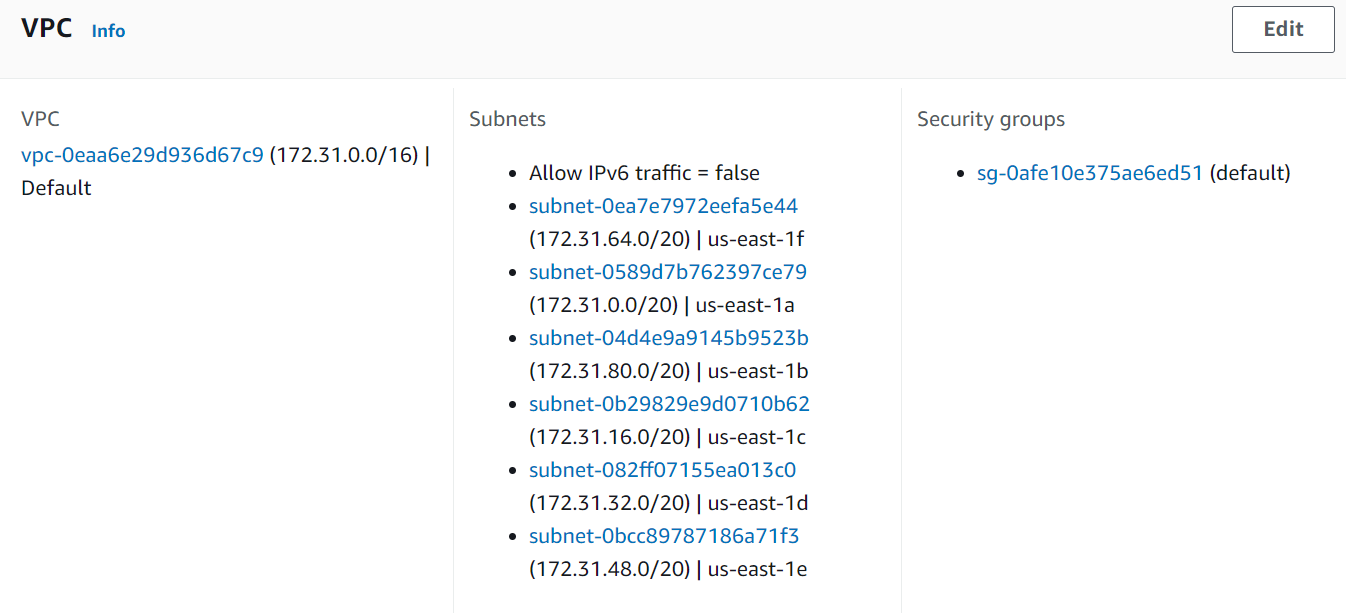
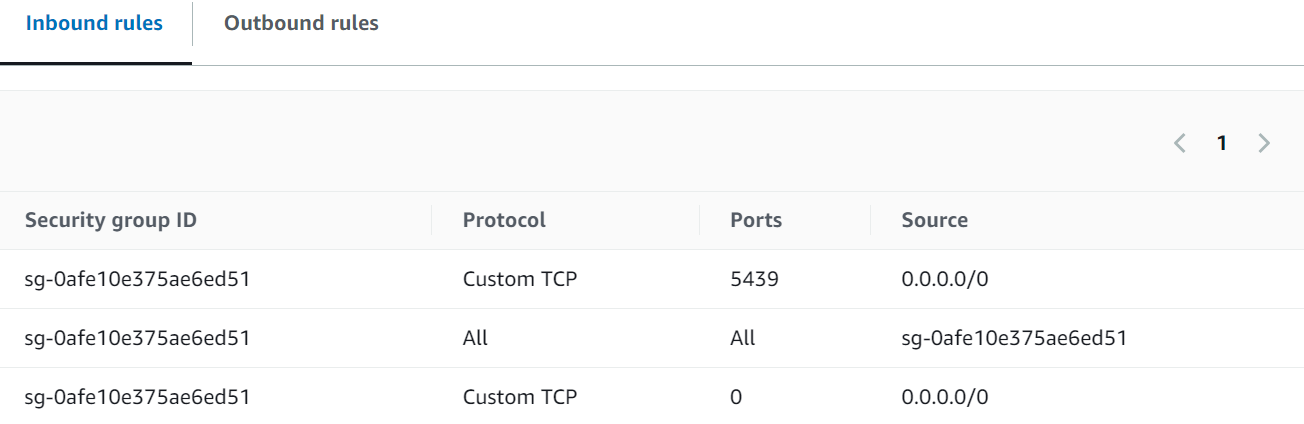
**AWS S3 to RedShift Data Transfer using AWS Lambda Function Triggered by S3 Events: -**

