularly splitting up a dataset based on a particular key, such as a date. For example, we may have daily partitions where the data is set up to be automatically divided and stored by days.

When our data is partitioned, the SQL engine can perform better optimization by looking at relevant partitions. This leads to a direct improvement in reducing the amount of scanned data, reducing overall cost.

# AWS Athena Vs Redshift

|  |  |  |
| --- | --- | --- |
|  | **Athena** | **Redshift** |
| Data structure | Can analyze unstructured data | Works best for structured data |
| Data location | Queries data directly in S3 | Requires data to be stored in clusters |
| Set up | No initial setup required | Requires setting up clusters |
| Pricing | Flat fee based on data scanned | Varies based on cluster configuration and other factors |
| Performance | Good for simple read and aggregated queries | Good for complex queries and large datasets |
| Integration | Integrates with other AWS services | Integrates into the AWS ecosystem |

## AWS RDS

* AWS RDS is short for Amazon Relational Database Service.
* Amazon RDS is a Relational Database Cloud Service
* Amazon RDS supports **PostgreSQL**, **MySQL**, **Maria DB**, **Oracle**, **SQL Server**, and **Amazon Aurora**

1. [MySQL.](https://https/www.geeksforgeeks.org/mysql-introdution/)
2. [PostgreSQL.](https://www.geeksforgeeks.org/what-is-postgresql-introduction/)
3. Oracle.
4. [SQL Server.](https://www.geeksforgeeks.org/what-is-sql/)

* The backup of the data and the infrastructure will be taken care of by the AWS

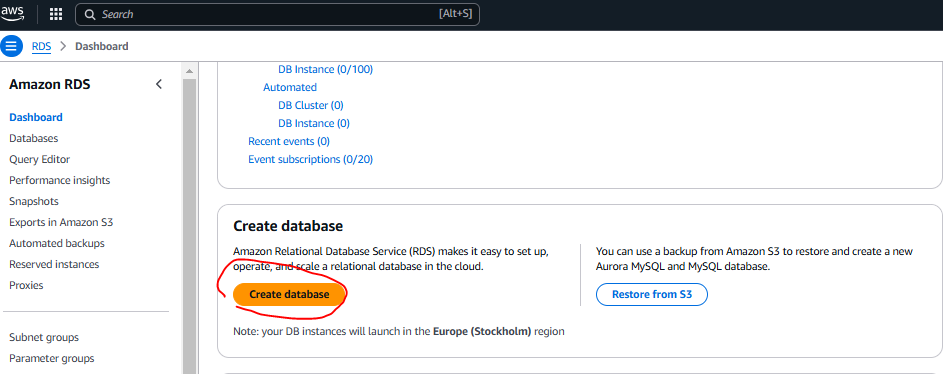
### Use Cases Of Amazon RDS (AWS)

1. **WebApplication:**The Amazon RDS is mainly used for the backend for web applications where it can support maximum no.of in and output operation. And also is easy to scale up and down.
2. **Managed Database:**Instead of you managing the database AWS will provide Amazon RDS as a service by just doing some configuration your database will be available to perform the operations.
3. **Isolation:**You can integrate and configure multiple applications with secure isolation by protecting the data of each application’s customers while managing the underlying infrastructure.
4. **Highly Secured:**You can use Amazon RDS for domains like health care and banking because the data used in this type of application is highly secure which can be achieved with the help of AWS RDS.

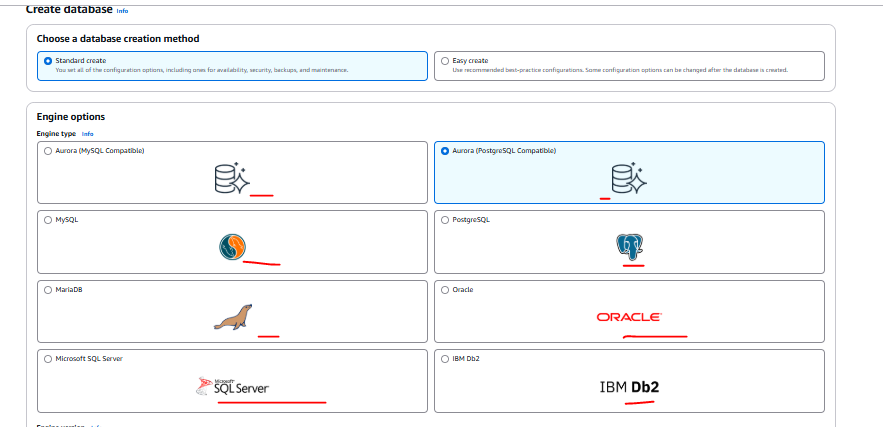
### Ways to Interact with Amazon RDS

The AWS Management Console provides an easy-to-use web-based interface to manage and monitor your RDS instances including creating databases, managing backups and scaling instances

Step1: Search **RDS** in aws management console



Step2: Click on “Create Database”



# AWS S3—Simple Storage Service

AWS comes up with different types of storage services for maintaining

* highly confidential data,
* frequently accessed data,
* often accessed storage data.

