1. Introduction

We would need to set up a **new AWS account** to connect to the Collision 24 production database for performing the cutover data ingestion and segregation activity on production data.

This document serves as a guide for building and deploying the architecture needed to securely connect to and migrate the production data into a separate AWS account.

The architecture allows configuring AWS Glue in such a way that it gets assigned a public IP address which can be whitelisted on the Collision Grotrex DB.

This document contains the following sections:

1. **Architecture:** Describes how to build the AWS infrastructure needed to perform the migration. This includes configuring an AWS Glue JDBC connector in a private subnet which can connect to sources outside the VPC through a public IP through a NAT Gateway.

2. **IAM Configuration:** Creation of a single IAM user which will be used to perform the migration. Additionally IAM role permissions required by AWS Glue and other services are described here.

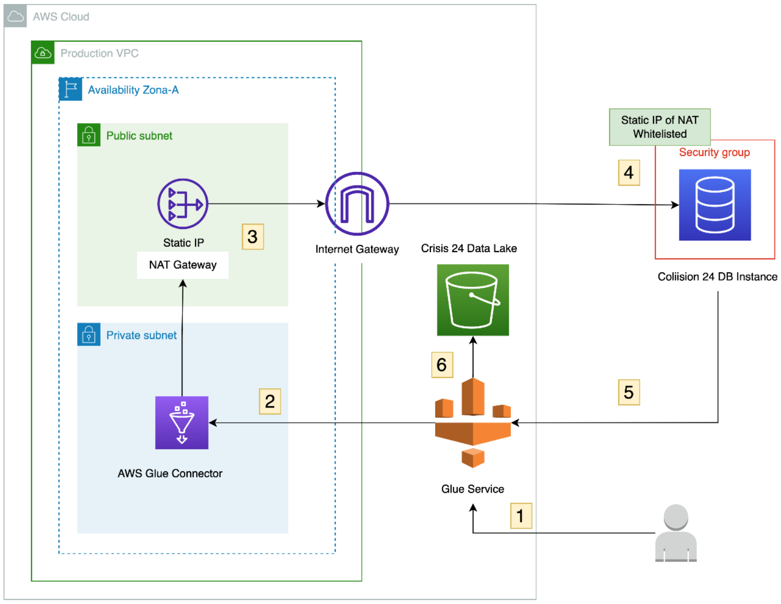
2. Infrastructure Setup

The infrastructure has the following components:

1. AWS Glue
2. AWS Glue Network Connector
3. VPC with Private and Public Subnets
4. NAT Gateway in the public subnet
5. AWS S3

The Region should be London (**eu-west-2**) for provisioning all resources.

The architecture is shown and described in the diagram below:



1. A user calls the migration script in the AWS Glue Service
2. The Glue Service uses a Glue Connector deployed inside a private subnet inside a VPC to make the connection to the database
3. The Glue Connector routes the connection through a NAT Gateway with a Static IP address assigned to it.
4. This Static IP is whitelisted on the Collision DB's Firewall settings
5. The Script is run inside the Collision DB and results returned to Glue Service
6. The Glue Service writes these results to the S3 bucket

For enhanced security, we need to configure a **static IP** for the **glue connector** to connect to Collision SQL Server, so that this IP can be directly whitelisted on the SQL server. This section walks you through a step by step process to configure the glue connector.

2.1. VPC Creation:

The first step is to create a VPC which includes the following:

1. A private subnet for hosting the Glue Connector
2. A public subnet that contains a **NAT Gateway with a static IP** attached to it which the Glue Connector uses to communicate with the internet
3. Configuration of route tables to enable this connectivity

