

In this Hands-on will learn how to create and use Redshift data warehouse

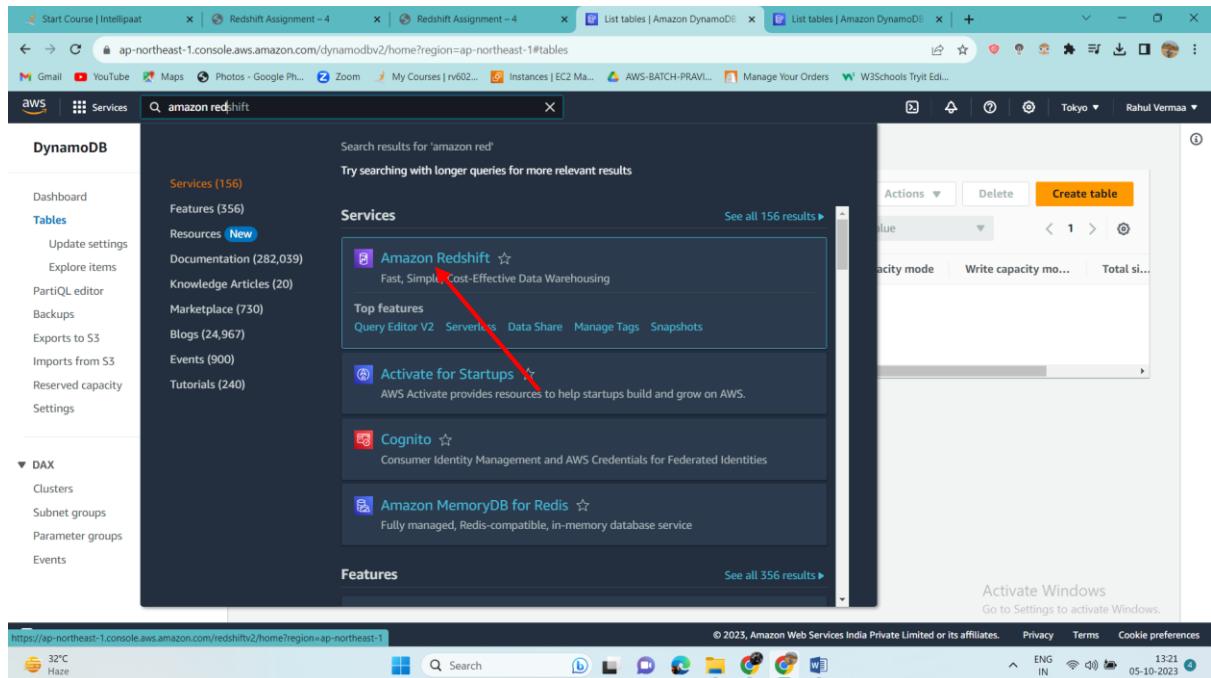
Problem Statement:

You work for XYZ Corporation. Their application requires a database service that can store data which can be retrieved if required. Implement suitable service for the same.

While migrating, you are asked to perform the following tasks:

1. Create a Redshift data warehouse.
2. Using the query editor:
 - a. Load some data
 - b. Query the data

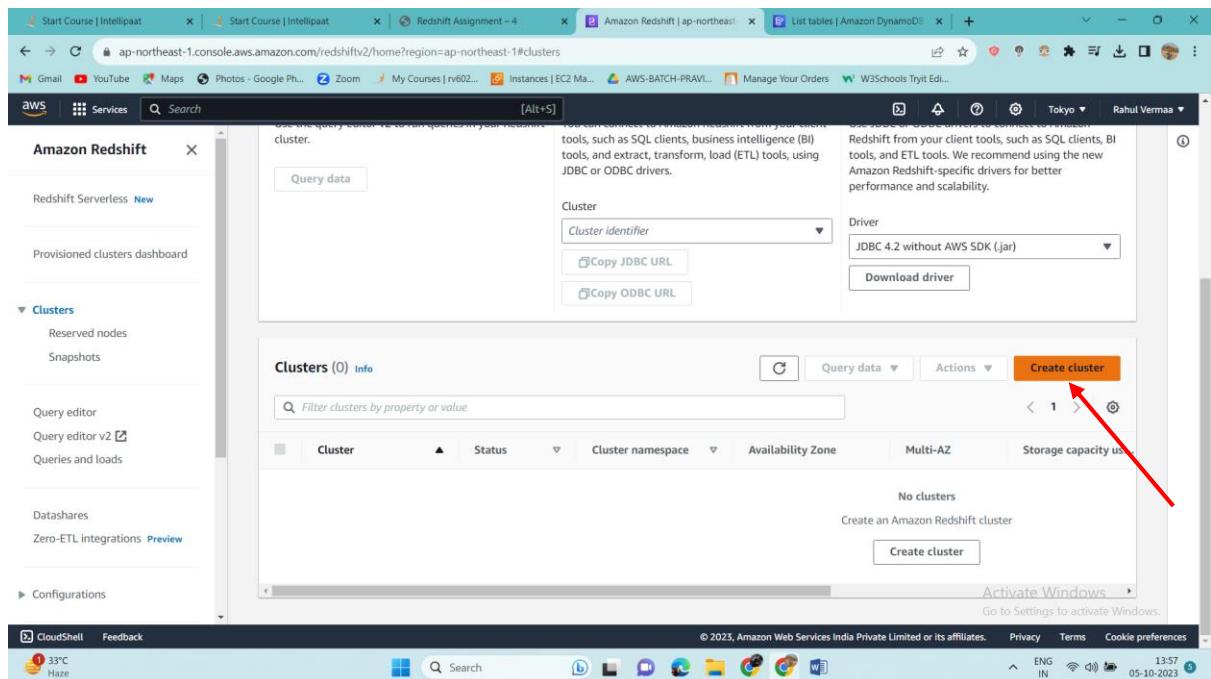
Step 1: First let's search for amazon redshift in our AWS console



The screenshot shows the AWS search interface with the query 'amazon redshift'. The results are categorized into 'Services' and 'Features'. The 'Amazon Redshift' service is highlighted with a red arrow pointing to it. The service card includes a star icon, a brief description: 'Fast, Simple, Cost-Effective Data Warehousing', and a 'Top features' section with links to 'Query Editor V2', 'Servers', 'Data Share', 'Manage Tags', and 'Schemas'. Below the service card, there are cards for 'Activate for Startups', 'Cognito', and 'Amazon MemoryDB for Redis'.

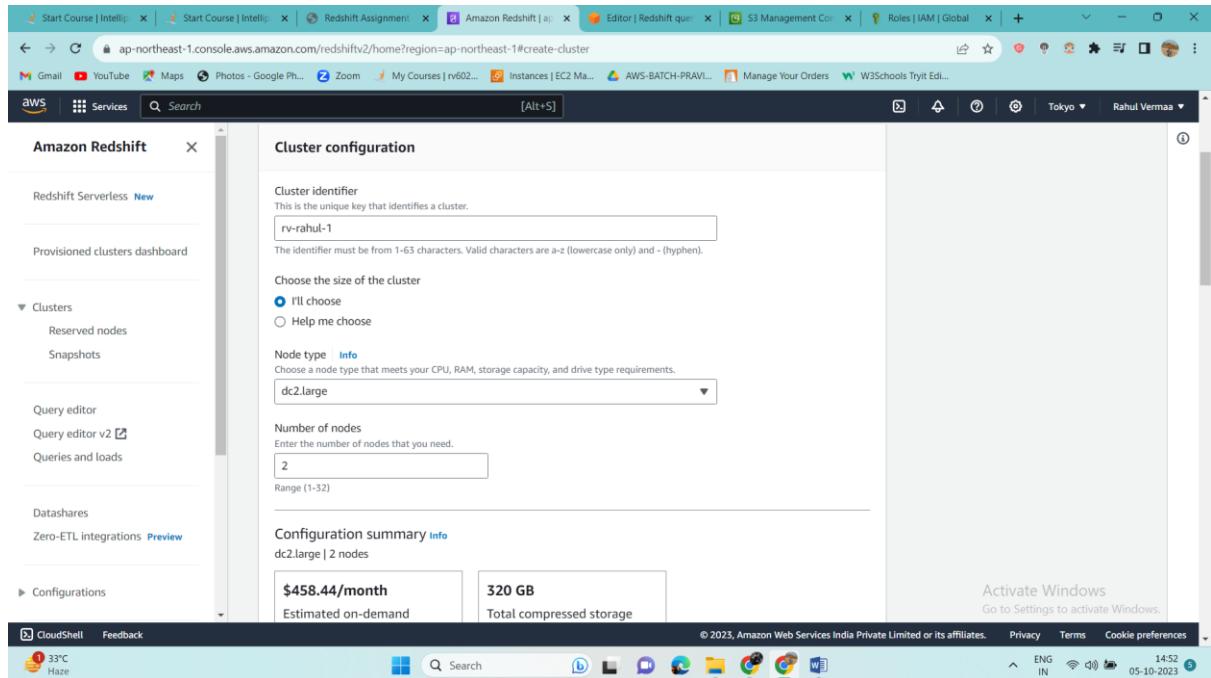
And here we have to create cluster

(We can say group of cluster is warehouse)



The screenshot shows the 'Amazon Redshift' service dashboard. The left sidebar has 'Clusters' selected, showing 'Reserved nodes' and 'Schemas'. The main area is titled 'Create a new cluster' and contains a 'Cluster identifier' input field, a 'Copy JDBC URL' button, and a 'Copy ODBC URL' button. Below this is a 'Clusters (0) Info' table with columns: Cluster, Status, Cluster namespace, Availability Zone, Multi-AZ, and Storage capacity. A red arrow points to the 'Create cluster' button at the top right of the table. The status bar at the bottom indicates '33°C Haze'.