### **1. Creating a Redshift Cluster**

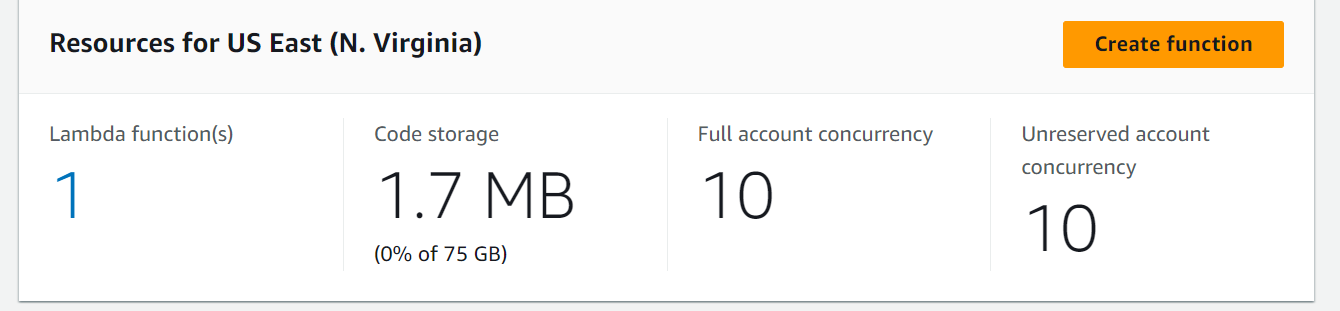
#### **Steps:**

1. **Sign** **in to the AWS Management Console**.
2. **Navigate to Amazon Redshift**.
3. **Create Cluster**:
   1. Click on "Create cluster".
   2. Configure cluster settings (e.g., cluster identifier, node type, number of nodes).
   3. Choose database settings (e.g., database name, master username, and password).
   4. Set up VPC and security groups in AWS Lambda as well as RedShift Cluster (for connecting to tableau)
      1. VPC Connection - (Create new inline policies **AllowCreateNetwork**, **DescribeNetwork**, **DeleteNetwork**) in lambda connection role in IAM (for e.g. - **lambda-demo-boto3**)
      2. Security Group – Edit Inbound Rule to add RedShift (TCP) for Tableau.





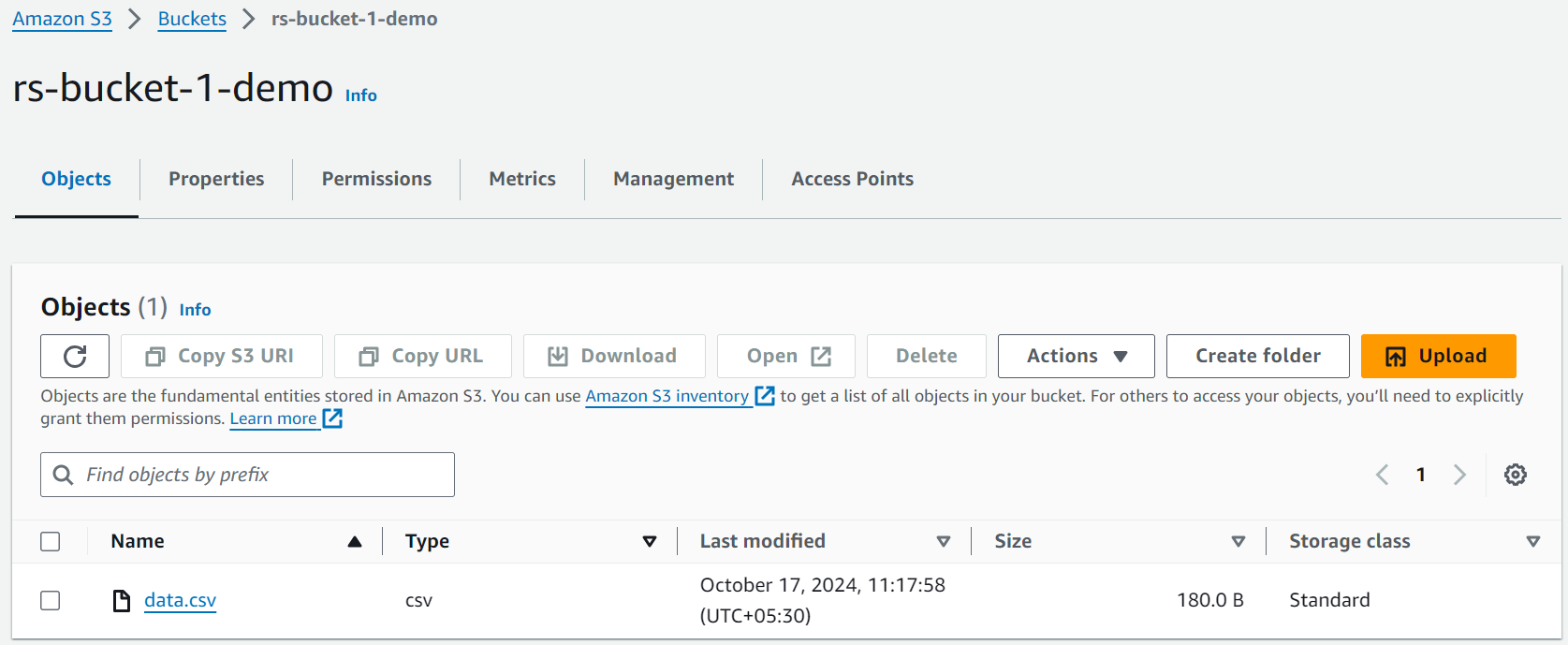
1. **Launch the Cluster** and wait for it to be available.