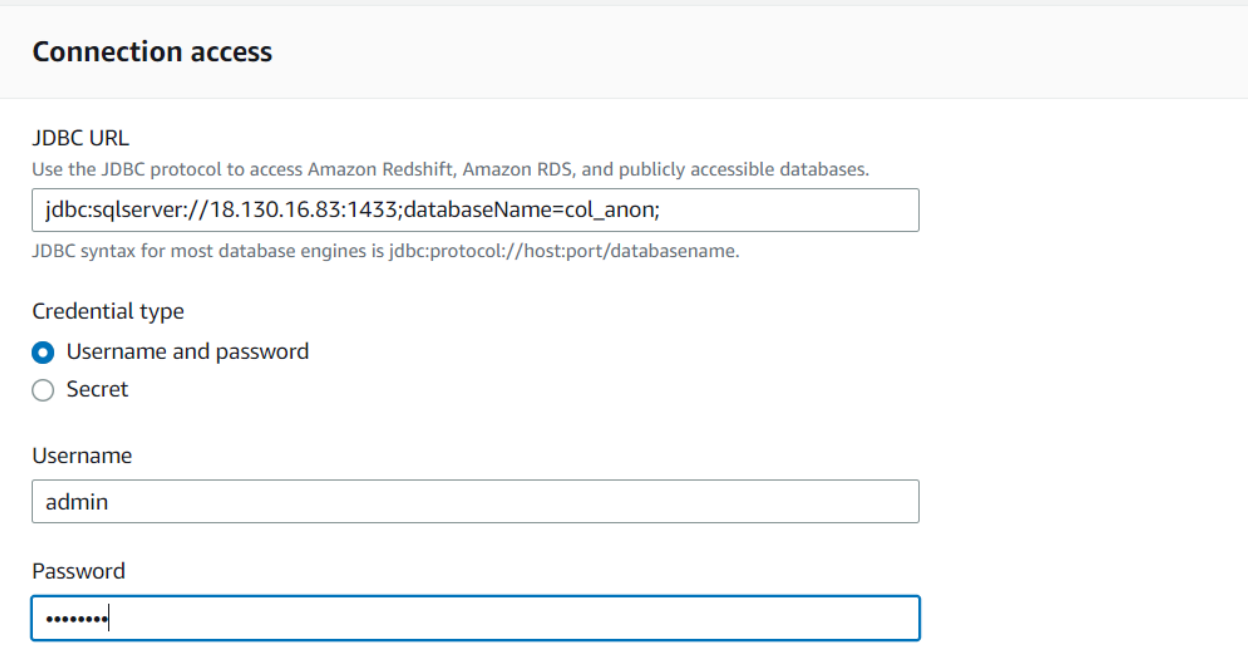
This architecture can be provisioned using a cloudformation template from the following link:

Download template from link :
https://docs.aws.amazon.com/codebuild/latest/userguide/cloudformation-vpc-template.html

2.2. Setup Glue connection:

After the VPC has been deployed, a Glue JDBC connection must be set up. Please follow the following steps to do so:

1. Select AWS Glue service, then select create a connection.
2. Enter a name for the connection, and select the Connection type of JDBC.
3. Enter JDBC URL `(Collision 24 DB Url, example: jdbc:sqlserver://<ip address / hostname>;databaseName=<db name>)`, DB Username and DB Password
4. Select the VPC , the private subnet and select Security group (Create new security group with allow inbound and outbound traffic)
5. Select Create Connection.



Connection access

JDBC URL

Use the JDBC protocol to access Amazon Redshift, Amazon RDS, and publicly accessible databases.

jdbc:sqlserver://18.130.16.83:1433;databaseName=col_anon;

JDBC syntax for most database engines is jdbc:protocol://host:port/databasename.

Credential type

● Username and password

○ Secret

Username

admin

Password

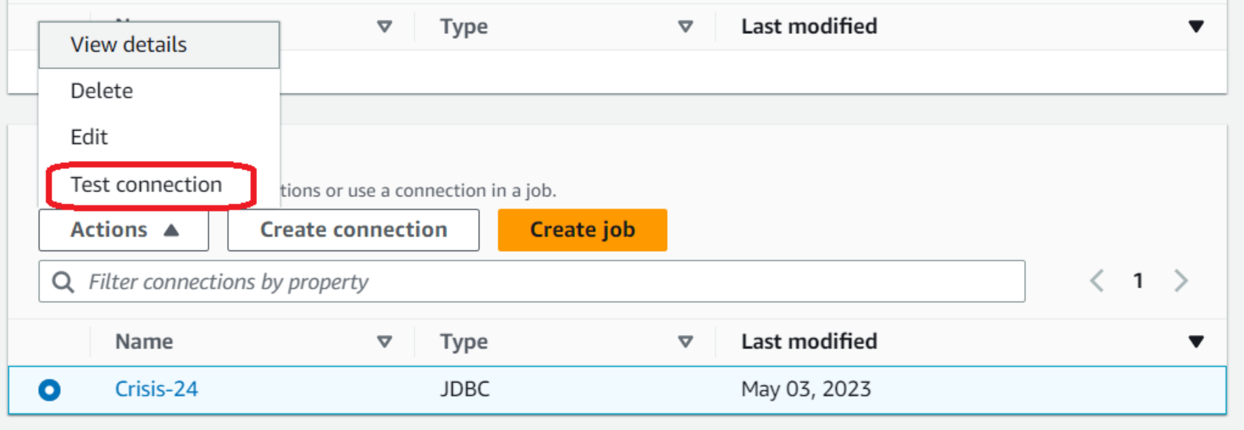
••••••••

2.3. Whitelisting the NAT IP on DB Firewall:

Select the private subnet you attached the Glue connection and then goto NAT gateway and note down the public IP of the NAT Gateway. Add that public IP into the inbound firewall rules of the Collision Database.

