Capstone Project Use Case 1

Data preparation using an Amazon RDS for MySQL database with AWS Glue DataBrew

Simran Agarwal

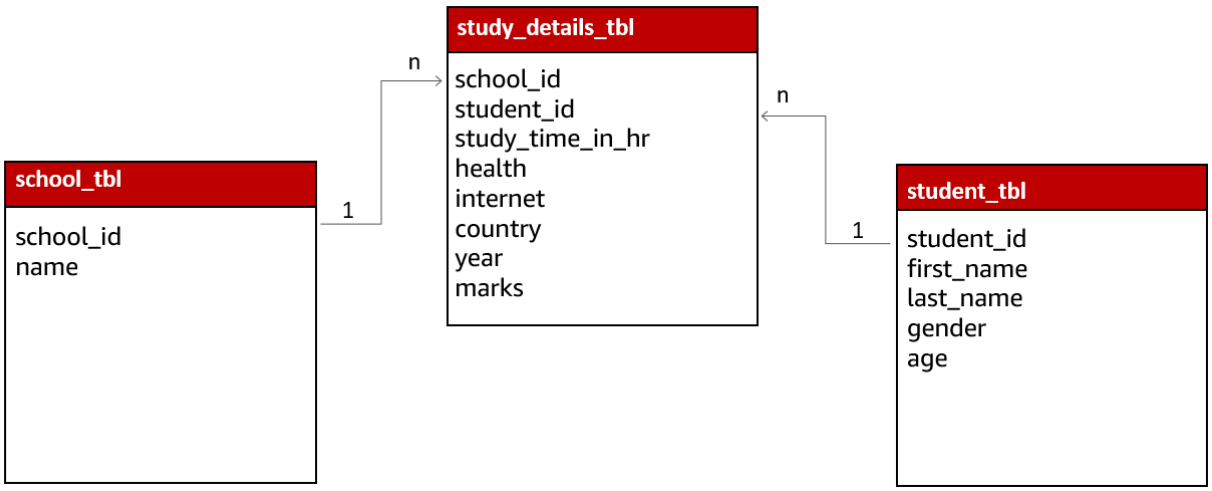
February 05, 2024

Chapter 1: Use case overview

For our use case, we use three datasets:

- A school dataset that contains school details like school ID and school name
- A student dataset that contains student details like student ID, name, and age
- A student study details dataset that contains student study time, health, country, and more

The following diagram shows the relation of these tables.

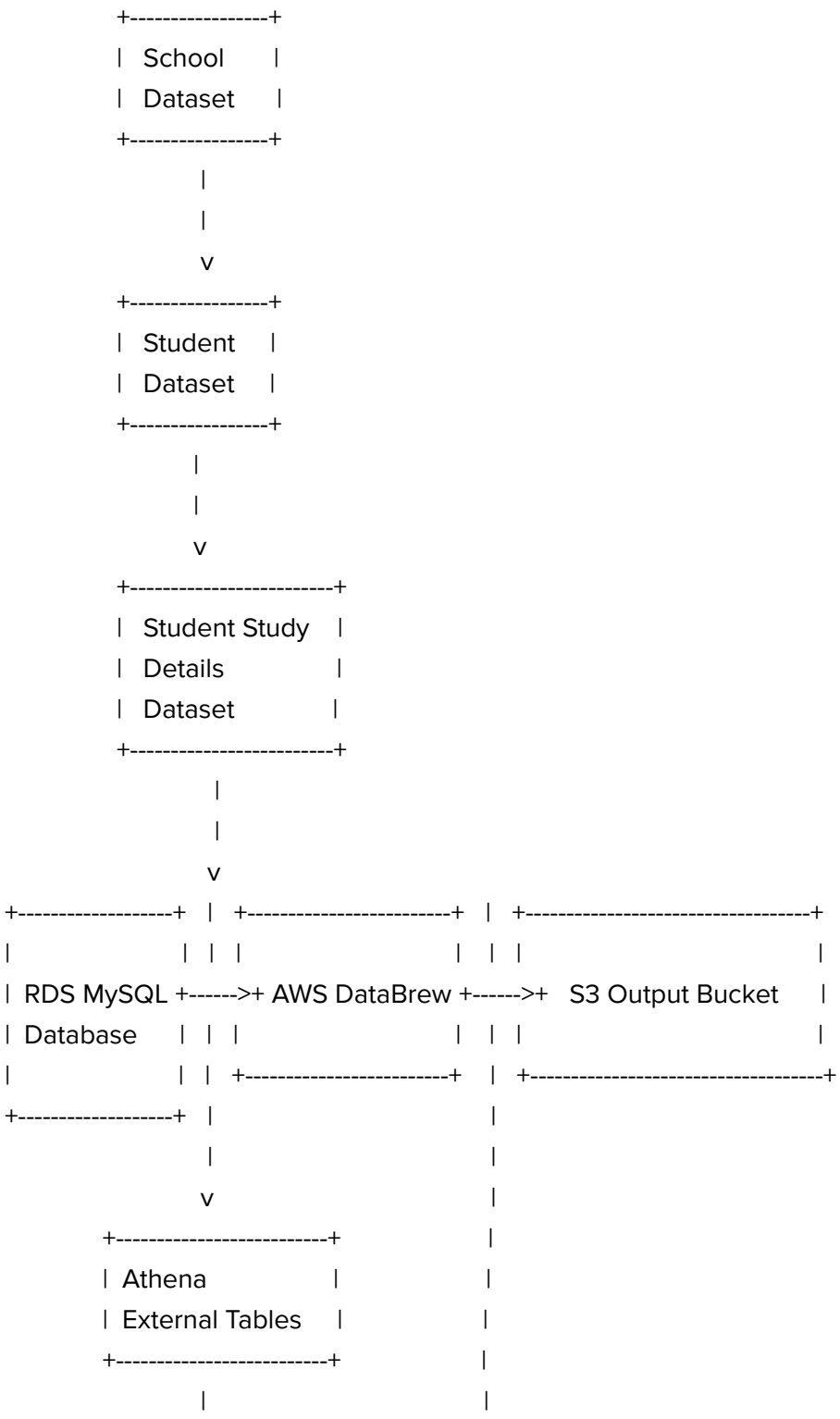


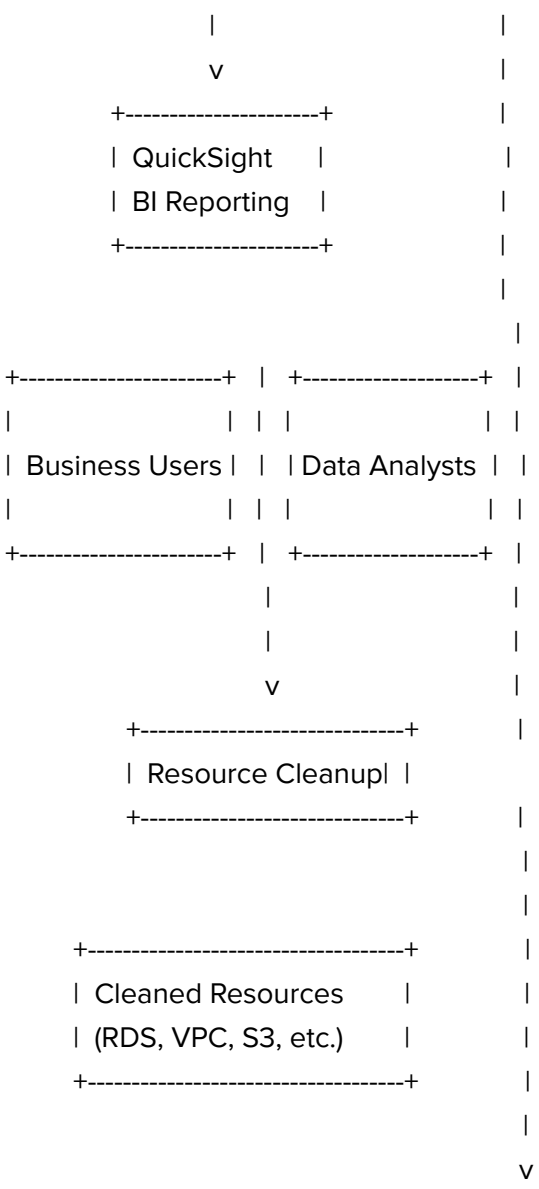
Workflow:

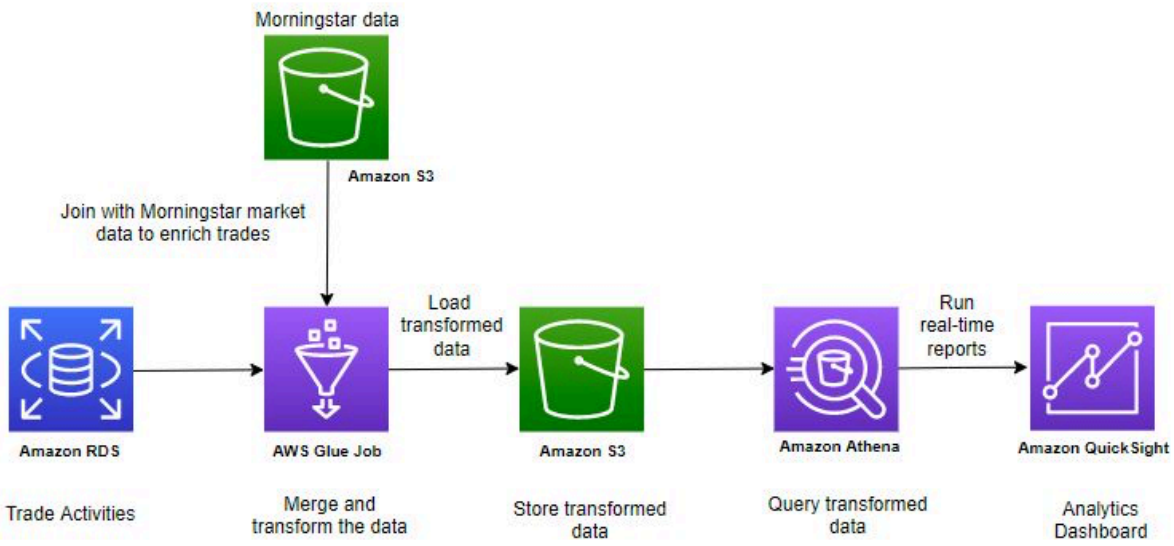
The workflow includes the following steps:

1. Create a JDBC connection for RDS and a DataBrew project. DataBrew does the transformation to find the top-performing students across all the schools considered for analysis.
2. The DataBrew job writes the final output to our S3 output bucket.
3. After the output data is written, we can create external tables on top of it with Athena create table statements.
4. Business users can use QuickSight for BI reporting, which fetches data through Athena. Data analysts can also use Athena to analyze the complete refreshed dataset.

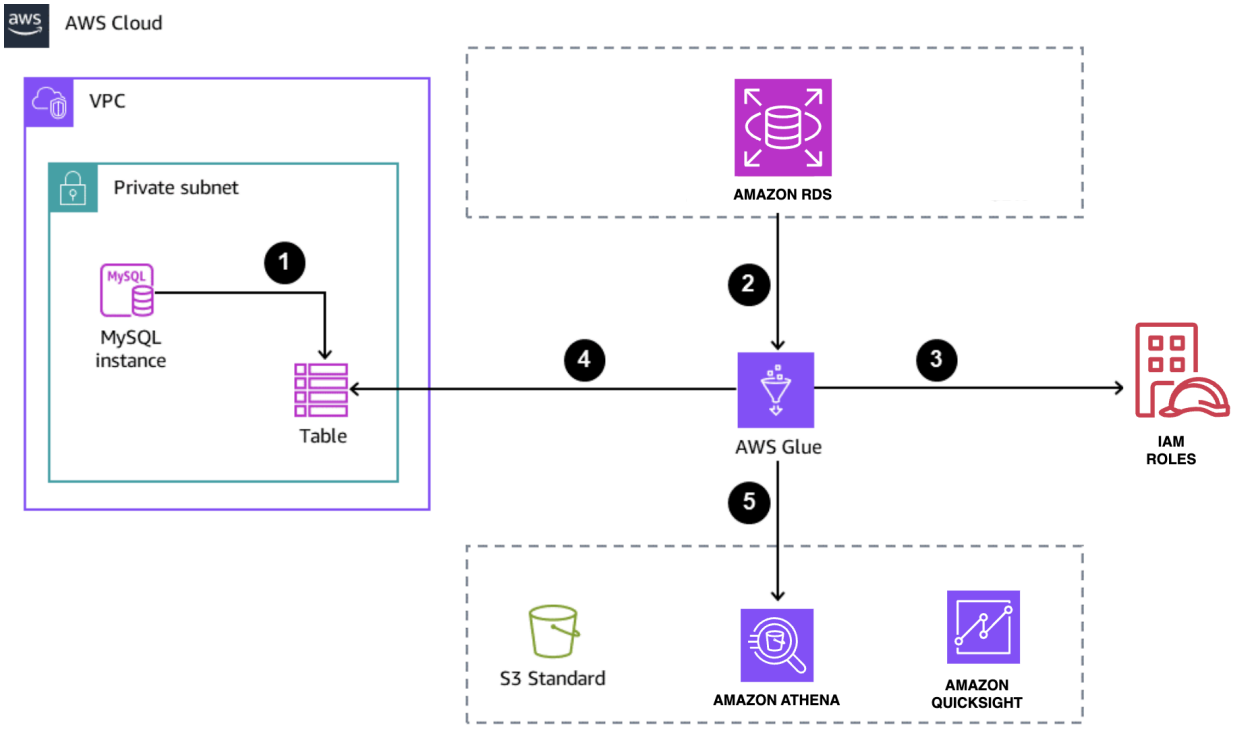
Chapter 2: Architecture Diagram




QuickSight
BI Reporting
Business Users
Data Analysts
Resource Cleanup
Cleaned Resources
(RDS, VPC, S3, etc.)

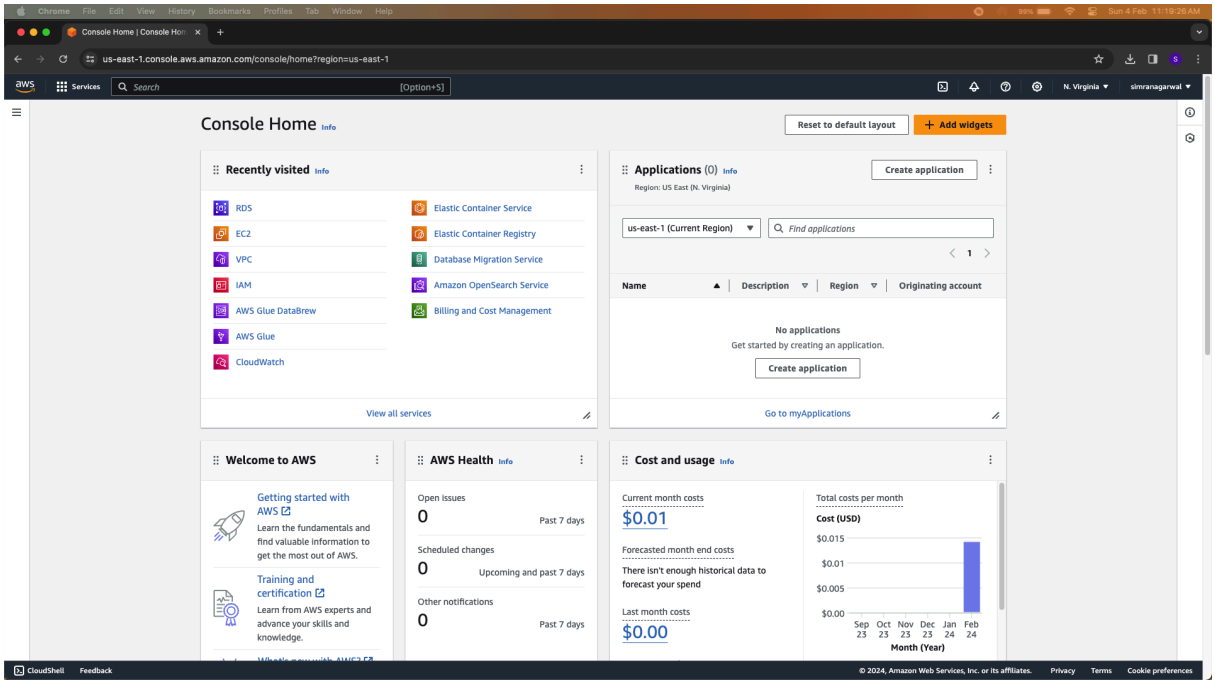


Final Architectural Diagram:



Chapter 3: Pre-requisites

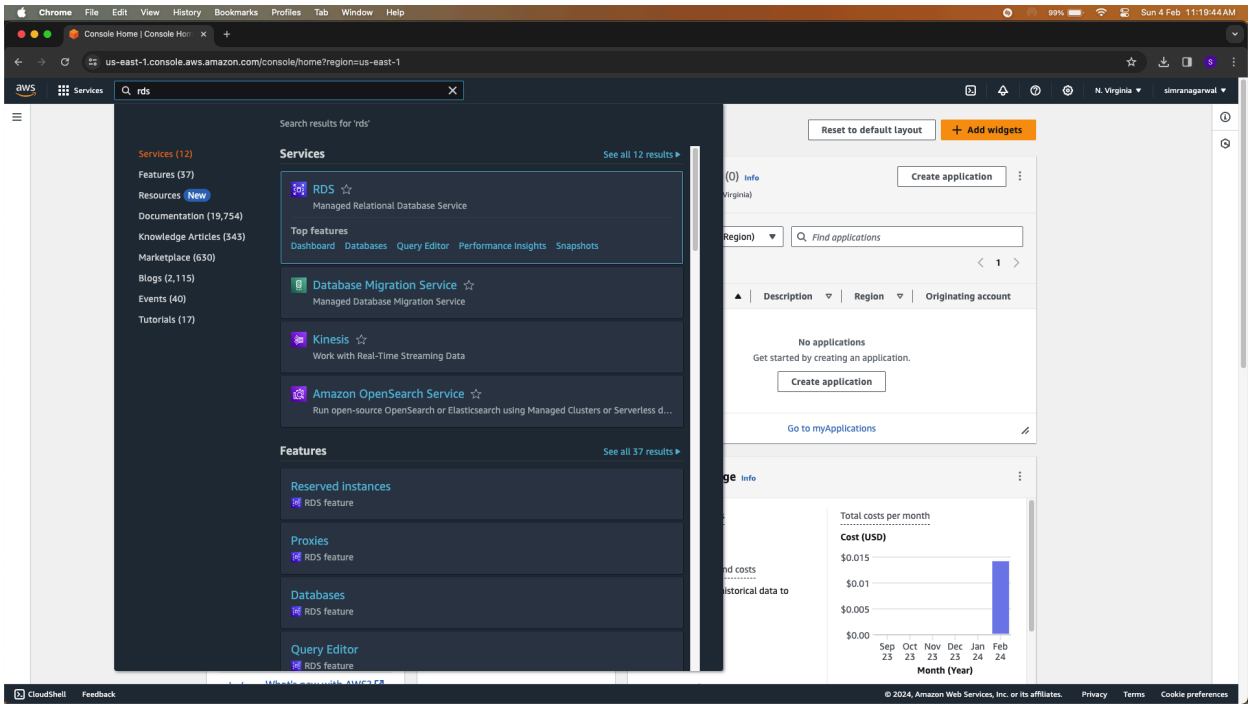
- To complete this solution you should have an ***Amazon AWS*** Account.



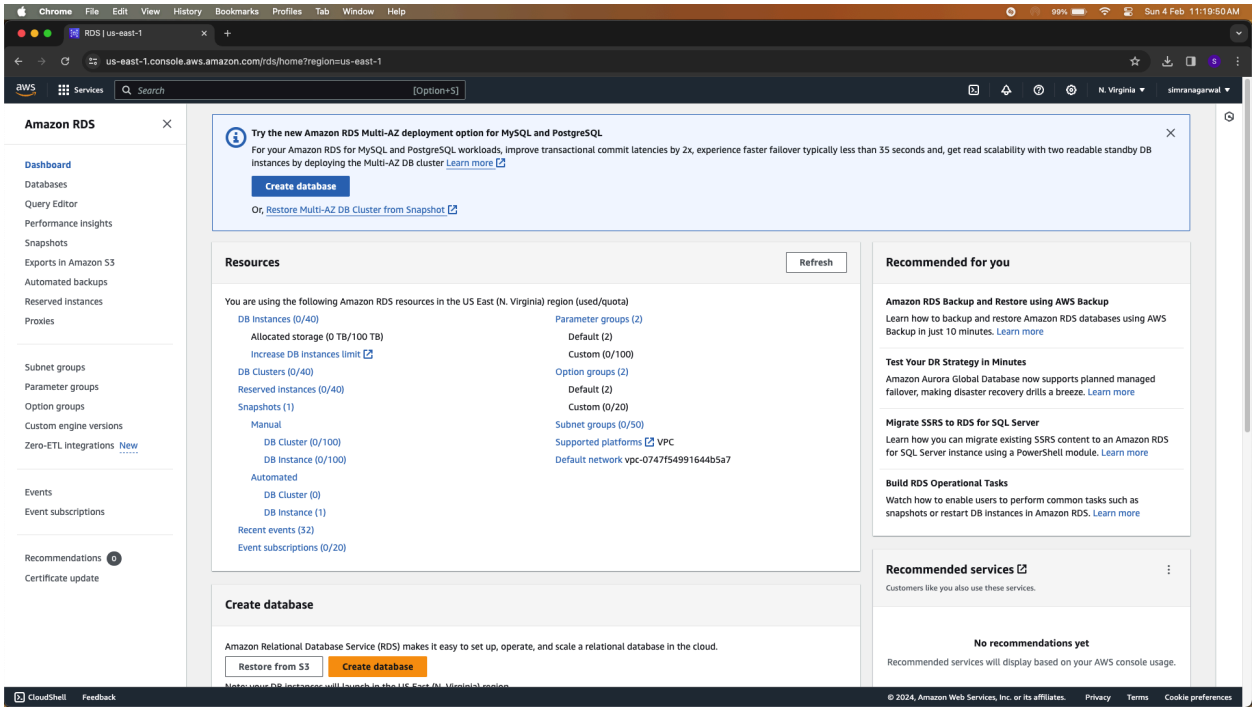
Chapter 4: Lab Setup

1. Create the RDS for MySQL instance to capture the student health data.

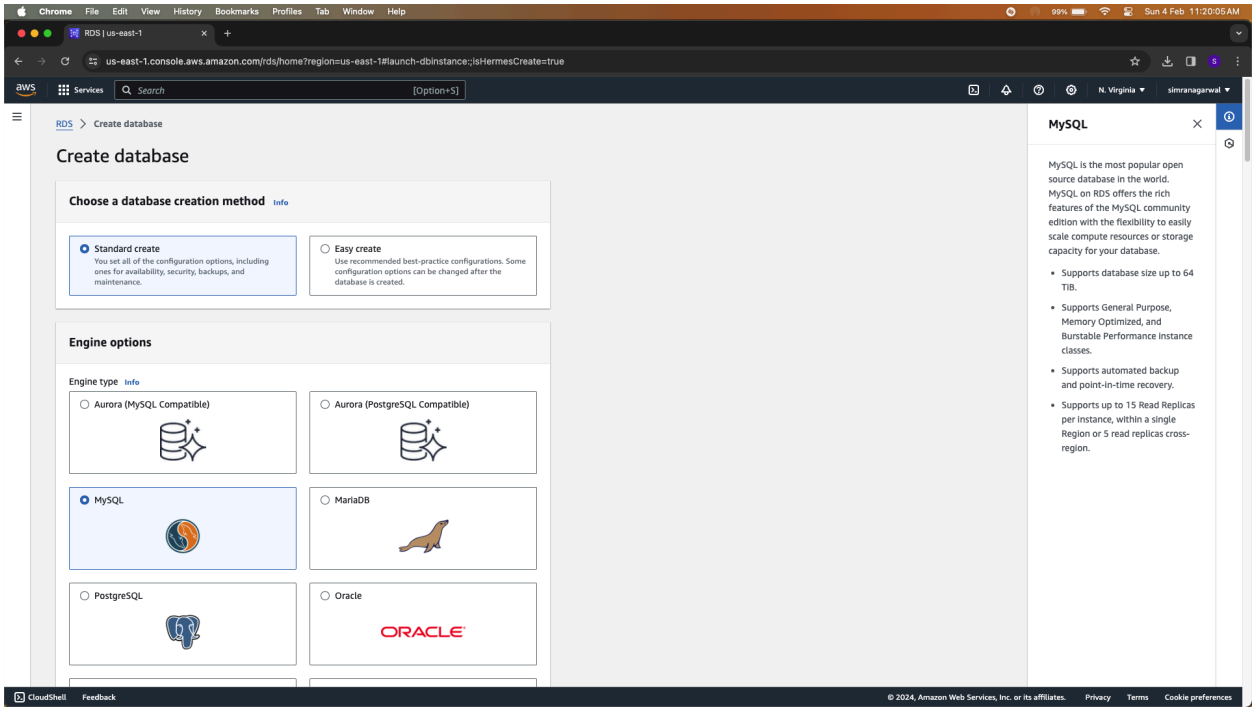
   (Search for RDS in AWS Management Console and go to the RDS Dashboard)



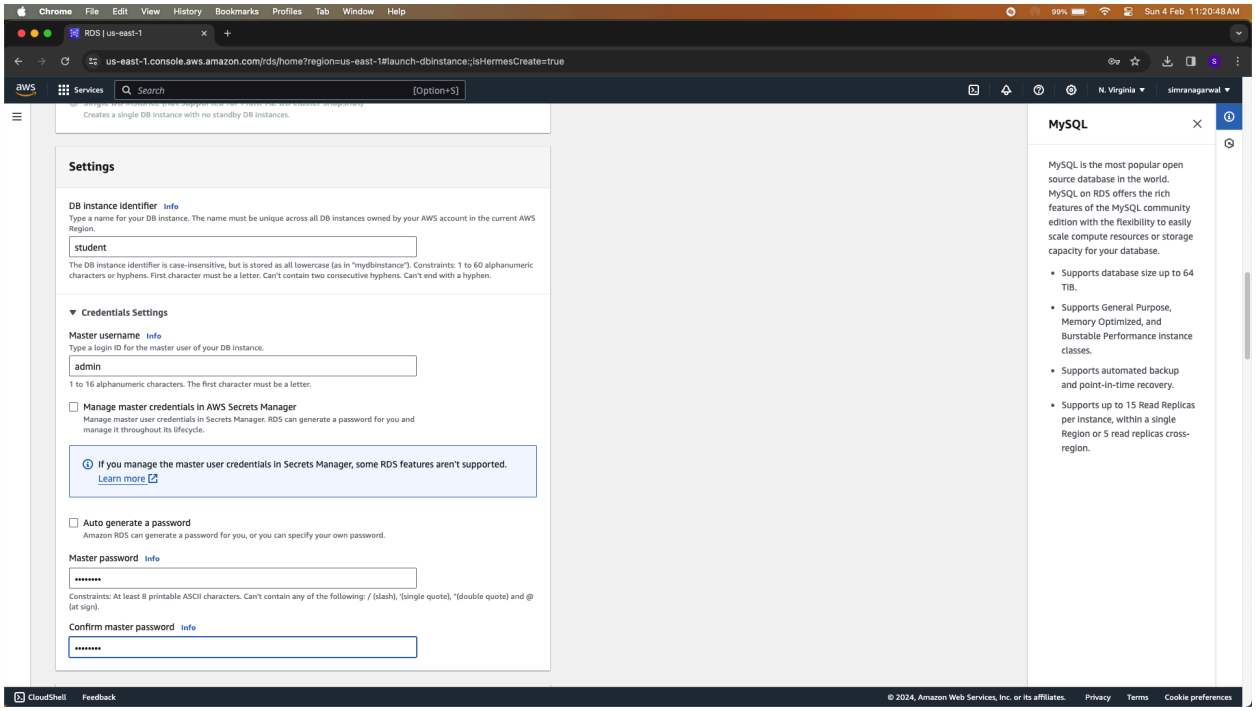
2. Click on Create Database.