Amazon S3 is a Simple Storage Service in[AWS](https://www.geeksforgeeks.org/aws-tutorial/)that stores files of different types like **Photos**, **Audio**, and **Videos** as Objects providing more scalability and security to. It allows the users to store and retrieve any amount of data at any point in time from anywhere on the web.

## What is Amazon S3 Used for?

**Data Storage:**Amazon s3 acts as the best option for scaling both small and large storage applications. It helps in storing and retrieving the data-intensitive applications as per needs in ideal time.

**Backup and Recovery:**Many Organizations are using Amazon S3 to backup their critical data and maintain the data durability and availability for recovery needs.

**Hosting Static Websites:**Amazon S3 facilitates in storing HTML, CSS and other web content from Users/developers allowing them for hosting Static Websites

**Data Archiving:**[Amazon S3 Glacie](https://www.geeksforgeeks.org/what-is-amazon-glacier/)r service integration helps as a cost-effective solution for long-term data storing which are less frequently accessed applications.

**Big Data Analytics:**Amazon S3 is often considered as data lake because of its capacity to store large amounts of both structured and unstructured data offering seamless integration with other AWS Analytics and AWS Machine Learning Services.

## Amazon S3 bucket?

Amazon S3 bucket is a fundamental Storage Container feature in AWS S3 Service. It provides a secure and scalable repository for storing of Objects such as **Text data**, **Images**, **Audio** and **Video** files over AWS Cloud. Each S3 bucket name should be named globally unique and should be configured with ACL (Access Control List).

* Each bucket will have its own set of policies and configurations.
* This enables users to have more control over their data.
* Bucket Names must be unique
* The maximum size of an AWS S3 bucket is 5TB

### How to Access Amazon S3 Bucket?

You can work and access the Amazon S3 bucket by using any one of the following methods

1. **AWS Management Console**
2. **AWS CLI Commands**
3. Programming Scripts ( Using **boto3** library of Python )

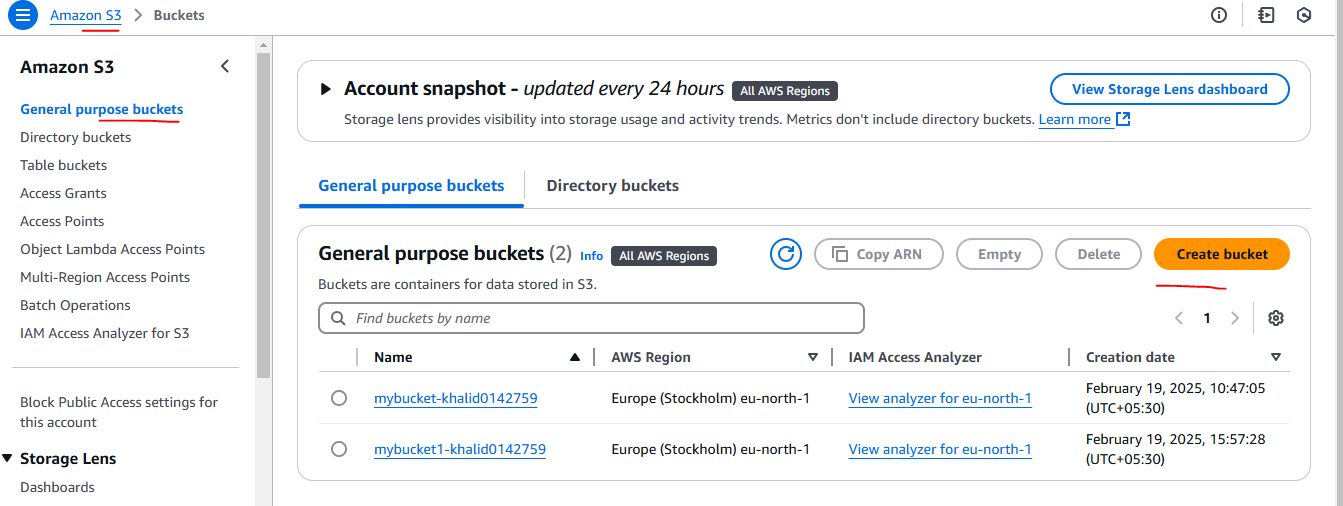
<https://www.youtube.com/watch?v=MwFH65uTMXw> --How to create s3bucket and upload file-playlist