Step 2: Enter details and select options



Cluster configuration

Cluster identifier: rv-rahul-1

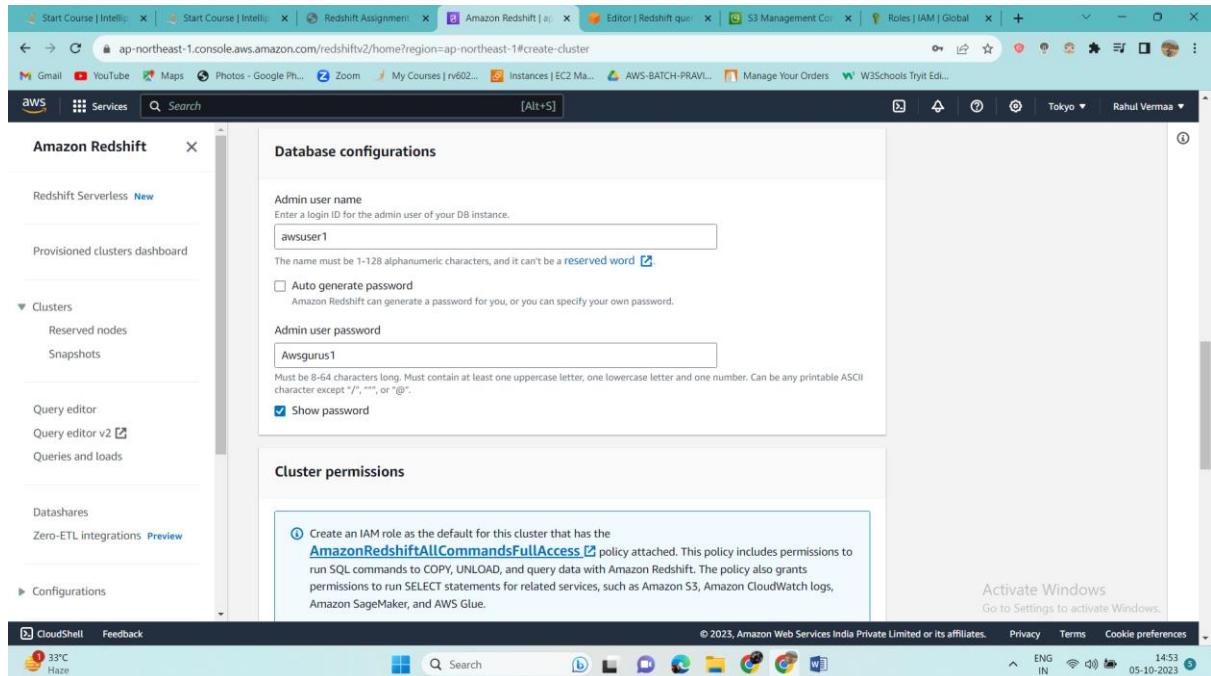
Choose the size of the cluster: I'll choose

Node type: dc2.large

Number of nodes: 2

Configuration summary: \$458.44/month, 320 GB

Password- Awsgurus1



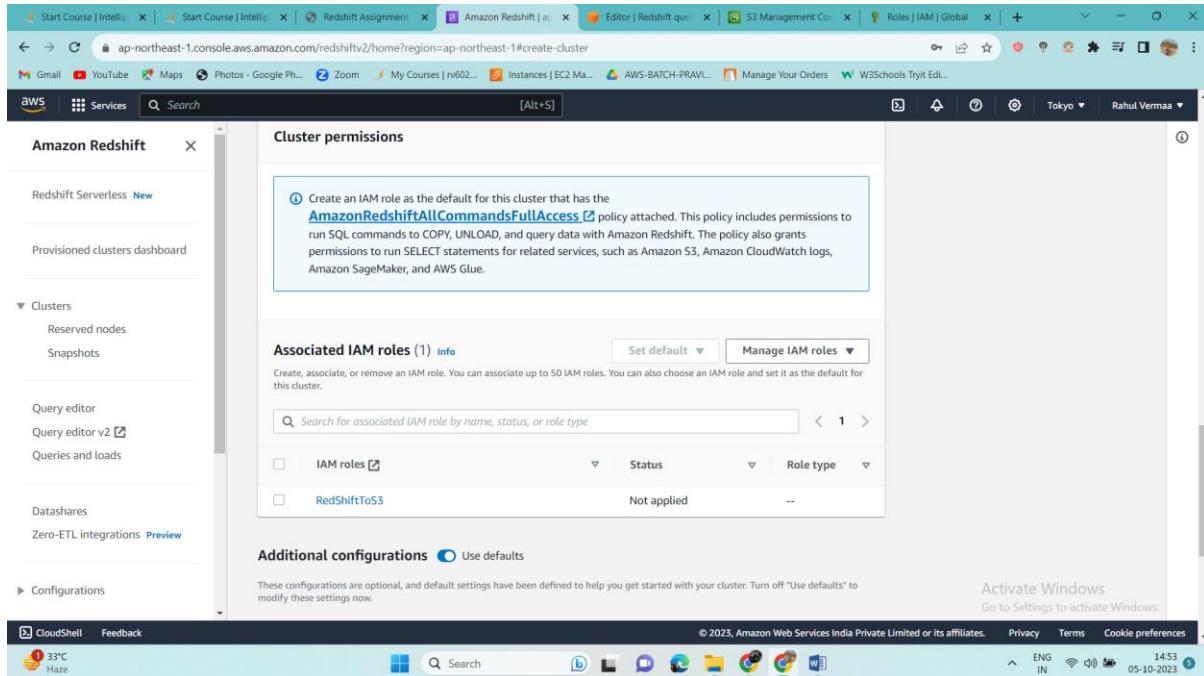
Database configurations

Admin user name: awsuser1

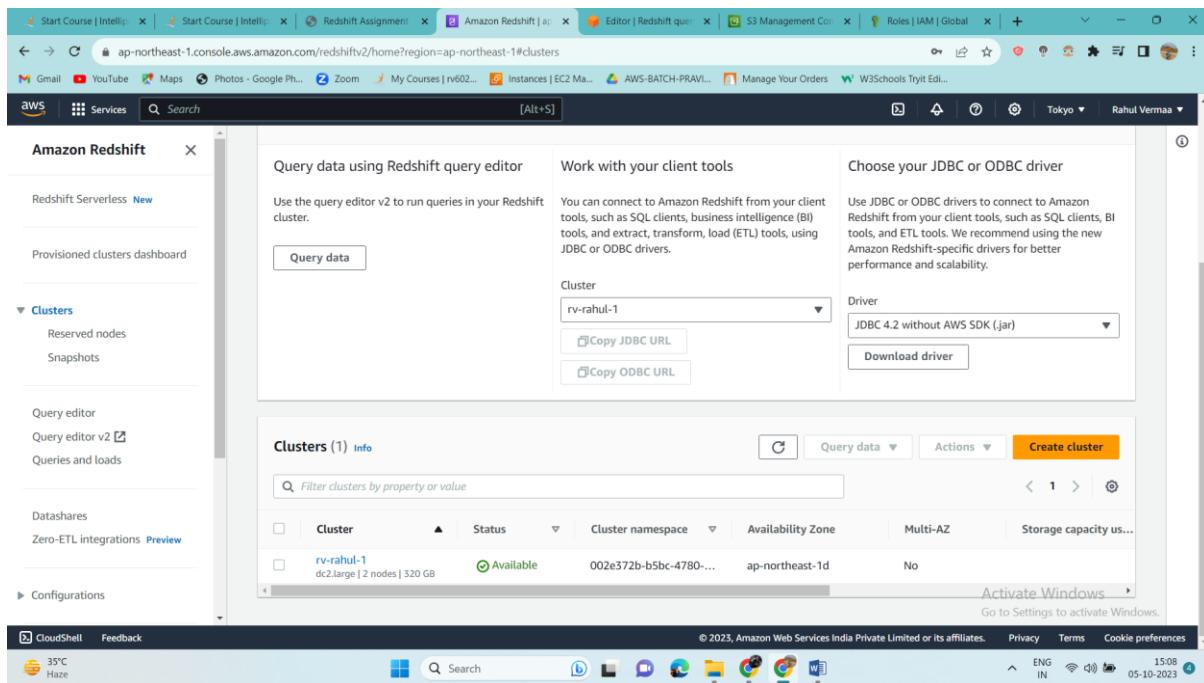
Admin user password: Awsgurus1

Cluster permissions: Create an IAM role as the default for this cluster that has the [AmazonRedshiftAllCommandsFullAccess](#) policy attached. This policy includes permissions to run SQL commands to COPY, UNLOAD, and query data with Amazon Redshift. The policy also grants permissions to run SELECT statements for related services, such as Amazon S3, Amazon CloudWatch logs, Amazon SageMaker, and AWS Glue.

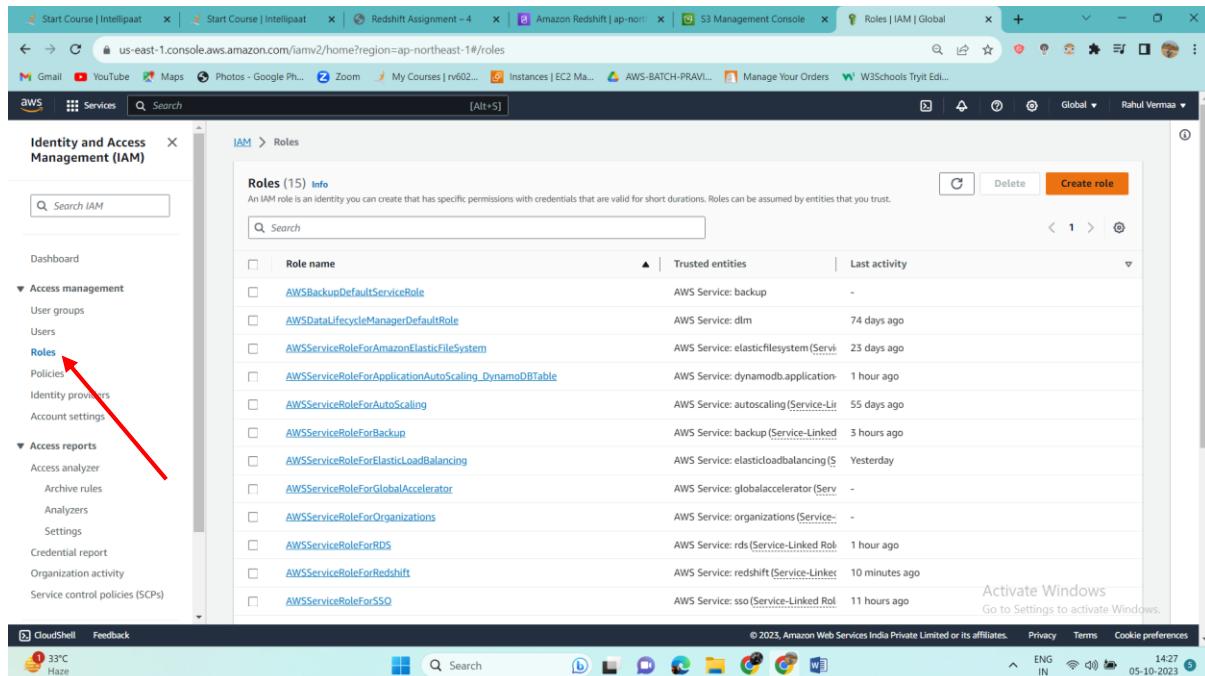
Step 3: We will associate IAM Role which we have already created, steps for creating IAM role I'll add after creation of clusters.