2.4. Verify connection:

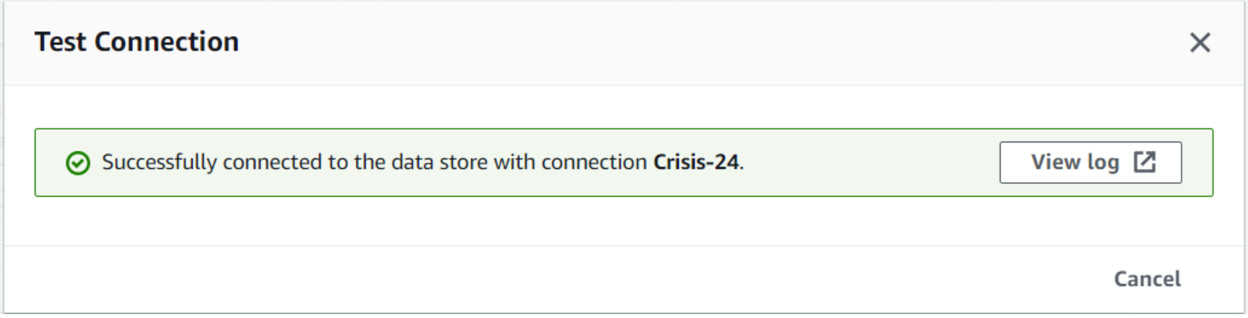
Select the connection, click on Action button and select test connection



tions or use a connection in a job.

	Name	Type	Last modified
●	Crisis-24	JDBC	May 03, 2023

Then select the IAM role from Glue connection test



Successfully connected to the data store with connection **Crisis-24**.

2.5. S3 Bucket Creation

Create an S3 bucket by the name of **crisis-24** in the London (eu-west-2) region.

3. IAM Configuration

3.1. IAM User Permissions

The migration from the production database would be performed through a single IAM user. The Crisis 24 team would have access to the IAM user and Addo would guide the Crisis 24 team through a screen sharing session.

The IAM user should be named: **migration-user**

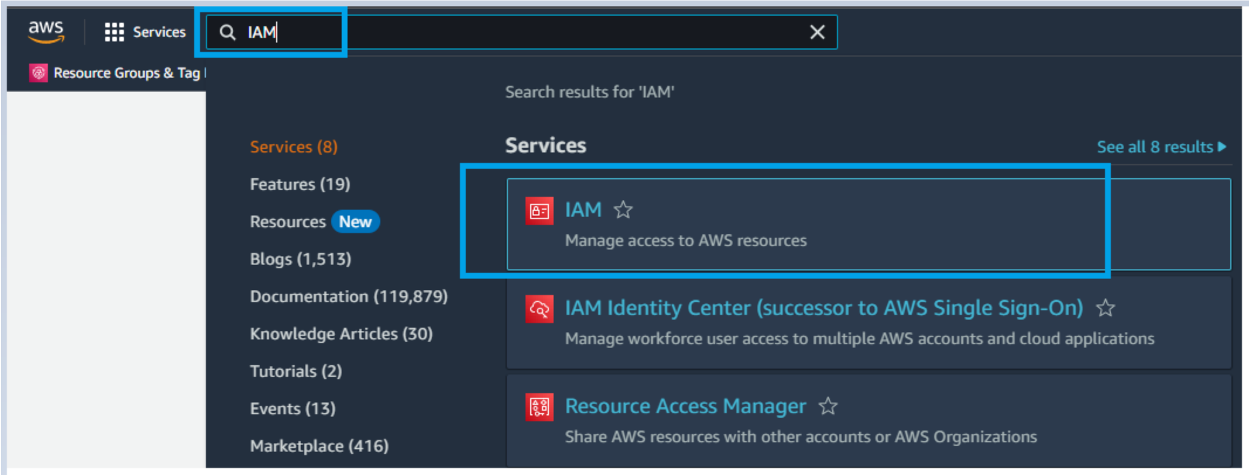
The user should be granted the following permissions:

1. `AmazonS3FullAccess`
2. `AmazonAthenaFullAccess`
3. `AWSGlueConsoleFullAccess`
4. `AWSLakeFormationDataAdmin`
5. `CloudWatchReadOnlyAccess`
6. `IAMUserChangePassword`
7. `IAMUsersAllowManagedMFA`
8. `AmazonAppFlowFullAccess`

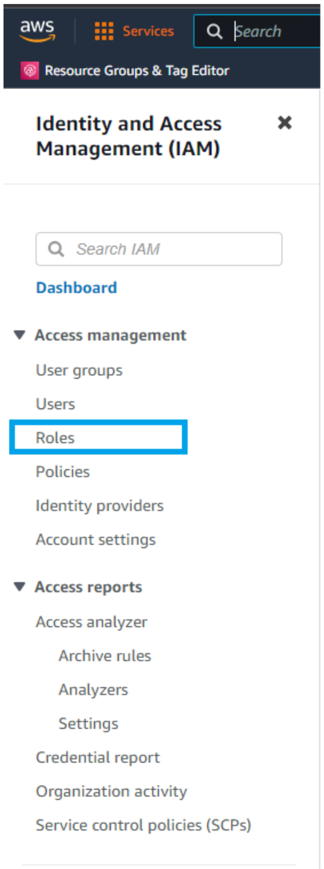
3.2. Glue IAM Role Permissions

Please follow the following instructions to get an IAM role created for Glue through which it can access S3 and other services:

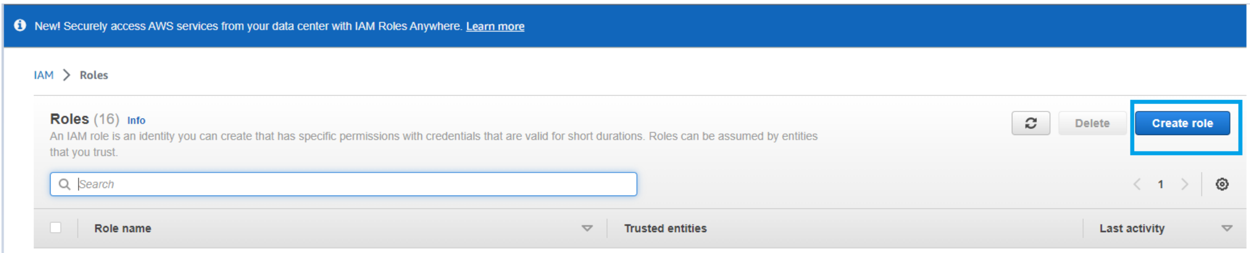
1. Go to AWS Console and search for IAM


aws
Services
IAM
Resource Groups & Tag
Search results for 'IAM'
Services (8)
Features (19)
Resources New
Blogs (1,513)
Documentation (119,879)
Knowledge Articles (30)
Tutorials (2)
Events (13)
Marketplace (416)
Services
See all 8 results ▸
IAM ☆
Manage access to AWS resources
IAM Identity Center (successor to AWS Single Sign-On) ☆
Manage workforce user access to multiple AWS accounts and cloud applications
Resource Access Manager ☆
Share AWS resources with other accounts or AWS Organizations