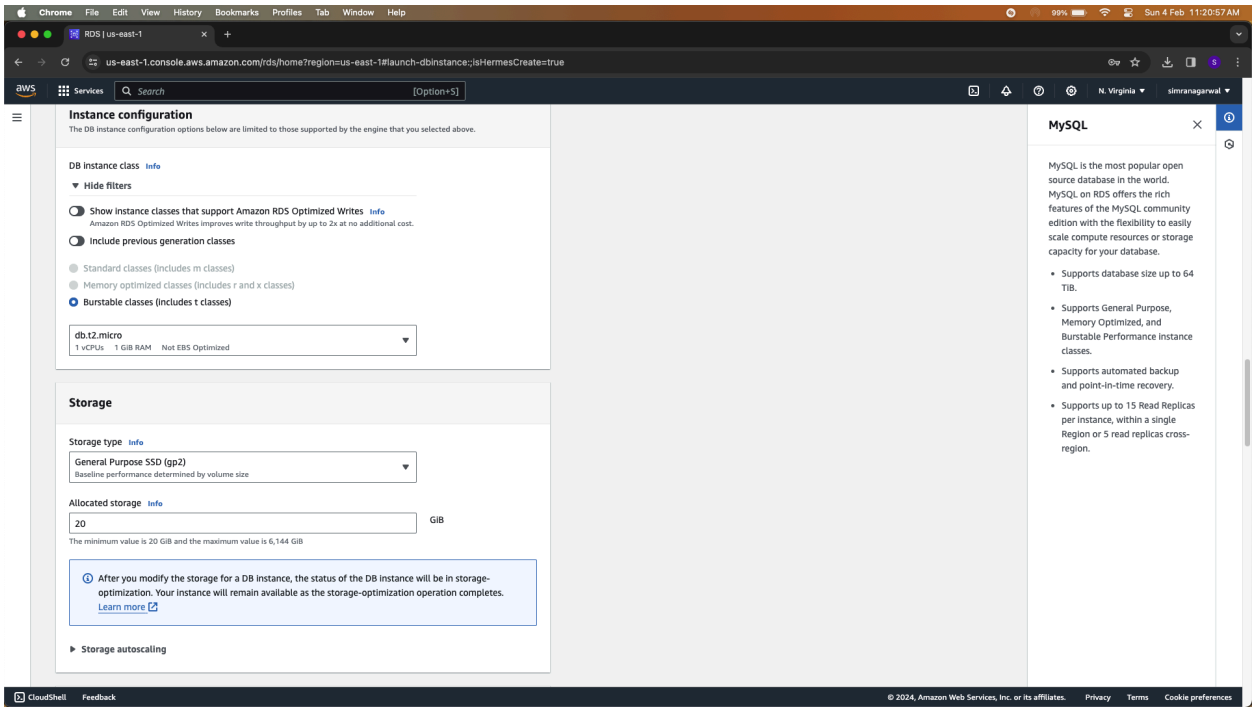


3. Choose a database creation method - “ Standard Create “ and Engine option as “MySQL“.

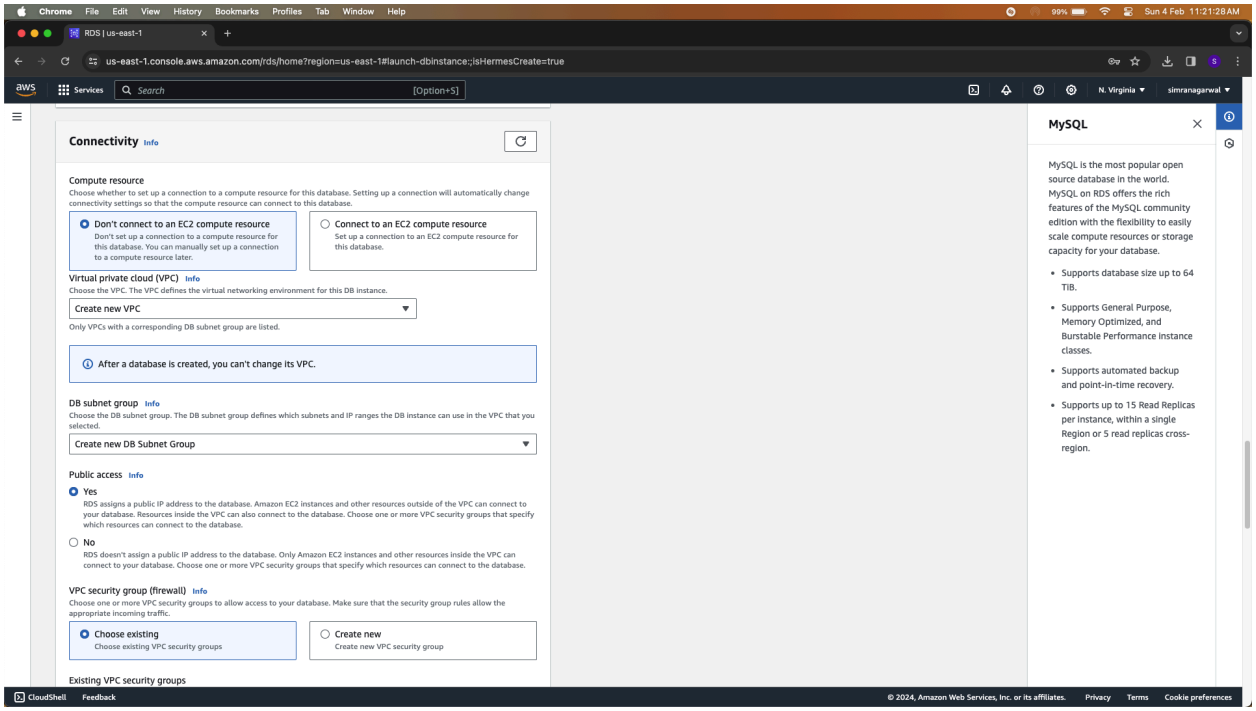


4. Check the Engine version and populate the fields (as shown in the diagrams below).



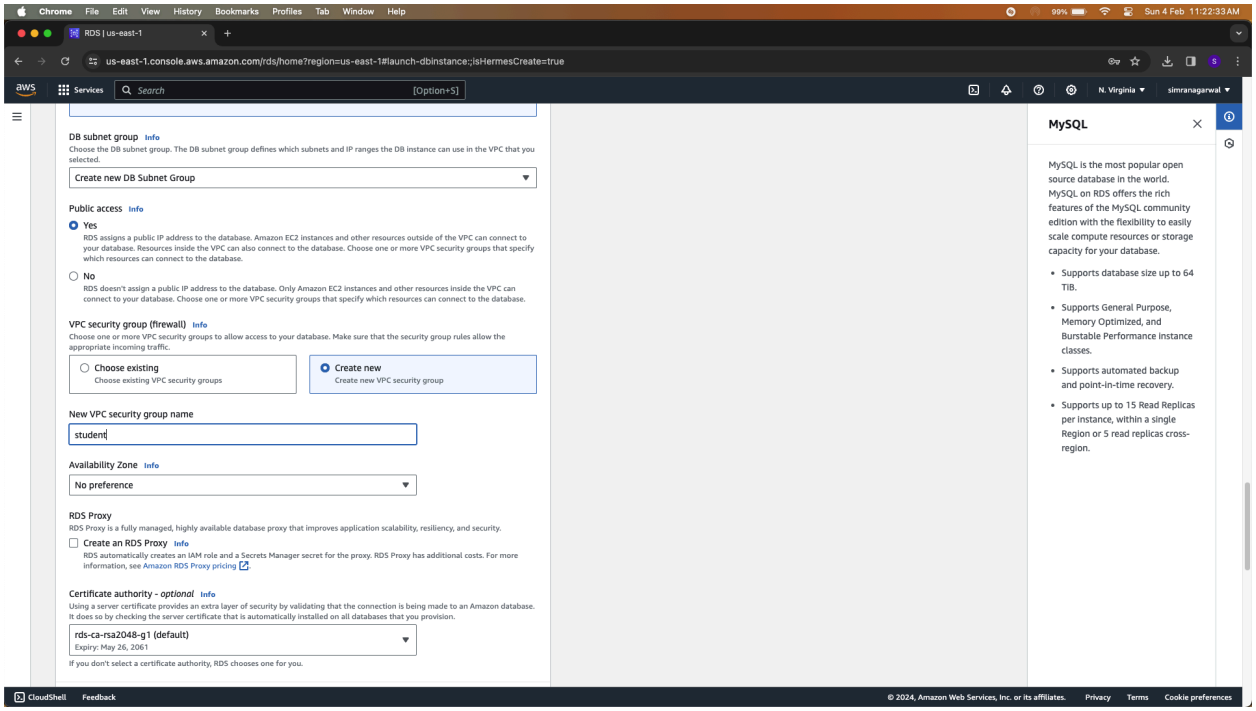


5. Don’t connect to an EC2 compute resource (not at all needed as we will be connecting to RDS MySQL database through VS Code.



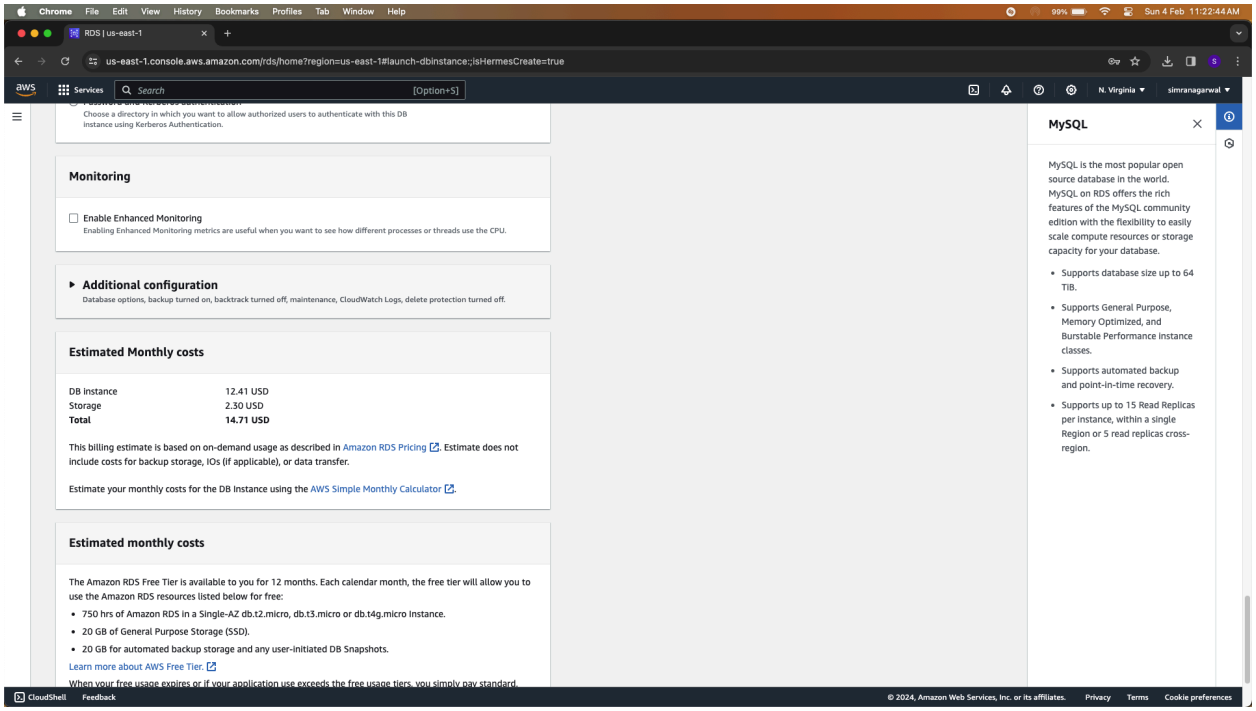
Also, allow the public access to access it from an external application.

6. Create a new subnet and VPC as well.

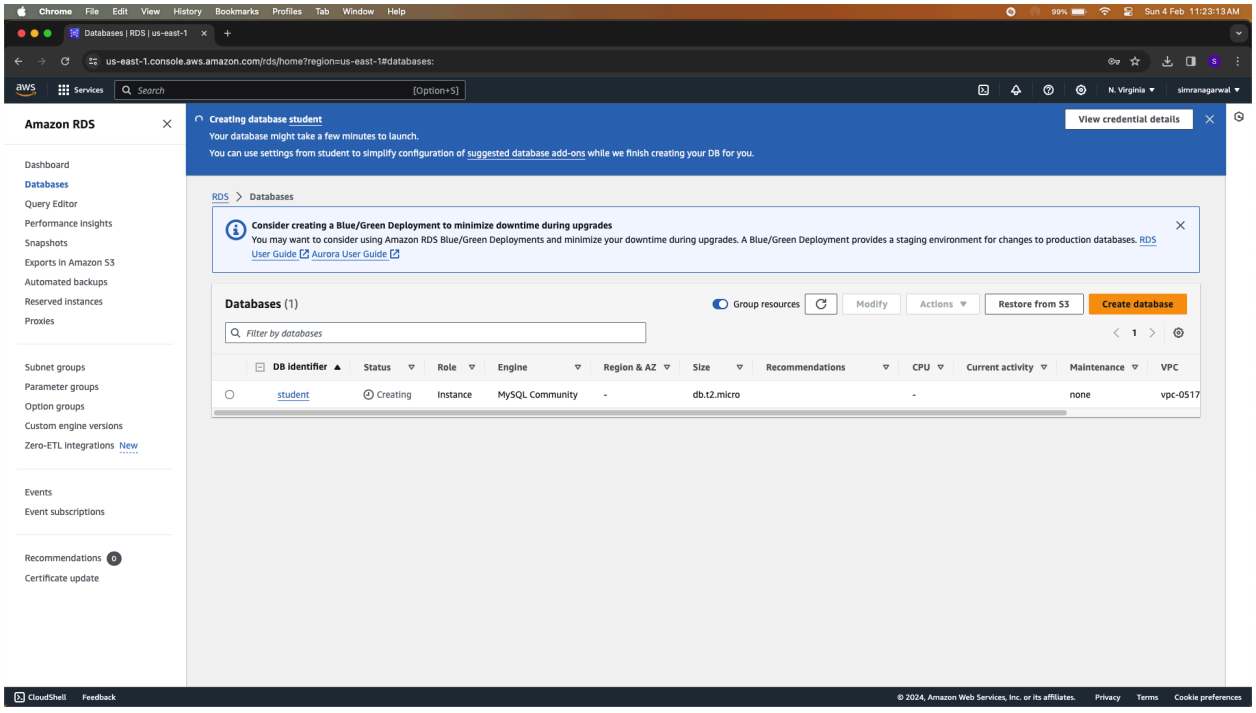


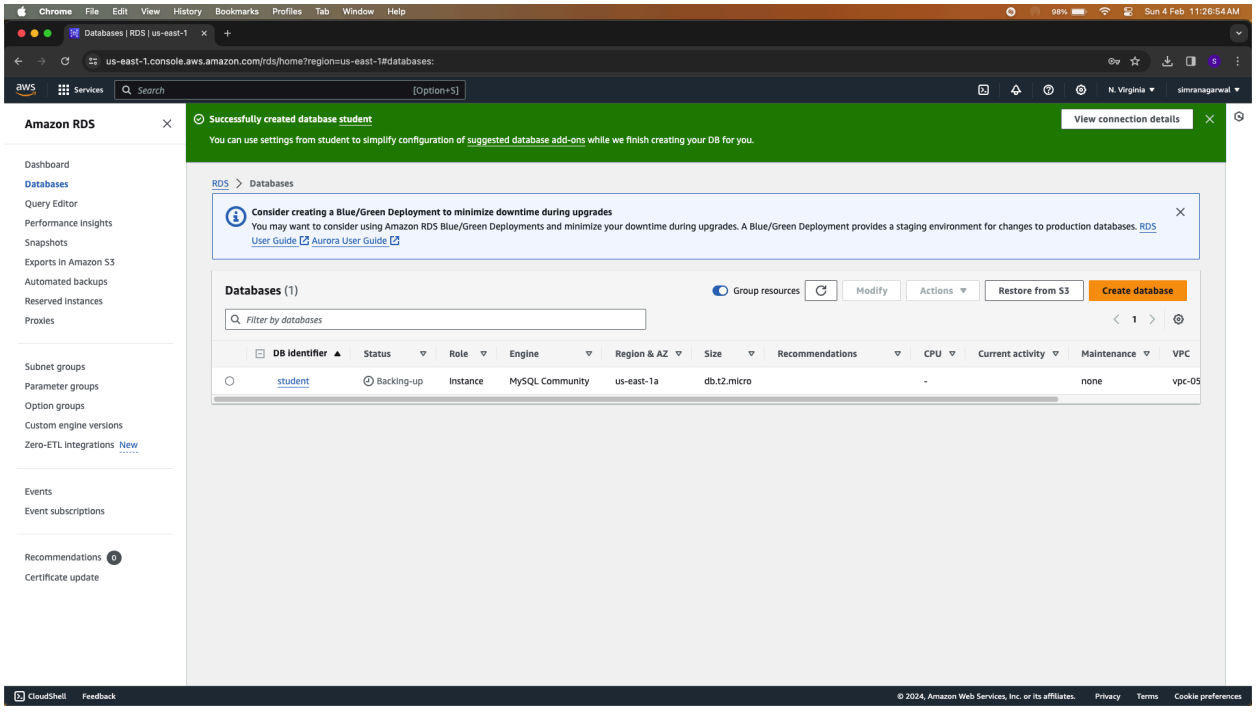
This will control the accessibility of the database from unknown sources (check the VPC and subnet settings in the VPC console).

7. Check the cost of your newly created RDS Database and click on the Create button at the bottom.

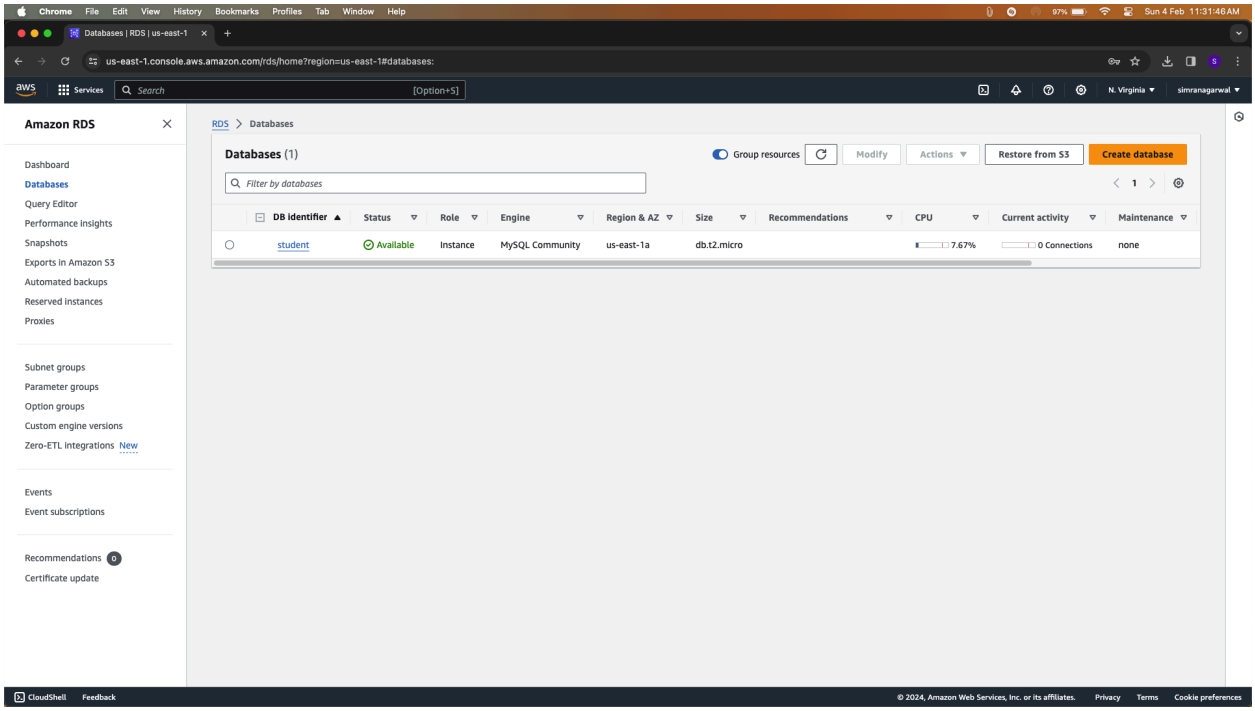


RDS Database creation might take a few minutes to be successfully available for use.