Cluster is created successfully

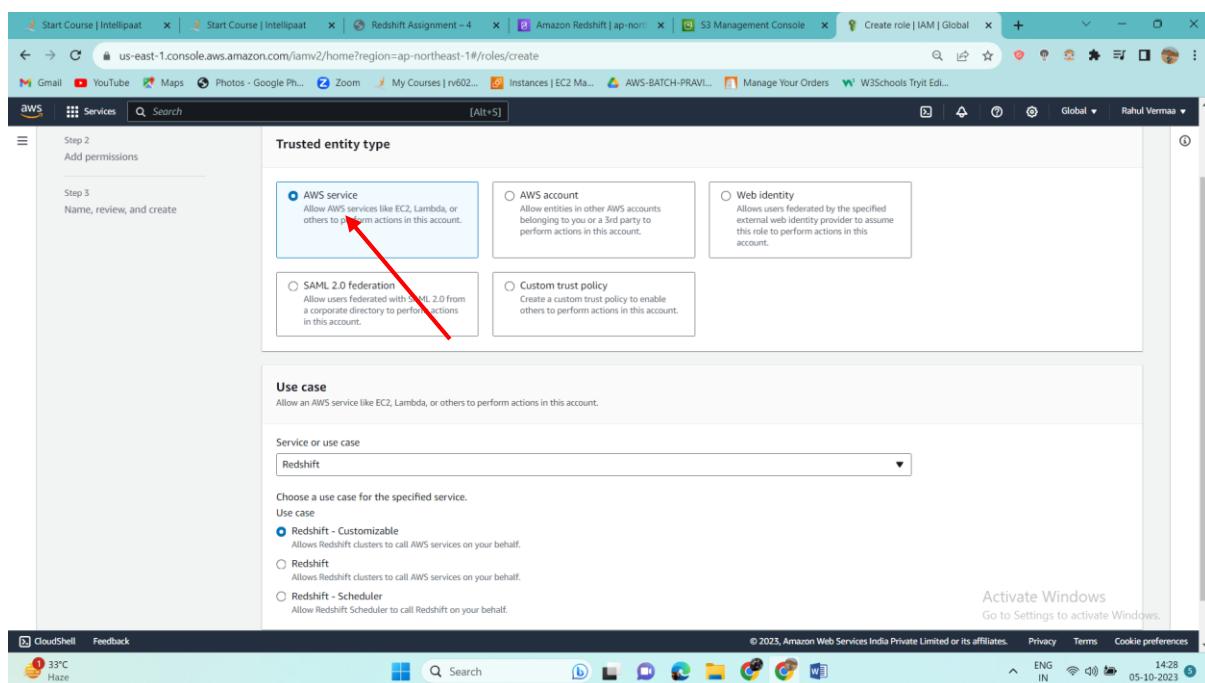


Step 4: So to access privately we need to create one IAM role. So just search IAM in search bar, go to roles and click it on create role



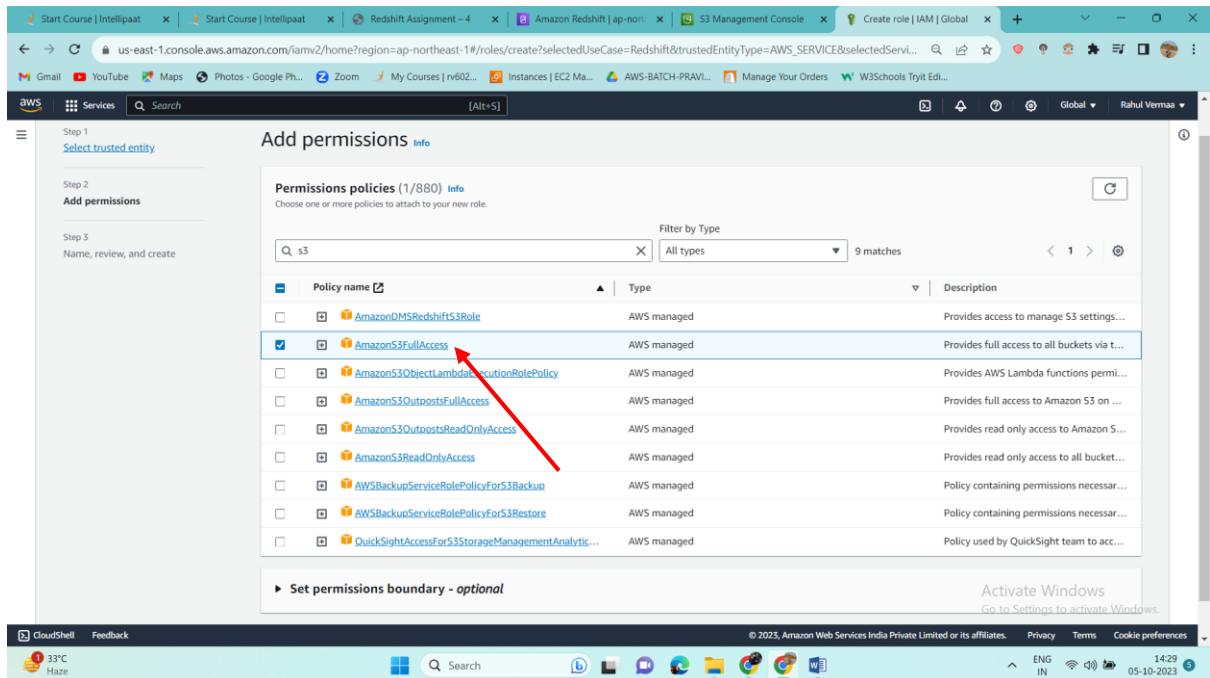
The screenshot shows the AWS IAM Roles list page. The left sidebar is expanded, showing 'Access management' with 'Roles' selected. A red arrow points to the 'Roles' link. The main content area shows a table of 15 existing IAM roles, each with a 'Role name', 'Trusted entities', and 'Last activity'. The roles listed include AWSBackupDefaultServiceRole, AWSDataLifecycleManagerDefaultRole, AWSServiceRoleForAmazonElasticFileSystem, AWSRoleForApplicationAutoScaling_DynamoDBTable, AWSRoleForAutoScaling, AWSRoleForBackup, AWSRoleForElasticLoadBalancing, AWSRoleForGlobalAccelerator, AWSRoleForOrganizations, AWSRoleForRDS, AWSRoleForRedshift, and AWSRoleForSSO. The table has columns for 'Role name', 'Trusted entities', and 'Last activity'. The 'Create role' button is visible in the top right corner of the table area.

So these are the options I'm going to select for my IAM role



The screenshot shows the 'Create role' wizard, Step 2: Add permissions. The left sidebar shows 'Step 2: Add permissions' and 'Step 3: Name, review, and create'. The main content area is titled 'Trusted entity type' and contains four options: 'AWS Service' (selected), 'AWS account', 'Web Identity', and 'SAML 2.0 federation'. Below this is a 'Use case' section with a dropdown for 'Service or use case' set to 'Redshift'. Under 'Use case', there are three options: 'Redshift - Customizable' (selected), 'Redshift', and 'Redshift - Scheduler'. The 'Activate Windows' watermark is visible in the bottom right corner.

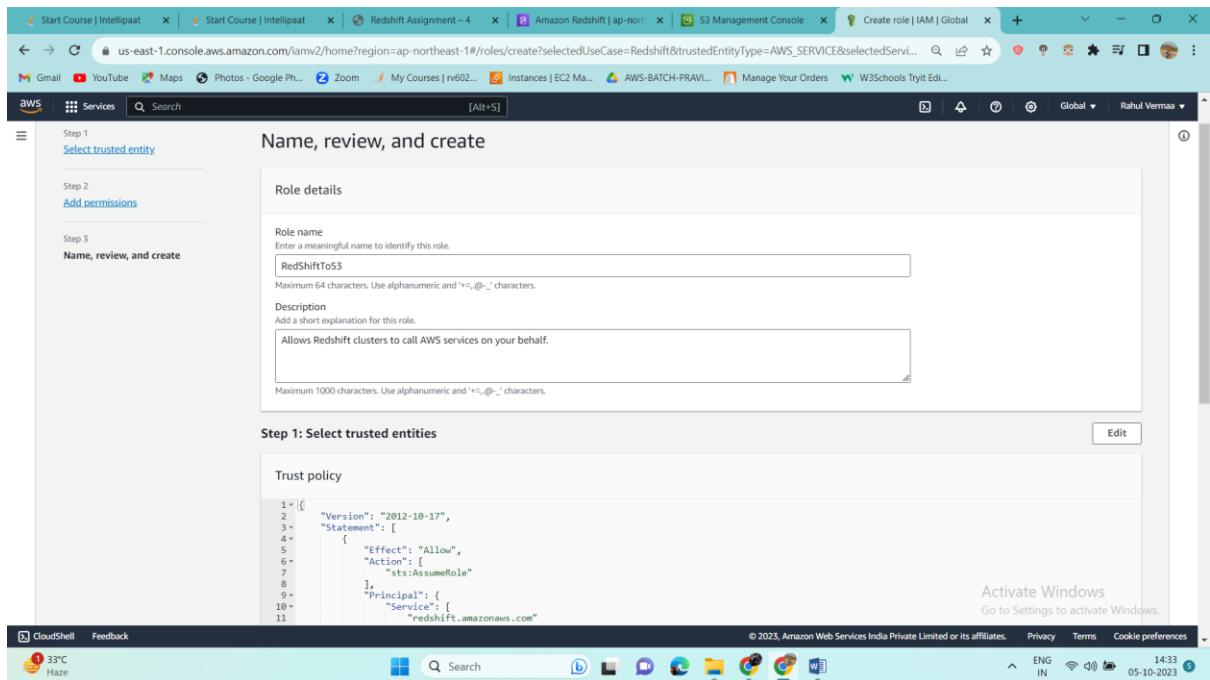
Step 5: Now in permissions policies will give AmazonS3FullAccess



The screenshot shows the 'Add permissions' step in the AWS IAM console. The search bar at the top has 's3' typed into it. Below the search bar, a table lists various AWS managed policies. The 'AmazonS3FullAccess' policy is highlighted with a red arrow pointing to its checkbox. The table includes columns for 'Policy name', 'Type', and 'Description'.

