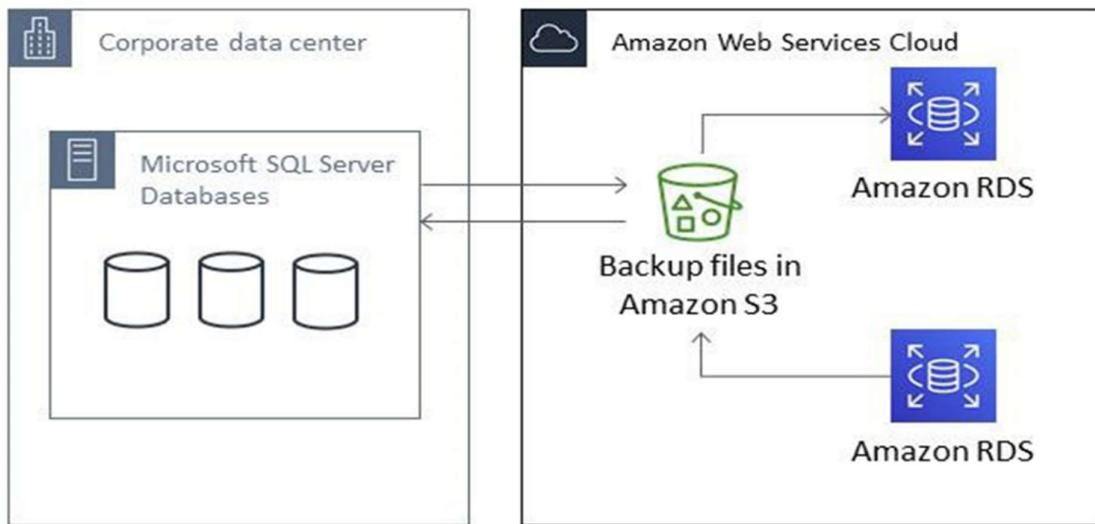


## Architecture Diagram: -



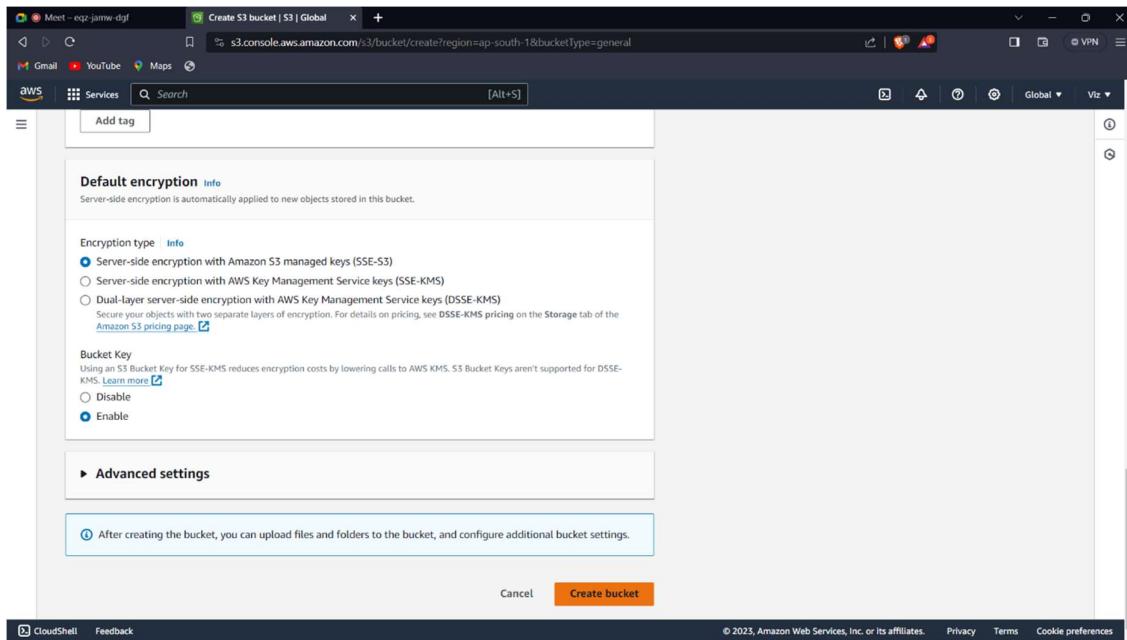
## Implementation:-

**STEP1:- Login to your AWS console and navigate to Amazon Simple Storage Service (Amazon S3).**

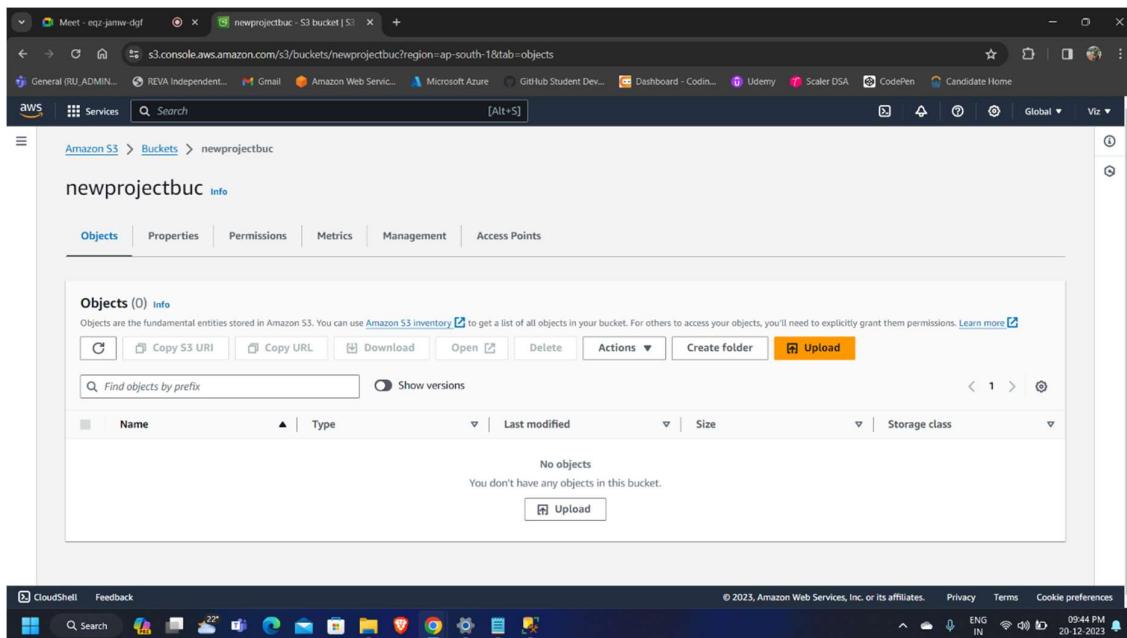
The screenshot shows the AWS Console Home page for the 'ap-south-1' region. The top navigation bar includes links for General, REVA Independent, Gmail, Amazon Web Services, Microsoft Azure, GitHub Student Dev..., Dashboard - Codin..., Udemy, Scaler DSA, CodePen, Candidate Home, Mumbai, and Viz. The main content area features a 'Build a solution' section with various quick-launch options like Launching a virtual machine, Registering a domain, Starting a development project, Building a web app, Deploying SAP on AWS, Deploying a serverless microservice, and Building using virtual servers. To the right is the 'Cost and usage' dashboard, which displays current month costs (\$517.27), total costs per month (\$86.21), and a forecasted month end costs chart. Below these are sections for Trusted Advisor, Latest announcements, Explore AWS, and Security.

This screenshot shows the same AWS Console Home page as above, but with a search query 's3' entered in the search bar. The search results for 'Services' are displayed, with 'S3' being the top result under 'Top features'. Other results include S3 Glacier, AWS Snow Family, and Storage Gateway. The rest of the interface remains consistent with the first screenshot, showing the cost dashboard and other navigation links.

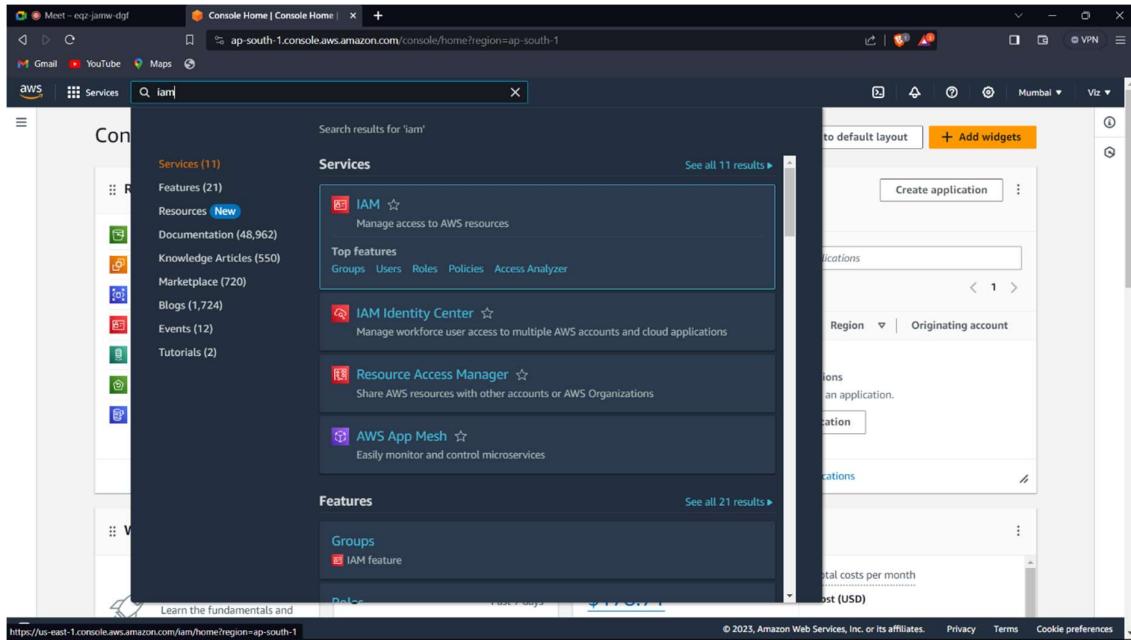
## STEP2:- Provide the suitable name and create a S3 bucket.



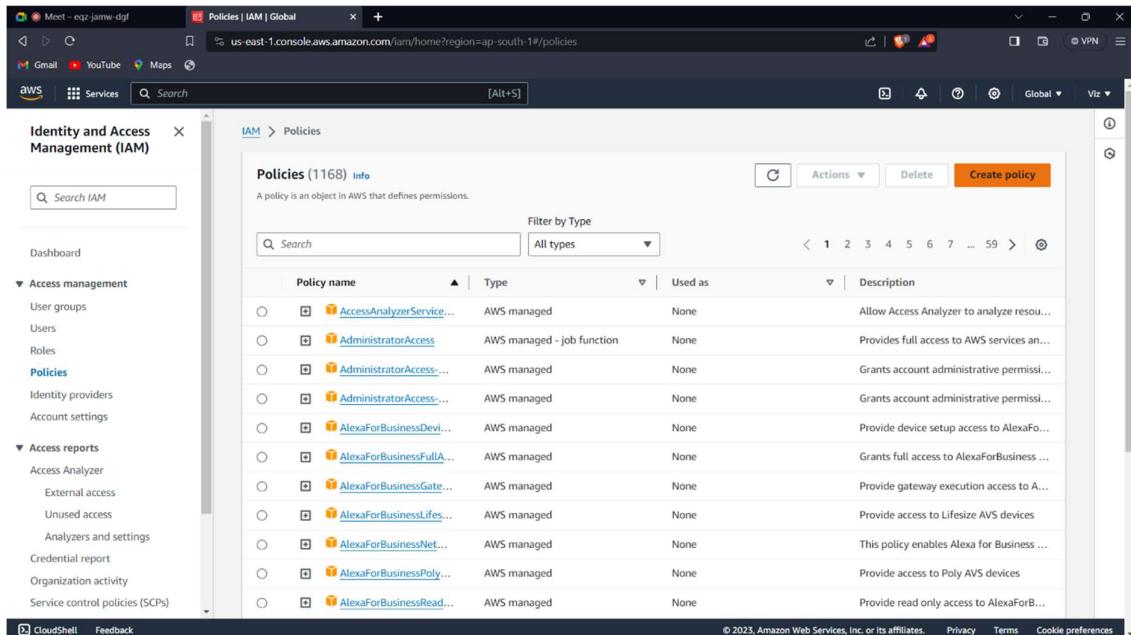
## STEP3:- Make a folder in S3 bucket.



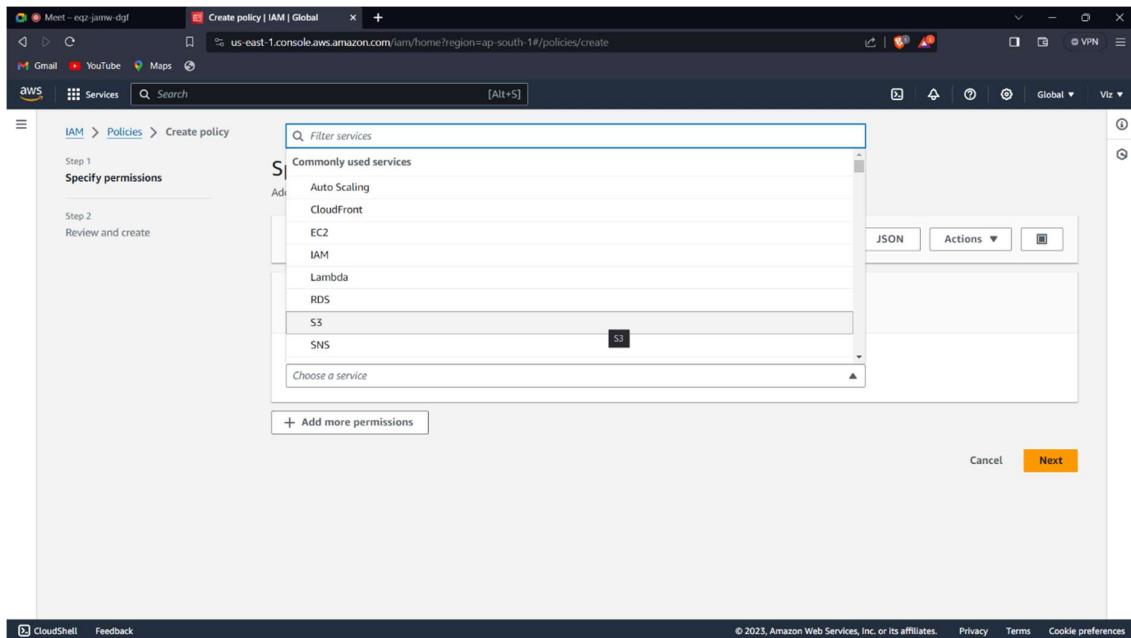
## STEP4:- Search IAM in dashboard.