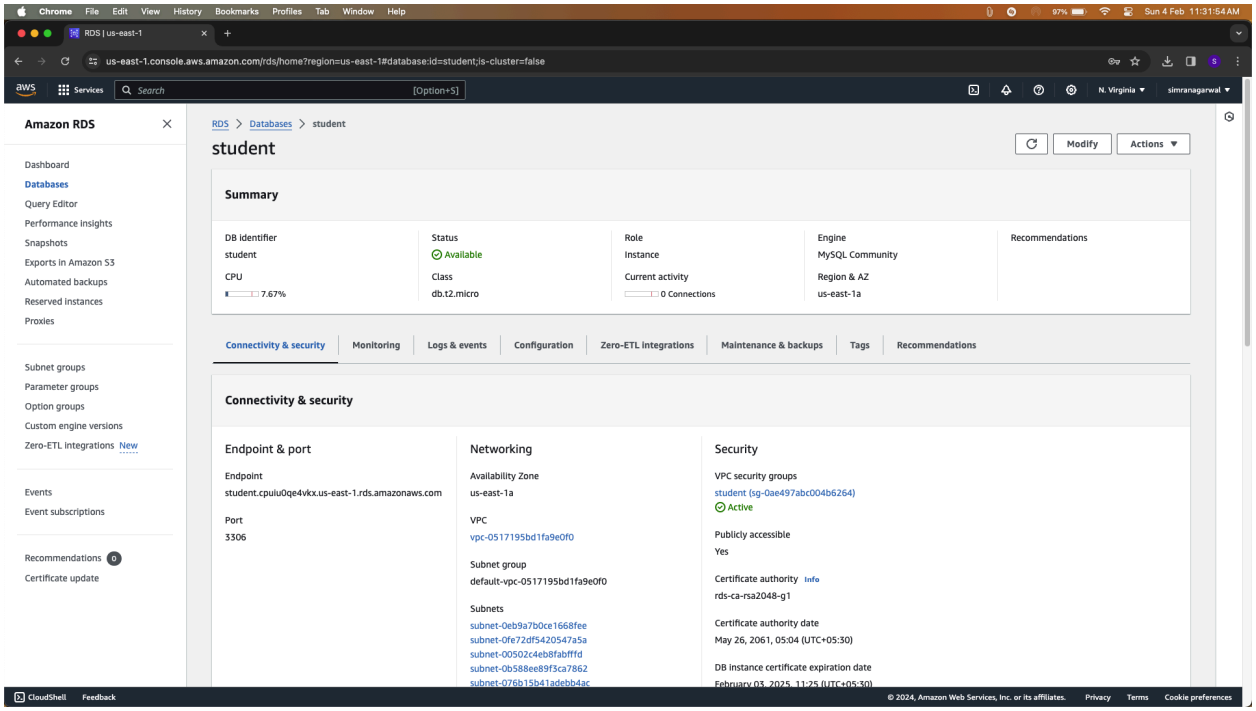

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sun 4 Feb 11:23:13 AM
Databases | RDS | us-east-1
us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#databases:
Services Search [Option+S]
N. Virginia
simranagarwal
Amazon RDS
Dashboard
Databases
Query Editor
Performance insights
Snapshots
Exports in Amazon S3
Automated backups
Reserved instances
Proxies
Subnet groups
Parameter groups
Option groups
Custom engine versions
Zero-ETL integrations New
Events
Event subscriptions
Recommendations 0
Certificate update
Creating database student
Your database might take a few minutes to launch.
You can use settings from student to simplify configuration of suggested database add-ons while we finish creating your DB for you.
View credential details
RDS > Databases
Consider creating a Blue/Green Deployment to minimize downtime during upgrades
You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. RDS User Guide Aurora User Guide
Databases (1)
Group resources
Modify
Actions
Restore from S3
Create database
Filter by databases
DB identifier
Status
Role
Engine
Region & AZ
Size
Recommendations
CPU
Current activity
Maintenance
VPC
student
Creating
Instance
MySQL Community
-
db.t2.micro
-
none
vpc-0517
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sun 4 Feb 11:26:54 AM
Databases | RDS | us-east-1
us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#databases:
Services Search [Option+S]
N. Virginia
simranagarwal
Amazon RDS
Dashboard
Databases
Query Editor
Performance insights
Snapshots
Exports in Amazon S3
Automated backups
Reserved instances
Proxies
Subnet groups
Parameter groups
Option groups
Custom engine versions
Zero-ETL integrations New
Events
Event subscriptions
Recommendations 0
Certificate update
Successfully created database student
You can use settings from student to simplify configuration of suggested database add-ons while we finish creating your DB for you.
View connection details
RDS > Databases
Consider creating a Blue/Green Deployment to minimize downtime during upgrades
You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. RDS User Guide Aurora User Guide
Databases (1)
Group resources
Modify
Actions
Restore from S3
Create database
Filter by databases
DB identifier
Status
Role
Engine
Region & AZ
Size
Recommendations
CPU
Current activity
Maintenance
VPC
student
Backing-up
Instance
MySQL Community
us-east-1a
db.t2.micro
-
none
vpc-05
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

8. Our RDS MySQL Database is ready to use.



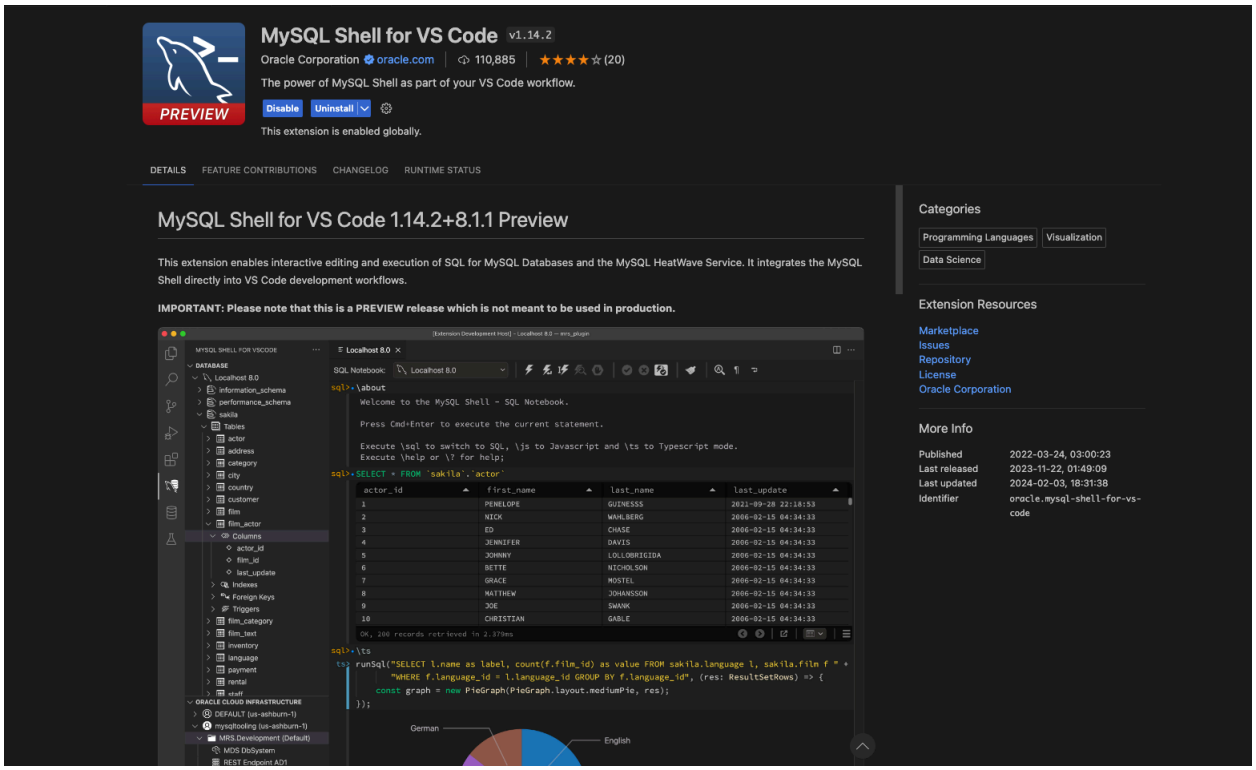
You can see the detailed configuration of the RDS instance created here.



9. Meanwhile, install Visual Studio Code and login through your Microsoft or GitHub account (whichever you feel is preferable)
   As an alternative, you can also use MySQL Workbench.

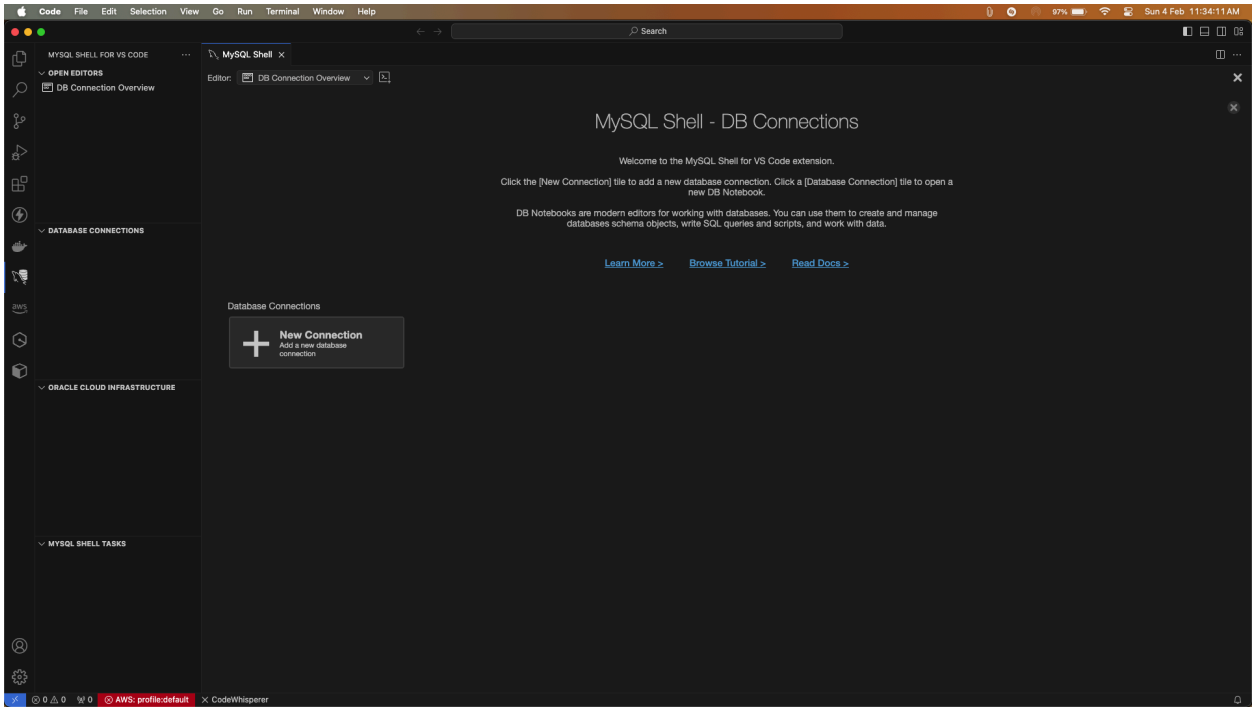
   After installing and setting up VS Code, now you are ready to use the extensions.

10. Now go to the extensions tab and search for MySQL Shell for VS Code.
    (There are many options available, but it is preferred that we use the one for Oracle Corporation.)

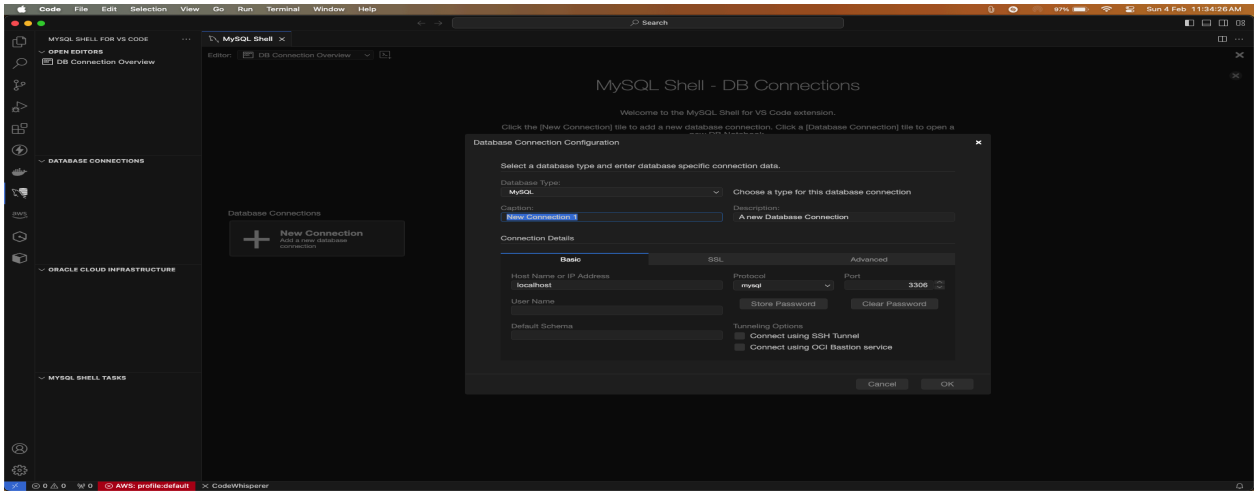


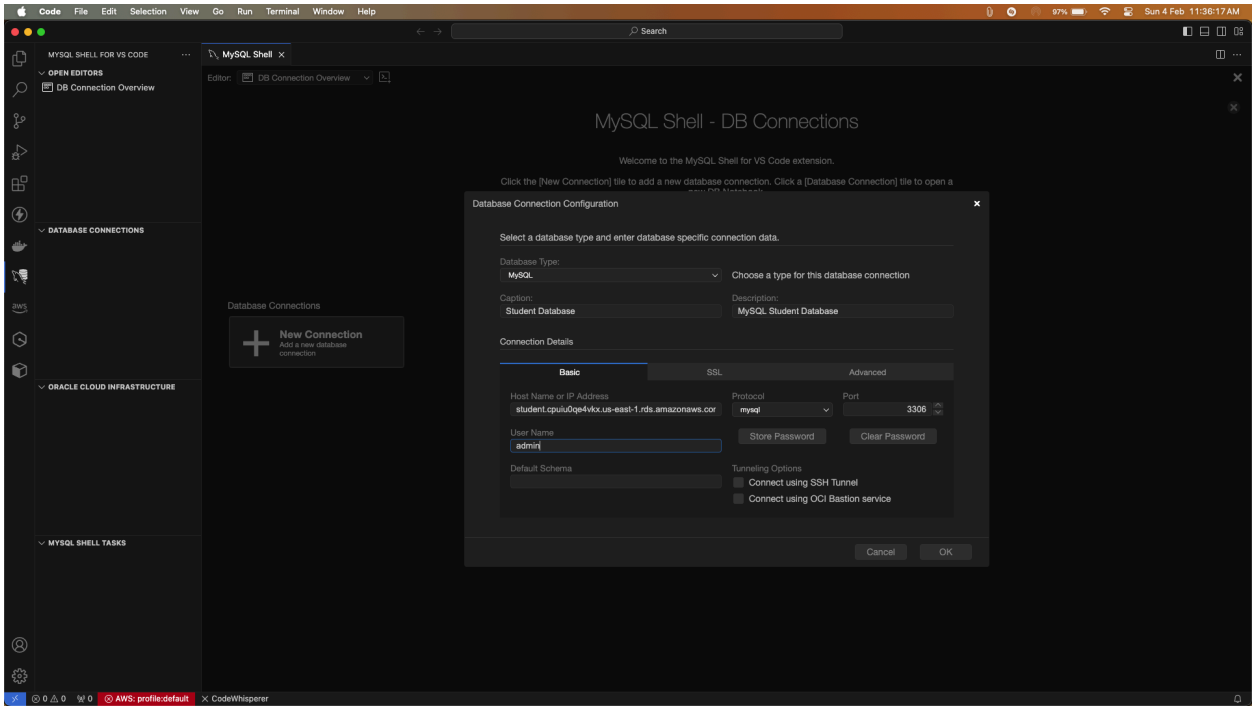
Click on install and wait for the installation to complete.