Policy name	Type	Description
AmazonDMSRedshiftS3Role	AWS managed	Provides access to manage S3 settings...
AmazonS3FullAccess	AWS managed	Provides full access to all buckets via t...
AmazonS3ObjectLambdaExecutionRolePolicy	AWS managed	Provides AWS Lambda functions permi...
AmazonS3OutpostsFullAccess	AWS managed	Provides full access to Amazon S3 on ...
AmazonS3OutpostsReadOnlyAccess	AWS managed	Provides read only access to Amazon S...
AmazonS3ReadOnlyAccess	AWS managed	Provides read only access to all bucket...
AWSBackupServiceRolePolicyForS3Backup	AWS managed	Policy containing permissions necessar...
AWSBackupServiceRolePolicyForS3Restore	AWS managed	Policy containing permissions necessar...
QuickSightAccessForS3StorageManagementAnalytic...	AWS managed	Policy used by QuickSight team to acc...

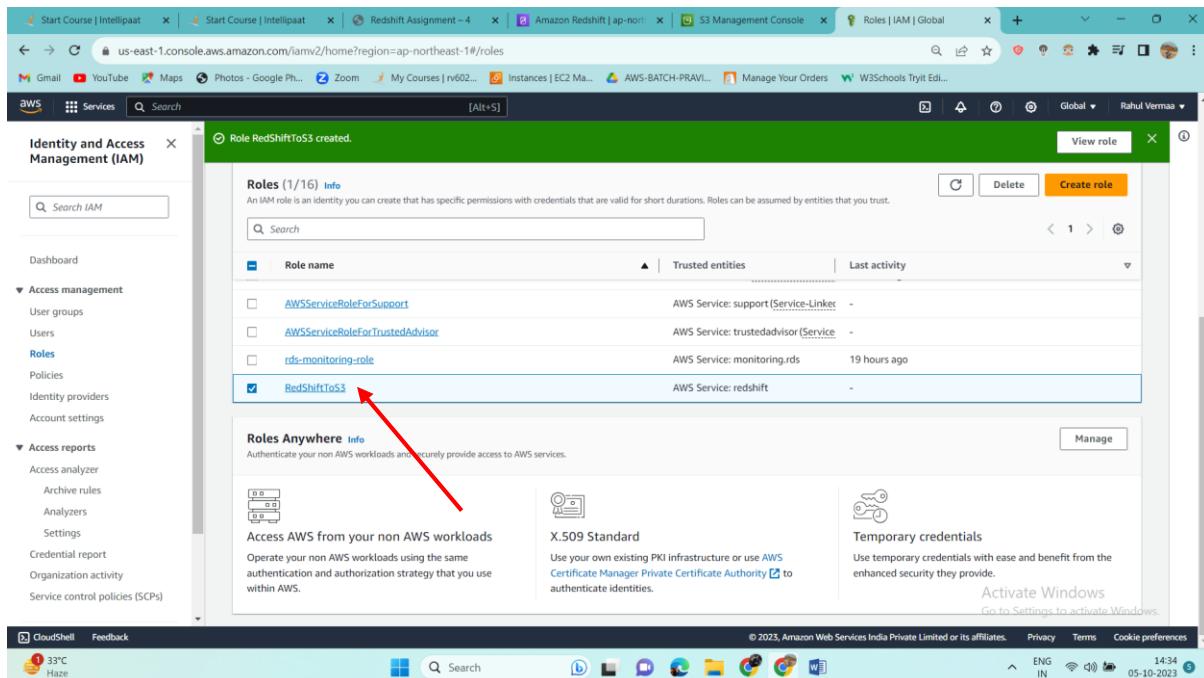
Now enter details, scroll down and click on create role button



The screenshot shows the 'Name, review, and create' step in the AWS IAM console. In the 'Role details' section, the 'Role name' is set to 'RedShiftToS3' and the 'Description' is 'Allows Redshift clusters to call AWS services on your behalf.'. In the 'Step 1: Select trusted entities' section, a JSON trust policy is displayed. The policy allows the 'sts:AssumeRole' action for the principal 'redshift.amazonaws.com'.

```
1+ [{  
2+     "Version": "2012-10-17",  
3+     "Statement": [  
4+         {  
5+             "Effect": "Allow",  
6+             "Action": [  
7+                 "sts:AssumeRole"  
8+             ],  
9+             "Principal": [  
10+                 "redshift.amazonaws.com"  
11+             ]  
12+         }  
13+     ]  
14+ }
```

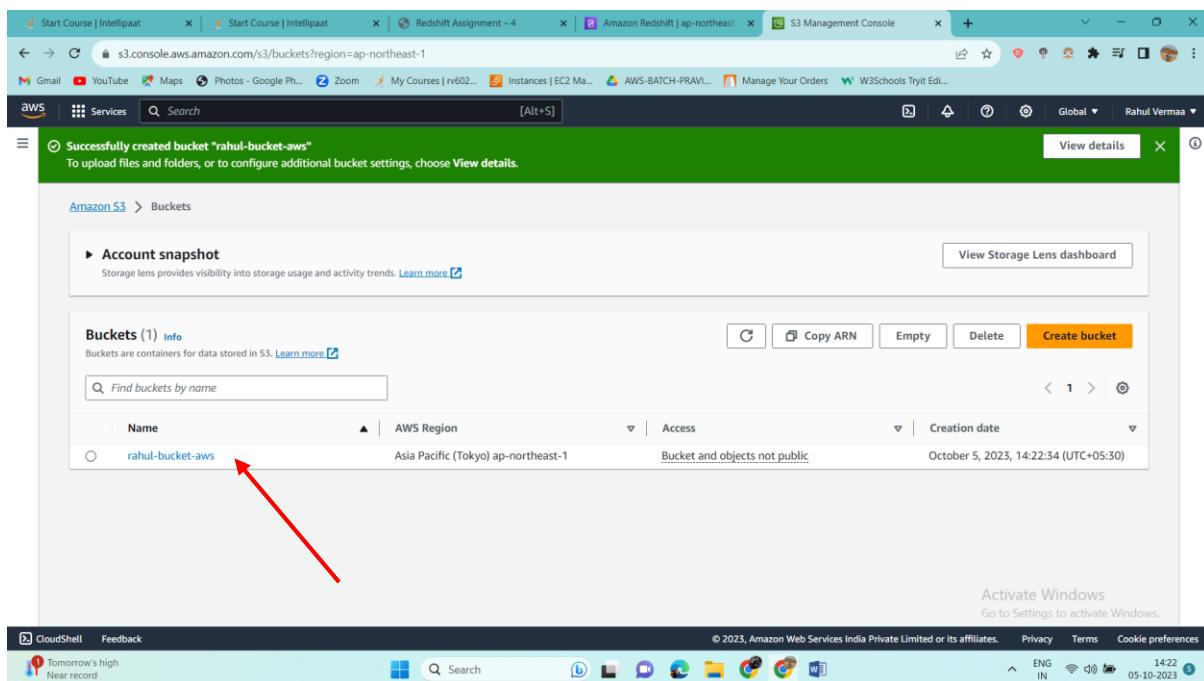
Our role is created-> RedShiftToS3



The screenshot shows the AWS IAM Roles page. A red arrow points to the 'RedShiftToS3' role in the list, which is highlighted with a blue border. The role details are as follows:

Role name	Trusted entities	Last activity
AWSServiceRoleForSupport	AWS Service: support (Service-Linked)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked)	-
rds-monitoring-role	AWS Service: monitoring.rds	19 hours ago
RedShiftToS3	AWS Service: redshift	-

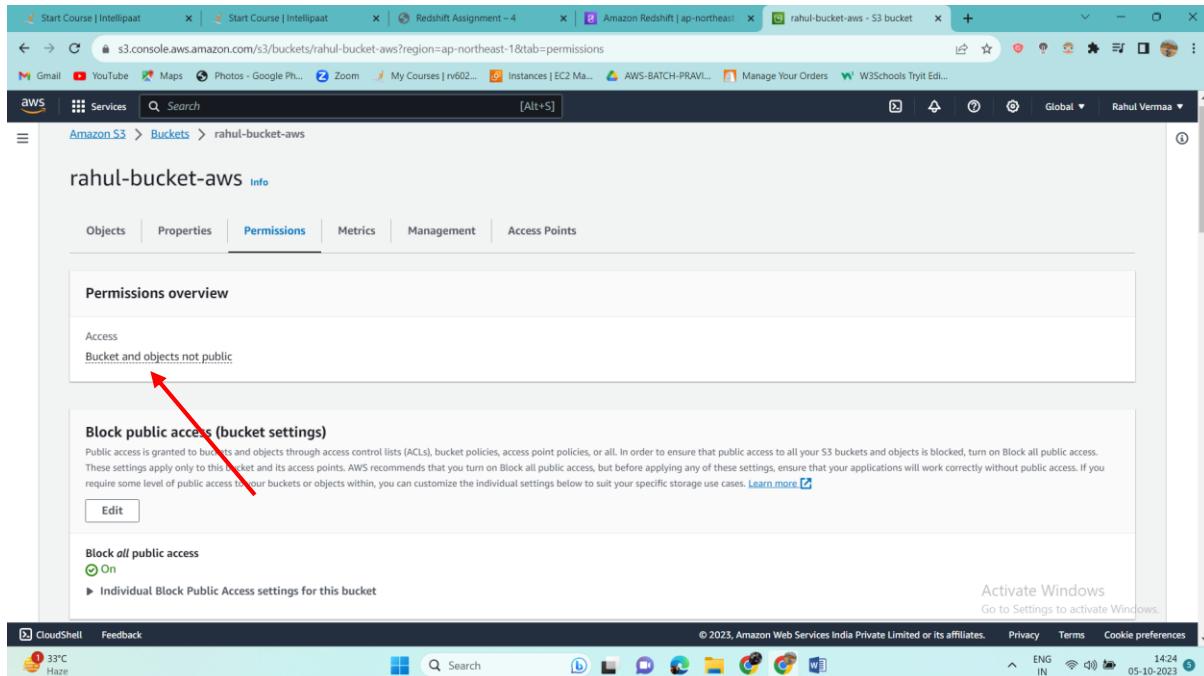
Step 6: Now we will create one s3 bucket and upload some data in that bucket-> Rahul-bucket-aws



The screenshot shows the AWS S3 Buckets page. A red arrow points to the 'rahul-bucket-aws' bucket in the list, which is highlighted with a blue border. The bucket details are as follows:

Name	AWS Region	Access	Creation date
rahul-bucket-aws	Asia Pacific (Tokyo) ap-northeast-1	Bucket and objects not public	October 5, 2023, 14:22:34 (UTC+05:30)