#### Create S3 bucket and upload file

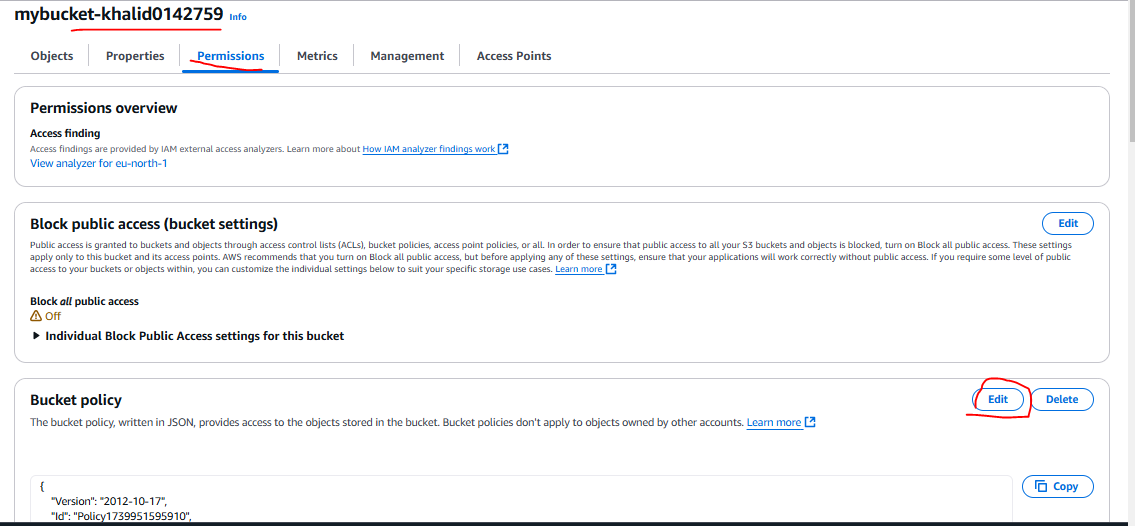
Step1: open AWS management console—web application

Step2: search “S3”

Step3: Click on “Create bucket”



Step4: Select created bucket and goto permission and edit policy as per above youtube video



Step5: Make sure to put “/\*” at the end of URL as per below json content

{

"Version": "2012-10-17",

"Id": "Policy1739951595910",

"Statement": [

{

"Sid": "Stmt1739951593390",

"Effect": "Allow",

"Principal": "\*",

"Action": "s3:\*",

"Resource": "arn:aws:s3:::mybucket-khalid0142759**/\***"

}

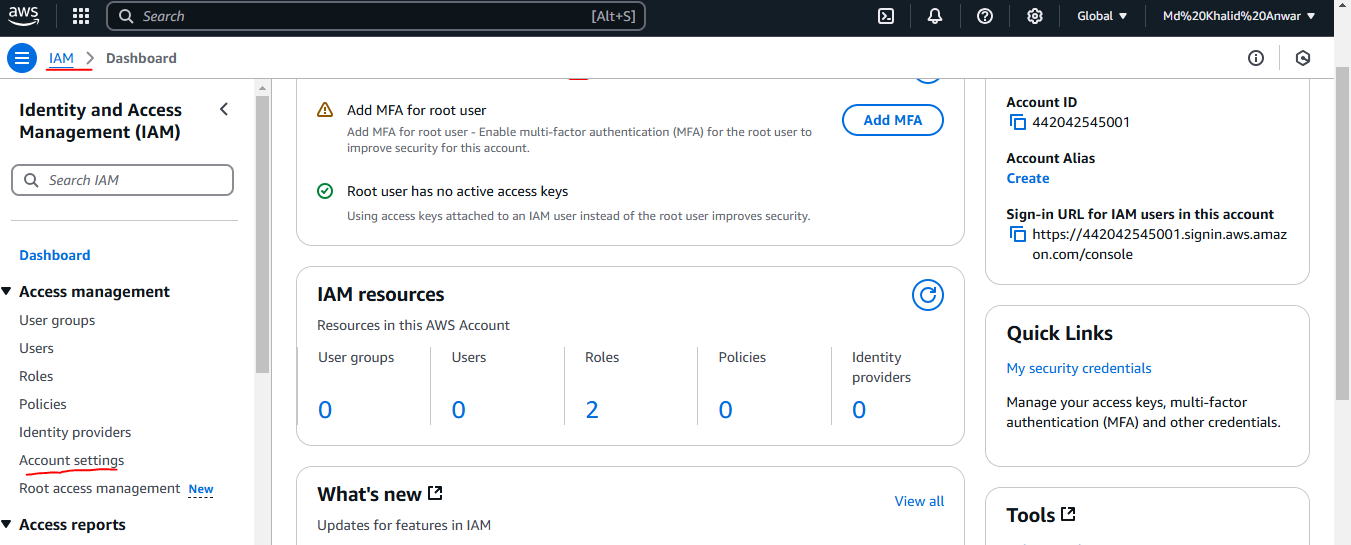
]