11. Now, the installation of MySQL Shell for VS Code is complete and we are ready to connect to a new database.

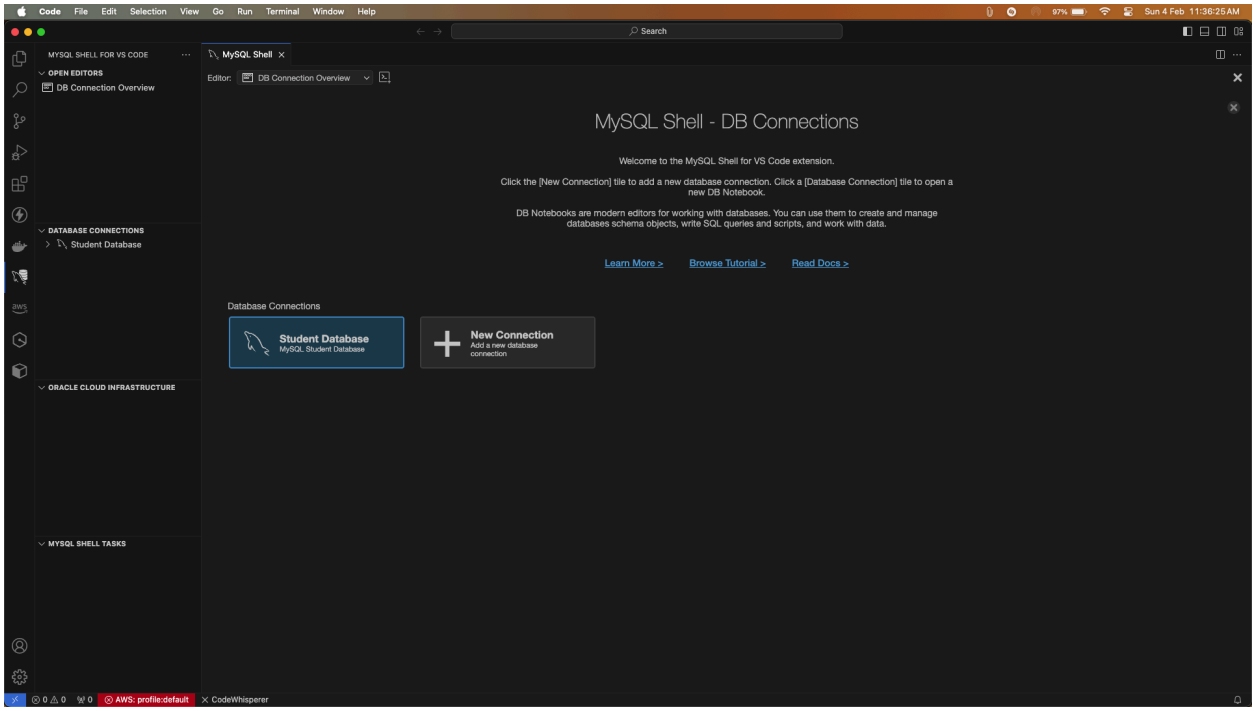


12. Fill in the details as follows:
    a. Database Type: MySQL
    b. Caption: The name of the connection (Alias) and also the description.
    c. Hostname: This will be the endpoint of the RDS instance created.
    d. Username: As provided while filling up the form for creating the RDS Database (in our case it is **admin**)
    e. Default Schema: Leave it as empty as we don't have any default schema set as of now.
    f. Port: Port number for the MySQL connection. (3306).

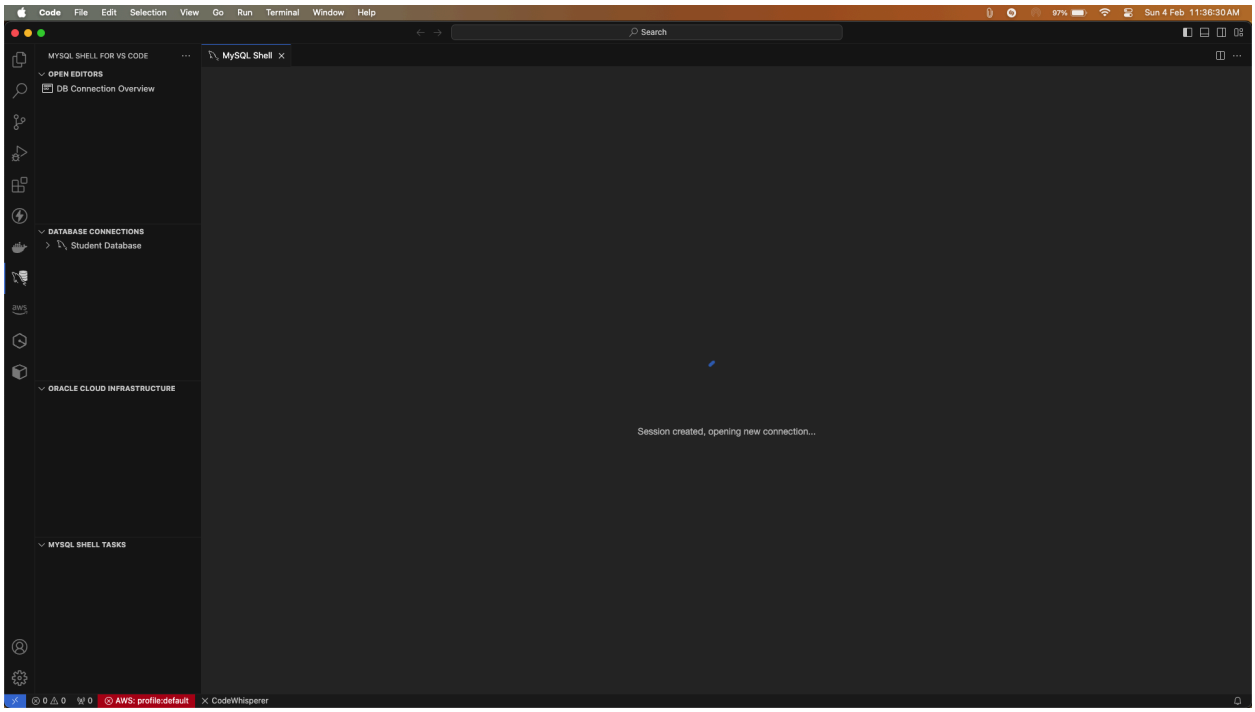




13. As you can see we have successfully created a connection. Now, it's time to connect to the RDS Database.

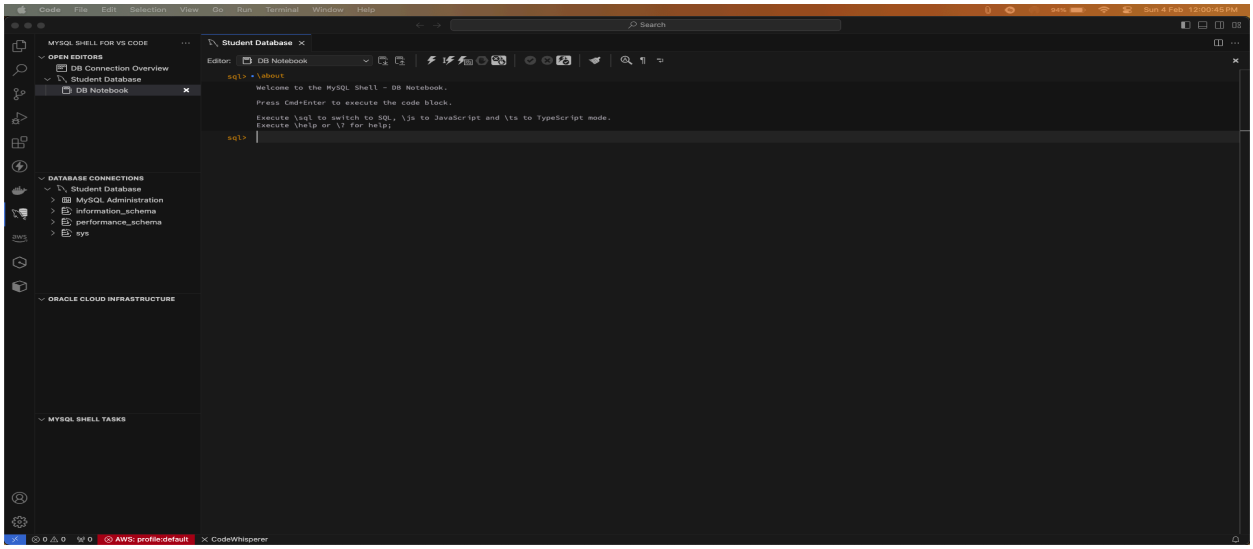


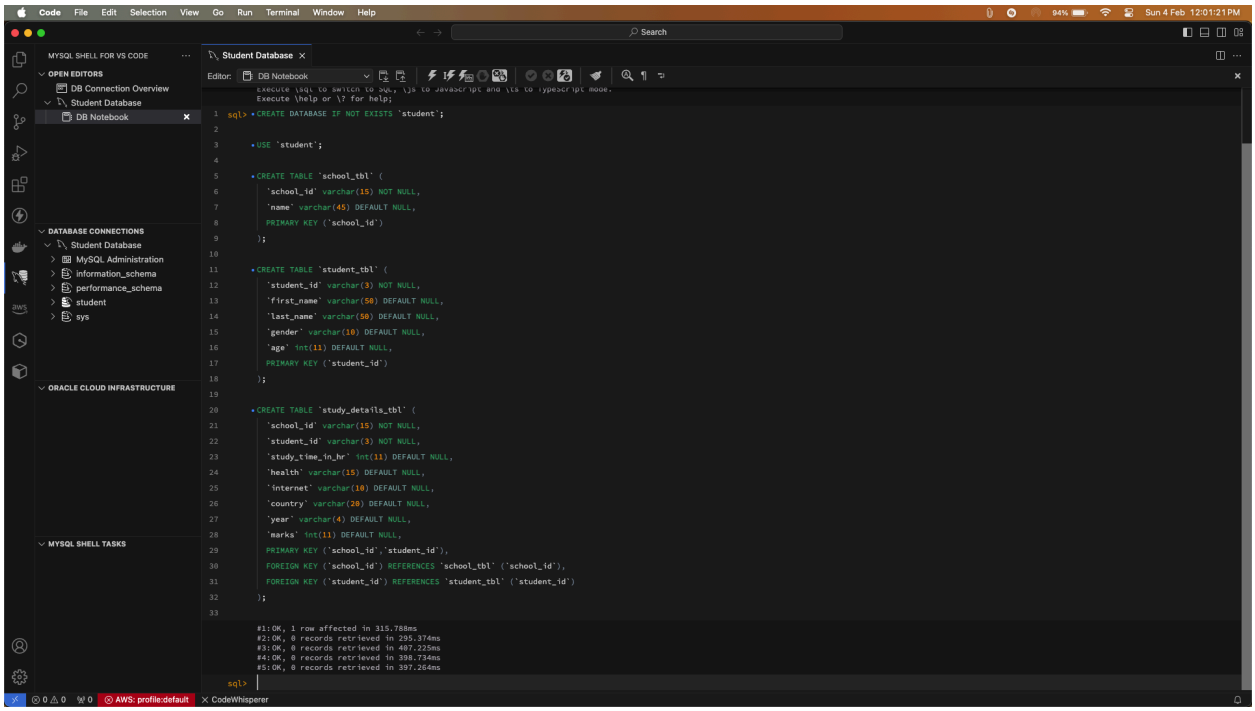
14. Now click on it and enter the password, this will connect you to a database connection that we created using AWS RDS (if you have followed the process correctly so far, you will be able to reach and connect to the database)



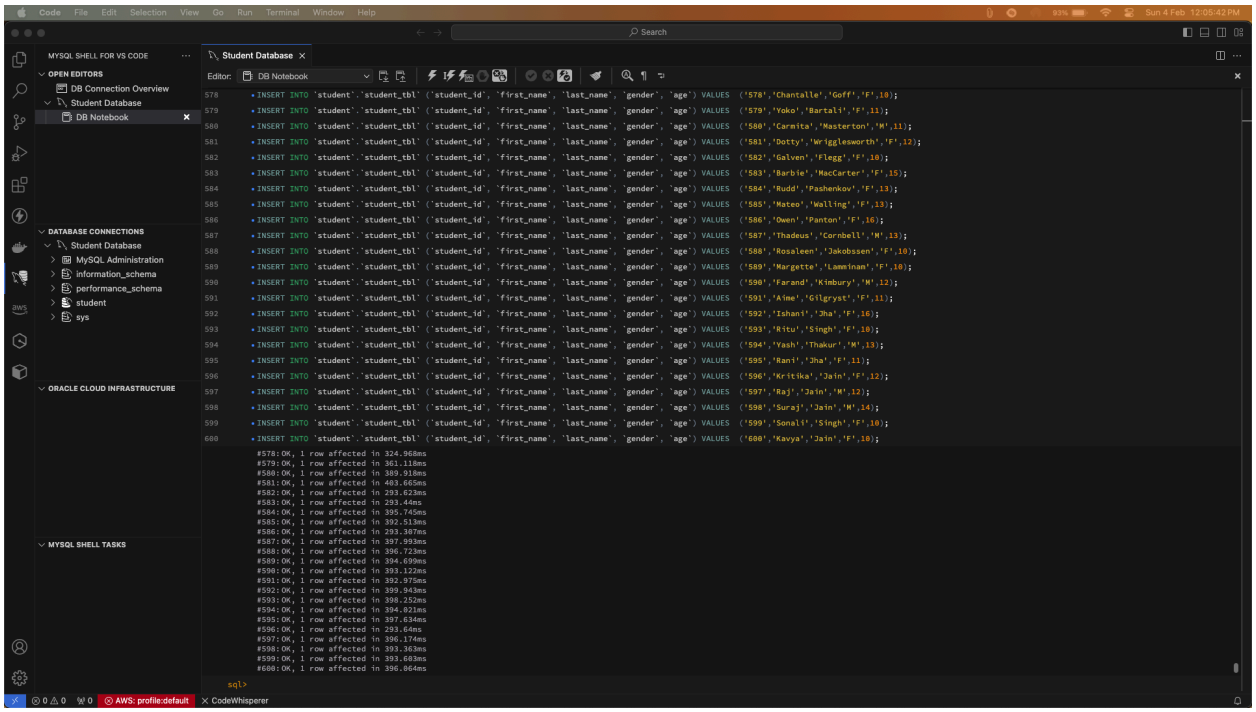
15. Now create a database named “**student**” and create the three tables as follows with the following parameters as mentioned in the screenshot:
    a. school_tbl
    b. student_tbl
    c. Study_details_tbl

    If you run the query correctly, you should see the command executed and the number of rows affected message as shown in the figure below.



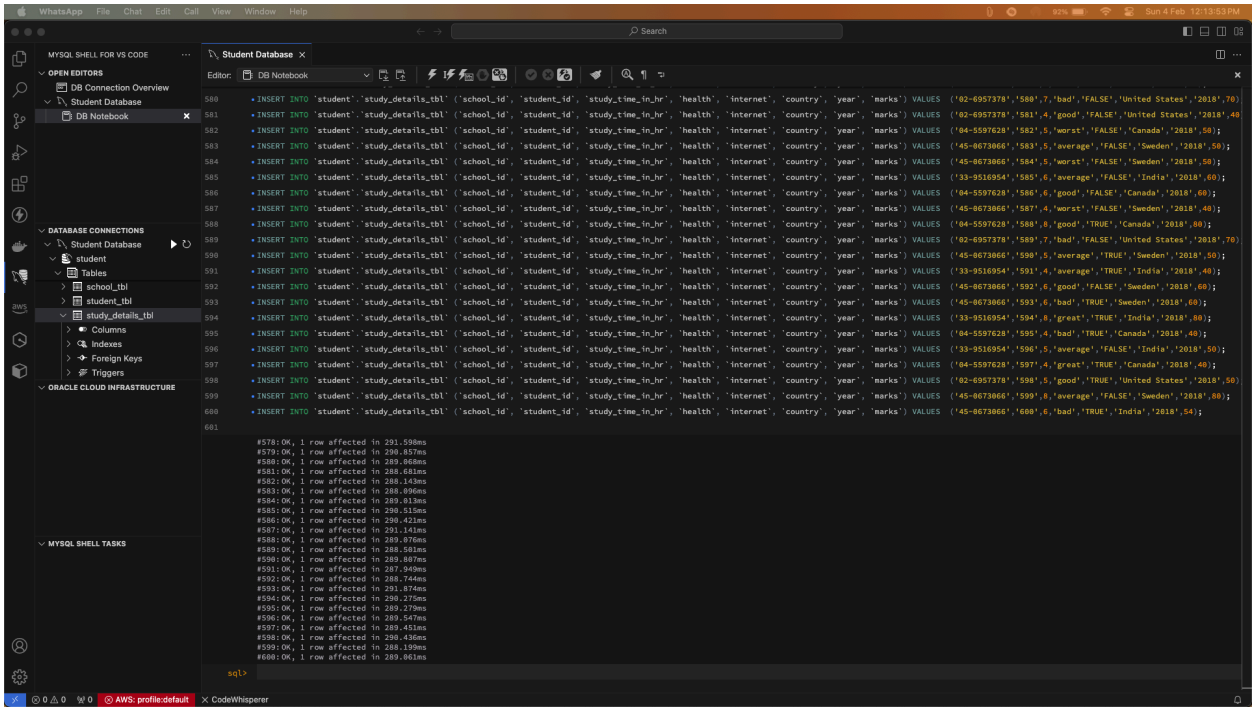


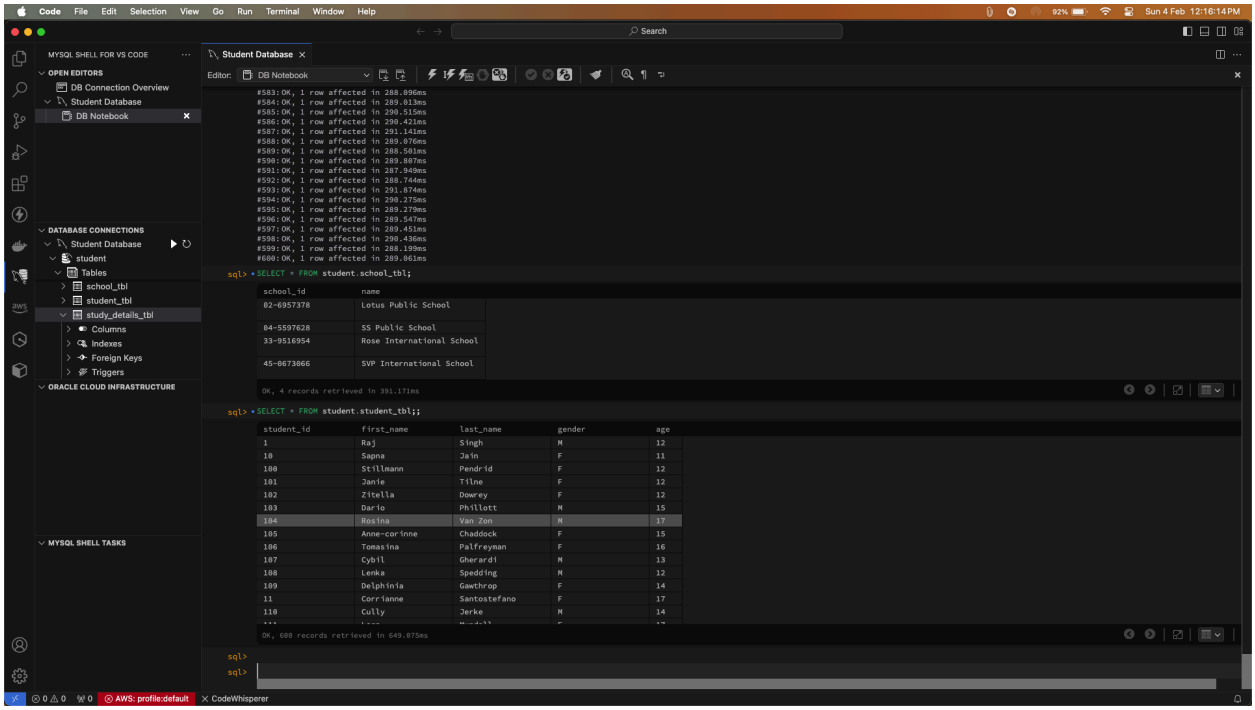
16. Now carefully insert the data into the three tables in order (as if you miss the order and try to perform the insertion in other tables first then it will give you a foreign key constraint error as all the tables are connected with a foreign key constraint).