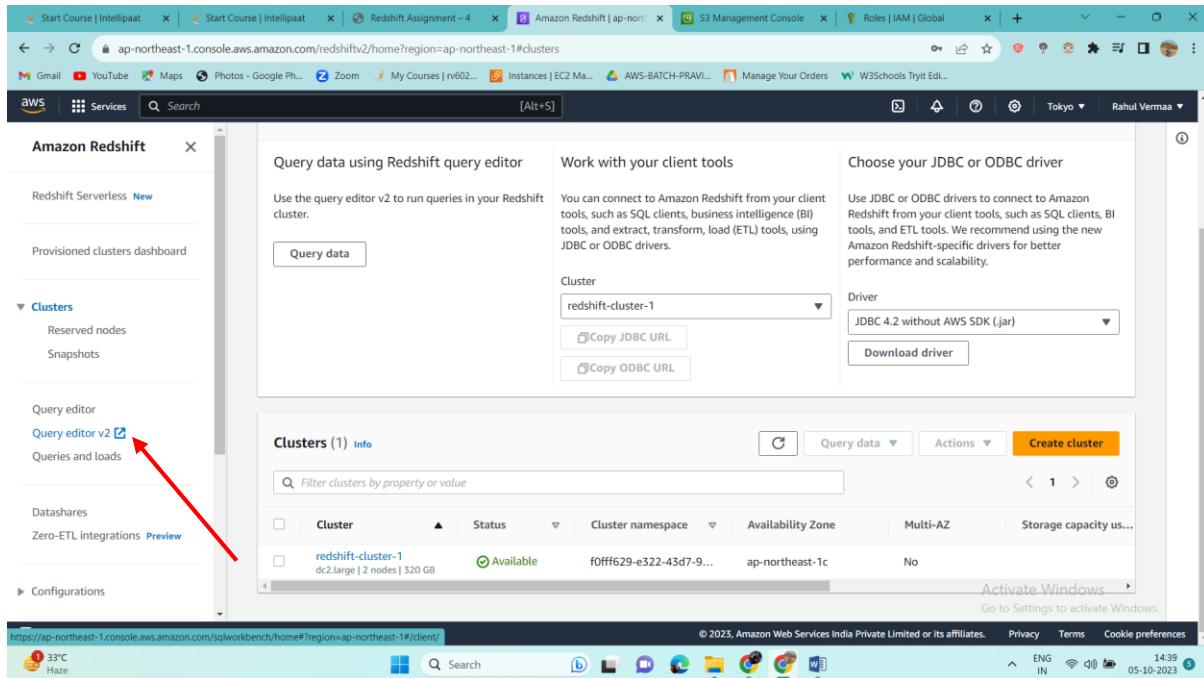
Our bucket is private



The screenshot shows the AWS S3 console with the 'Permissions' tab selected for the 'rahul-bucket-aws' bucket. The 'Access' section displays the message 'Bucket and objects not public'. A red arrow points to this message. Below it, the 'Block public access (bucket settings)' section is shown, with the 'Block all public access' toggle set to 'On'. The status bar at the bottom indicates 'CloudShell Feedback' and the date '05-10-2023'.

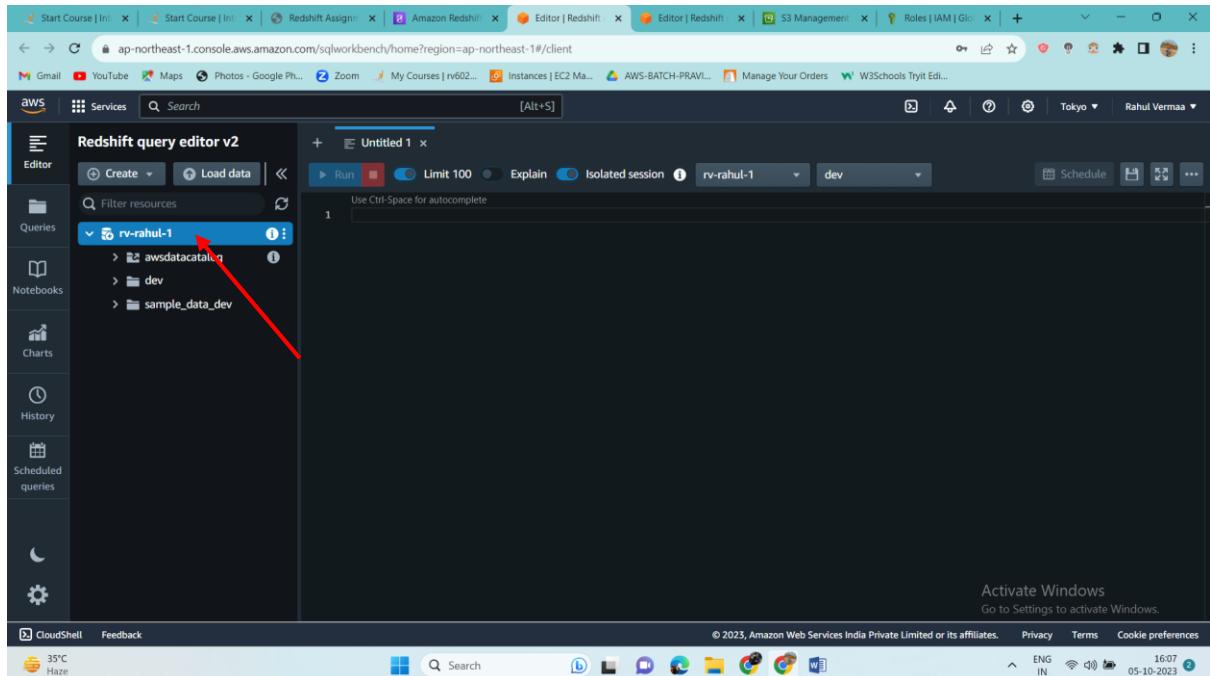
Step 7: Now to get inside of our cluster will use

Query editor v2 (Here we need not have to do connection manually)

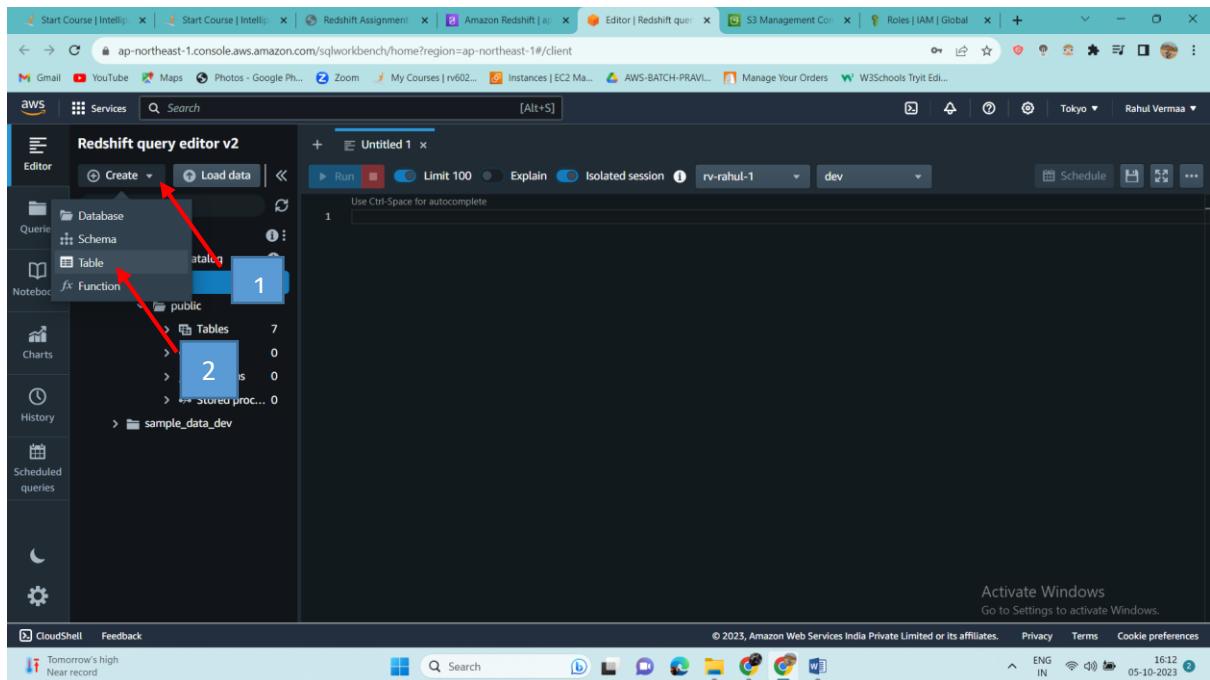


The screenshot shows the AWS Redshift Management Console. The sidebar on the left has a 'Clusters' section with 'Query editor v2' highlighted. A red arrow points to this link. The main content area shows the 'Query data using Redshift query editor' section and a 'Clusters (1) Info' table. The table lists a single cluster: 'redshift-cluster-1' (Status: Available, Cluster namespace: f0fff629-e322-43d7-9..., Availability Zone: ap-northeast-1c, Multi-AZ: No). The status bar at the bottom indicates 'CloudShell Feedback' and the date '05-10-2023'.