}

## Create policy

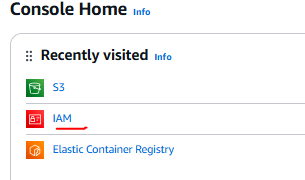
<https://docs.matillion.com/metl/docs/2044579/#create-an-external-stage>

Goto IAM and then policies in left hand side



#### Create Policies like below

<https://www.youtube.com/watch?v=NXaRxwax9qA>



### Policy json

Policy is all about what actions can be performed. Few actions

* PutObject—uploading file in s3 bucket
* GetObject—Reading File from S3 bucket
* GetObjectVersion—get version file if we are maintaining file versioning
* DeleteObject ---delete file from S3 bucket
* DeleteObjectVersion—delete versioned file from S3 bucket

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"s3:PutObject",

"s3:GetObject",

"s3:GetObjectVersion",

"s3:DeleteObject",

"s3:DeleteObjectVersion"

],

"Resource": "arn:aws:s3:::**mybucket-khalid0142759/\***"

},

{

"Effect": "Allow",

"Action": [

"s3:ListBucket",

"s3:GetBucketLocation"

],

"Resource": "arn:aws:s3:::mybucket-khalid0142759",

"Condition": {

"StringLike": {

"s3:prefix": [

"\*"

]