### **2. Creating an S3 Bucket**

#### **Steps:**

1. **Navigate to Amazon S3** in the AWS Management Console.
2. **Create Bucket**:
   1. Click on "Create bucket".
   2. Provide a unique name and select the appropriate region.
   3. Configure options (e.g., versioning, encryption) as needed (optional/default).
3. **Set Permissions**:
   1. Make sure the bucket policy allows access from Redshift and Lambda.

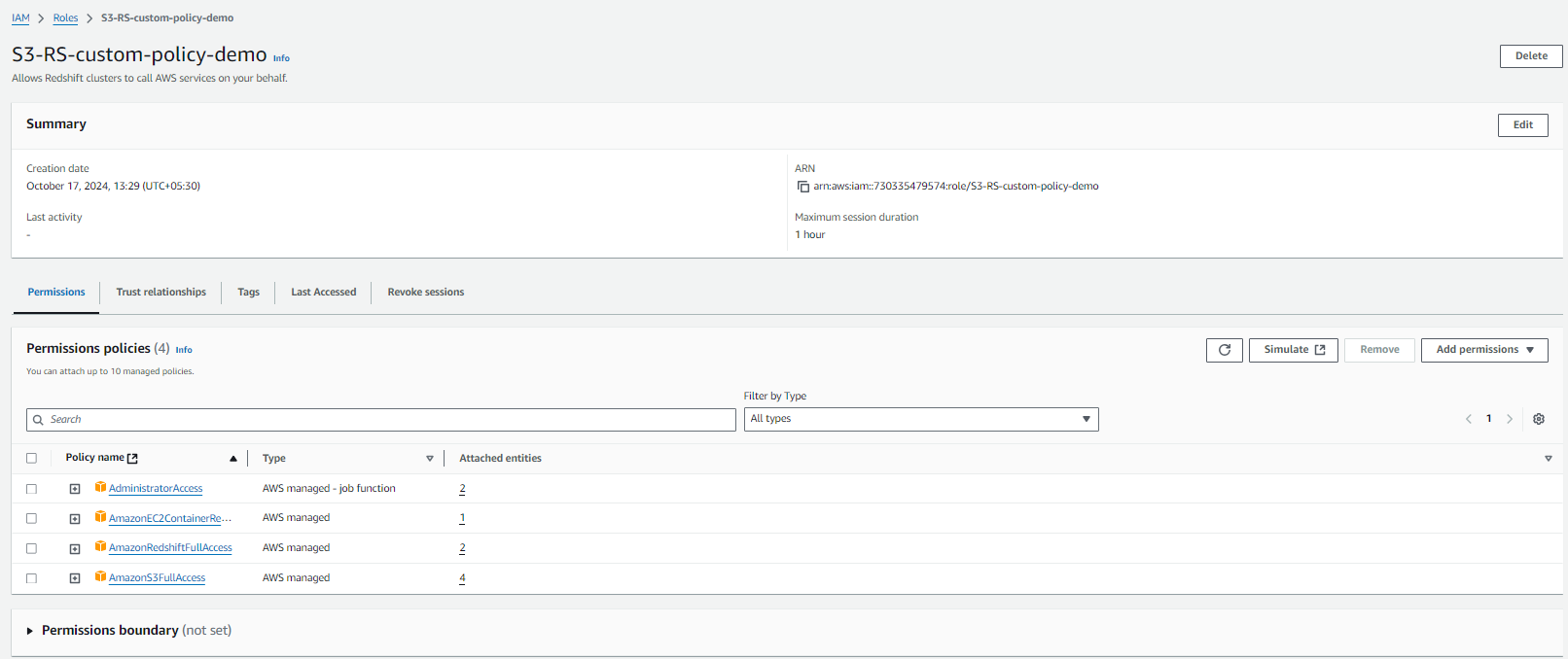


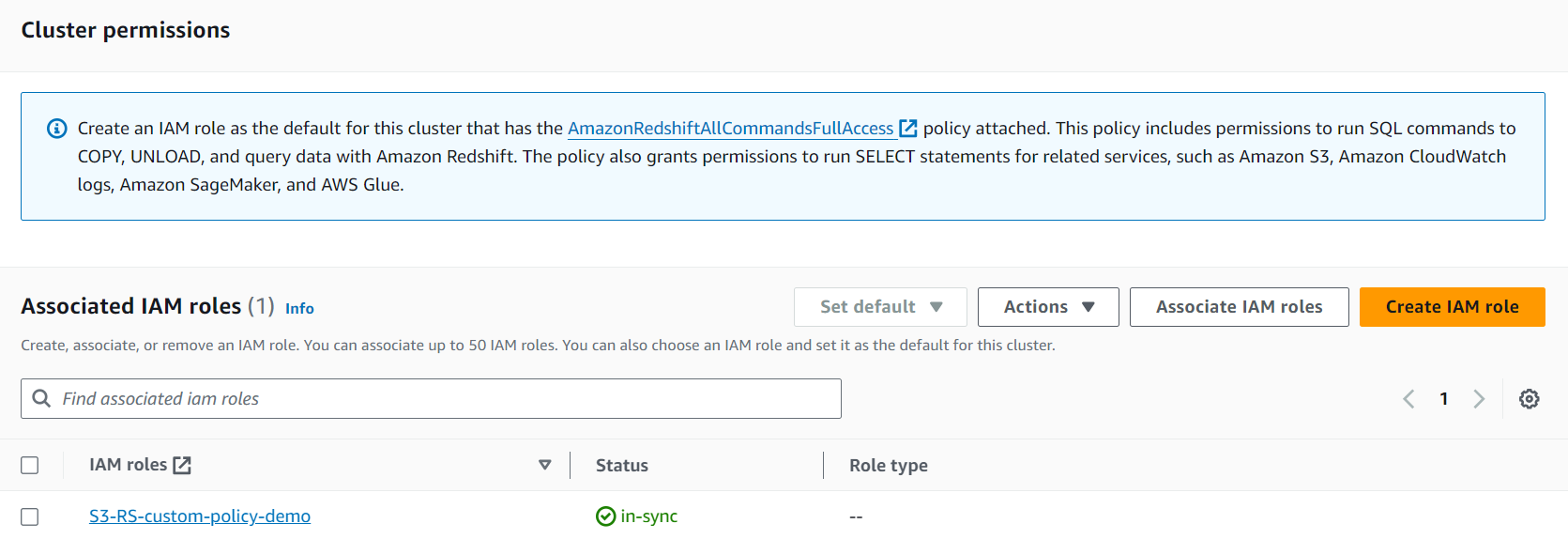
### **3. Setting Up IAM Roles**

#### **Roles Needed:**

1. **IAM Role for Redshift**:
   1. Allows Redshift to access the S3 bucket.
   2. Attach the AmazonS3ReadOnlyAccess policy or create a custom policy.
2. **IAM Role for Lambda**:
   1. Allows Lambda to execute and access necessary resources (S3, Redshift).
   2. Attach the following policies:
      1. AWSLambdaBasicExecutionRole
      2. AmazonS3ReadOnlyAccess
      3. Custom policy for Redshift access.
         1. S3-RS-custom-policy-demo(create Redshift policy from IAM and attach other policies like)
            1. [AmazonS3FullAccess](https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/policies/details/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAmazonS3FullAccess)
            2. [AmazonRedshiftFullAccess](https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/policies/details/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAmazonRedshiftFullAccess)
            3. [AmazonEC2ContainerRegistry](https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/policies/details/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAmazonEC2ContainerRegistryPowerUser)
            4. [AdministratorAccess](https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/policies/details/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAdministratorAccess)