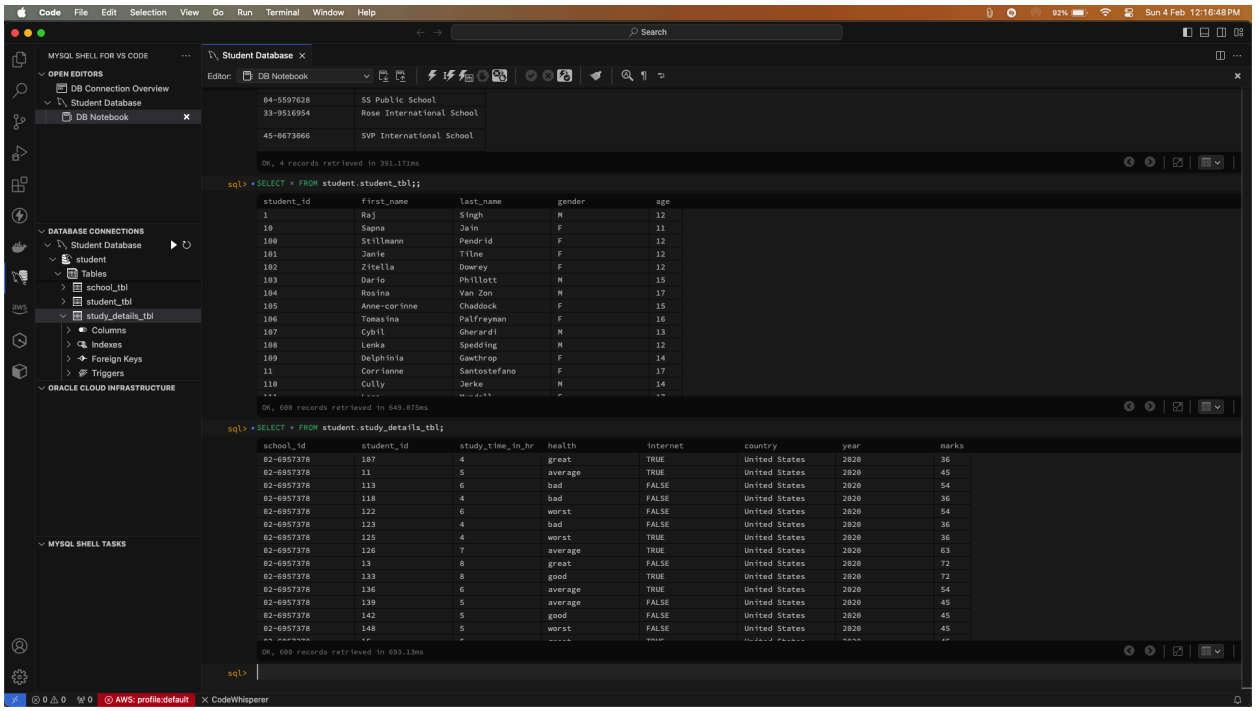


First, you need to insert the data in:

a. school_tbl
b. student_tbl
c. study_details_tbl


MYSQL SHELL FOR VS CODE
OPEN EDITORS
DB Connection Overview
Student Database
DB Notebook
DATABASE CONNECTIONS
Student Database
student
Tables
school_tbl
student_tbl
study_details_tbl
Columns
Indexes
Foreign Keys
Triggers
ORACLE CLOUD INFRASTRUCTURE
MYSQL SHELL TASKS
Student Database
Editor: DB Notebook
#578: OK, 1 row affected in 291.598ms
#579: OK, 1 row affected in 290.857ms
#580: OK, 1 row affected in 289.868ms
#581: OK, 1 row affected in 288.681ms
#582: OK, 1 row affected in 288.143ms
#583: OK, 1 row affected in 288.896ms
#584: OK, 1 row affected in 289.013ms
#585: OK, 1 row affected in 290.515ms
#586: OK, 1 row affected in 290.421ms
#587: OK, 1 row affected in 291.141ms
#588: OK, 1 row affected in 289.076ms
#589: OK, 1 row affected in 288.581ms
#590: OK, 1 row affected in 289.807ms
#591: OK, 1 row affected in 287.949ms
#592: OK, 1 row affected in 288.744ms
#593: OK, 1 row affected in 291.874ms
#594: OK, 1 row affected in 290.275ms
#595: OK, 1 row affected in 289.279ms
#596: OK, 1 row affected in 289.547ms
#597: OK, 1 row affected in 289.451ms
#598: OK, 1 row affected in 290.436ms
#599: OK, 1 row affected in 288.199ms
#600: OK, 1 row affected in 289.061ms
sql>
AWS: profile:default
CodeWhisperer


MYSQL SHELL FOR VS CODE
OPEN EDITORS
DB Connection Overview
Student Database
DB Notebook
DATABASE CONNECTIONS
Student Database
student
Tables
school_tbl
student_tbl
study_details_tbl
Columns
Indexes
Foreign Keys
Triggers
ORACLE CLOUD INFRASTRUCTURE
MYSQL SHELL TASKS
Student Database
Editor: DB Notebook
sql> SELECT * FROM student.school_tbl;
school_id name
02-6957378 Lotus Public School
04-5597628 SS Public School
33-9516954 Rose International School
45-0673066 SVP International School
OK, 4 records retrieved in 391.171ms
sql> SELECT * FROM student.student_tbl;;
student_id first_name last_name gender age
1 Raj Singh M 12
10 Sapna Jain F 11
100 Stillmann Pendrid F 12
101 Janie Tilne F 12
102 Zitella Dowrey F 12
103 Dario Phillott M 15
104 Rosina Van Zon M 17
105 Anne-corinne Chaddock F 15
106 Tomasina Palfreyman F 16
107 Cybil Gherardi M 13
108 Lenka Spedding M 12
109 Delphinia Gawthrop F 14
11 Corrianne Santostefano F 17
110 Cully Jerke M 14
OK, 609 records retrieved in 649.075ms
sql>
sql>
AWS: profile:default
CodeWhisperer

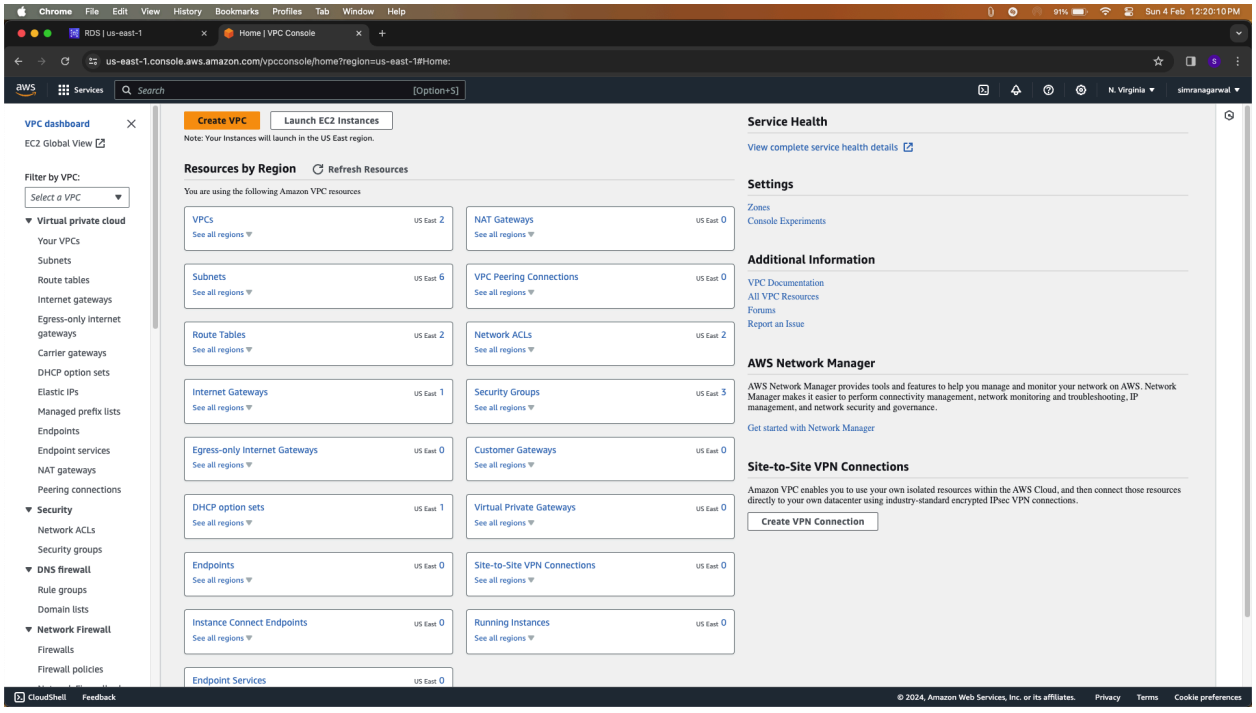


17. Now after inserting the data in all three tables check all the data in all three tables using the following queries.

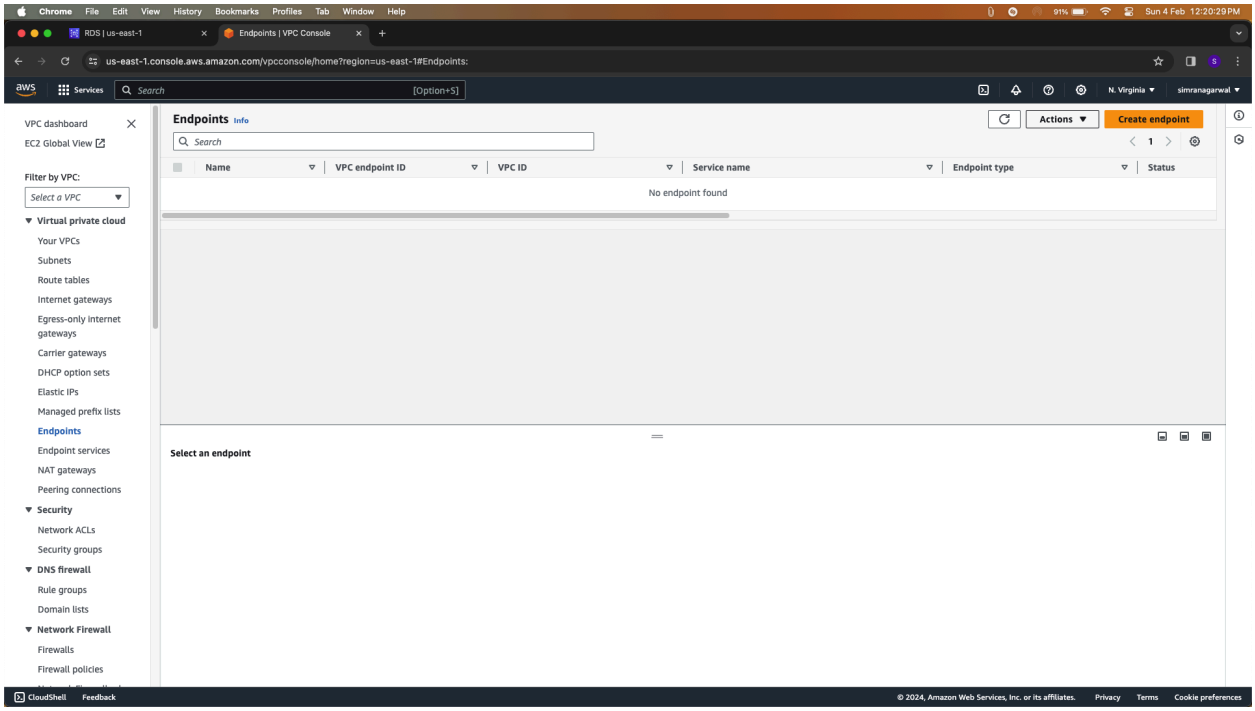
a. select * from student.school_tbl;
b. select * from student.student_tbl;
c. select * from student.study_details_tbl;

If you find the data in all three tables, then congratulations you are good to go.

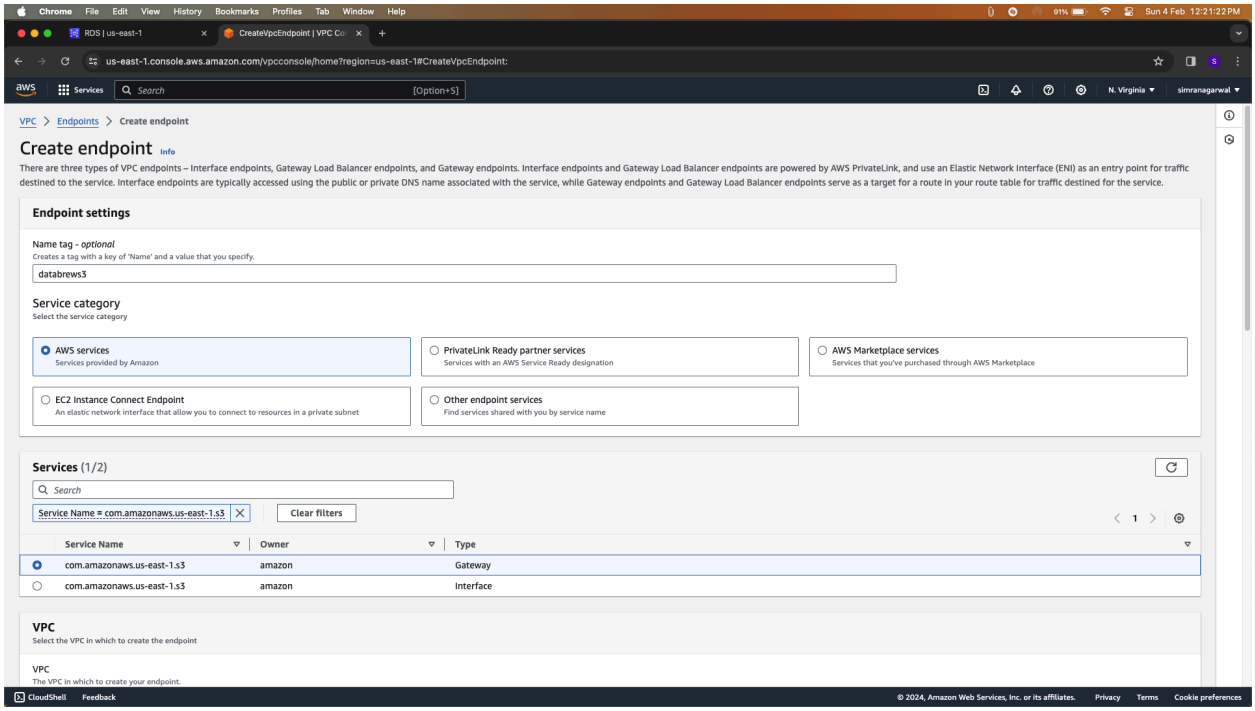
18. Goto, VPC Console by typing VPC in the search bar of Amazon AWS and click on VPC from the search results.