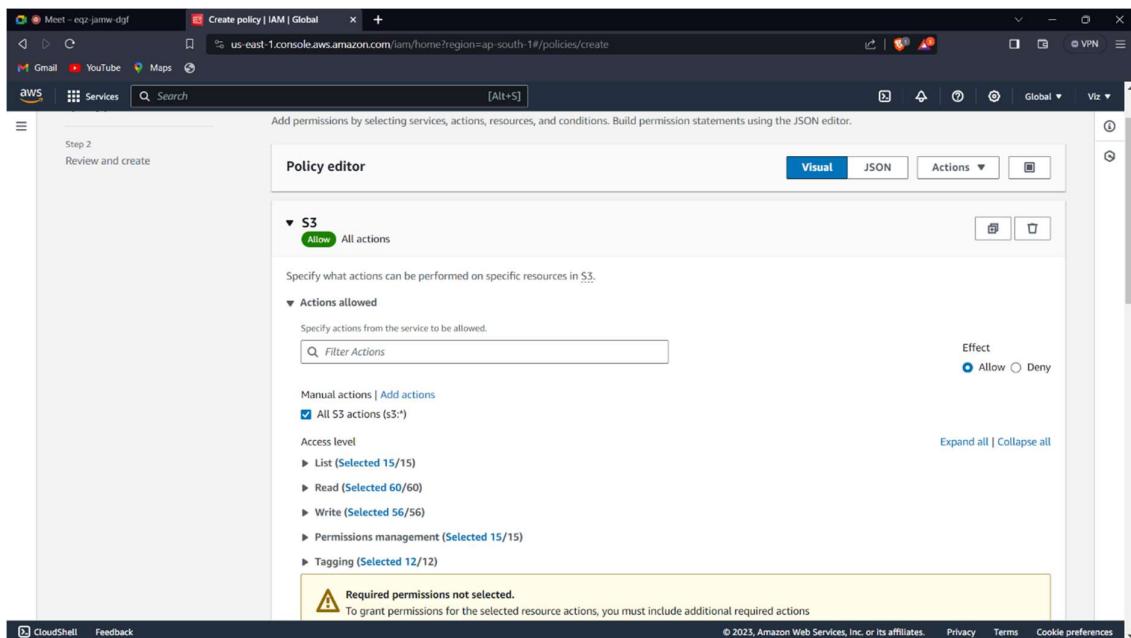
## STEP5:- Create a IAM Policy.



## STEP6:- Choose S3 and RDS as policy service.



## STEP7:- Grant Full Access of this services to policy.



## STEP8:- Go to the S3 service and chose the bucket created and copy the ARN of bucket.

The screenshot shows the AWS S3 console with the 'Buckets' page open. On the left, there's a sidebar with options like 'Access Grants', 'Object Lambda Access Points', and 'Storage Lens'. The main area displays a table of buckets. One row for 'newprojectbuc' is selected, and a tooltip 'Bucket ARN copied' is overlaid on the 'Copy ARN' button. The table includes columns for Name, AWS Region, Access, and Creation date.

Name	AWS Region	Access	Creation date
newprojectbuc	Asia Pacific (Mumbai) ap-south-1	Objects can be public	December 14, 2023, 22:54:55 (UTC+0:30)

## STEP9:- Provide the copied ARN in the bucket and object action.

The screenshot shows the 'Create policy' wizard in the AWS IAM console. It's on the 'Specify resource ARNs for these actions' step. The 'Specific' option is selected. A list of ARN types is shown, each with a description and a checkbox for 'Any in this account' or 'Any'. The ARN types include: accessgrant, accessgrantsinstance, accessgrantslocation, accesspoint, bucket, job, multiregionaccesspoint, multiregionaccesspointrequestarn, and object.

The screenshot shows the AWS IAM 'Create policy' interface. In the top navigation bar, there are tabs for 'Create policy | IAM | Global' and 'S3 buckets | S3 | Global'. Below the tabs, there's a search bar and a 'CloudShell' button. The main content area is titled 'Specify resource ARNs for these actions.' and contains a list of actions:

- accessgrant**: Specified accessgrant resource ARN for the DeleteAccessGrant and 3 more actions.  Any in this account
- accessgrantsinstance**: Specified accessgrantsinstance resource ARN for the AssociateAccessGrantsIdentityCenter and 11 more actions.  Any in this account
- accessgrantslocation**: Specified accessgrantslocation resource ARN for the CreateAccessGrant and 5 more actions.  Any in this account
- accesspoint**: Specified accesspoint resource ARN for the CreateAccessPoint and 5 more actions.  Any in this account
- bucket**: Specified bucket resource ARN for the CreateBucket and 48 more actions.  Any
- job**: Specified job resource ARN for the DeleteJobTagging and 5 more actions.  Any in this account
- multiregionaccesspoint**: Specified multiregionaccesspoint resource ARN for the CreateMultiRegionAccessPoint and 7 more actions.  Any in this account
- multiregionaccesspointrequestarn**: Specified multiregionaccesspointrequestarn resource ARN for the DescribeMultiRegionAccessPointOperation action.  Any in this account
- object**: Specified object resource ARN for the AbortMultipartUpload and 32 more actions.  Any

At the bottom of the page, there are links for 'CloudShell', 'Feedback', and copyright information: '© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

**STEP10:- Paste the ARN in the same way as displayed, and click add ARN.**

The screenshot shows the 'Specify ARN(s)' dialog box, which is a modal window overlaid on the main IAM policy creation page. The dialog has two tabs: 'Visual' (selected) and 'Text'. The 'Visual' tab allows you to select resources from a list, while the 'Text' tab allows you to enter ARNs directly. The dialog contains fields for 'Resource bucket name' (with a dropdown menu showing 'Any bucket name') and 'Resource ARN' (with a text input field containing 'jimaws.s3::newprojectbuc'). At the bottom right of the dialog is an 'Add ARNs' button.

## STEP 11:- Provide RDS access and the new policy is now created.

The screenshot shows the AWS IAM Policy editor interface. The left pane displays the JSON code for the policy:

```

1  {
2      "Version": "2012-10-17",
3      "Statement": [
4          {
5              "Sid": "VisualEditor0",
6              "Effect": "Allow",
7              "Action": [
8                  "s3:ListAccessPointsForObjectLambda",
9                  "s3:GetAccessPoint",
10                 "s3:PutAccountPublicAccessBlock",
11                 "s3:ListAccessPoints",
12                 "s3:CreateStorageLensGroup",
13                 "s3:ListJobs",
14                 "s3:PutStorageLensConfiguration",
15                 "s3:ListMultiRegionAccessPoints",
16                 "s3:ListStorageLensGroups",
17                 "s3:ListStorageLensConfigurations",
18                 "s3:GetAccountPublicAccessBlock",
19                 "s3:ListAllMyBuckets",
20                 "s3:ListAccessGrantsInstances",
21                 "s3:PutAccessPointPublicAccessBlock",
22                 "s3:CreateJob",
23                 "rds:/*"
24             ],
25         },
26         {
27             "Resource": [
28                 "*",
29                 "*"
30             ]
31         }
32     ]
33 }

```

The right pane shows the policy statement being edited, with the "Add actions" section expanded. It lists various RDS actions under the "All services > RDS" category. A checkbox for "All actions (rds:\*)" is checked. Other available actions include:

- DescribeAccountAttributes Info
- DescribeBlueGreenDeployments Info
- DescribeCertificates Info
- DescribeDBClusterAutomatedBackups Info
- DescribeDBClusterBacktracks Info
- DescribeDBClusterEndpoints Info
- DescribeDBClusterParameterGroups Info
- DescribeDBClusterParameters Info

At the bottom right of the editor, there is an "Add" button.

The screenshot shows the AWS IAM Policies page. The left sidebar navigation includes "Identity and Access Management (IAM)" and "Policies". The main content area displays a table of policies:

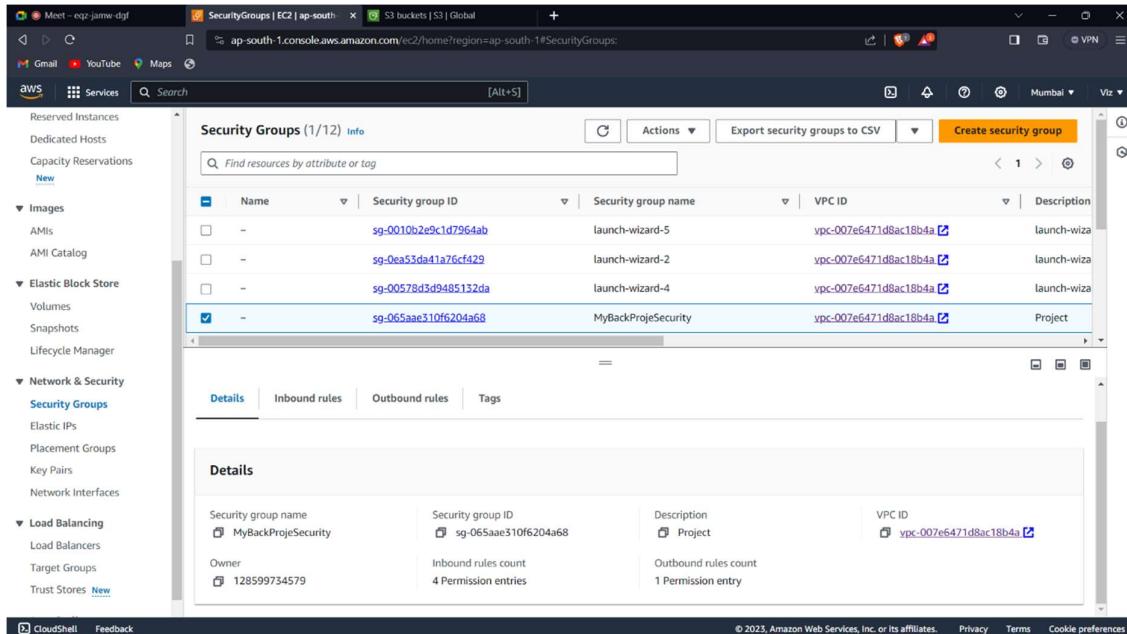
Policy name	Type	Used as	Description
<a href="#">Backup_policy</a>	Customer managed	None	-
<a href="#">NewProjectPolicy</a>	Customer managed	Permissions policy (1)	-
<a href="#">sql-backups-user</a>	Customer managed	Permissions policy (1)	-
<a href="#">sqlNativeBackup-170240...</a>	Customer managed	Permissions policy (1)	RDS SQL Server Native Backup Without E...

At the top right of the table, there are buttons for "Actions", "Delete", and "Create policy". The URL in the browser bar is <https://us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/policies/details/am%3Aaws%3Aiam%3A12859734579%3Apolicy%2FNewProjectPolicy>.