**Note:** This policy (Redshift) with attached policies must be associated with the policy used in the **RedShift Cluster** and use the same **ARN** of this policy in the lambda function IAM used





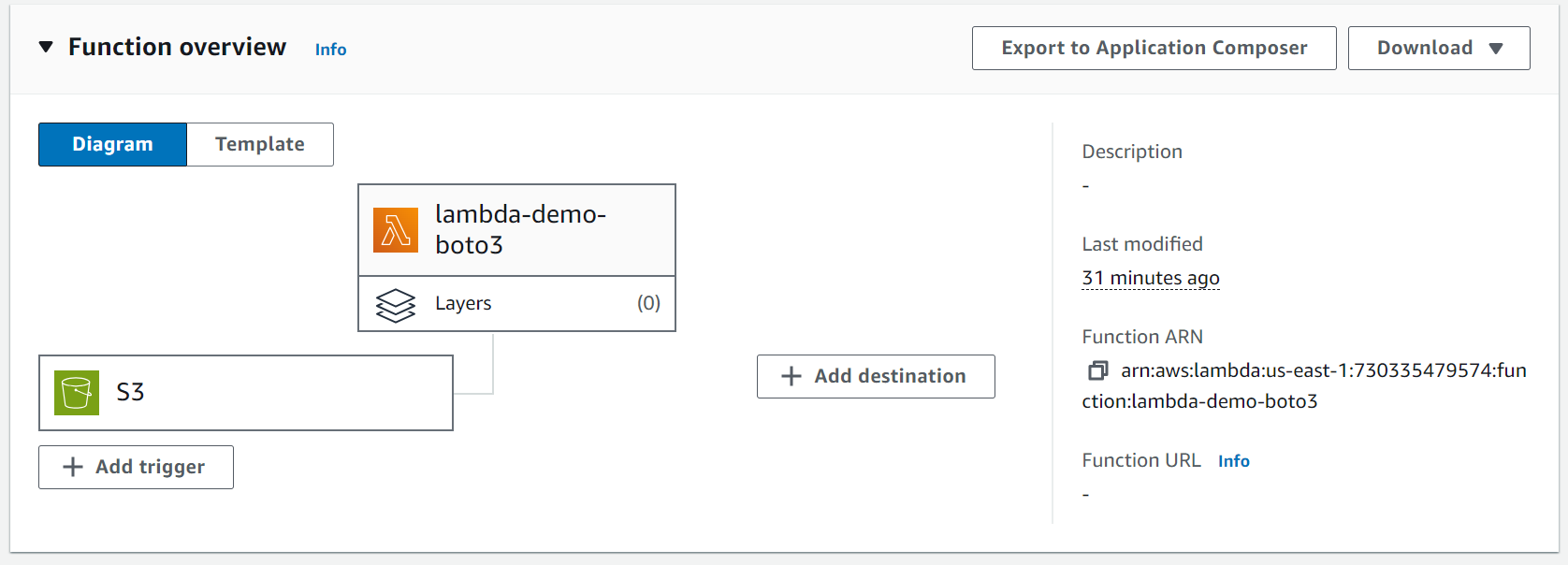
#### **Steps:**

1. **Navigate to IAM** in the AWS Management Console.
2. **Create a Role** for each service:
   1. Select the service (Redshift).
   2. Attach the necessary policies (S3, redshift, admin, ec2).
   3. Note the Role **ARN** for later use in lambda function.