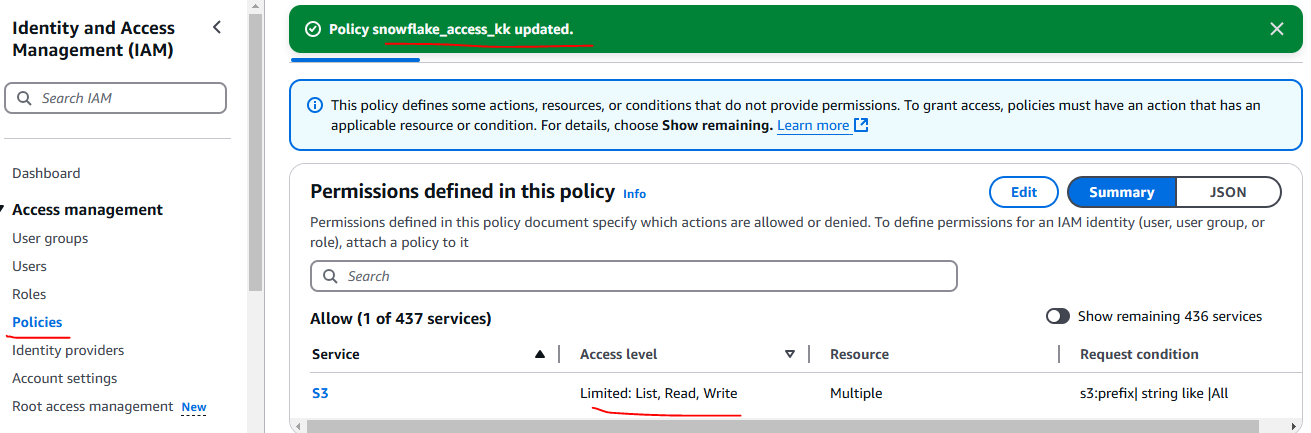
}

}

}

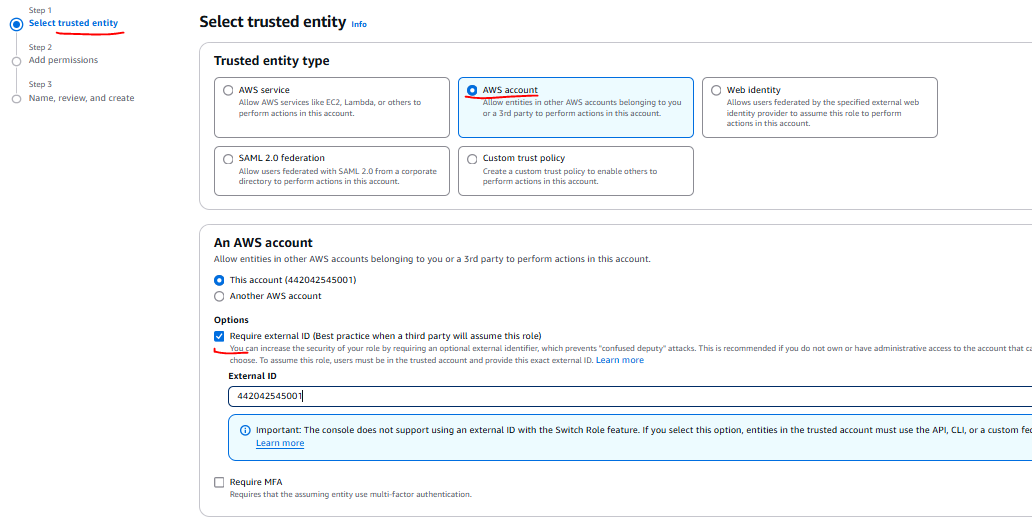
]

}

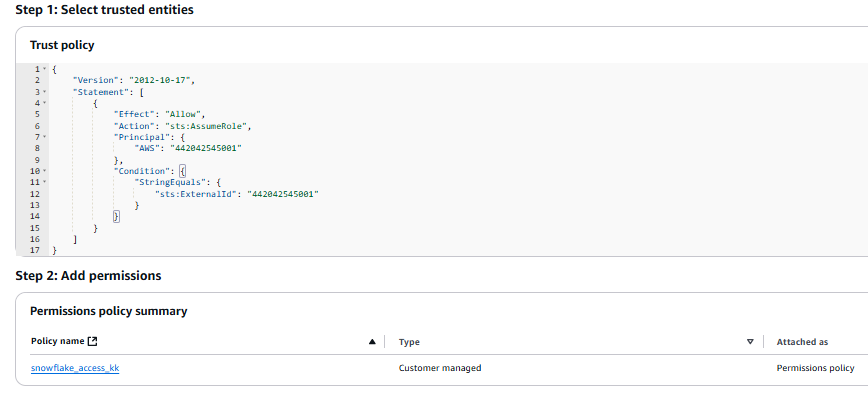


### Create Role

Create Role which will be used in third party app so select “**AWS account**” and created policy must be mapped to it.

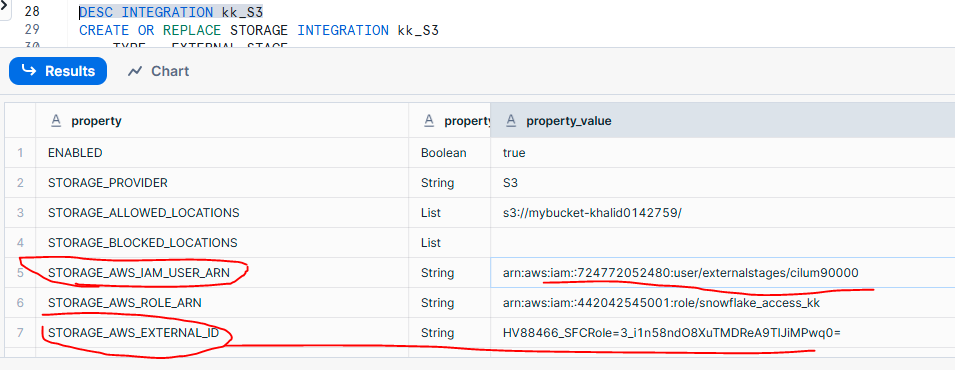


While creating, we are entering default account number and later this should be changed as per third party application



ARN—Application resource number

After creating Storage integration



Above two highlighted values should be copied and updated in AWS IAM role as below.

Goto to IAM>>Role>>Trusted relationship and make changes