2. After selecting IAM service, choose the option of ***Role***



3. Now select the option ***Create Role***

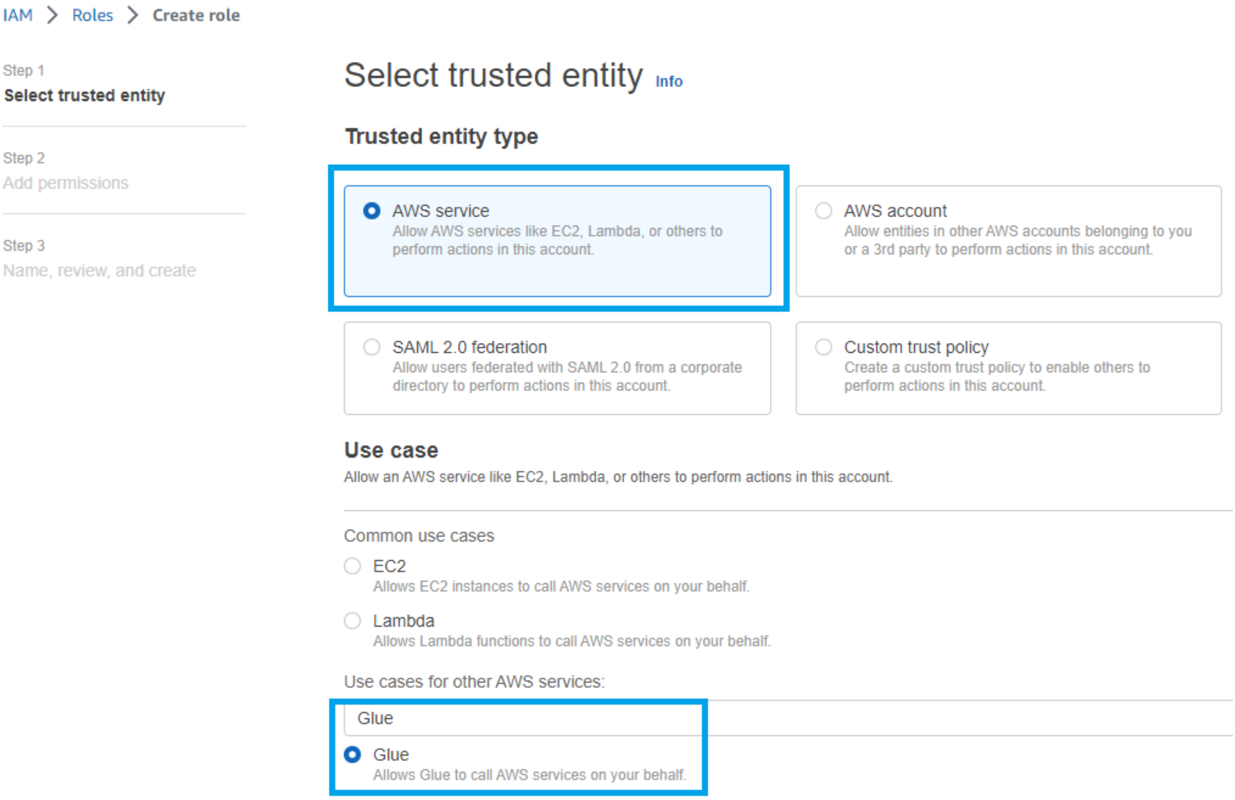


New! Securely access AWS services from your data center with IAM Roles Anywhere. Learn more

Roles (16) 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.

4. Now choose ***AWS service*** as the Trusted Entity Type and ***Glue*** as the service



IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Select trusted entity Info

Trusted entity type

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

○ AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

○ SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

○ Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

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

Common use cases

○ EC2
Allows EC2 instances to call AWS services on your behalf.

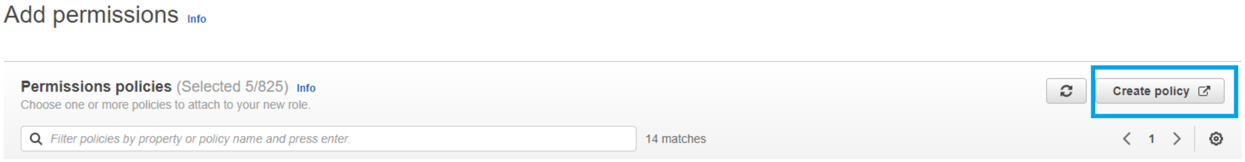
○ Lambda
Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:

Glue

● Glue
Allows Glue to call AWS services on your behalf.

5. Now Add permissions by first selecting the ***Create policy*** option to create a custom in line policy



Add permissions Info

Permissions policies (Selected 5/825) Info
Choose one or more policies to attach to your new role.

Filter policies by property or policy name and press enter.