### **4. Creating the Lambda Function**

#### **Steps:**

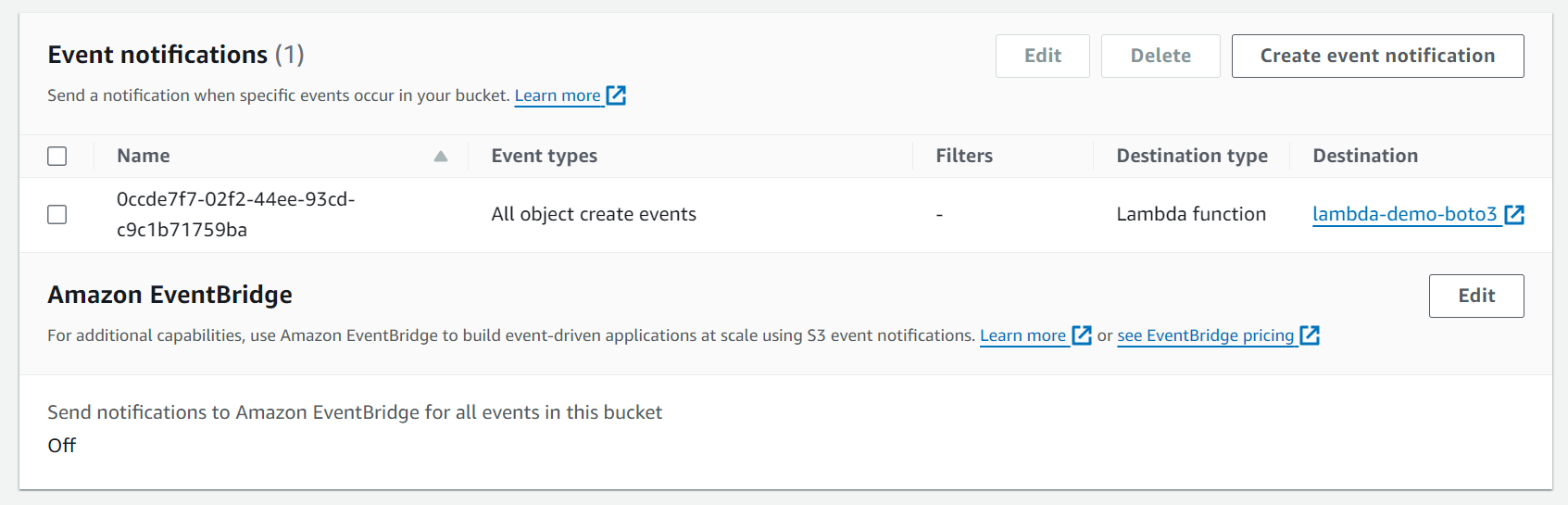
1. **Navigate to AWS Lambda** in the AWS Management Console.
2. **Create Function**:
   1. Click on "Create function".
   2. Choose "Author from scratch".
   3. Configure basic settings (function name, runtime, and role).



1. **Write the Function**:
   1. Use the following template for the Lambda function (using Boto3):