**Note:- This is how JSON file would look like.**

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "VisualEditor0",  
            "Effect": "Allow",  
            "Action": [  
                "s3>ListAccessPointsForObjectLambda",  
                "s3:GetAccessPoint",  
                "s3:PutAccountPublicAccessBlock",  
                "s3>ListAccessPoints",  
                "s3>CreateStorageLensGroup",  
                "s3>ListJobs",  
                "s3:PutStorageLensConfiguration",  
                "s3>ListMultiRegionAccessPoints",  
                "s3>ListStorageLensGroups",  
                "s3>ListStorageLensConfigurations",  
                "s3:GetAccountPublicAccessBlock",  
                "s3>ListAllMyBuckets",  
                "s3>ListAccessGrantsInstances",  
                "s3:PutAccessPointPublicAccessBlock",  
                "s3>CreateJob"  
            ],  
            "Resource": [  
                "*-*",  
                "*-*"  
            ]  
        },  
        {  
            "Sid": "VisualEditor1",  
            "Effect": "Allow",  
            "Action": "s3:*",  
            "Resource": [  
                "arn:aws:s3:::newprojectbuc",  
                "arn:aws:s3:::newprojectbuc/*"  
            ]  
        }  
    ]  
}
```

**STEP 12:- Go to EC2 dashboard and head to Security Groups and create a security group.**



## STEP 13:- Add the inbound rules and create security group.

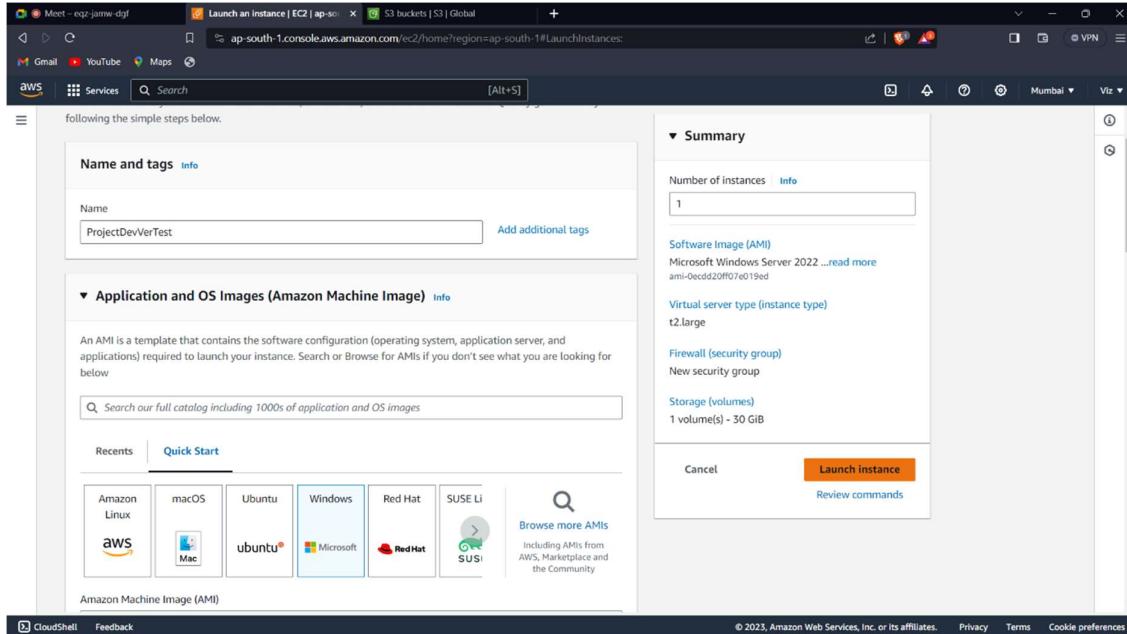
Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-0bfdf3e6051e2658	IPv4	All traffic	All	All
-	sgr-0b6247fec7e5f52f3	IPv4	MSSQL	TCP	1433
-	sgr-0c2cf4f76f655f584	IPv6	MSSQL	TCP	1433
-	sgr-042772103f5c93ade	IPv4	MSSQL	TCP	1433

## STEP 14:- Go to Instances and click on create instance.

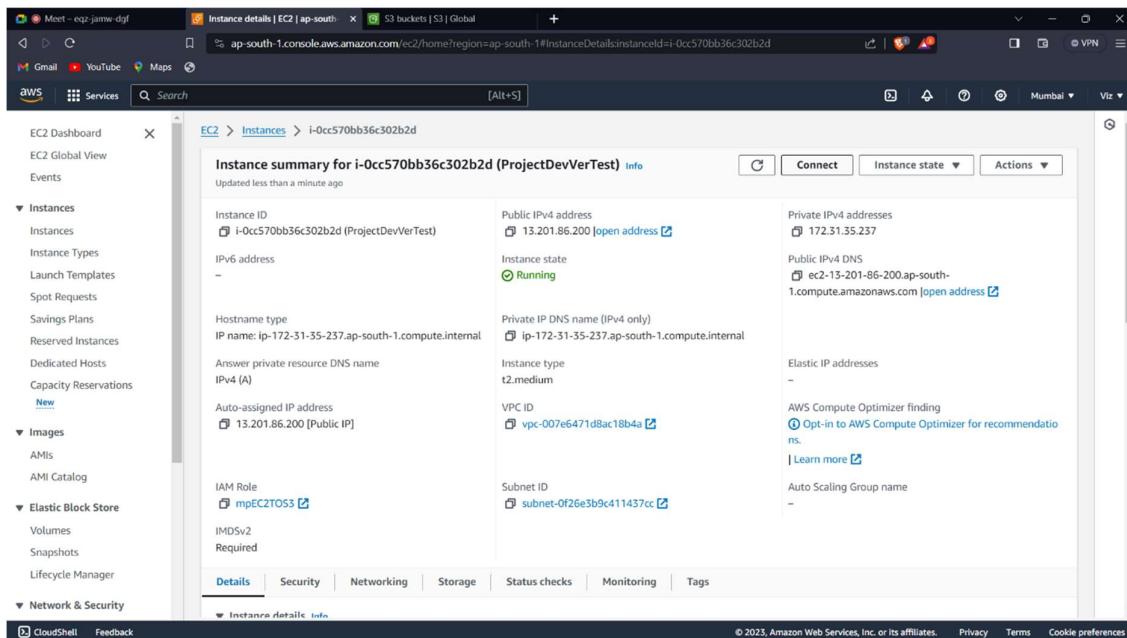
Search results for 'ec2'  
Try searching with longer queries for more relevant results

ARN
arn:aws:iam::128599734579:policy/NewProjectPolicy

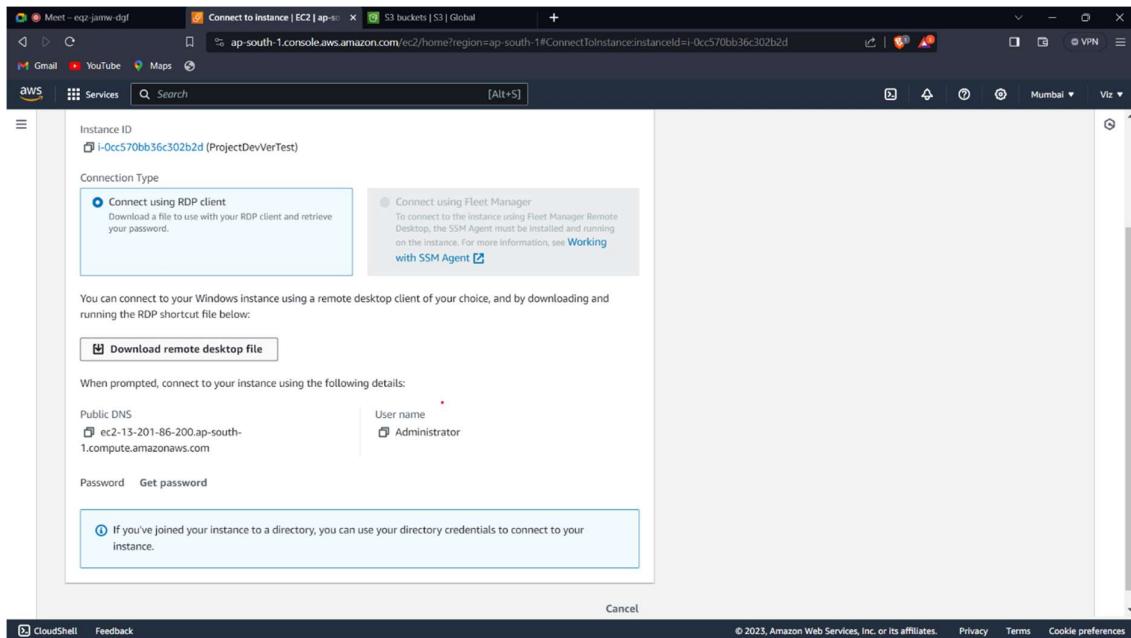
**STEP 15:- Provide the name and select Windows as OS and provide the needed storage and CPU and select the created security group and create instance.**



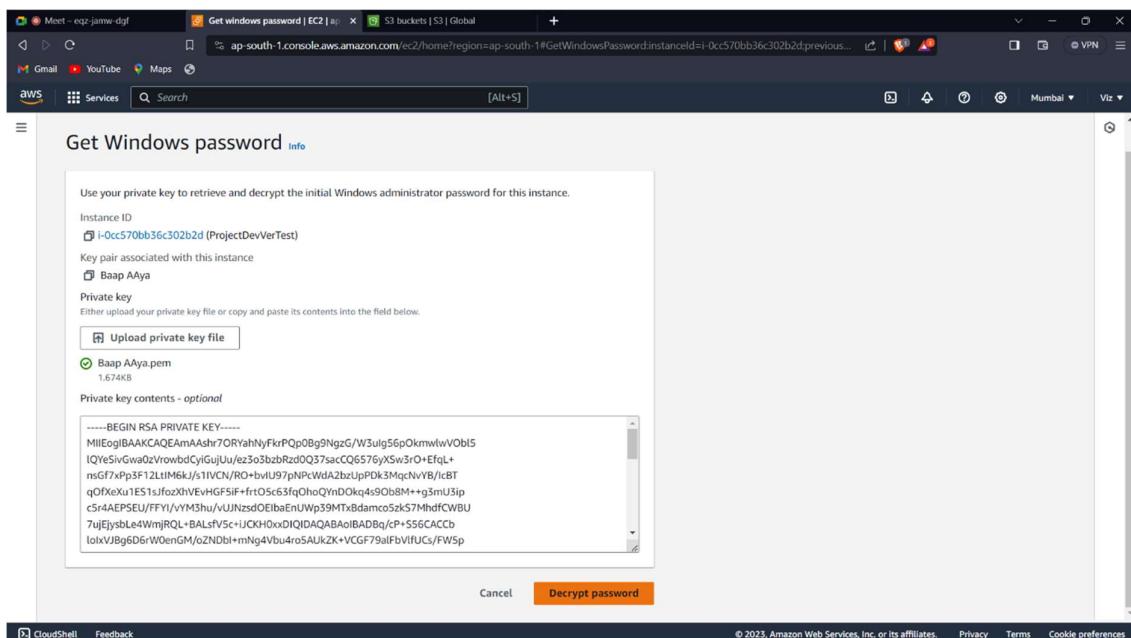
**STEP 16:- Your instance is ready to use.**



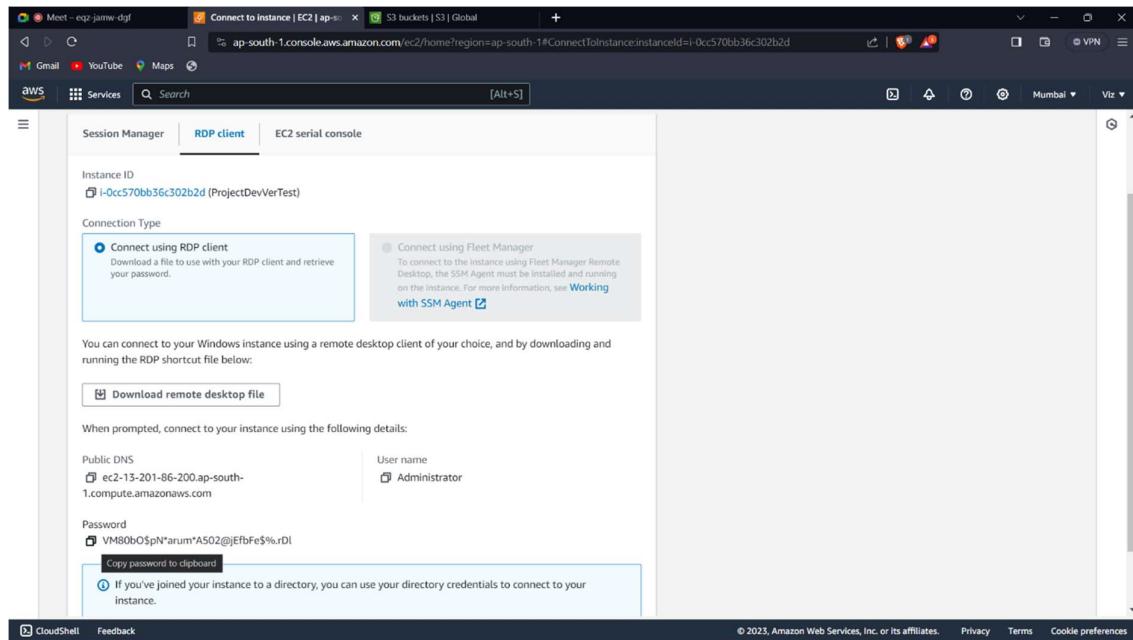
**STEP 17:- Select the EC2 Instance and click on connect and download the RDP file.**



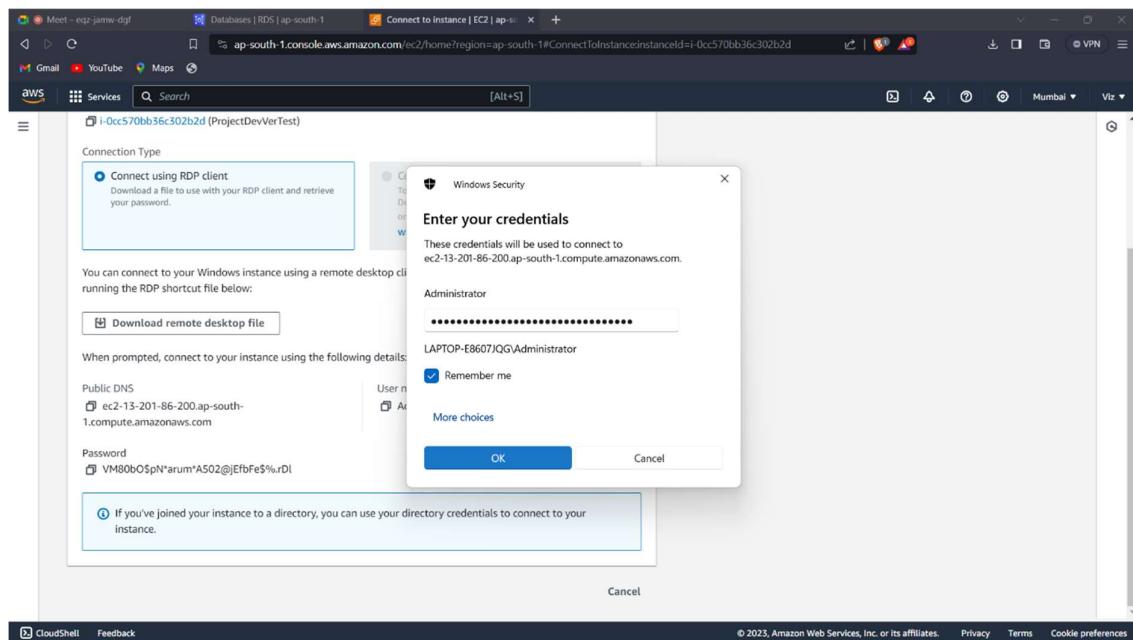
**STEP 18:- Click on Get Password and provide the key which you downloaded to get windows login password. Click on Decrypt password.**



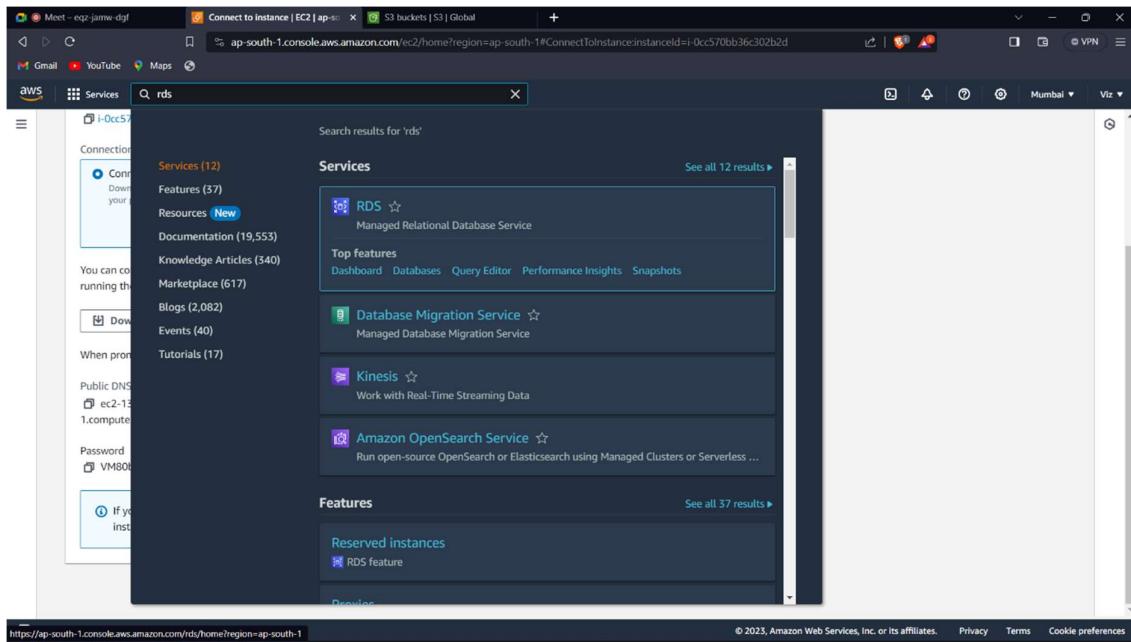
## STEP 19:- Copy the password.



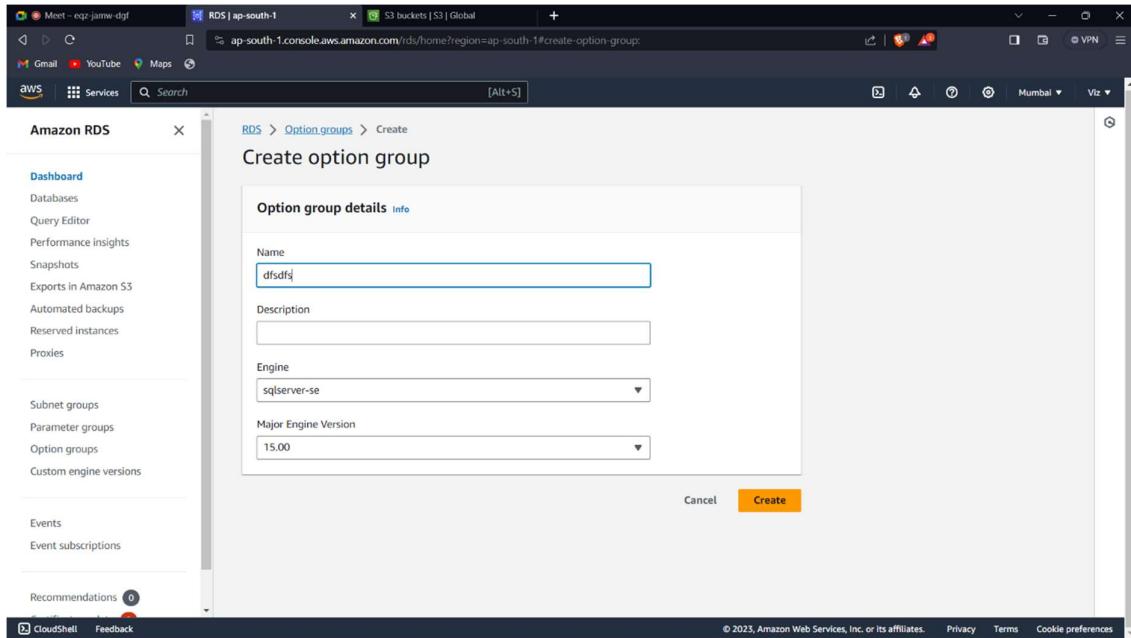
## STEP 20:- Run the RDP file and Paste the password and click on ok.



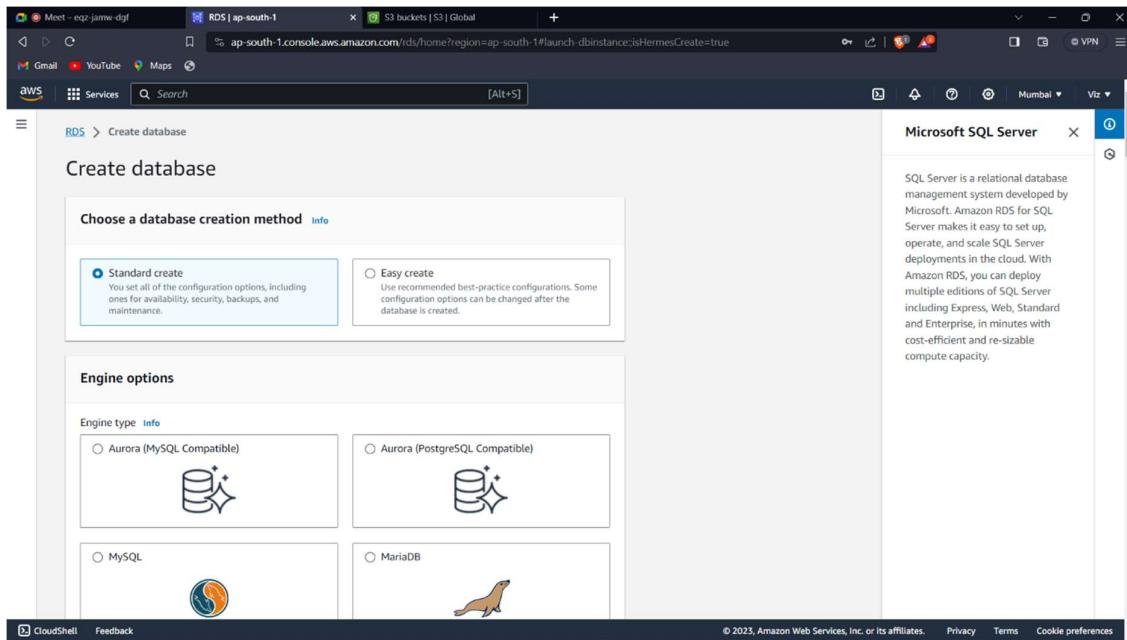
## STEP 21:- Search for the RDS in AWS dashboard.



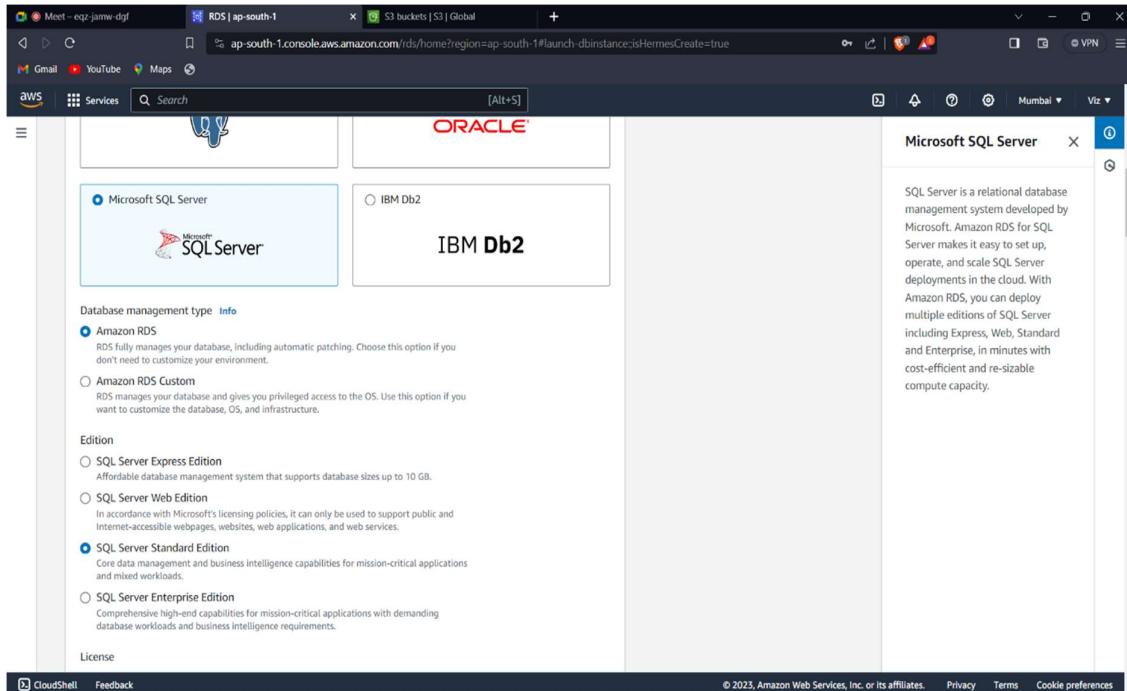