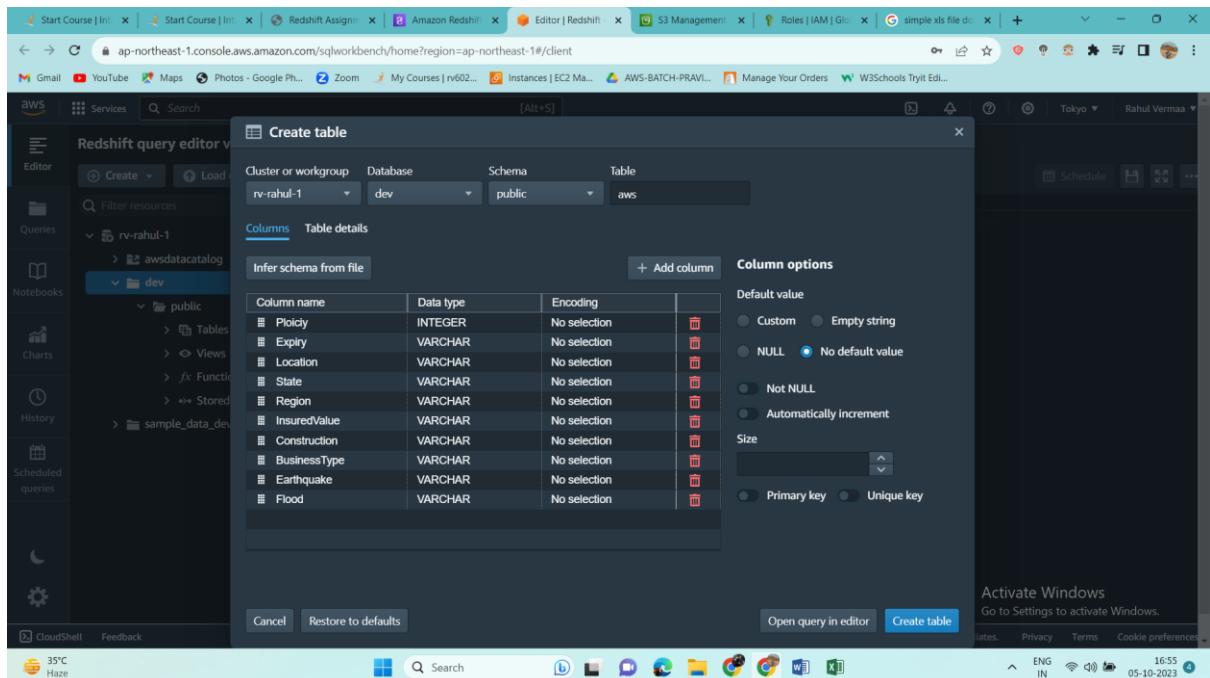
And you can see it's showing our cluster



Step 8: Now let's create one table

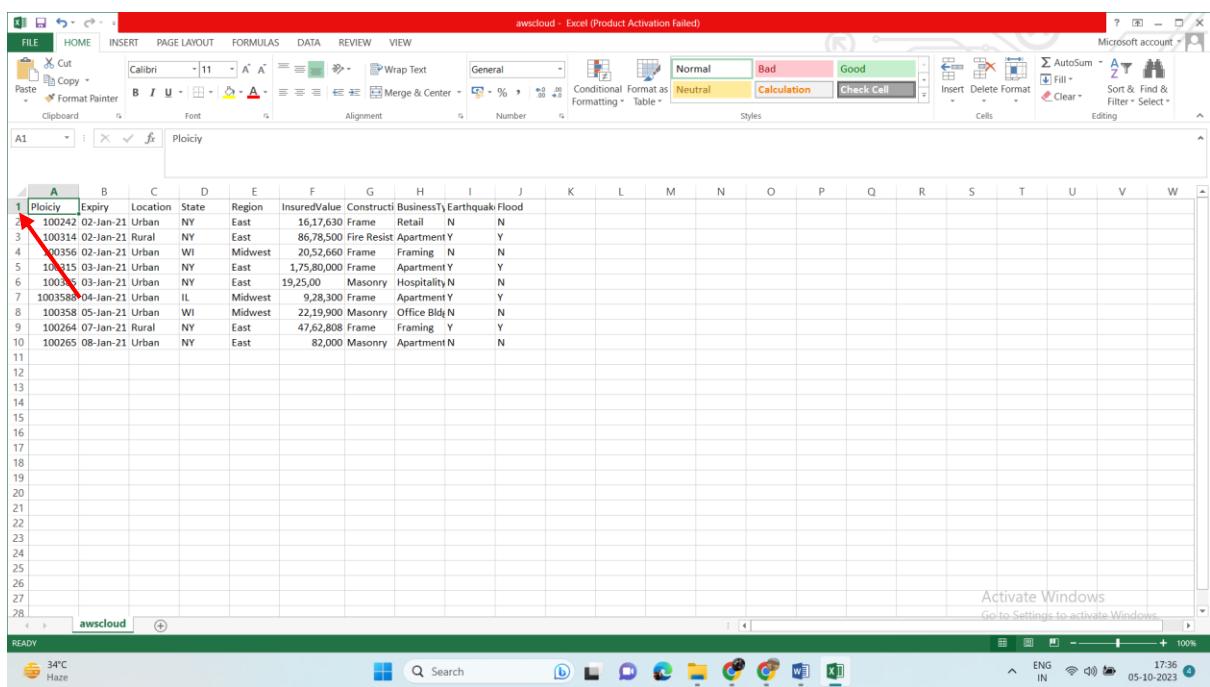


We have load one CSV file



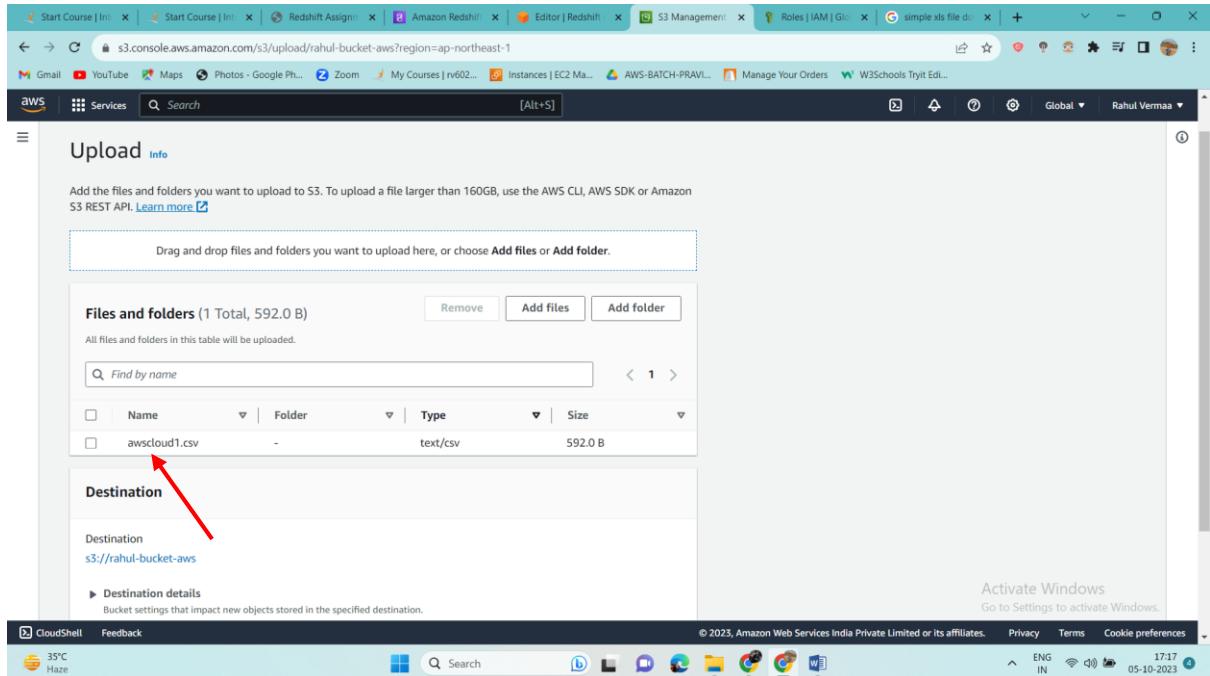
The screenshot shows the AWS Redshift Query Editor interface. A modal window titled 'Create table' is open. In the 'Table' section, the 'Cluster or workgroup' is 'rv-rahl-1', 'Database' is 'dev', 'Schema' is 'public', and the 'Table' is 'aws'. The 'Columns' section lists 11 columns: Ploicy, Expiry, Location, State, Region, InsuredValue, Constructi, BusinessTy, Earthquak, and Flood. The 'Column options' section includes settings for 'Default value' (radio buttons for 'Custom', 'NULL', 'Not NULL', and 'Automatically increment'), 'Size' (a dropdown menu), and checkboxes for 'Primary key' and 'Unique key'. The 'Create table' button is at the bottom right of the modal.

Content in our awscloud CSV file which we have uploaded in our table (1st row is our Header)



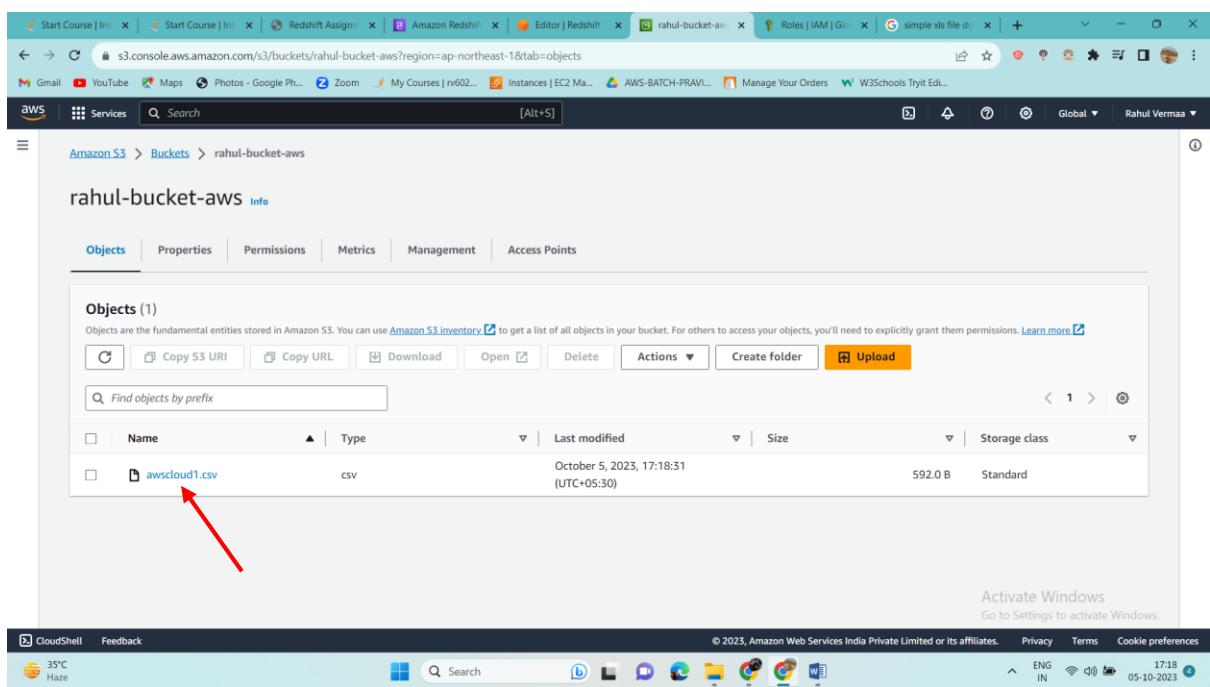
The screenshot shows a Microsoft Excel spreadsheet titled 'awscloud'. The first row contains the column headers: Ploicy, Expiry, Location, State, Region, InsuredValue, Constructi, BusinessTy, Earthquak, and Flood. The data starts from row 2, with the first data row (row 2) highlighted in yellow. The data rows show various policy details such as Expiry date, Location, State, Region, InsuredValue, and Construction type. The 'FILE', 'HOME', and 'VIEW' tabs are visible at the top, along with various Excel ribbon icons. The status bar at the bottom shows 'Activate Windows' and the date '05-10-2023'.