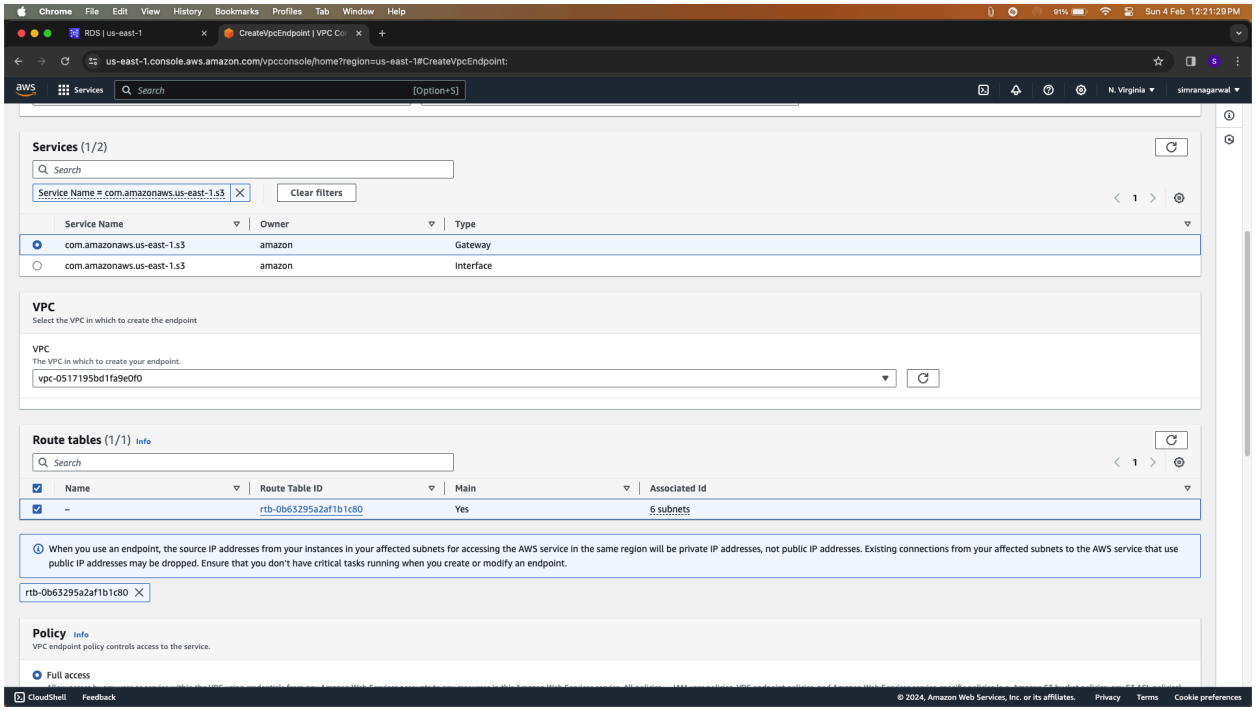
19. Now open the Endpoints tab in the sub-menu of VPC, and click on “Create Endpoint”.

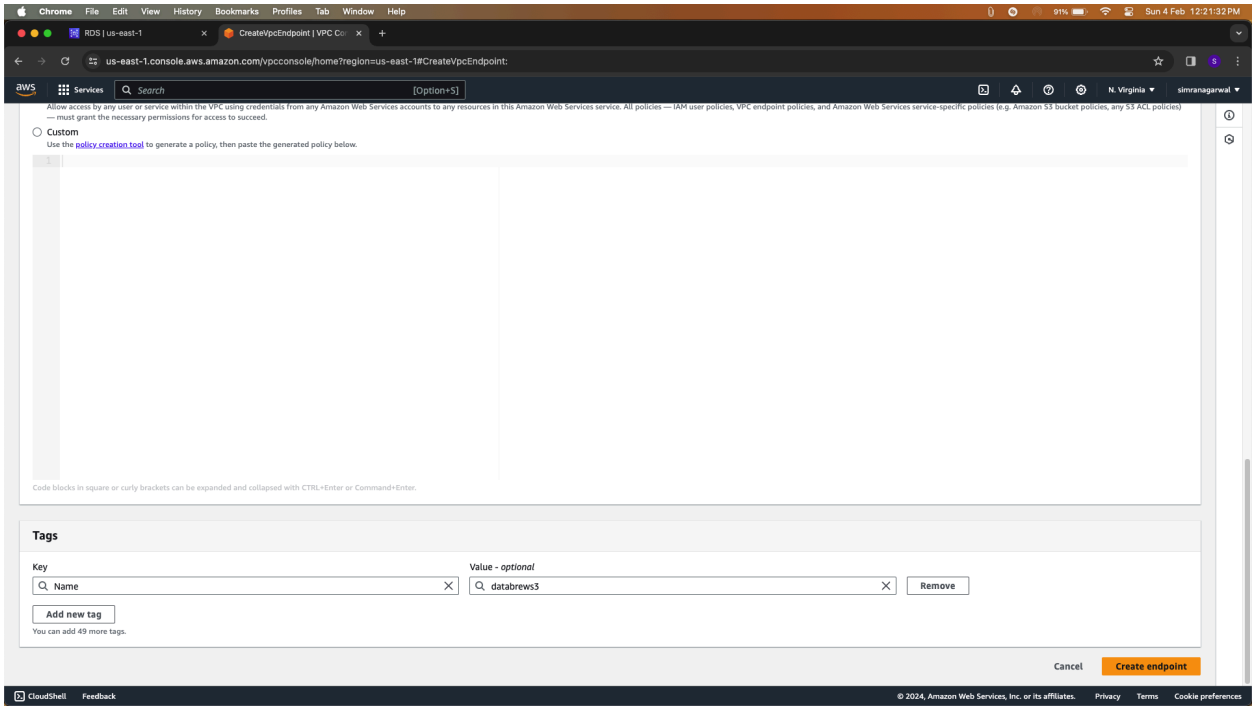


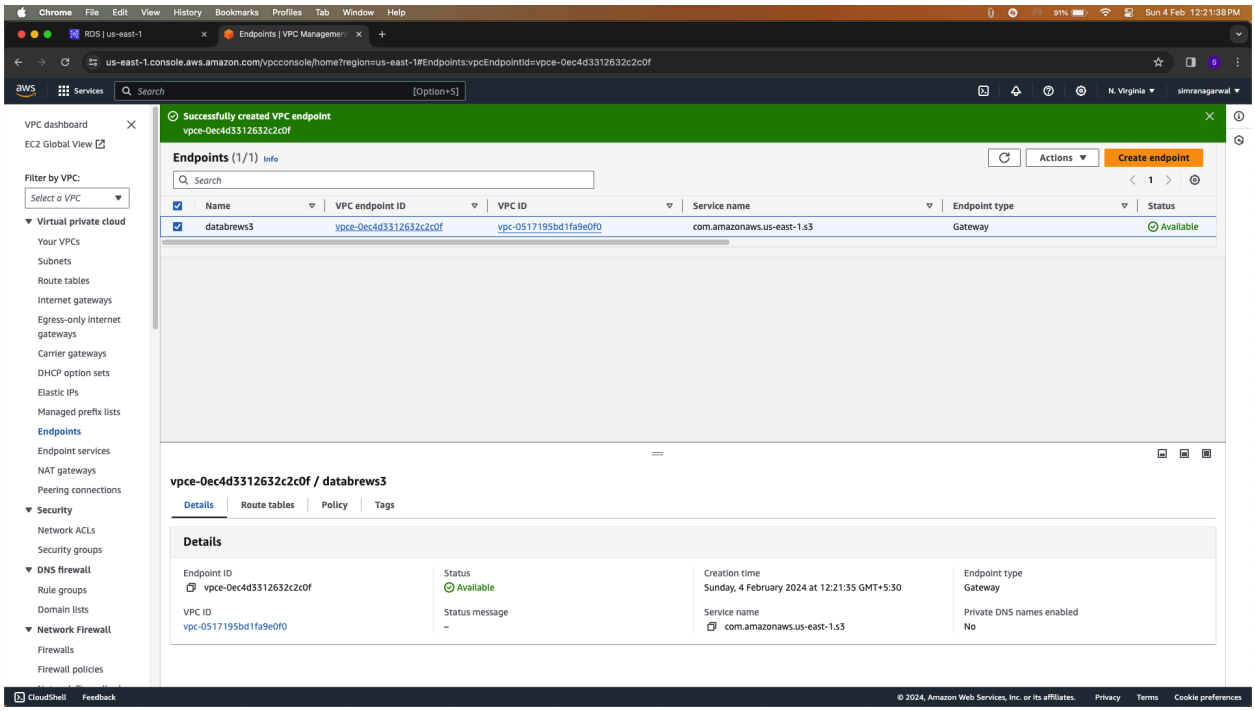
20. Under the service category Select “AWS Services” and fill out the form accordingly.



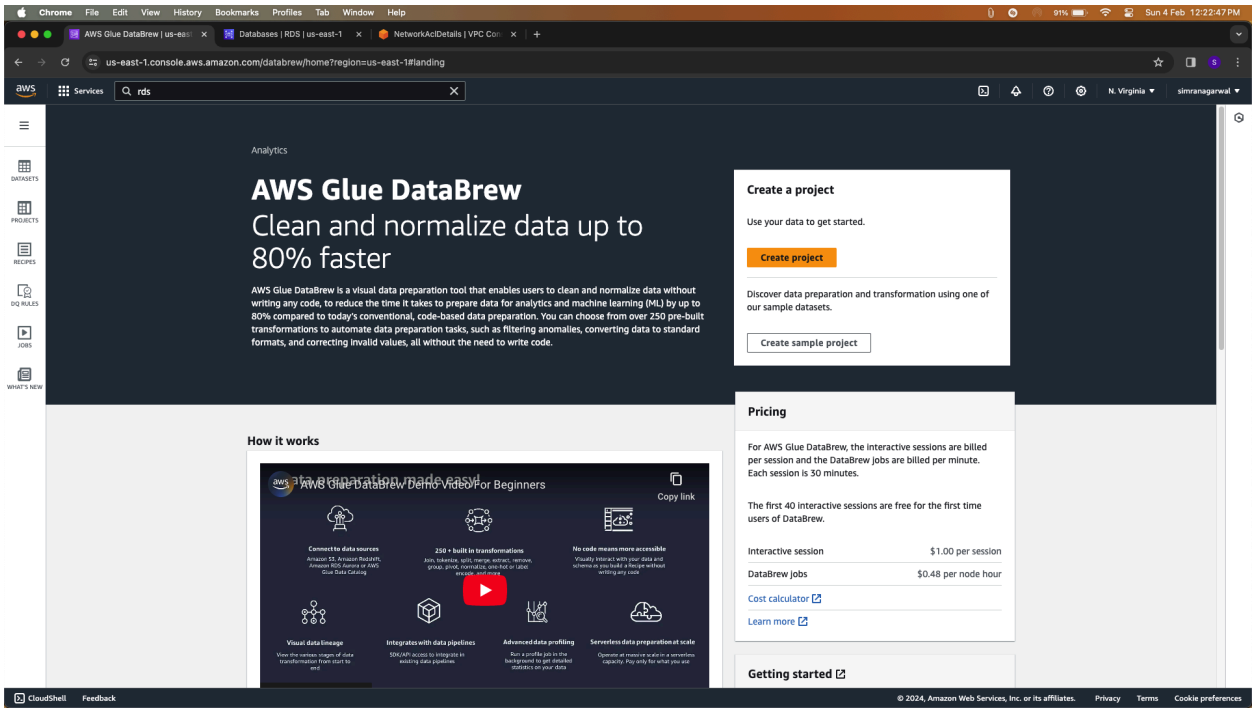
21. Now search for “s3” and select the one with Type - Gateway.


Services (1/2)
Search
Service Name = com.amazonaws.us-east-1.s3
Clear filters
Service Name
Owner
Type
com.amazonaws.us-east-1.s3
amazon
Gateway
com.amazonaws.us-east-1.s3
amazon
Interface
VPC
Select the VPC in which to create the endpoint
VPC
The VPC in which to create your endpoint.
vpc-0517195bd1fa9e0f0
Route tables (1/1) Info
Search
Name
Route Table ID
Main
Associated Id
–
rtb-0b63295a2af1b1c80
Yes
6 subnets
When you use an endpoint, the source IP addresses from your instances in your affected subnets for accessing the AWS service in the same region will be private IP addresses, not public IP addresses. Existing connections from your affected subnets to the AWS service that use public IP addresses may be dropped. Ensure that you don't have critical tasks running when you create or modify an endpoint.
rtb-0b63295a2af1b1c80
Policy Info
VPC endpoint policy controls access to the service.
Full access

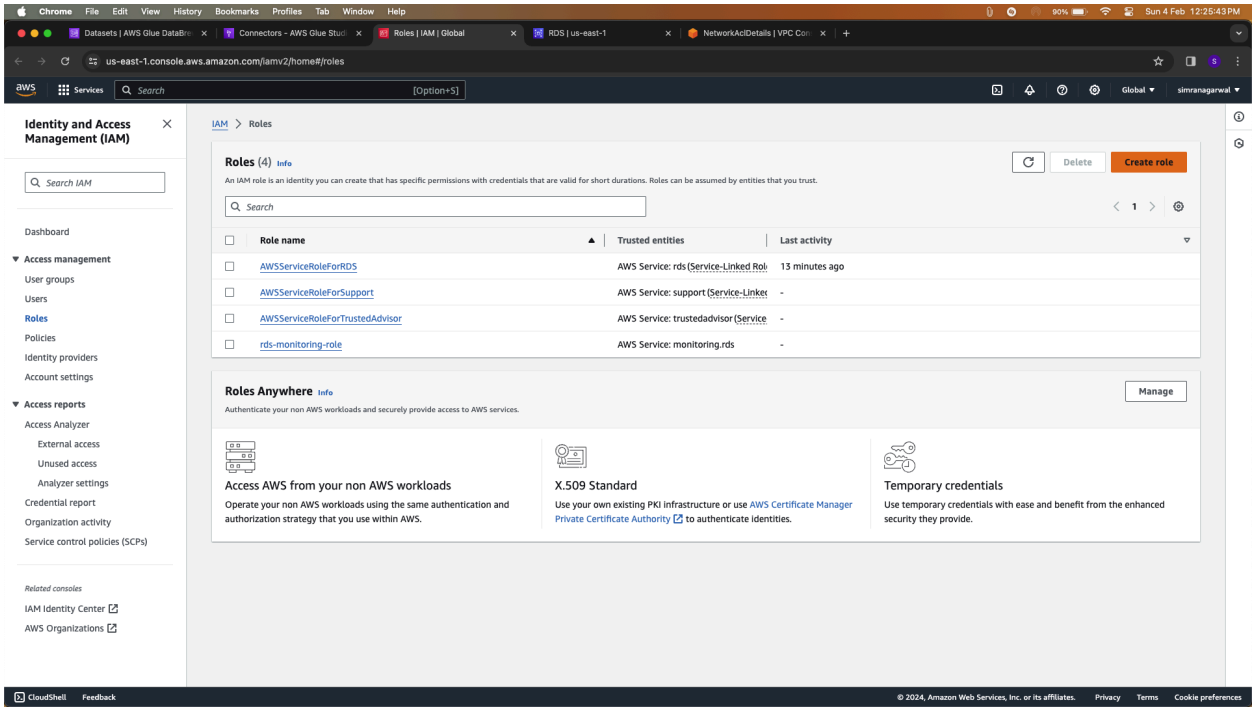

Custom
Use the policy creation tool to generate a policy, then paste the generated policy below.
Code blocks in square or curly brackets can be expanded and collapsed with CTRL+Enter or Command+Enter.
Tags
Key
Value - optional
Name
databrews3
Remove
Add new tag
You can add 49 more tags.
Cancel
Create endpoint