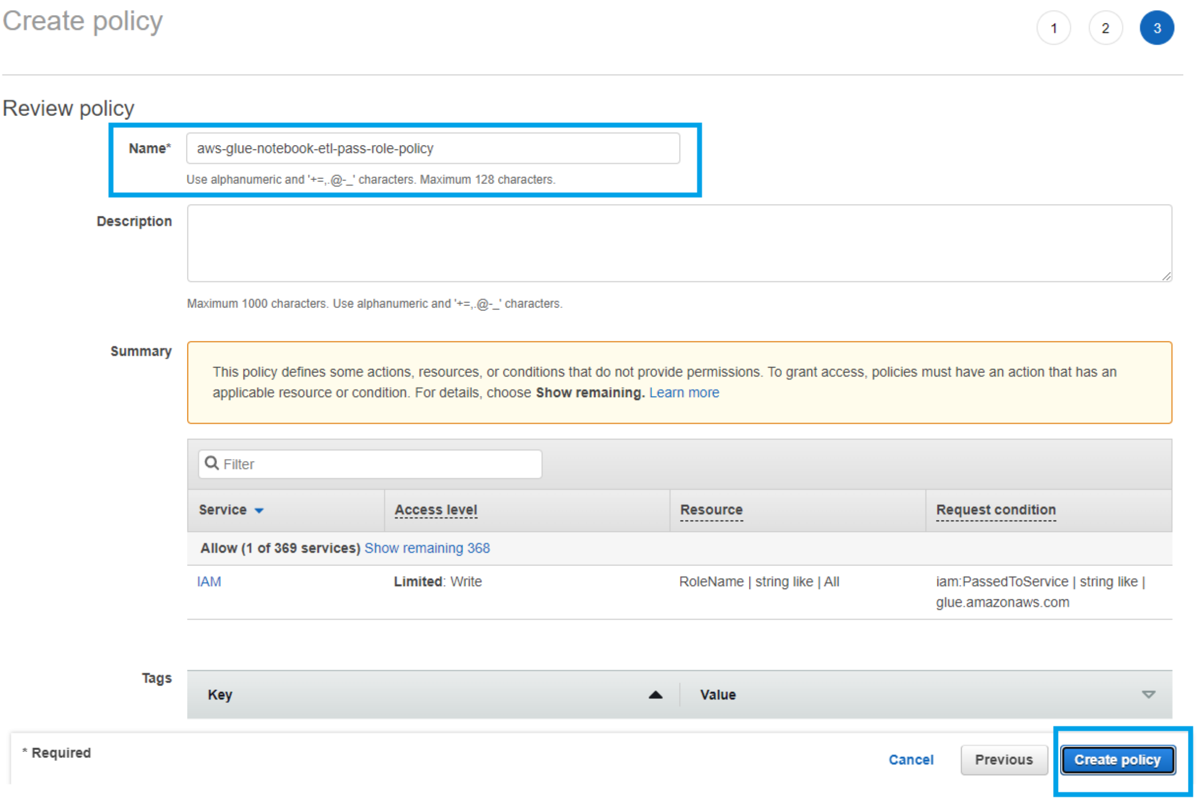
14 matches

6. Now choose the ***JSON*** option and enter the following *JSON* in the editor

JavaScript

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
           "Action": [
               "iam:GetRole",
               "iam:PassRole"
           ],
           "Effect": "Allow",
           "Resource": [
               "arn:aws:iam::*:role/*"
           ],
           "Condition": {
               "StringLike": {
                   "iam:PassedToService": [
                       "glue.amazonaws.com"
                   ]
               }
           }
       }
   ]
}
```

7. Now select option ***Next:Review*** and give a descriptive name to the custom policy such as : *aws-glue-notebook-etl-pass-role-policy* and select ***Create Policy*** option



Create policy

Review policy

Name* aws-glue-notebook-etl-pass-role-policy

Use alphanumeric and '+=,.@-_' characters. Maximum 128 characters.

Description

Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Summary

This policy defines some actions, resources, or conditions that do not provide permissions. To grant access, policies must have an action that has an applicable resource or condition. For details, choose **Show remaining.** Learn more

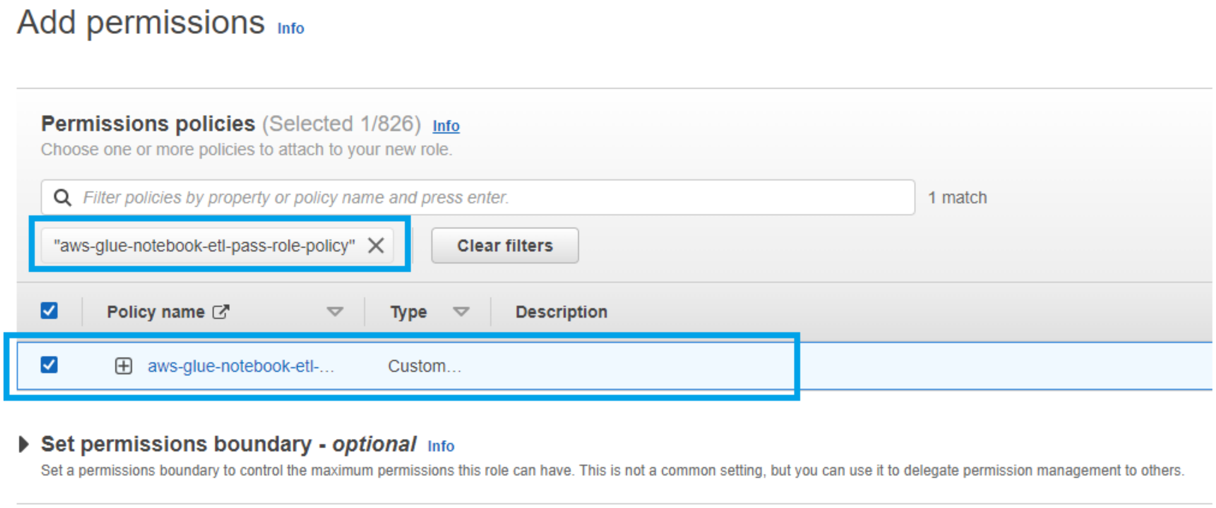
Service	Access level	Resource	Request condition
Allow (1 of 369 services) Show remaining 368			