Step 9: Now let's add some file to our s3 bucket and will try to load that file on our cluster from s3



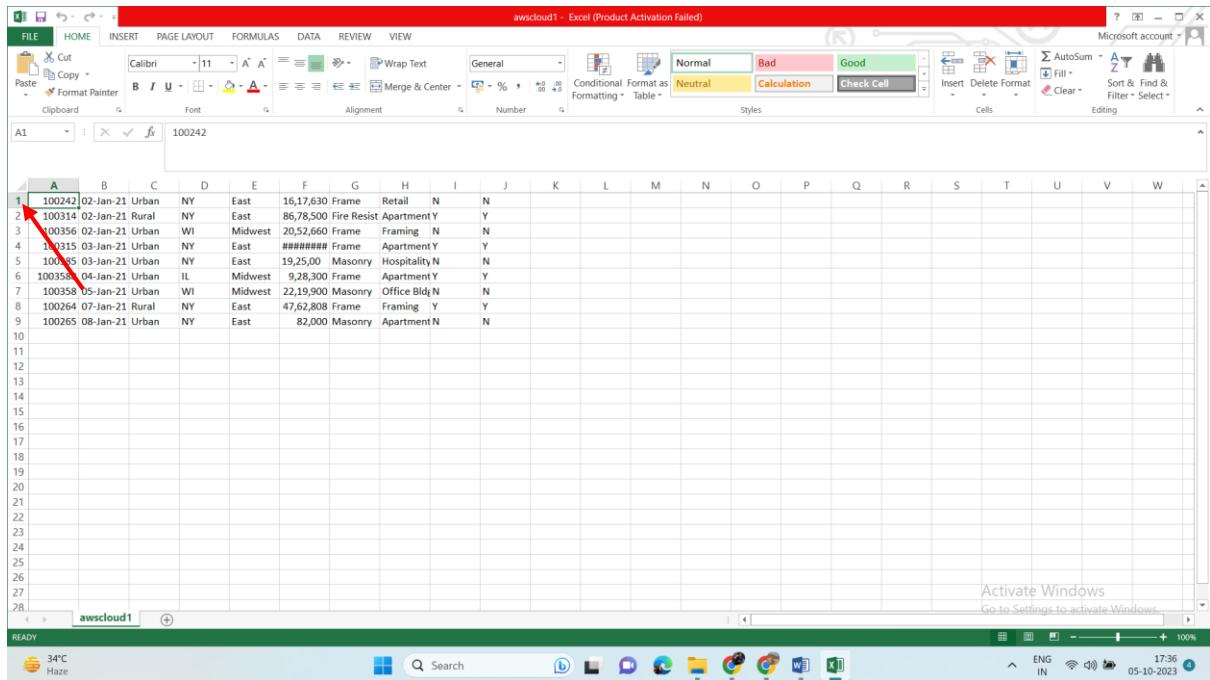
The screenshot shows the AWS S3 console with the 'Upload' interface. A red arrow points to the 'awscloud1.csv' file in the 'Files and folders' list, which contains one item: 'awscloud1.csv' (text/csv, 592.0 B). The 'Destination' section shows 's3://rahul-bucket-aws'.

File is uploaded successfully



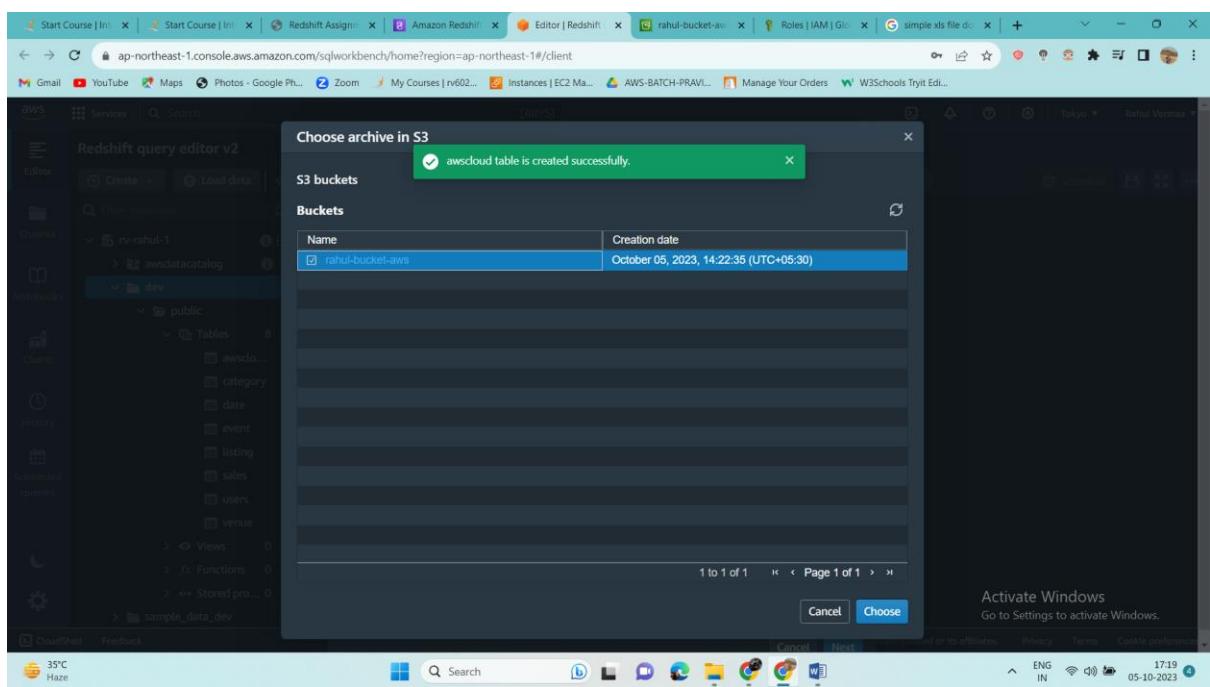
The screenshot shows the AWS S3 console with the 'Objects' tab selected. A red arrow points to the 'awscloud1.csv' file in the 'Objects (1)' list, which contains one item: 'awscloud1.csv' (csv, 592.0 B, Standard storage class). The file was last modified on October 5, 2023, at 17:18:31 (UTC+05:30).

Content in our awscloud1 csv file which we have uploaded in s3 bucket (Header is not there)



1	100242	02-Jan-21	Urban	NY	East	16,17,630	Frame	Retail	N	N
2	100314	02-Jan-21	Rural	NY	East	86,78,500	Fire Resist	Apartment	Y	Y
3	100356	02-Jan-21	Urban	WI	Midwest	20,52,660	Frame	Framing	N	N
4	100315	03-Jan-21	Urban	NY	East	#####	Frame	Apartment	Y	Y
5	100385	03-Jan-21	Urban	NY	East	19,25,000	Masonry	Hospitality	N	N
6	100359	04-Jan-21	Urban	IL	Midwest	9,28,300	Frame	Apartment	Y	Y
7	100358	05-Jan-21	Urban	WI	Midwest	22,19,900	Masonry	Office Bld	N	N
8	100264	07-Jan-21	Rural	NY	East	47,62,808	Frame	Framing	Y	Y
9	100265	08-Jan-21	Urban	NY	East	82,000	Masonry	Apartment	N	N
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24										
25										
26										
27										
28										

Step 10: Now will browse s3 and select our s3 bucket



Choose archive in S3

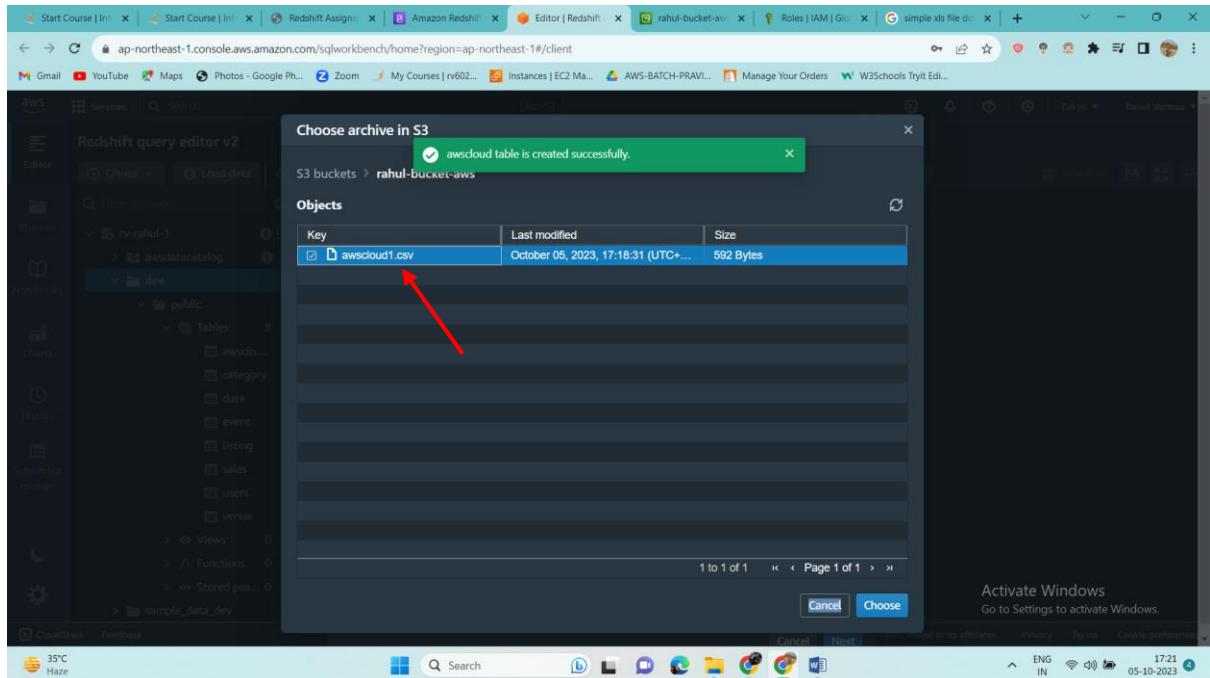
awscloud table is created successfully.