## STEP 22:- Search for option group and create a option group and select the database and the Engine version and click create.



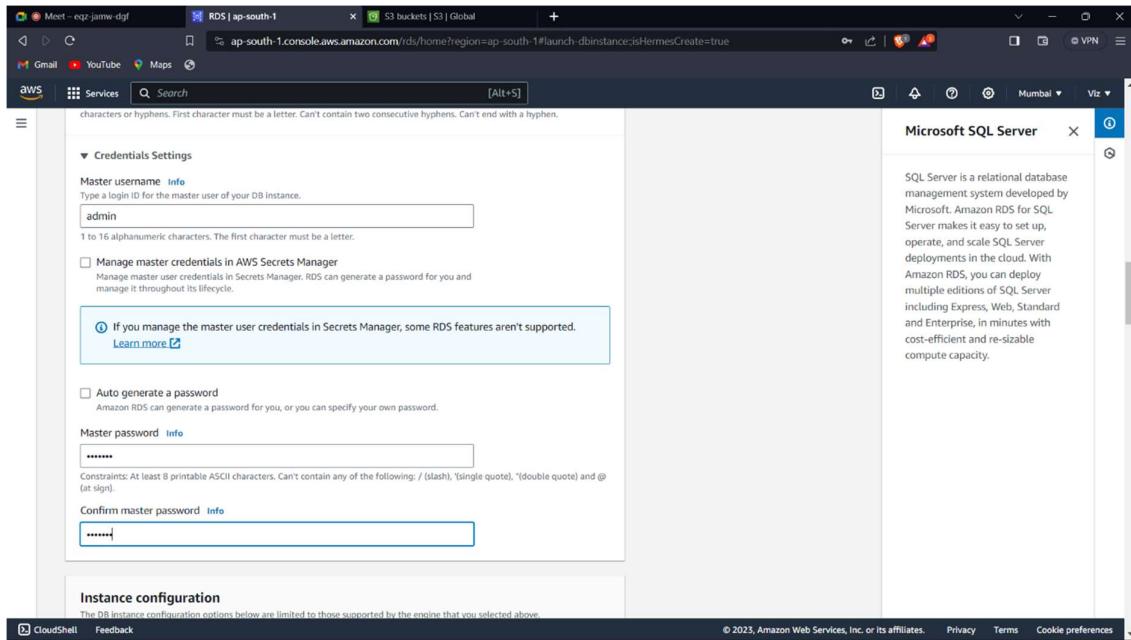
## STEP 23:- Create Database chose standard create option.



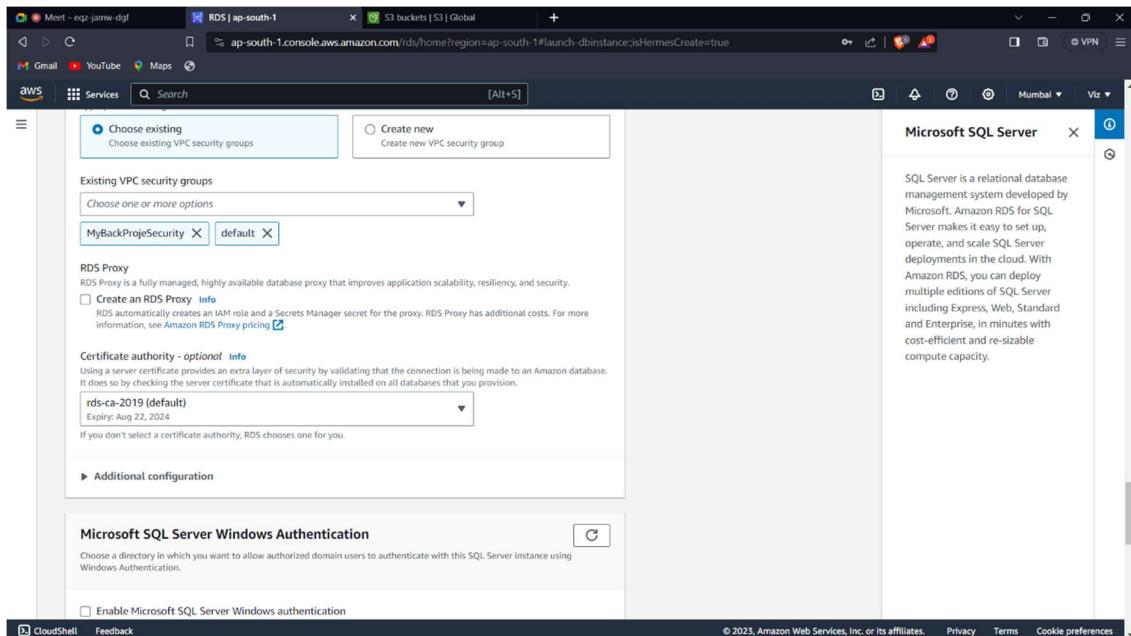
## STEP 24:- Choose the MS SQL Server Database and select SQL Server Standard Edition.



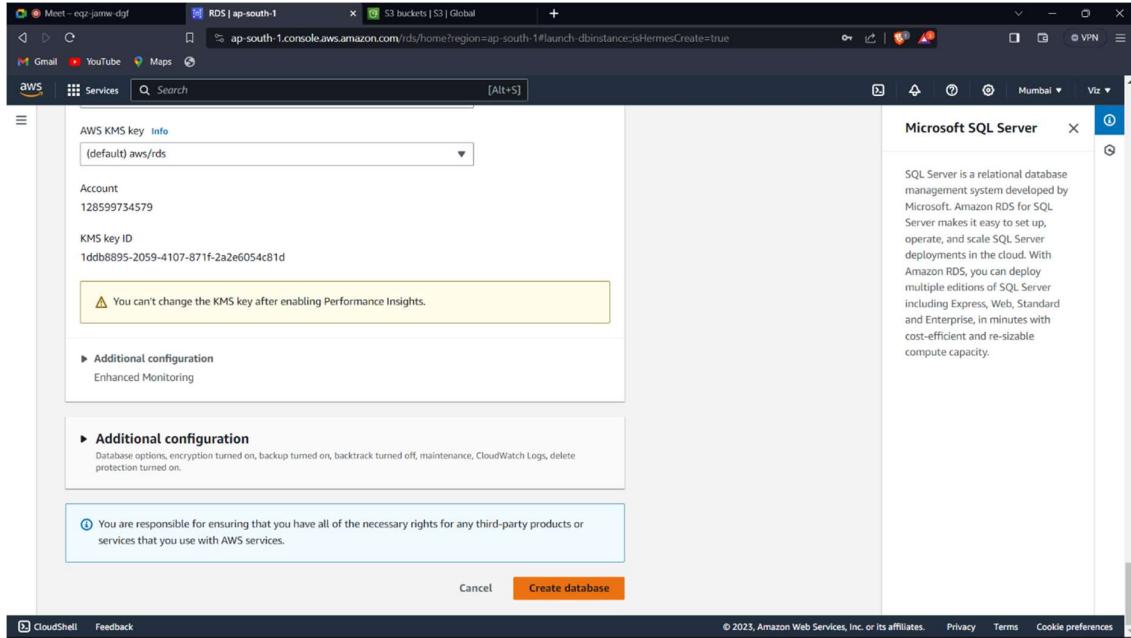
## STEP 25:- Provide the database identifier as database-2000 and master username as admin and provide the password.



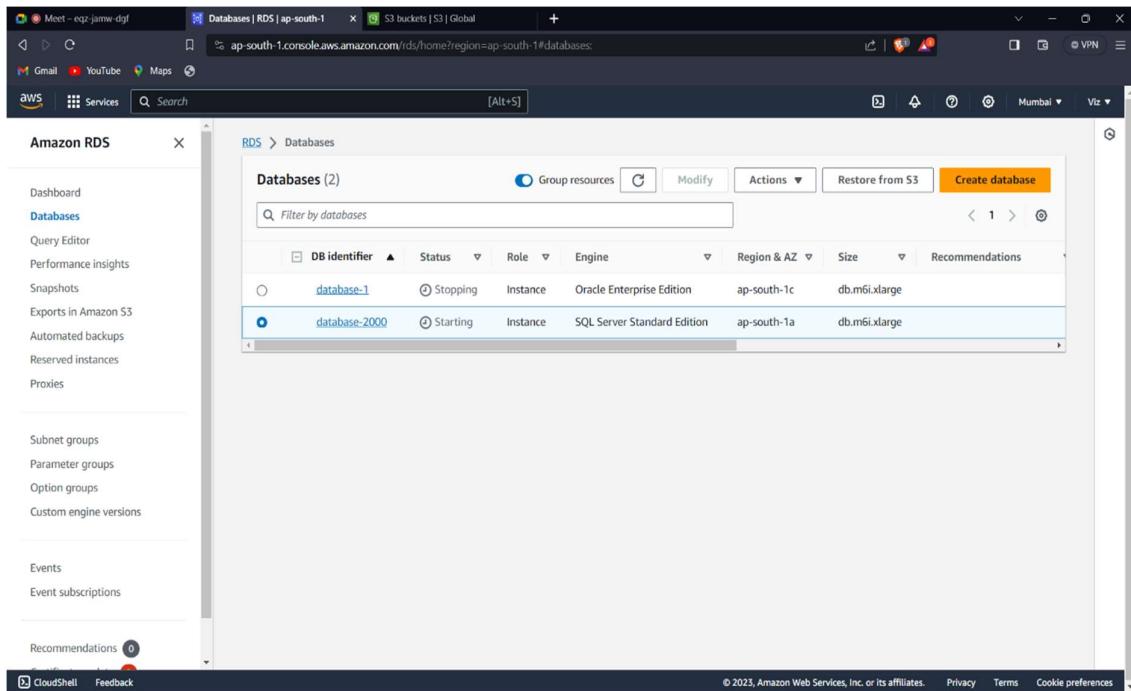
## STEP 26:- Choose the created security group and default together.



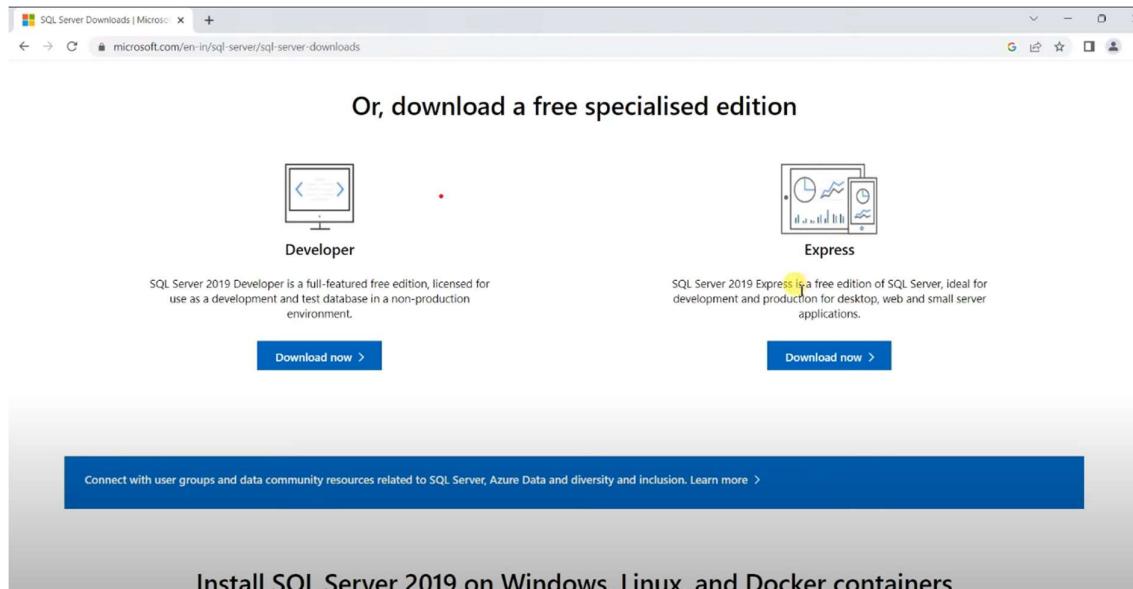
## STEP 27:- Click on create database.



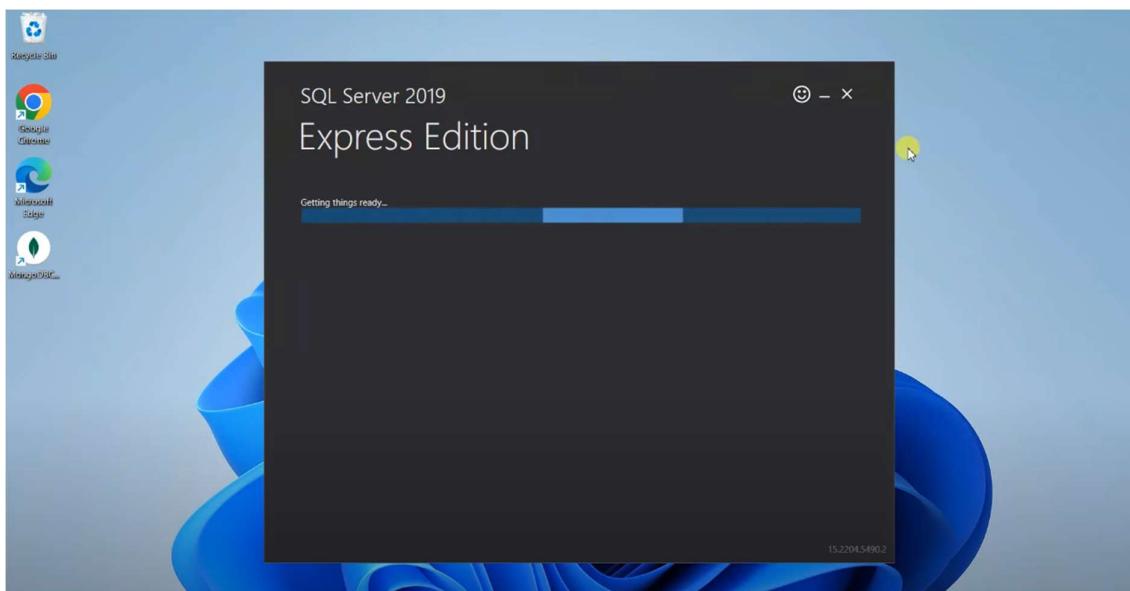
## STEP 28:- Check the status of database and once its Available its ready to use.



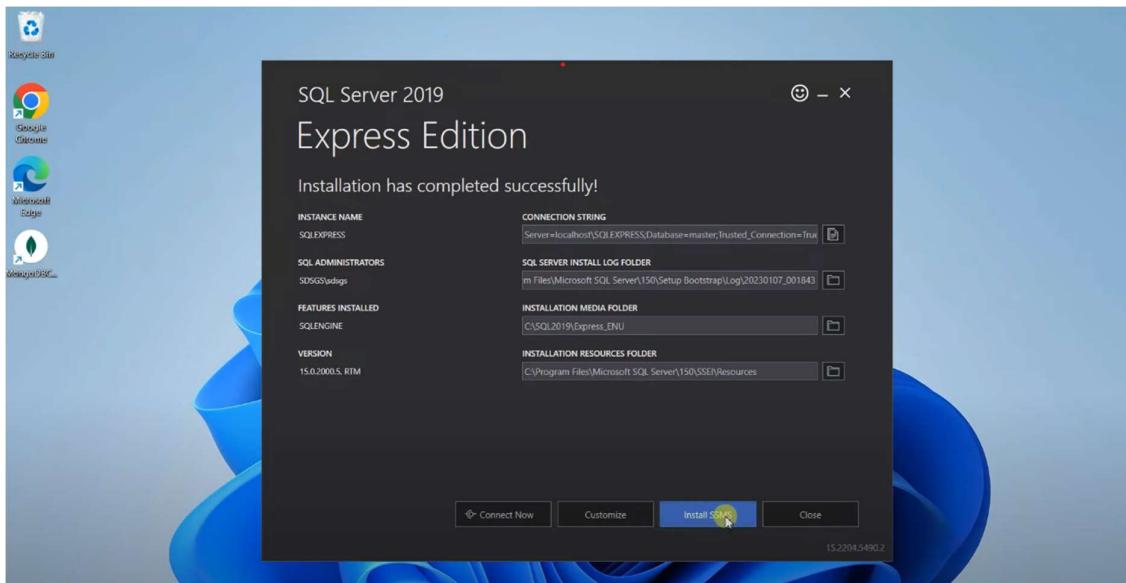
**STEP 29:- Go to your logged in EC2 instance And download the Microsoft SQL Server Management Studio Express Edition.**



**STEP 30:- Install The Downloaded SQL server express edition.**



## STEP 31:- After the installation of SQL Server, Install SSMS.



## STEP 32:- Download and install the SSMS.

A screenshot of a Microsoft web browser window displaying the download page for SQL Server Management Studio (SSMS). The URL in the address bar is "learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16".

The left sidebar shows a navigation menu under "Version":

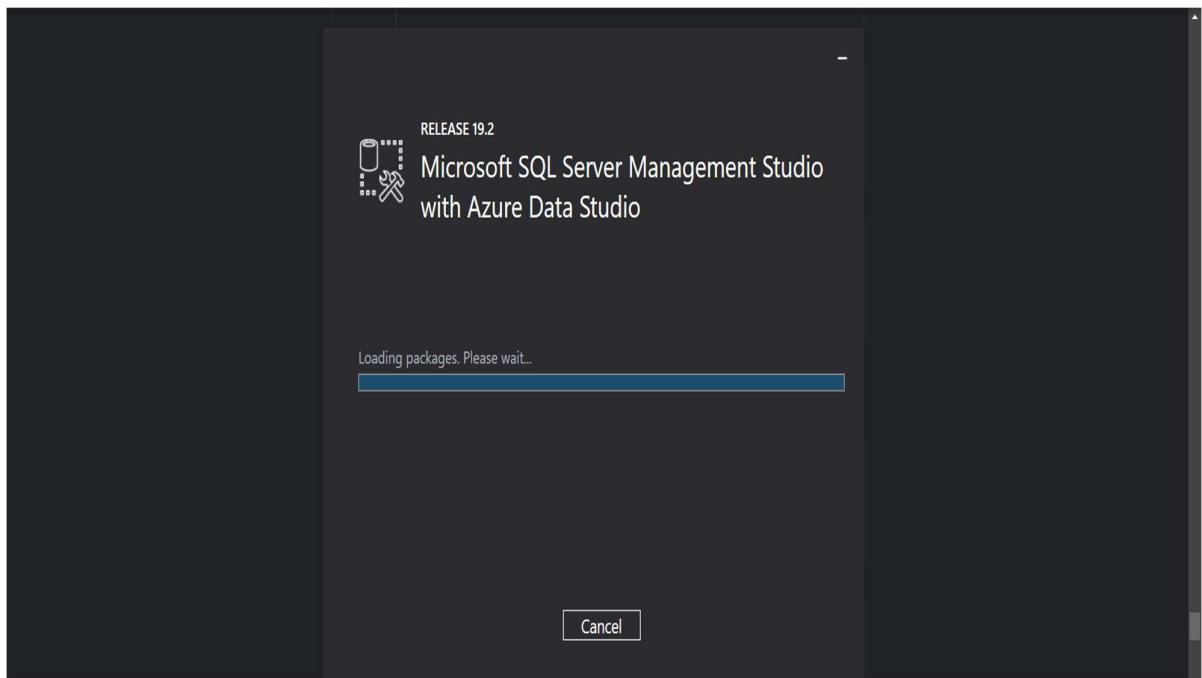
- SQL Server 2022
- Filter by title
- Download SSMS
  - Release notes
  - Overview
  - Quickstarts
  - Tutorials
  - Concepts
  - How-to
  - References
  - Resources
  - SqlPackage
  - SQL Server Profiler
  - Visual Studio native helpers
  - Extended Features
  - Visual Studio Code
  - Tutorials
  - SQL Server on Linux
  - SQL on Azure
  - Azure Arc
  - Resources
  - Download PDF

The main content area features a section titled "Download SSMS" with a link to "Free Download for SQL Server Management Studio (SSMS) 19.2". Below this, a paragraph explains the latest general availability (GA) version (19.2) and provides release notes:

- Release number: 19.2
- Build number: 19.2.56.2
- Release date: November 13, 2023

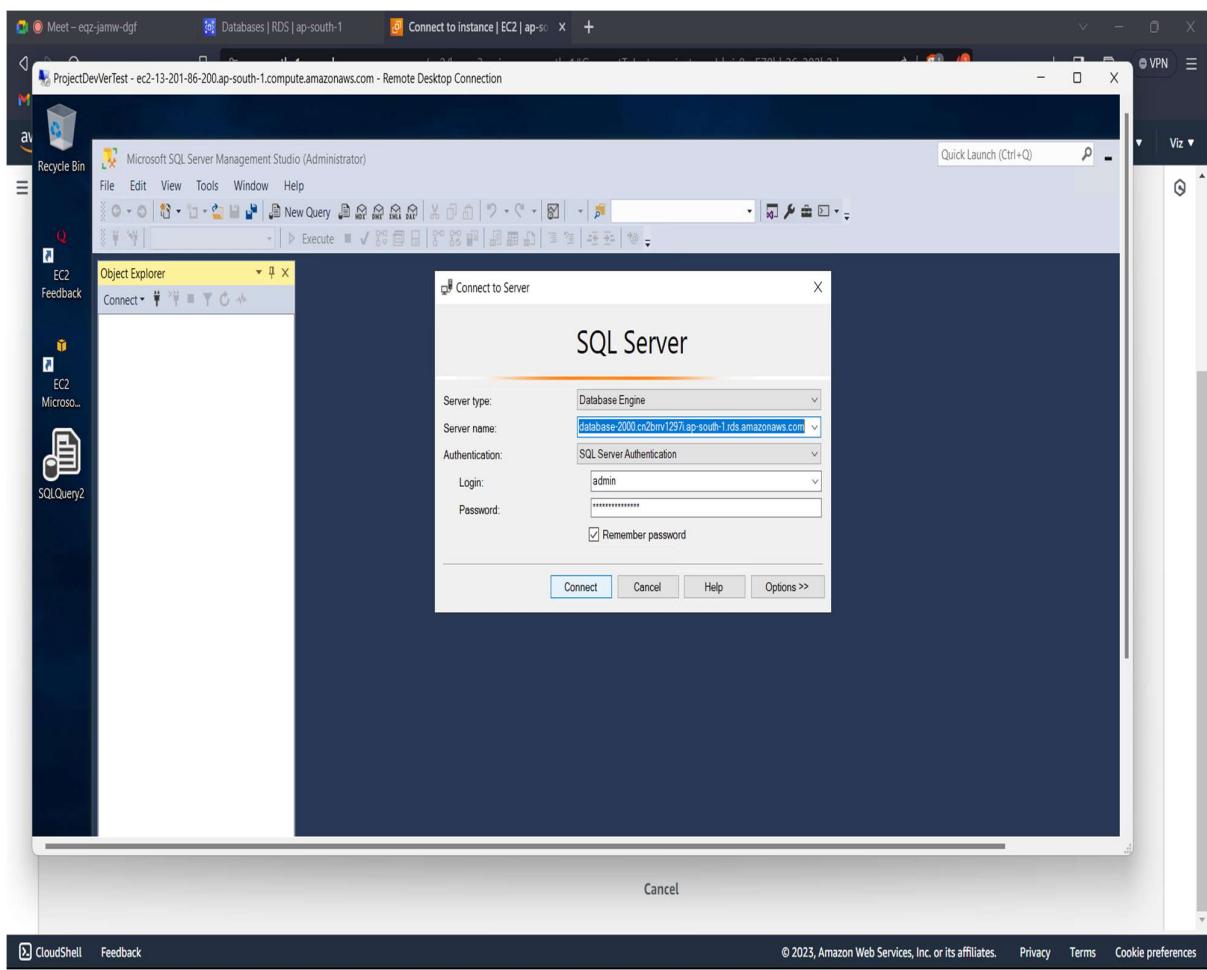
At the bottom of the main content, there is a note about agreeing to the license terms and privacy statement, followed by a link to "https://aka.ms/ssmstfullsetup".

The right sidebar contains "Additional resources" and "Documentation" sections, which include links to "Training", "Certification", and "Release notes for (SSMS) - SQL Server Management Studio (SSMS)".



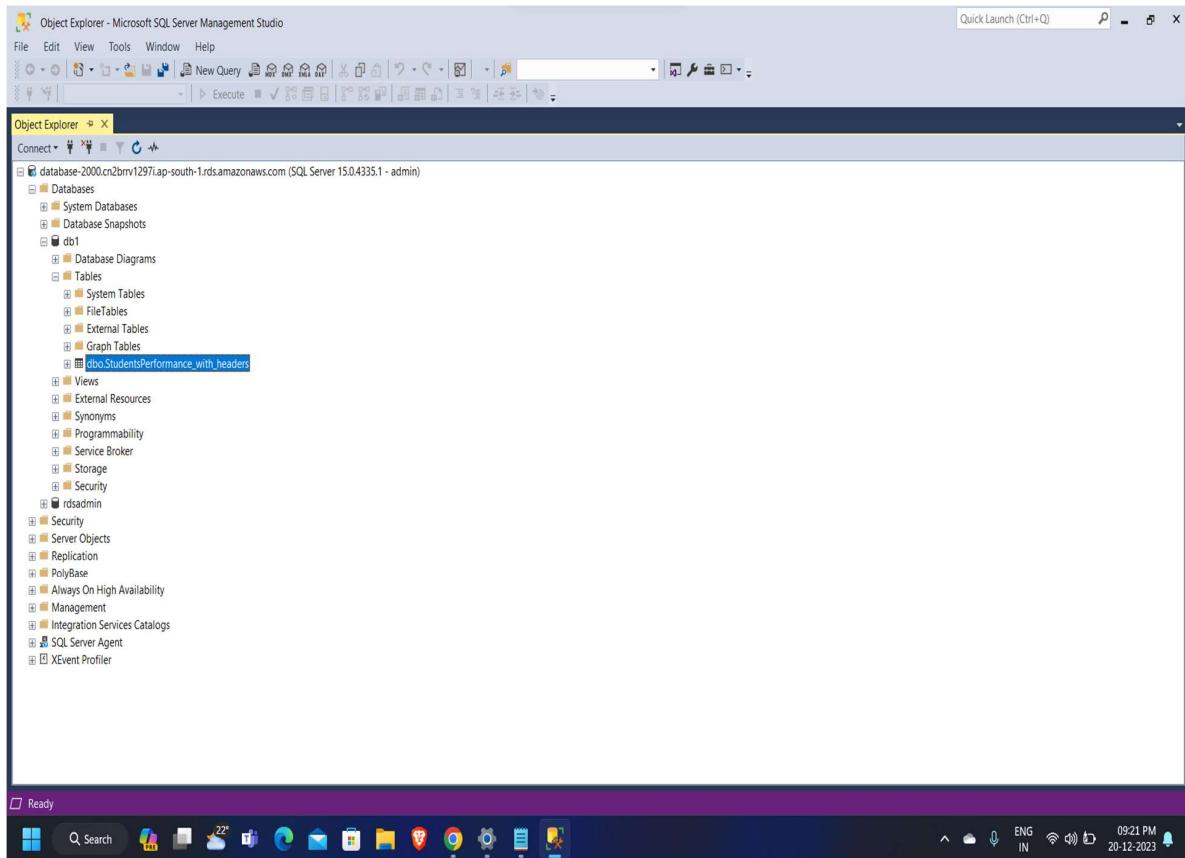
## **STEP 33:- Open The SQL Server Management Studio And Login Using.**

- 1] In Server Name section provide the created Database's Endpoint, and login to SQL server database.**
- 2] Provide the admin name which you have chosen.**