IAM	**Limited**: Write	RoleName | string like | All	iam:PassedToService | string like | glue.amazonaws.com

Tags

Key	Value

* Required

8. Return to the original Add permissions page and select the custom policy created above



Add permissions Info

Permissions policies (Selected 1/826) Info

Choose one or more policies to attach to your new role.

1 match

"aws-glue-notebook-etl-pass-role-policy"

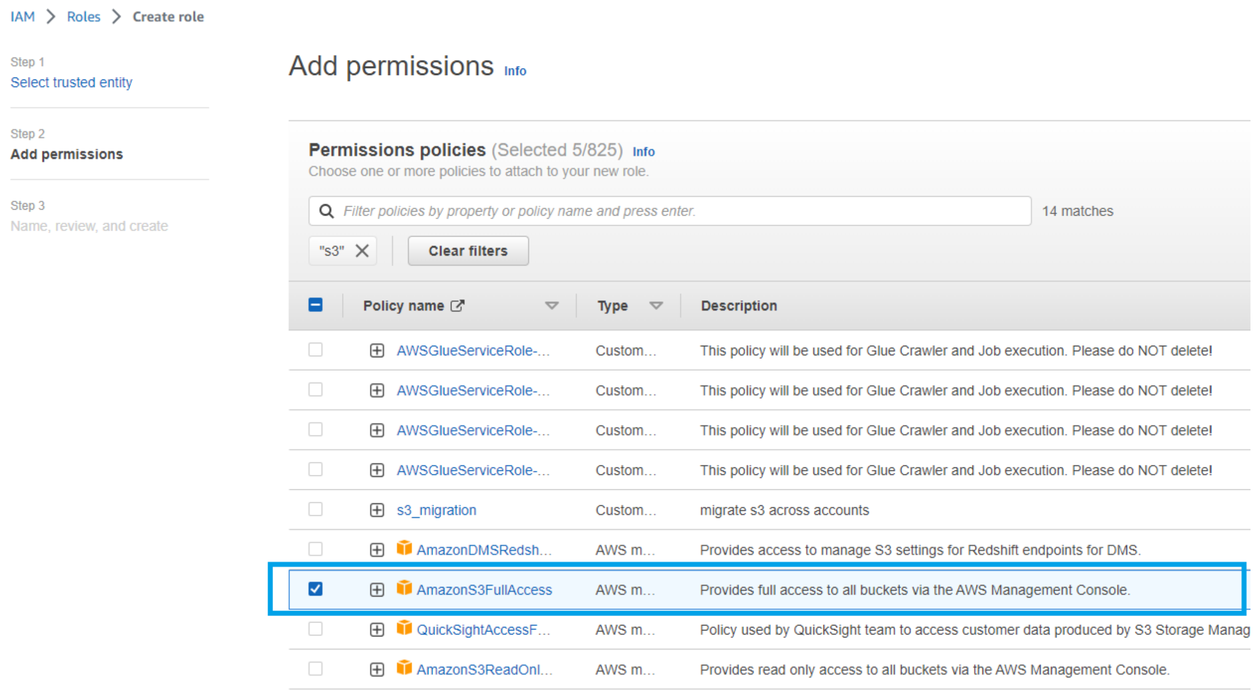
☑	Policy name	Type	Description
☑	aws-glue-notebook-etl-...	Custom...	

Set permissions boundary - *optional* Info

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.

9. Now add the following built in policies by selecting them:

- AmazonS3FullAccess
- AWSGlueServiceNotebookRole
- AWSGlueServiceRole
- AWSGlueConsoleFullAccess
- AmazonRDSFullAccess
- AmazonAthenaFullAccess



IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Add permissions Info

Permissions policies (Selected 5/825) Info
Choose one or more policies to attach to your new role.

Filter policies by property or policy name and press enter. 14 matches

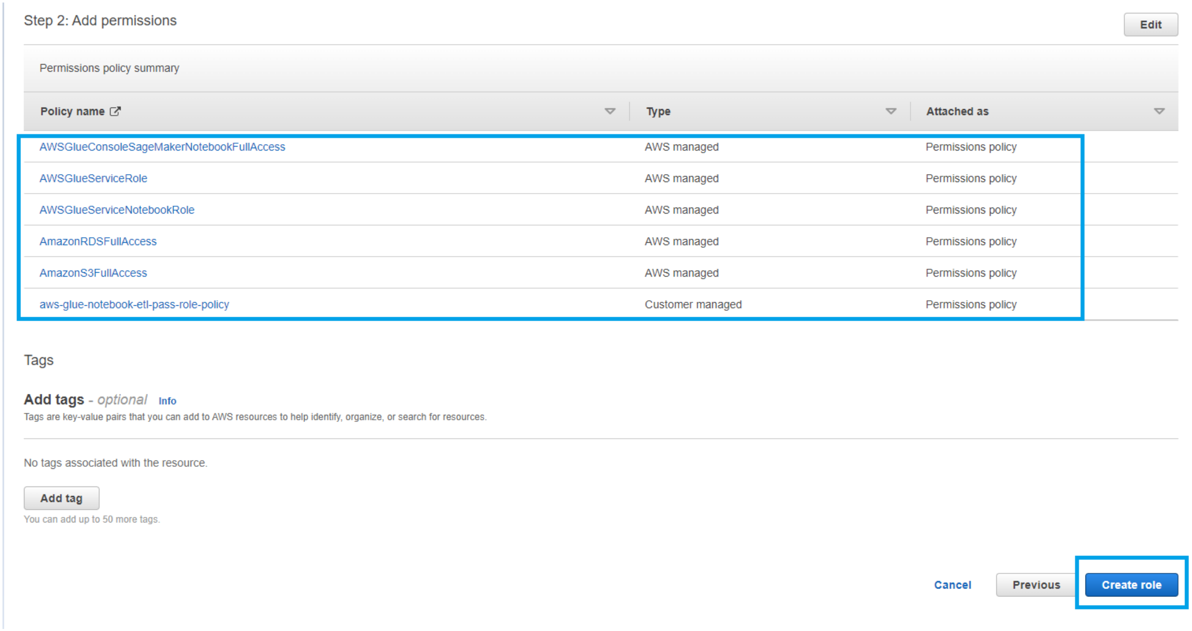
"s3" ✕ Clear filters

▣	Policy name	Type	Description
□	AWSGlueServiceRole-...	Custom...	This policy will be used for Glue Crawler and Job execution. Please do NOT delete!
□	AWSGlueServiceRole-...	Custom...	This policy will be used for Glue Crawler and Job execution. Please do NOT delete!
□	AWSGlueServiceRole-...	Custom...	This policy will be used for Glue Crawler and Job execution. Please do NOT delete!