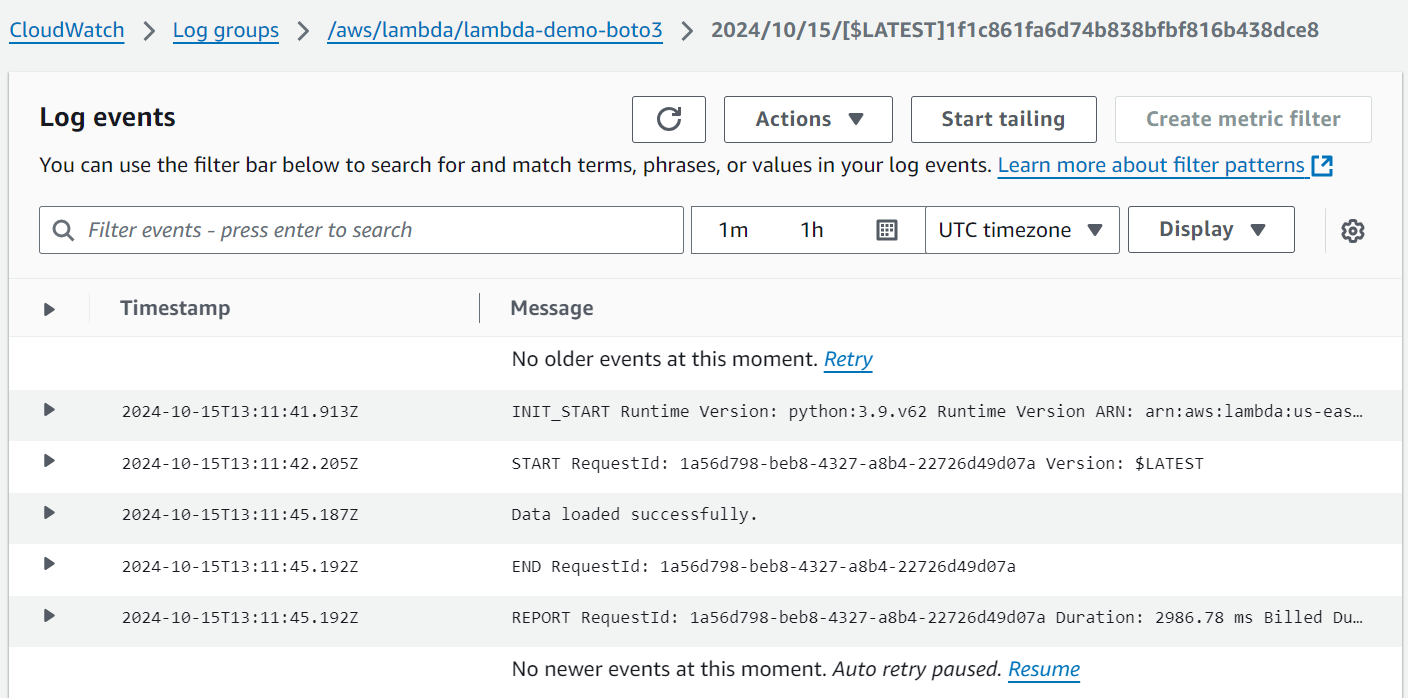


1. **Set Environment Variables**:
   1. Add S3\_BUCKET with the name of the S3 bucket.
2. **Configure Trigger**:
   1. Add an S3 trigger for the Lambda function.
   2. Specify the event type (e.g., "All object create events/PUT/POST").