- 3] In the password section provide the Master password obtained from RDS.**

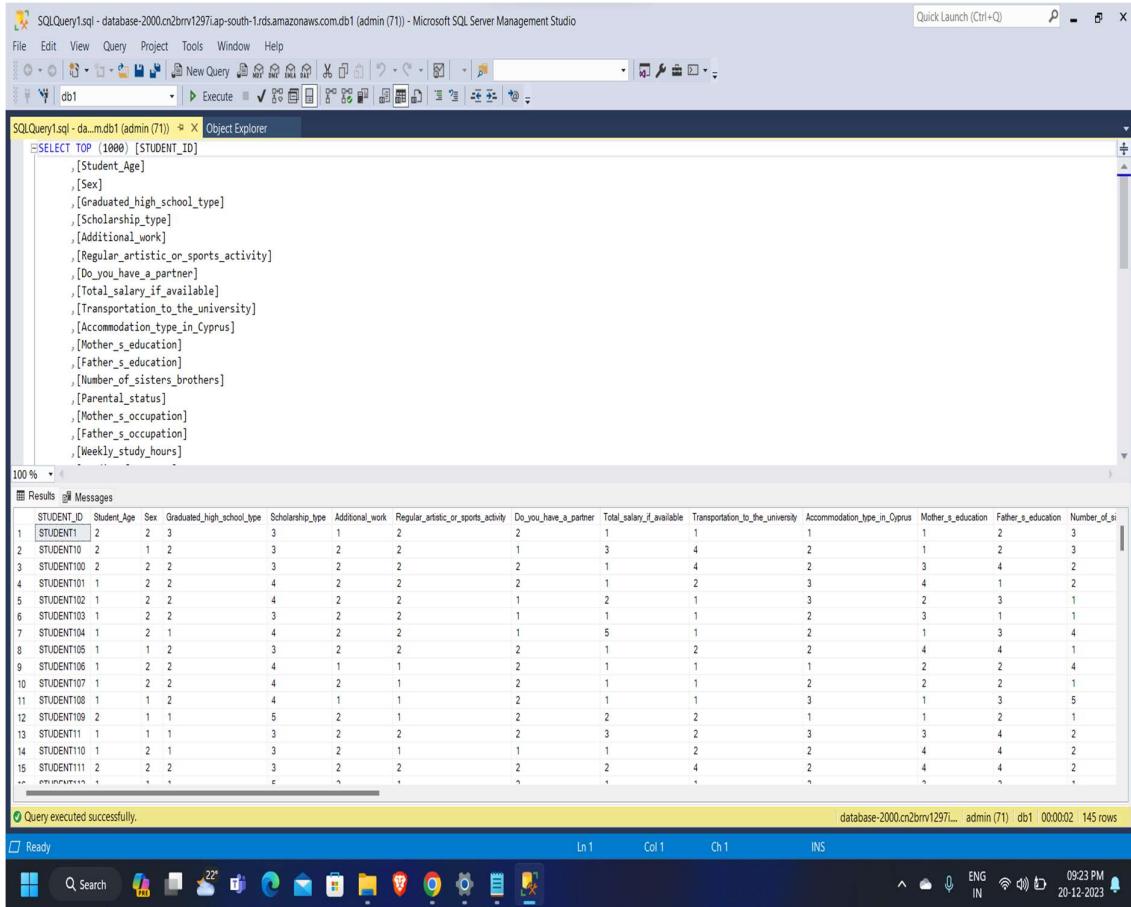


**STEP 34:- Once you have logged in the SQL server.**

- 1] Create a Database named “db1”.**
- 2] Create Tables, Rows, Columns in the database.**
- 3] Provide the data and create a whole large set of SQL Table.**



**STEP 35:- Create a student table with different rows and columns and chose primary key .**



The screenshot shows the Microsoft SQL Server Management Studio interface. A query window is open with the following T-SQL code:

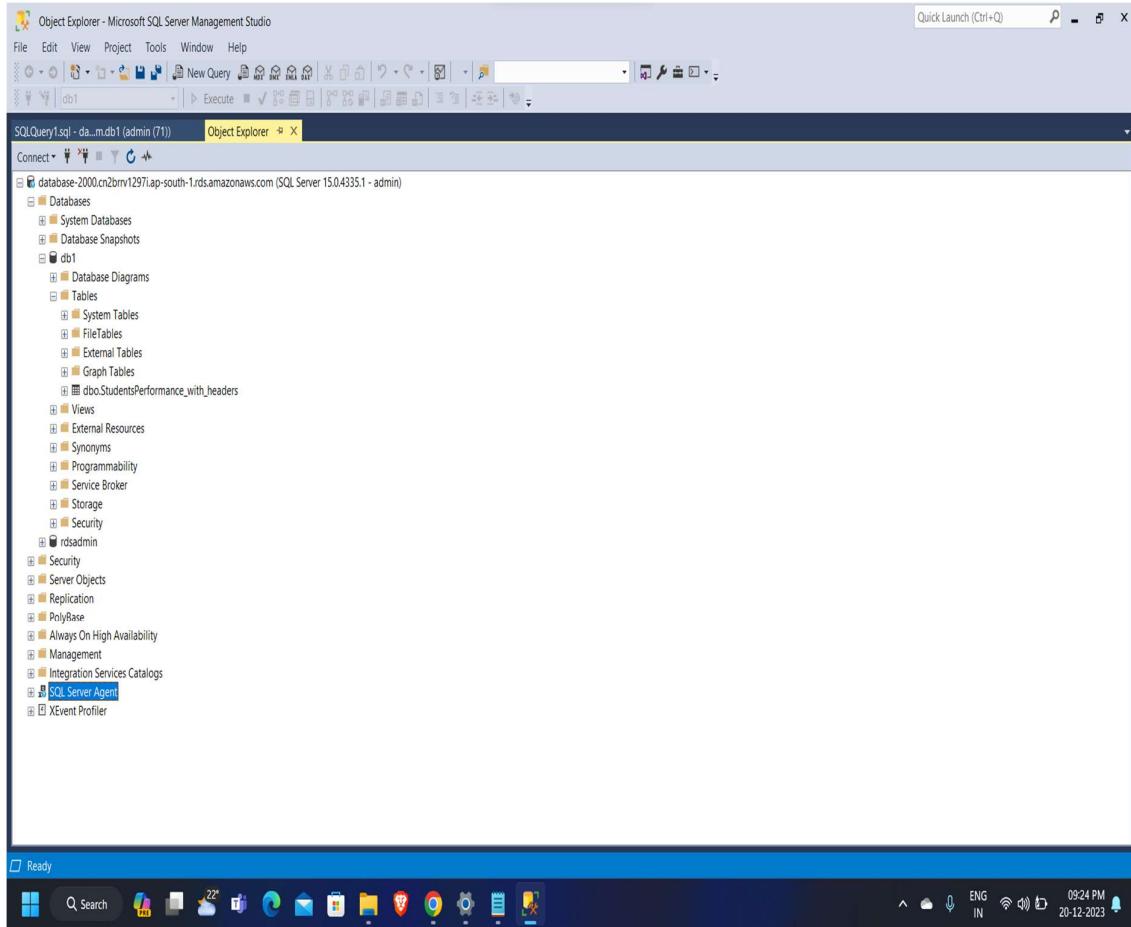
```
SELECT TOP (1000) [STUDENT_ID]
      ,[Student_Age]
      ,[Sex]
      ,[Graduated_high_school_type]
      ,[Scholarship_type]
      ,[Additional_work]
      ,[Regular_artistic_or_sports_activity]
      ,[Do_you_have_a_partner]
      ,[Total_salary_if_available]
      ,[Transportation_to_the_university]
      ,[Accommodation_type_in_Cyprus]
      ,[Mother_s_education]
      ,[Father_s_education]
      ,[Number_of_sisters_brothers]
      ,[Parental_status]
      ,[Mother_s_occupation]
      ,[Father_s_occupation]
      ,[Weekly_study_hours]
```

The results grid displays 145 rows of data from the STUDENT table. The columns correspond to the selected fields in the query. The data includes various student characteristics such as age, sex, education level, and family background.

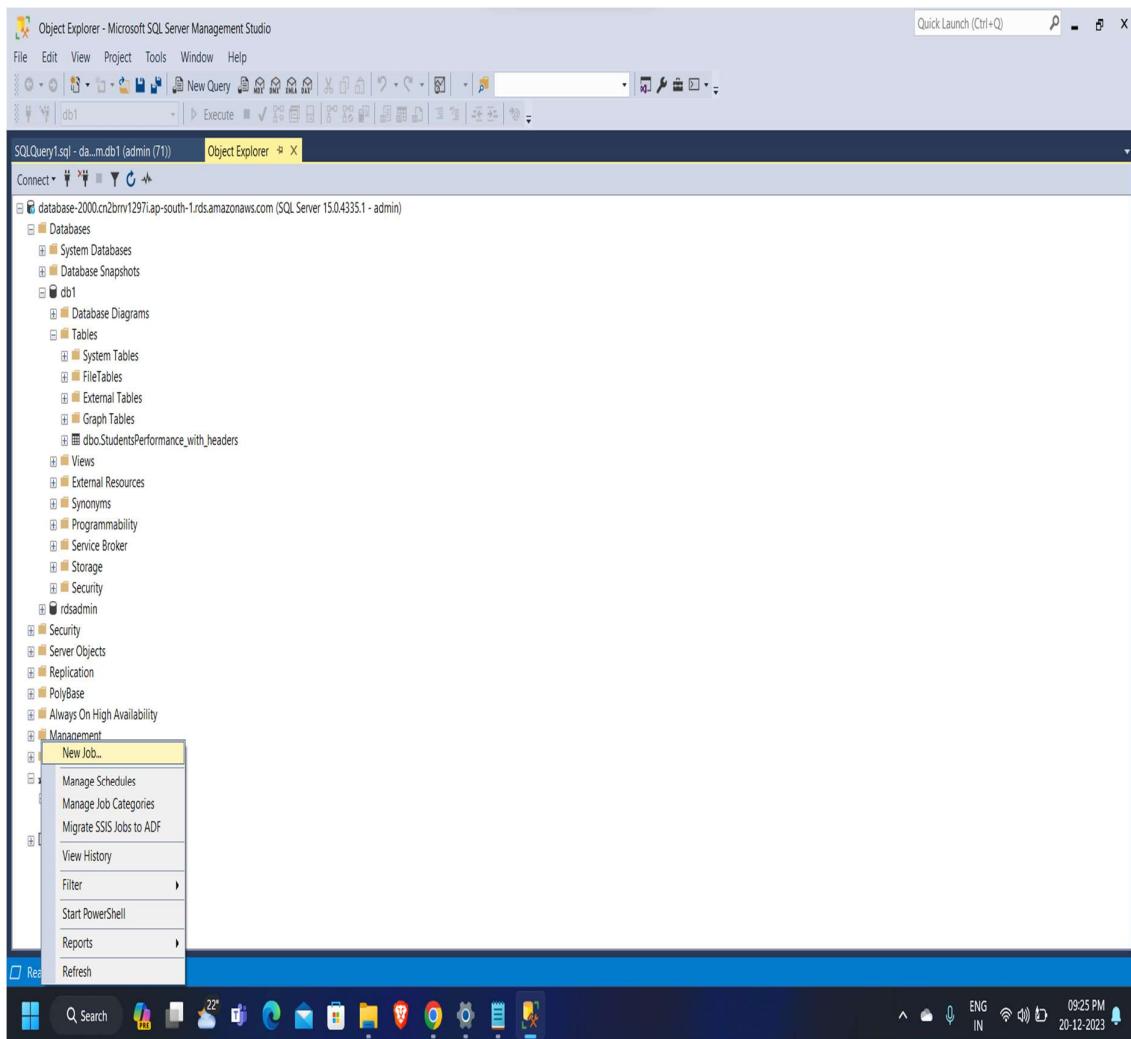
STUDENT_ID	Student_Age	Sex	Graduated_high_school_type	Scholarship_type	Additional_work	Regular_artistic_or_sports_activity	Do_you_have_a_partner	Total_salary_if_available	Transportation_to_the_university	Accommodation_type_in_Cyprus	Mother_s_education	Father_s_education	Number_of_sisters_brothers
1	STUDENT1	2	3	3	1	2	2	1	1	1	1	2	3
2	STUDENT10	2	1	2	3	2	2	1	3	4	2	1	2
3	STUDENT100	2	2	2	3	2	2	2	1	4	2	3	2
4	STUDENT101	1	2	2	4	2	2	2	1	2	3	4	1
5	STUDENT102	1	2	2	4	2	2	1	2	1	3	2	3
6	STUDENT103	1	2	2	3	2	2	1	1	1	2	3	1
7	STUDENT104	1	2	1	4	2	2	1	5	1	2	1	3
8	STUDENT105	1	1	2	3	2	2	2	1	2	2	4	1
9	STUDENT106	1	2	2	4	1	1	2	1	1	1	2	2
10	STUDENT107	1	2	2	4	2	1	2	1	1	2	2	1
11	STUDENT108	1	1	2	4	1	1	2	1	1	3	1	5
12	STUDENT109	2	1	1	5	2	1	2	2	2	1	1	2
13	STUDENT110	1	1	1	3	2	2	2	3	2	3	4	2
14	STUDENT111	1	2	1	3	2	1	1	1	2	2	4	2
15	STUDENT112	2	2	2	3	2	2	2	2	4	2	4	2
..	STUDENT113	1	*	*	e	*	*	*	*	*	*	*	*

Query executed successfully.

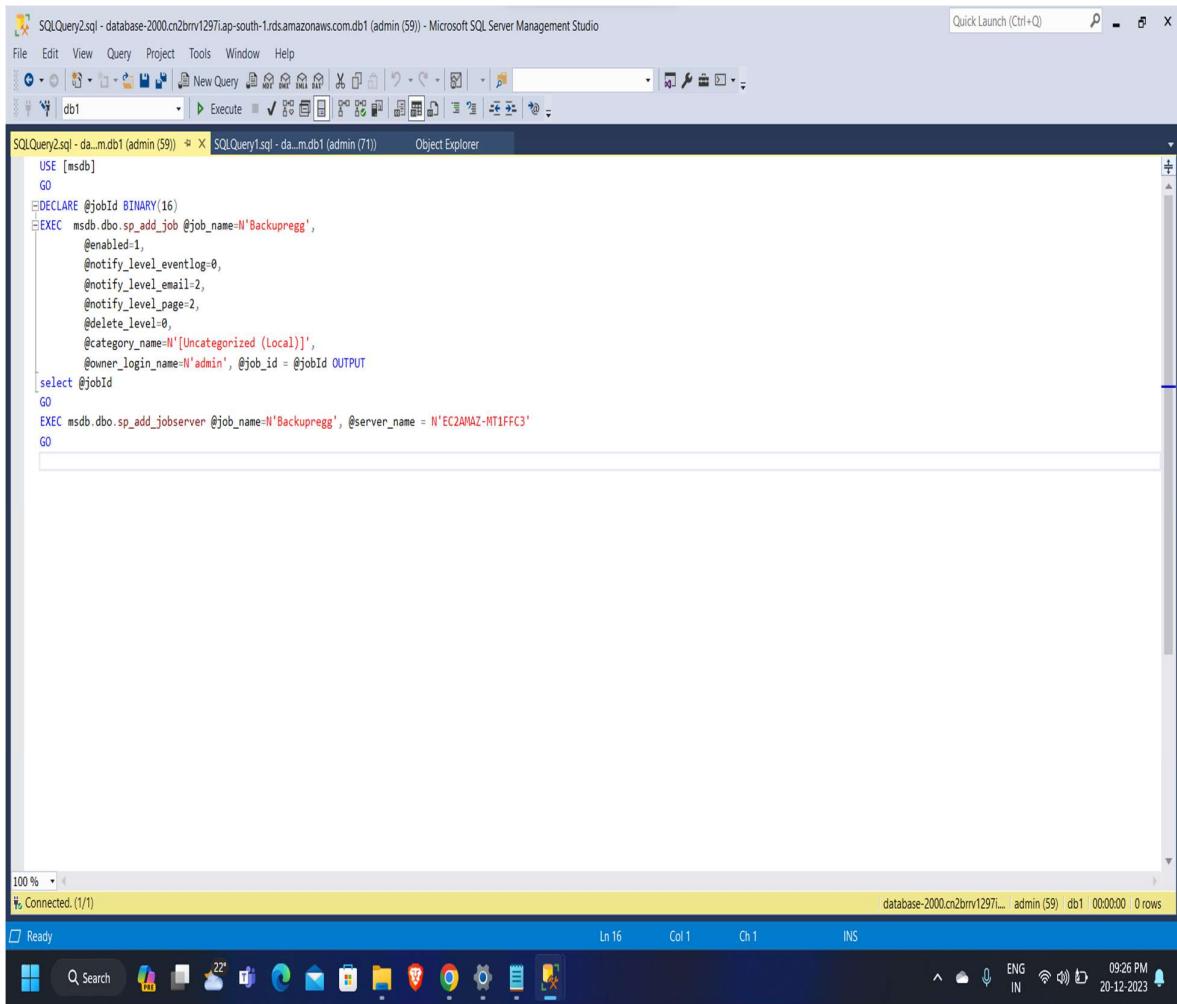
**STEP 36:- Once the database has been created successfully. Search for the SQL Server Agent in the object explorer of the SQL server.**



**STEP 37:- Expand the SQL Server Agent and you will find Jobs and click on jobs.  
Click on New job.**



## STEP 38:- Create a backup user profile for creating a Backup Script.



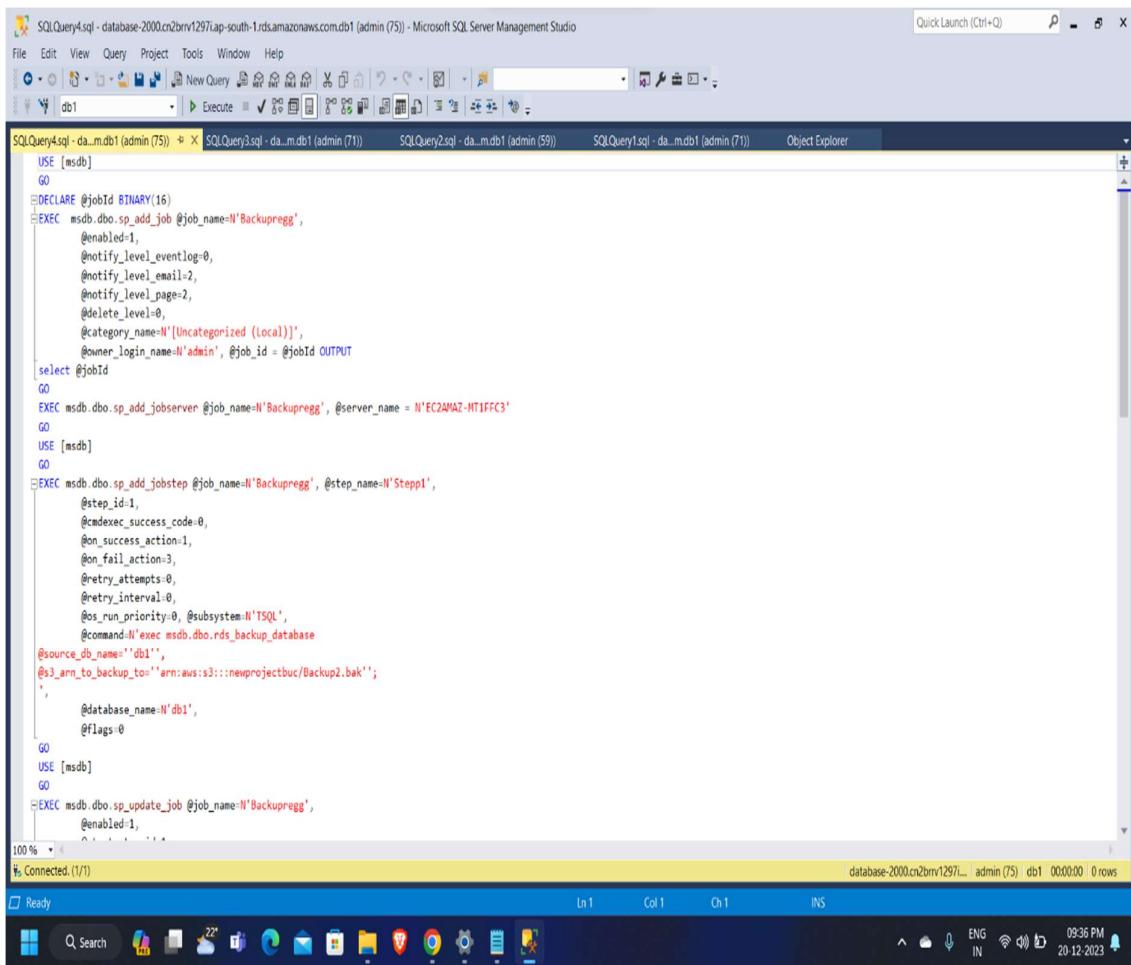
The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor window displays the following T-SQL script:

```
USE [msdb]
GO
DECLARE @jobId BINARY(16)
EXEC msdb.dbo.sp_add_job @job_name=N'Backupregg',
    @enabled=1,
    @notify_level_eventlog=0,
    @notify_level_email=2,
    @notify_level_page=2,
    @delete_level=0,
    @category_name=N'[Uncategorized (Local)]',
    @owner_login_name=N'admin', @job_id = @jobId OUTPUT
select @jobId
GO
EXEC msdb.dbo.sp_add_jobserver @job_name=N'Backupregg', @server_name = N'EC2AMAZ-MT1FFC3'
GO
```

The status bar at the bottom indicates "Connected. (1/1)" and "0 rows". The taskbar at the bottom right shows the date and time as "20-12-2023 09:26 PM".

**STEP 39:- Create a script which takes the created database and moves its replication to AWS S3 bucket regularly .**

- 1] Use the backup user profile which you created earlier.**
- 2] Provide the job step in the script, in job script write the commands which moves the data.**
- 3] Provide the extra steps like should a job stop when a failure occurs or it should retry again.**



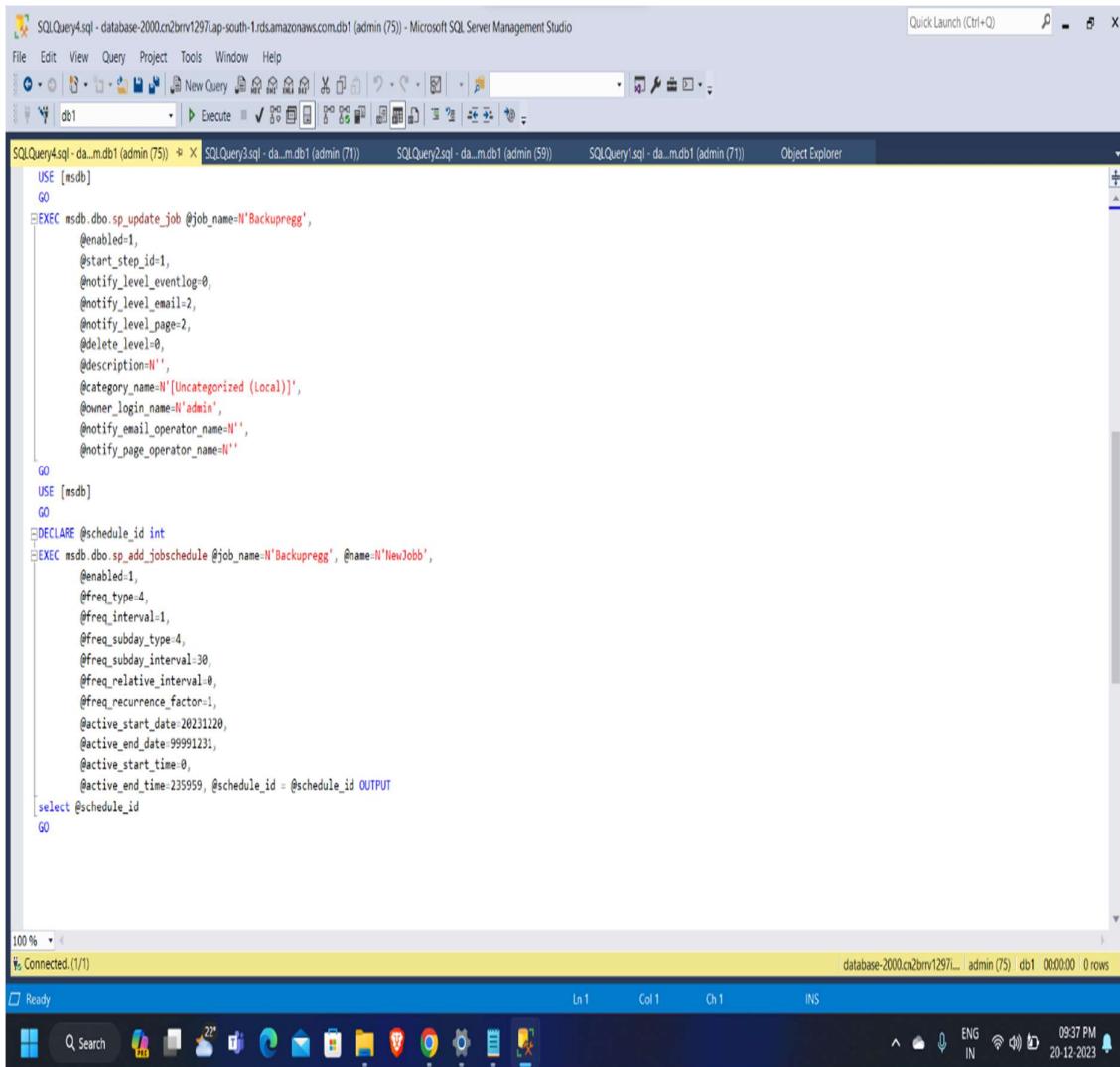
The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The title bar reads "SQLQuery4.sql - database-2000.cn2brv1297.ap-south-1.rds.amazonaws.com.db1 (admin (75)) - Microsoft SQL Server Management Studio". The main window displays a T-SQL script for creating a replication job:

```
USE [msdb]
GO
DECLARE @jobId BINARY(16)
EXEC msdb.dbo.sp_add_job @job_name=N'Backupregg',
    @enabled=1,
    @notify_level_eventlog=0,
    @notify_level_email=2,
    @notify_level_page=2,
    @delete_level=0,
    @category_name=N'[Uncategorized (Local)]',
    @owner_login_name=N'admin', @job_id = @jobId OUTPUT
select @jobId
GO
EXEC msdb.dbo.sp_add_jobserver @job_name=N'Backupregg', @server_name = N'EC2AMAZ-HT1FFC3'
GO
USE [msdb]
GO
EXEC msdb.dbo.sp_add_jobstep @job_name=N'Backupregg', @step_name=N'Stepp1',
    @step_id=1,
    @cmdexec_success_code=0,
    @on_success_action=1,
    @on_fail_action=3,
    @retry_attempts=0,
    @retry_interval=0,
    @os_run_priority=0, @subsystem=N'TSQL',
    @command=N'exec msdb.dbo.rds_backup_database
@source_db_name=''db1'',
@s3_arn_to_backup_to='arn:aws:s3:::newprojectbuc/Backup2.bak';
',
    @database_name=N'db1',
    @flags=0
GO
USE [msdb]
GO
EXEC msdb.dbo.sp_update_job @job_name=N'Backupregg',
    @enabled=1,
```

The status bar at the bottom indicates "Connected. (1/1)" and "database-2000.cn2brv1297... admin (75) db1 00:00:00 0 rows". The system tray shows the date and time as "20-12-2023 09:35 PM".

**STEP 40:- Provide the additional functionalities in script such as job scheduling and backup timing.**

- 1] Schedule the timing to which a script should run like daily, weekly or monthly.**
- 2] Provide the timings from when a automatic backup can start.**
- 3] Provide the times when the backup should not be performed.**
- 4] Provide the command termination date.**

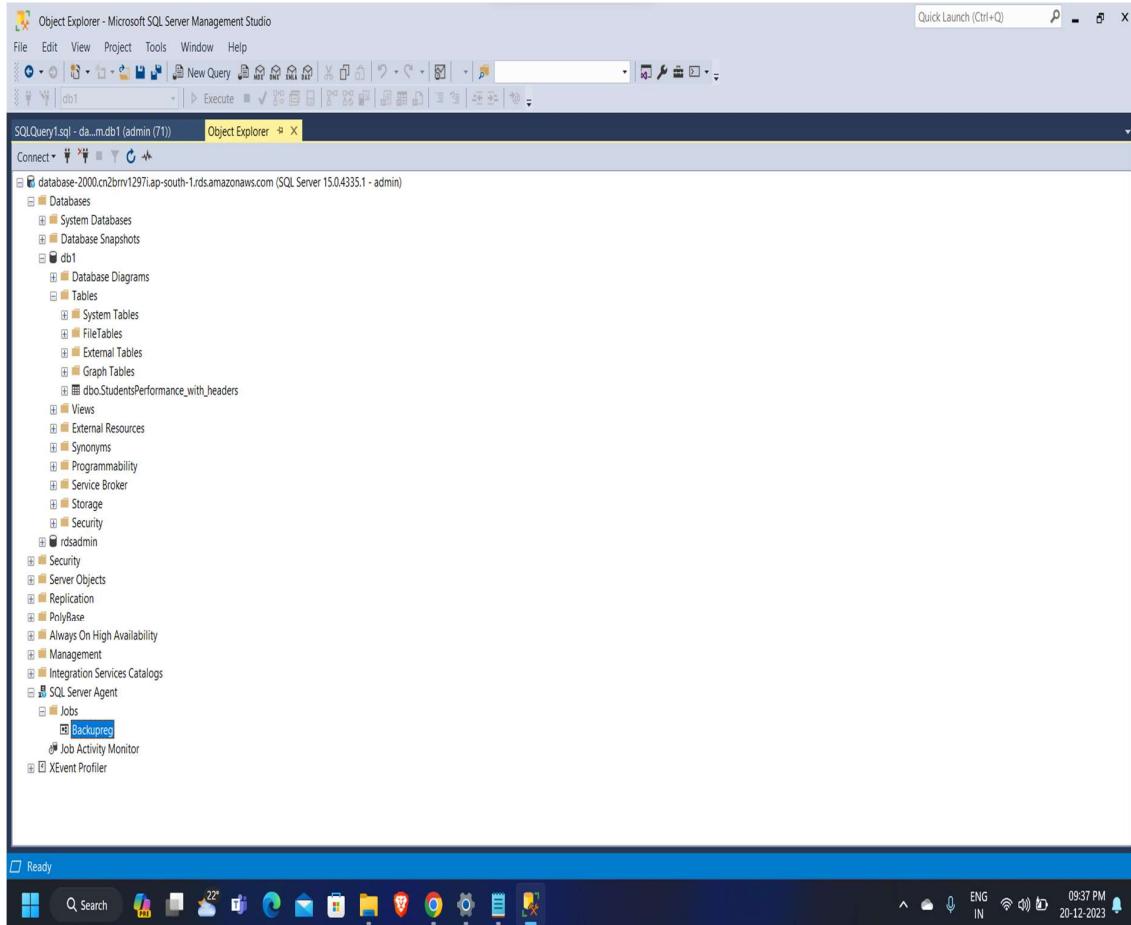


The screenshot shows a Microsoft SQL Server Management Studio (SSMS) interface. The query window contains the following T-SQL script:

```
USE [msdb]
GO
EXEC msdb.dbo.sp_update_job @job_name=N'Backupregg',
    @enabled=1,
    @start_step_id=1,
    @notify_level_eventlog=0,
    @notify_level_email=2,
    @notify_level_page=2,
    @delete_level=0,
    @description=N'',
    @category_name=N'[Uncategorized (Local)]',
    @owner_login_name=N'admin',
    @notify_email_operator_name=N'',
    @notify_page_operator_name=N''
GO
USE [msdb]
GO
DECLARE @schedule_id int
EXEC msdb.dbo.sp_add_jobschedule @job_name=N'Backupregg', @name=N'NewJobb',
    @enabled=1,
    @freq_type=4,
    @freq_interval=1,
    @freq_subday_type=4,
    @freq_subday_interval=30,
    @freq_relative_interval=0,
    @freq_recurrence_factor=1,
    @active_start_date=20231220,
    @active_end_date=99991231,
    @active_start_time=0,
    @active_end_time=235959, @schedule_id = @schedule_id OUTPUT
select @schedule_id
GO
```

The status bar at the bottom indicates "Connected (1/1)" and "database=2000.cn2brv1297... admin (75) db1 00:00:00 0 rows". The system tray shows the date and time as "20-12-2023 09:37 PM".

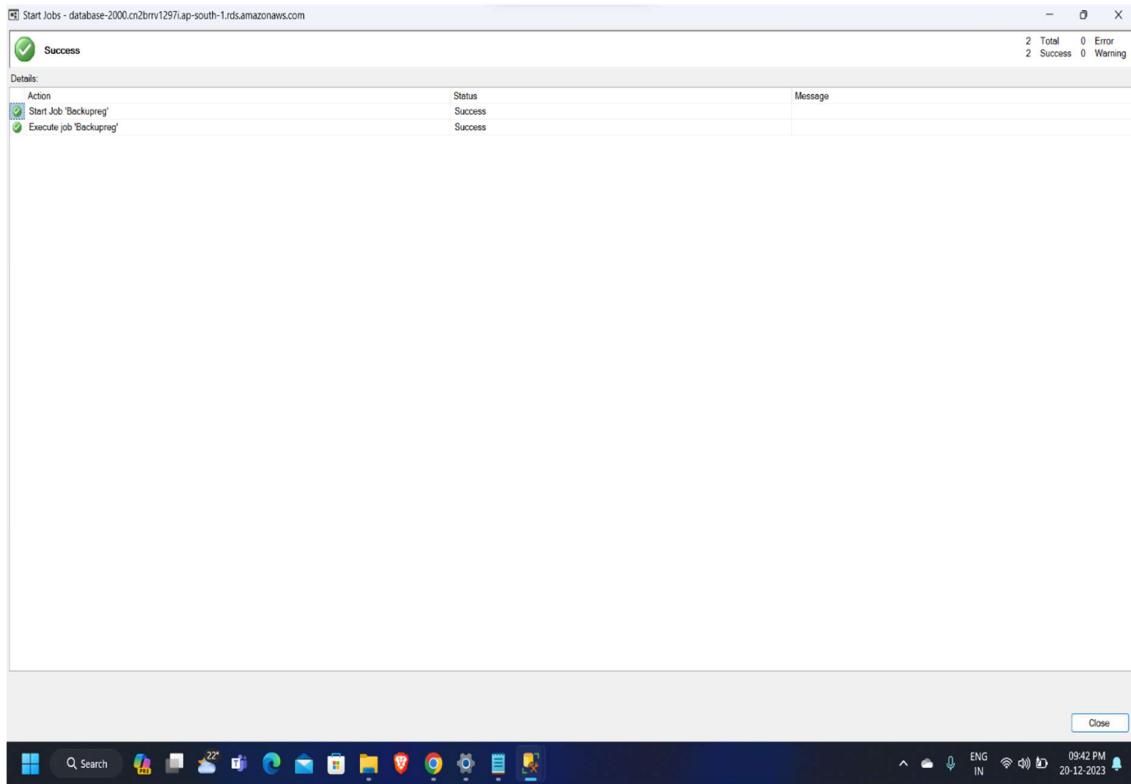
**STEP 41:- Now Save and Run the script and go back to the object explorer. Refresh the object explorer and go to SQL Server Agent and expand the Jobs you will find a new job with the name which was provided by user and the functionalities which were defined by the user.**



**STEP 42:- Once you find the job which was made by yourself, Try and Run that script manually once to check the success.**

**1] Start Job Action (If Pass Script Moves to execution or else script must be written again).**

**2] Execute Job Action (If the script passes this stage it means the script is ready to run automatically).**

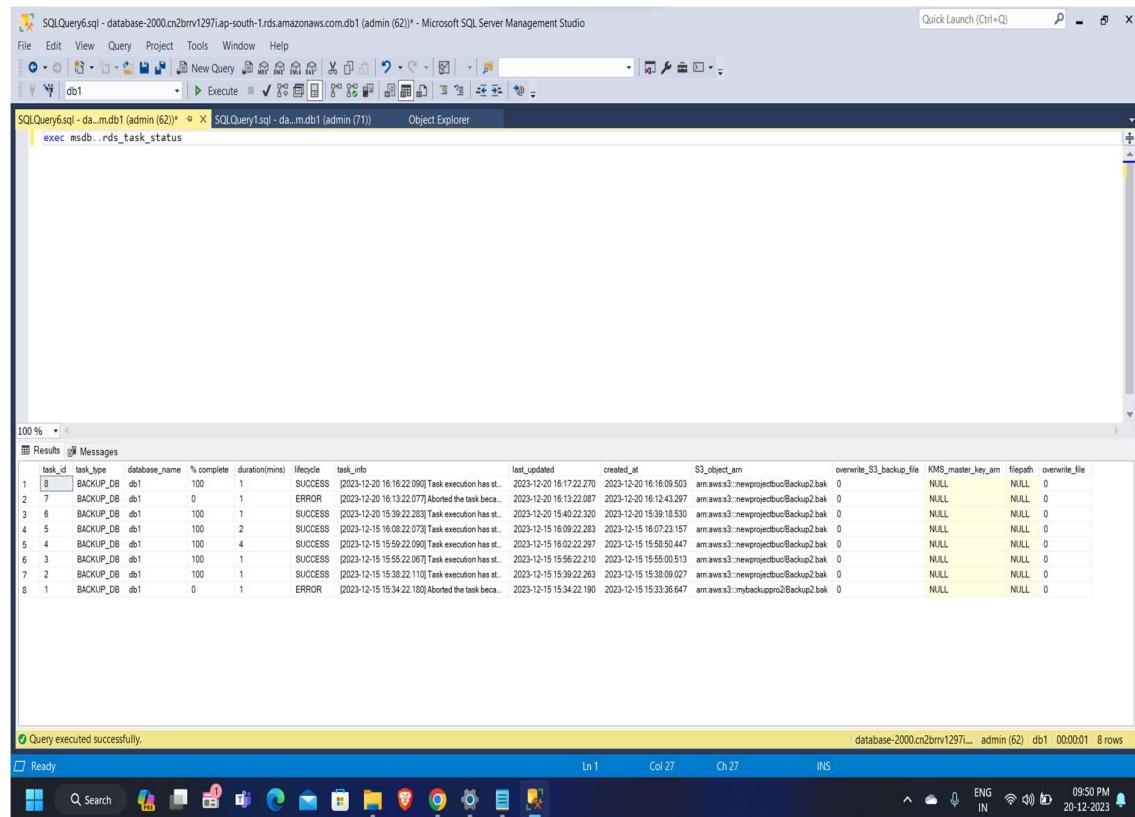


## TESTING: -

**Step 1:- Execute the below provided command to find the status of the automatically running scripts in background.**

```
# exec msdb..rds_task_status
```

**Execute this command to know the whether the script has passed or failed to backup the data.**



The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'SQLQuery6.sql' is open, displaying the results of the command '# exec msdb..rds\_task\_status'. The results grid shows 8 rows of task execution details. Below the grid, a message indicates 'Query executed successfully.' The system tray at the bottom right shows the date and time as 20-12-2023 09:50 PM.

task_id	task_type	database_name	% complete	duration(min)	life_cycle	task_info	last_updated	created_at	S3_object_arn	overwrite_S3_backup_file	KMS_master_key_arn	lifepath	overwrite_file
1	BACKUP_DB	db1	100	1	SUCCESS	(2023-12-20 16:18:22.090) Task execution has st...	2023-12-20 16:17:22.770	2023-12-20 16:18:09.503	am.aves.s3..newprojectus\$Backup2.bak	0	NULL	NULL	0
2	BACKUP_DB	db1	0	1	ERROR	(2023-12-20 16:13:22.077) Aborted the task beca...	2023-12-20 16:13:22.087	2023-12-20 16:14:23.297	am.aves.s3..newprojectus\$Backup2.bak	0	NULL	NULL	0
3	BACKUP_DB	db1	100	1	SUCCESS	(2023-12-20 15:39:22.283) Task execution has st...	2023-12-20 15:40:22.320	2023-12-20 15:39:18.530	am.aves.s3..newprojectus\$Backup2.bak	0	NULL	NULL	0
4	BACKUP_DB	db1	100	2	SUCCESS	(2023-12-19 16:09:22.073) Task execution has st...	2023-12-19 16:09:22.080	2023-12-19 16:07:23.157	am.aves.s3..newprojectus\$Backup2.bak	0	NULL	NULL	0
5	BACKUP_DB	db1	100	4	SUCCESS	(2023-12-19 15:59:22.090) Task execution has st...	2023-12-19 16:02:22.297	2023-12-19 15:58:50.447	am.aves.s3..newprojectus\$Backup2.bak	0	NULL	NULL	0
6	BACKUP_DB	db1	100	1	SUCCESS	(2023-12-19 15:59:22.067) Task execution has st...	2023-12-19 15:59:22.071	2023-12-19 15:59:22.053	am.aves.s3..newprojectus\$Backup2.bak	0	NULL	NULL	0
7	BACKUP_DB	db1	100	1	SUCCESS	(2023-12-19 15:39:22.110) Task execution has st...	2023-12-19 15:39:22.286	2023-12-19 15:38:09.027	am.aves.s3..newprojectus\$Backup2.bak	0	NULL	NULL	0
8	BACKUP_DB	db1	0	1	ERROR	(2023-12-19 15:34:22.180) Aborted the task beca...	2023-12-19 15:34:22.190	2023-12-19 15:33:36.647	am.aves.s3..mybackup02\$Backup2.bak	0	NULL	NULL	0

**STEP 2:- The script written by yourself runs perfectly and you can check and confirm this by going to your AWS console and search Aws S3 bucket and check the bucket whose ARN where used earlier to allow permissions. The bucket must contain the Backup copy of the database you wanted to be backed up.**