□	AWSGlueServiceRole-...	Custom...	This policy will be used for Glue Crawler and Job execution. Please do NOT delete!
□	s3_migration	Custom...	migrate s3 across accounts
□	AmazonDMSRedsh...	AWS m...	Provides access to manage S3 settings for Redshift endpoints for DMS.
☑	AmazonS3FullAccess	AWS m...	Provides full access to all buckets via the AWS Management Console.
□	QuickSightAccessF...	AWS m...	Policy used by QuickSight team to access customer data produced by S3 Storage Manag
□	AmazonS3ReadOnl...	AWS m...	Provides read only access to all buckets via the AWS Management Console.

10. Now, give the role a descriptive name such as : `AWSGlueServiceRoleMigration`

Note: It must begin with AWSGlueServiceRole as a prefix

11. Review the policies attached (both built in and custom policies) and select ***Create role*** option to get the role created



Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
AWSGlueConsoleSageMakerNotebookFullAccess	AWS managed	Permissions policy
AWSGlueServiceRole	AWS managed	Permissions policy
AWSGlueServiceNotebookRole	AWS managed	Permissions policy
AmazonRDSFullAccess	AWS managed	Permissions policy
AmazonS3FullAccess	AWS managed	Permissions policy
aws-glue-notebook-etl-pass-role-policy	Customer managed	Permissions policy

Tags

Add tags - *optional* Info

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.

You can add up to 50 more tags.