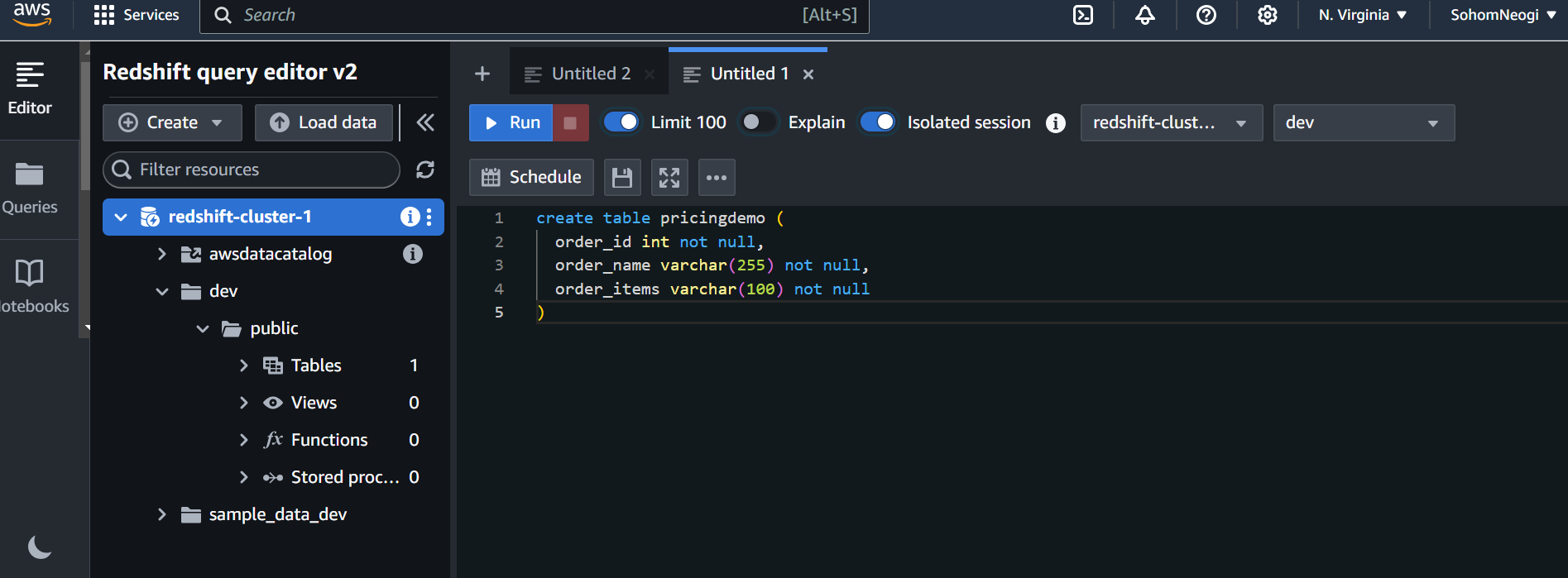


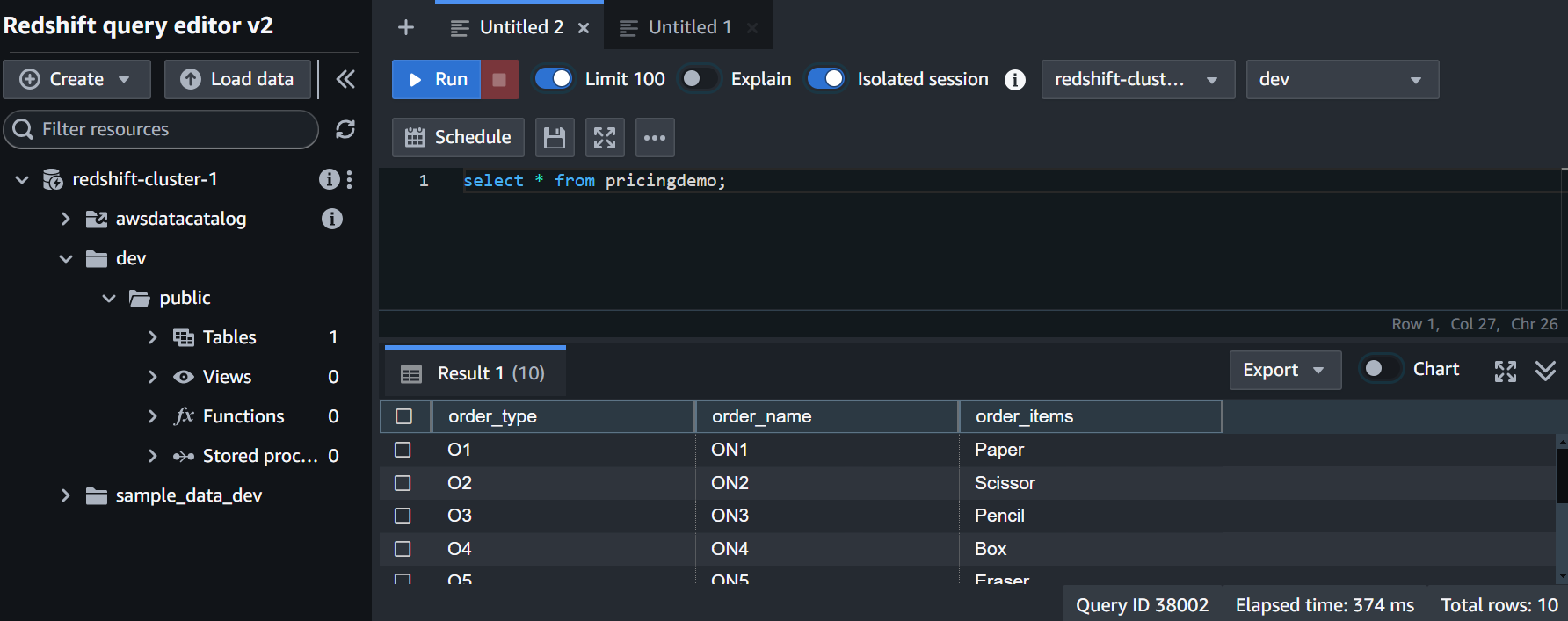
### **5. Testing the Setup**

1. **Upload a File to S3**:
   1. Place a CSV file in the S3 bucket to trigger the Lambda function.
2. **Check Lambda Execution**:
   1. Go to the AWS Lambda console and check the logs in CloudWatch for any errors or success messages.



1. **Verify Data in Redshift**:
   1. Use a SQL client or the Redshift Query Editor to verify that the data has been copied to the table.





### **6. Monitoring and Maintenance**

* **CloudWatch Logs**: Monitor logs for the Lambda function for any errors or issues.
* **Redshift Monitoring**: Use Redshift’s monitoring tools to check for performance and query issues.

### **Conclusion**

This process automates the transfer of SQL data from S3 to Amazon Redshift using AWS Lambda. By following these steps, one can set up a robust data pipeline (ETL Pipeline) that can handle incoming data efficiently. One can adjust the COPY command and IAM roles as needed based on specific requirements and security policies.

**Sources**  
  
 [S3-RS](https://medium.com/@ravi0dubey/etl-pipeline-between-amazon-s3-and-amazon-redshift-using-aws-lambda-function-135f1e3b8a2f)