S3 buckets

Name	Creation date
rahul-bucket-aws	October 05, 2023, 14:22:35 (UTC+05:30)

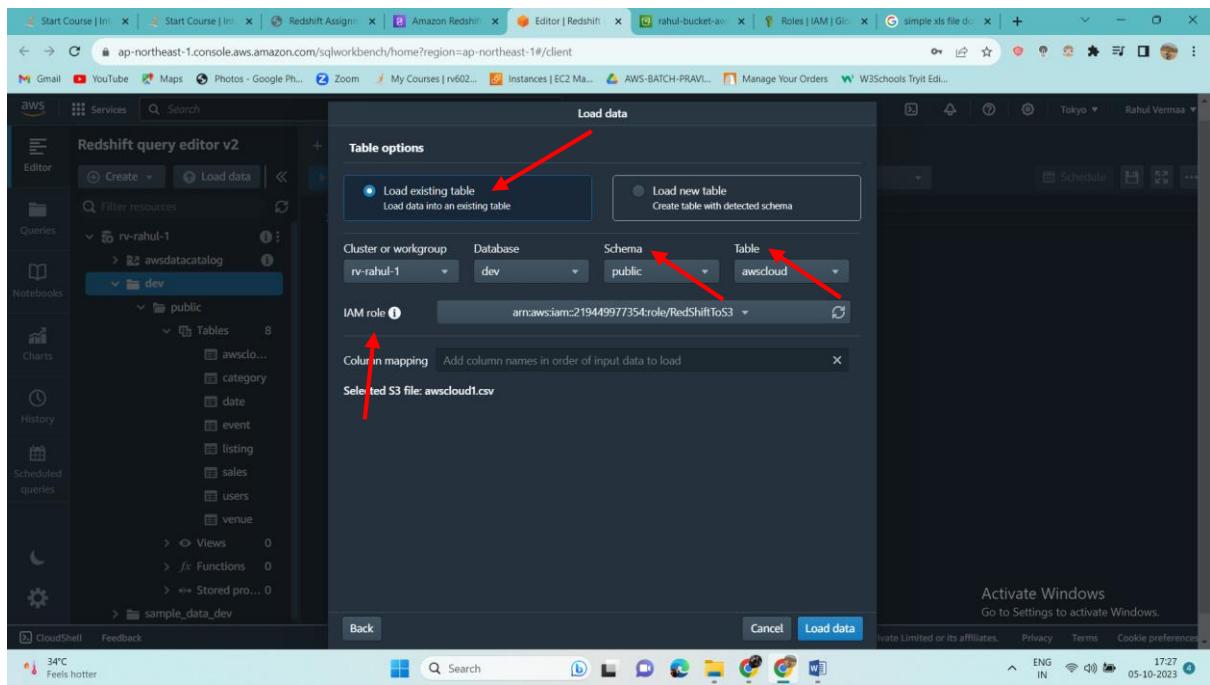
Cancel Choose

And will go inside s3 and select our uploaded file



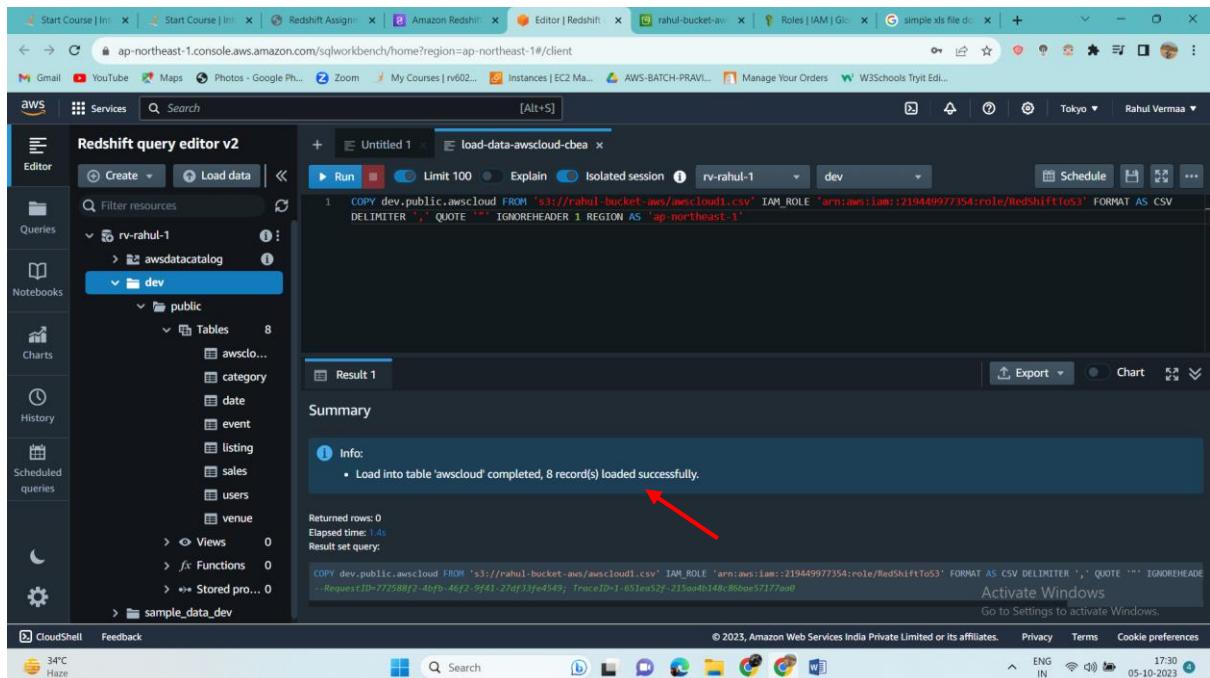
The screenshot shows the Redshift query editor interface. A modal window titled 'Choose archive in S3' is open, displaying a list of objects in the 'rahul-bucket-aws' bucket. The list includes a single item: 'awscloud1.csv'. A red arrow points to this file. The background shows the Redshift schema and table structure for the 'dev' database.

Step 11: And now will select schema- public, table- awscloud and will select our IAM role also and click on load data



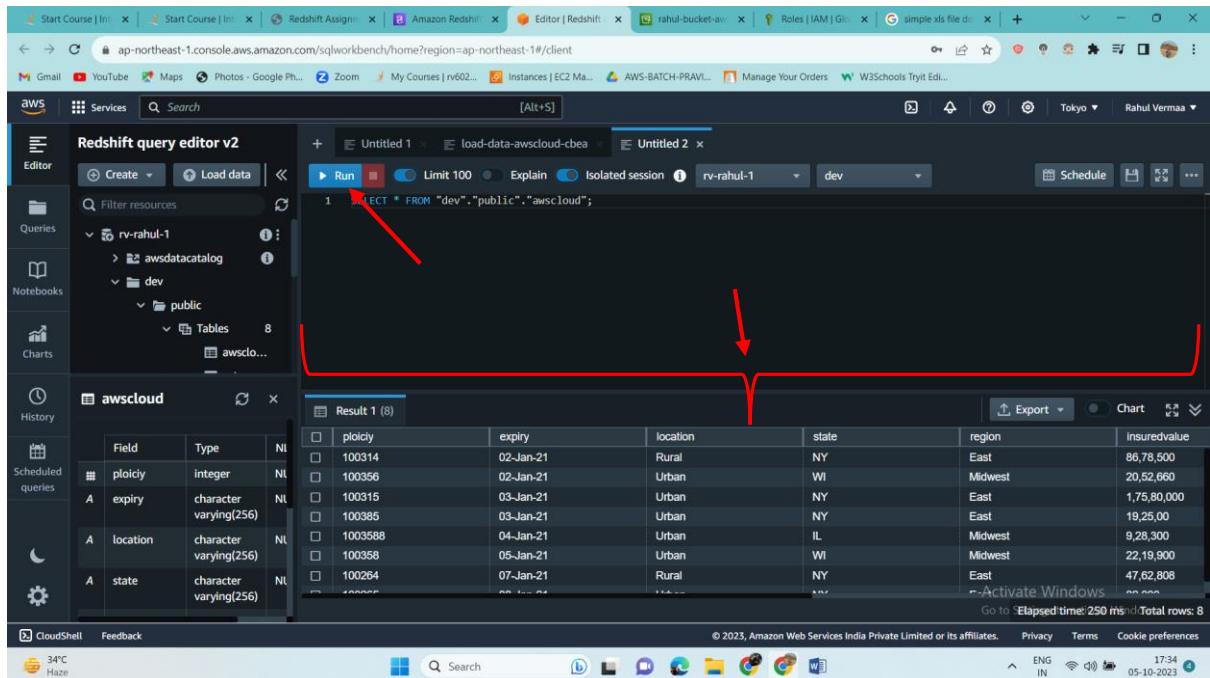
The screenshot shows the Redshift query editor interface with the 'Load data' dialog open. The dialog is configured to 'Load existing table' and specifies the 'Cluster or workgroup' as 'rv-rahul-1', 'Database' as 'dev', 'Schema' as 'public', and 'Table' as 'awscloud'. The 'IAM role' dropdown is set to 'arnaws:iam::21944977354:role/RedShiftToS3'. A red arrow points to the 'Schema' and 'Table' dropdowns, and another red arrow points to the 'IAM role' dropdown.

Now our data is loaded



The screenshot shows the AWS Redshift Query Editor v2. In the left sidebar, under the 'Tables' section of the 'dev' database, there is a table named 'awscloud'. The 'awscloud' table has 8 records. The 'Run' button is highlighted in red. The 'Result 1' pane shows a summary message: 'Info: Load into table 'awscloud' completed, 8 record(s) loaded successfully.' A red arrow points to this message. The status bar at the bottom right shows 'Elapsed time: 45s'.

Step 12: Now to query the data, go to your table double click on it and now click on RUN and you can see it is showing our data.



The screenshot shows the AWS Redshift Query Editor v2. The 'awscloud' table is selected in the left sidebar. A red arrow points to the 'Run' button. The 'Result 1' pane displays the query 'SELECT * FROM "dev"."public"."awscloud";' and the resulting data table. The data table has 8 rows and 7 columns: policy, expiry, location, state, region, and insuredvalue. A red arrow points to the data table. The status bar at the bottom right shows 'Elapsed time: 250 ms Total rows: 8'.

Field	Type	Field	Type	Field	Type	Field
policy	integer	expiry	character varying(256)	location	character varying(256)	state
100314	02-Jan-21	Rural	NY	East	86,78,500	
100356	02-Jan-21	Urban	WI	Midwest	20,52,660	
100315	03-Jan-21	Urban	NY	East	1,75,80,000	
100385	03-Jan-21	Urban	NY	East	19,25,00	
1003588	04-Jan-21	Urban	IL	Midwest	9,28,300	
100358	05-Jan-21	Urban	WI	Midwest	22,19,900	
100264	07-Jan-21	Rural	NY	East	47,62,808	