Successfully created VPC endpoint
vpce-0ec4d3312632c2c0f
Endpoints (1/1) Info
Actions
Create endpoint
Name
VPC endpoint ID
VPC ID
Service name
Endpoint type
Status
databrews3
vpce-0ec4d3312632c2c0f
vpc-0517195bd1fa9e0f0
com.amazonaws.us-east-1.s3
Gateway
Available
vpce-0ec4d3312632c2c0f / databrews3
Details
Route tables
Policy
Tags
Endpoint ID
vpce-0ec4d3312632c2c0f
Status
Available
Creation time
Sunday, 4 February 2024 at 12:21:35 GMT+5:30
Endpoint type
Gateway
VPC ID
vpc-0517195bd1fa9e0f0
Status message
–
Service name
com.amazonaws.us-east-1.s3
Private DNS names enabled
No

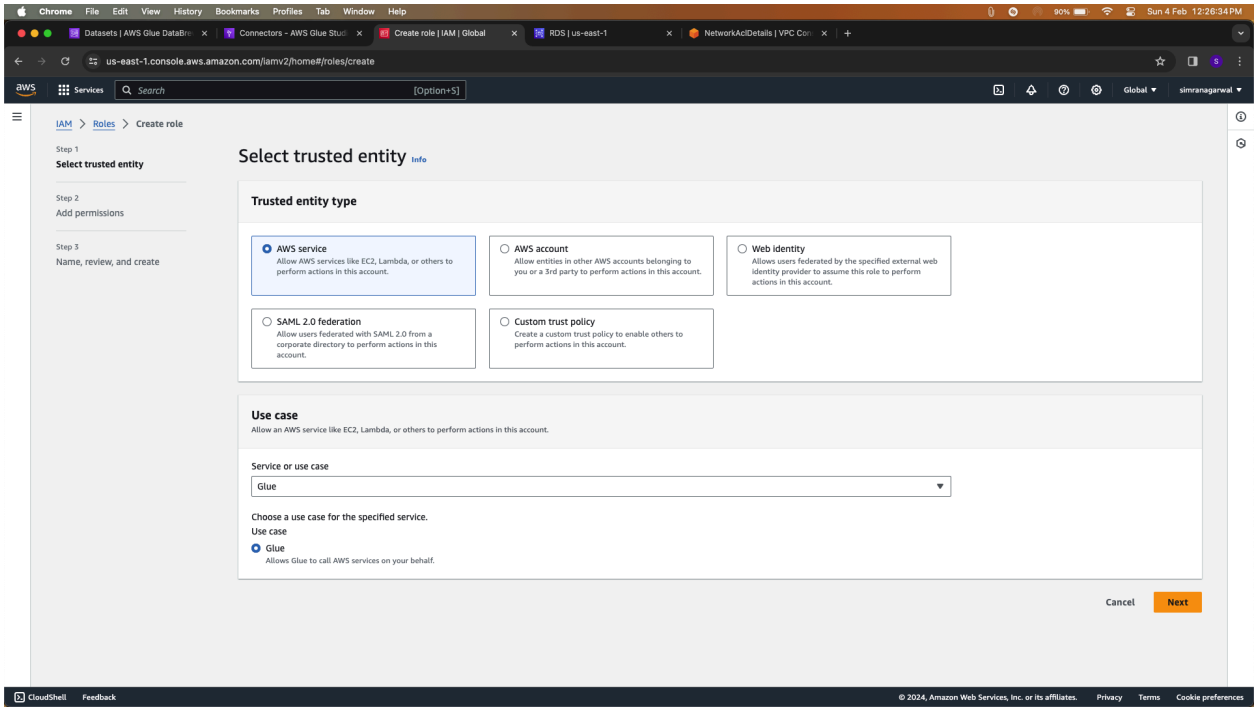
22. Now that your endpoint is created, You need to search for AWS Glue Databrew. (here we will try to create a data set and a project and then run a job which will put the output in the S3 Bucket).

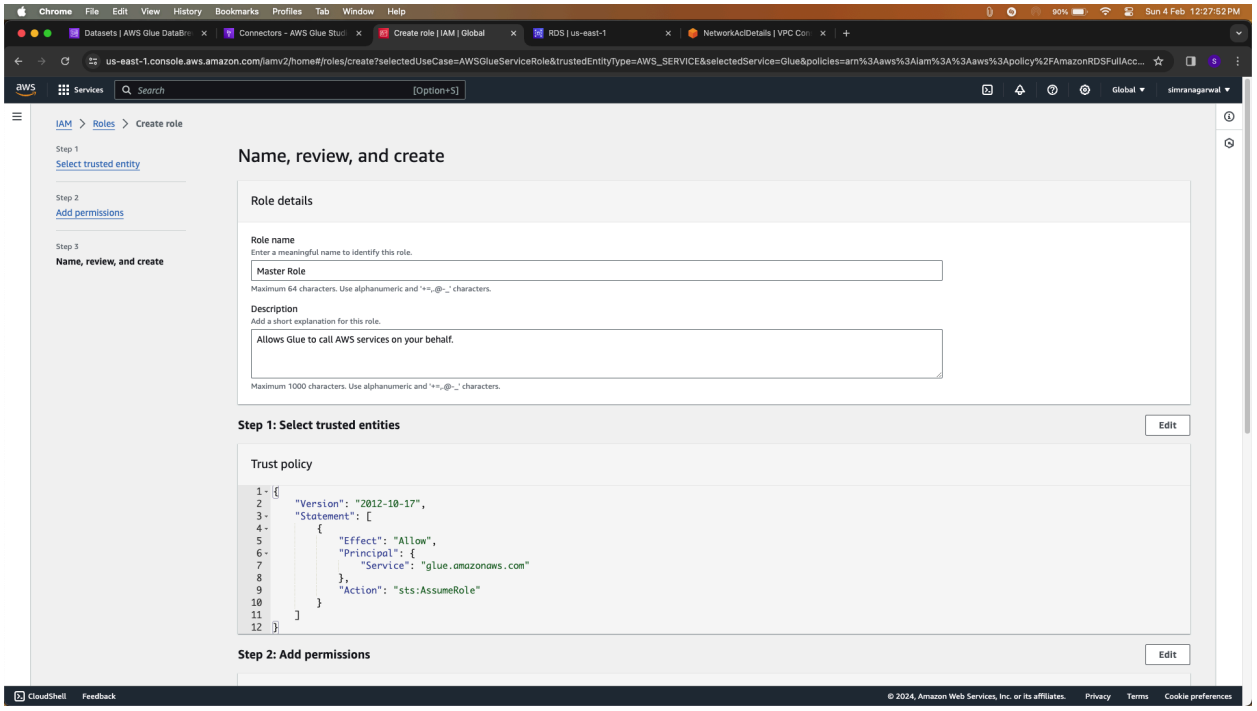


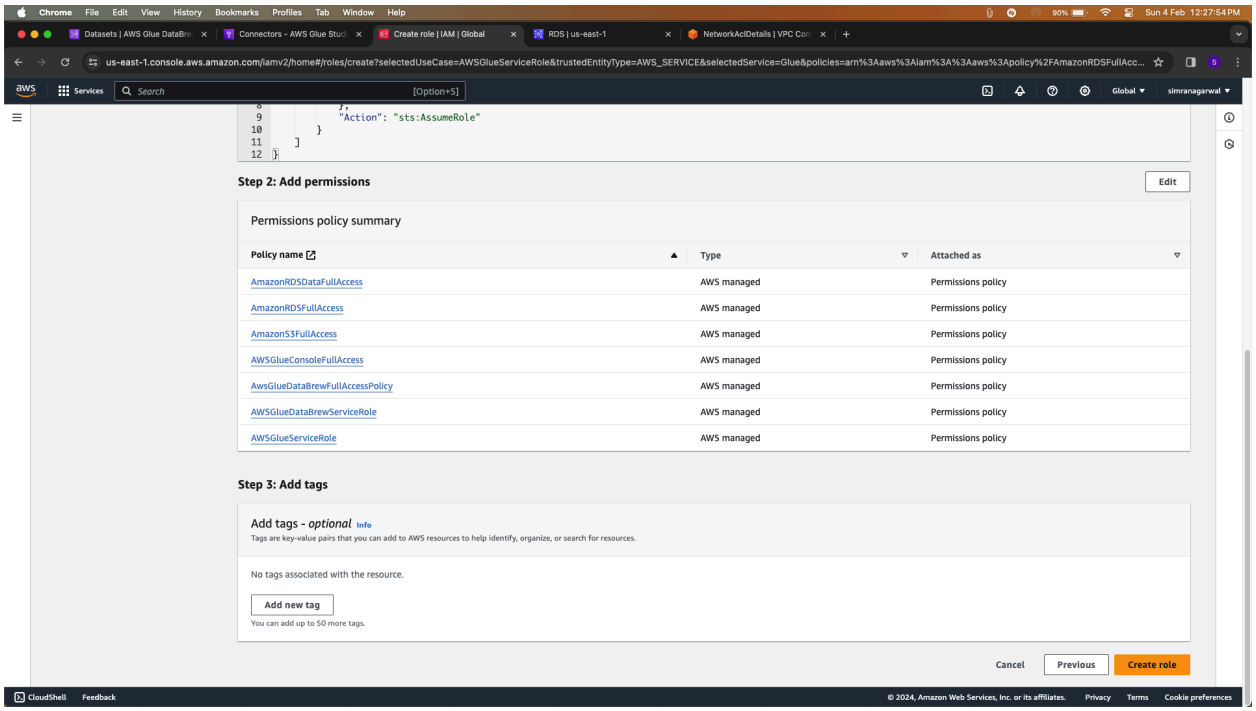
23. Before going to Databrew we need to take care and assign some roles for the Databrew. Goto IAM Console by searching in the search bar or through AWS Home Panel.



24. Click on “Create Role”.
Choose the trusted entity type as “AWS Service” and use case as **“Glue”** and then click Next.


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sun 4 Feb 12:26:34 PM
us-east-1.console.aws.amazon.com/iamv2/home#/roles/create
Services Search [Option+S]
Global simranagarwal
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.
Web identity
Allows users federated by the specified external web identity provider to assume this role 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.
Service or use case
Glue
Choose a use case for the specified service.
Use case
Glue
Allows Glue to call AWS services on your behalf.
Cancel
Next
CloudShell Feedback
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sun 4 Feb 12:27:52 PM
us-east-1.console.aws.amazon.com/iamv2/home#/roles/create?selectedUseCase=AWSGlueServiceRole&trustedEntityType=AWS_SERVICE&selectedService=Glue&policies=arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAmazonRDSFullAcc...
Services Search [Option+S]
Global simranagarwal
IAM > Roles > Create role
Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create
Name, review, and create
Role details
Role name
Enter a meaningful name to identify this role.
Master Role
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.
Description
Add a short explanation for this role.
Allows Glue to call AWS services on your behalf.
Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.
Step 1: Select trusted entities
Edit
Trust policy
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Principal": {
"Service": "glue.amazonaws.com"
},
"Action": "sts:AssumeRole"
}
]
}
Step 2: Add permissions
Edit
CloudShell Feedback
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



Review the policies assigned by you to that role.

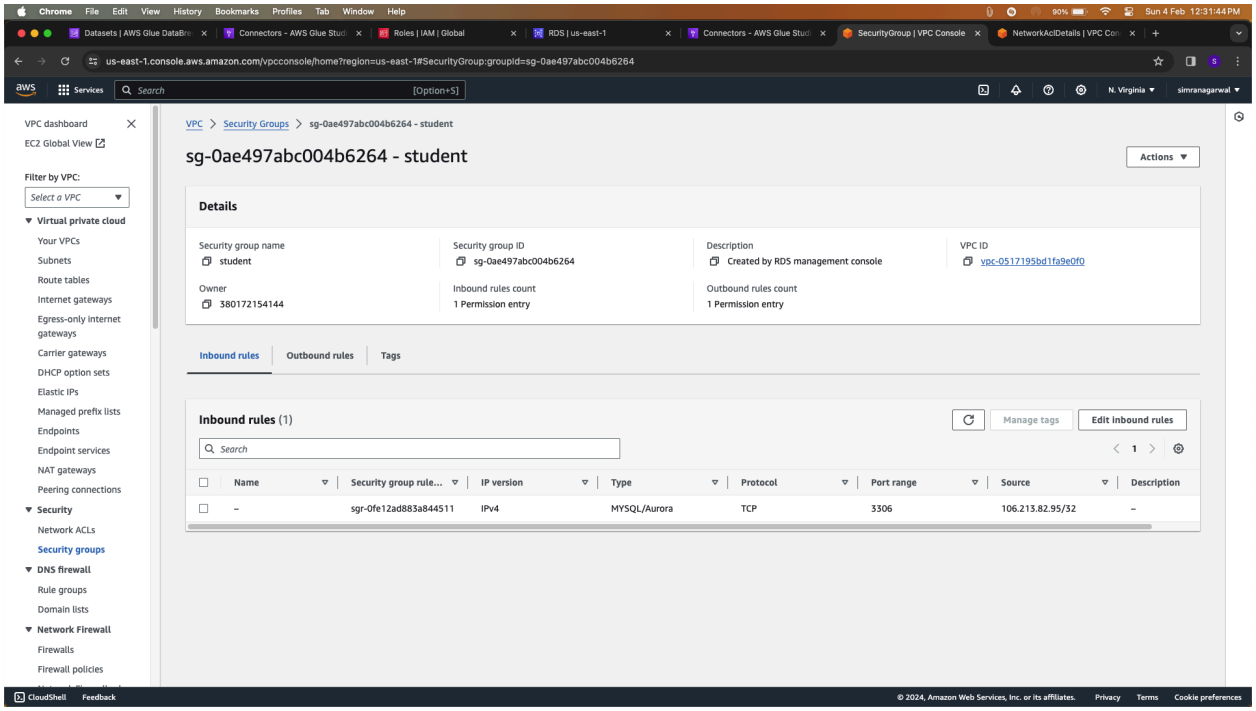
Essential ones are.

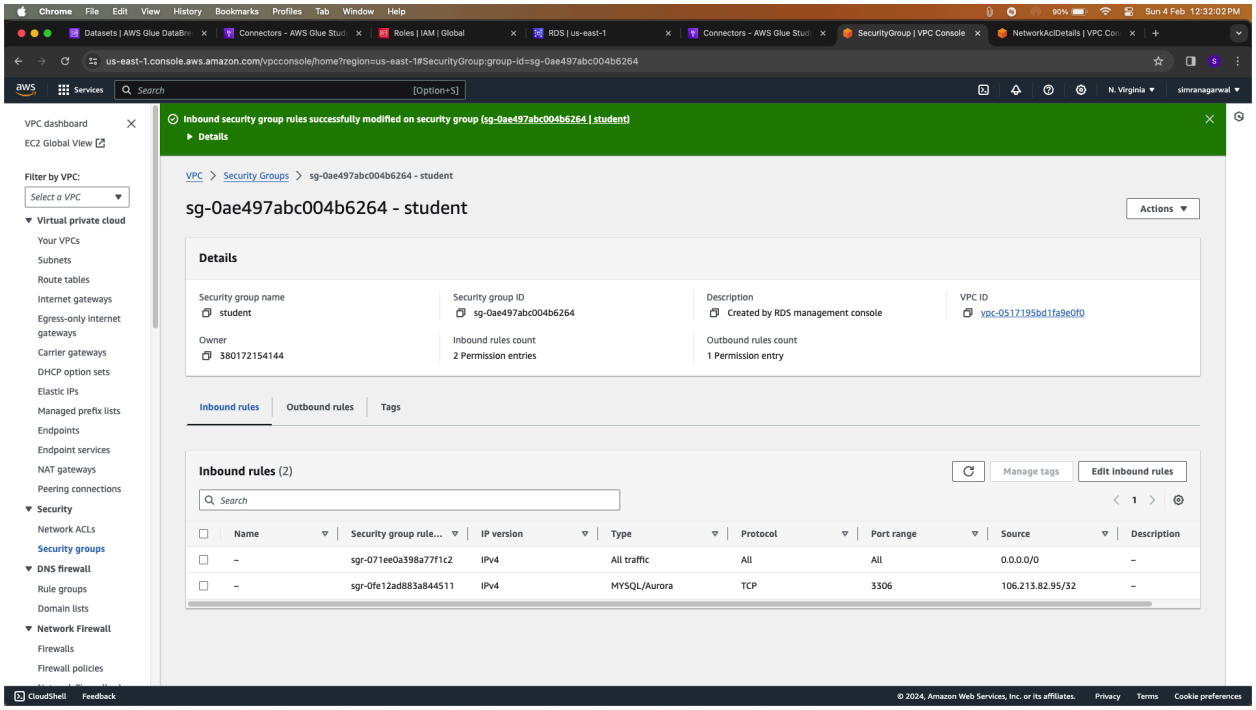
a. S3 Full Access Policy - will be used to output the dataset to an S3 Bucket
b. RDS Full Access Policy - will be used to read data from the RDS MySQL Database.
c. Glue Databrew Access Policy - will be used to allow connecting to the Databrew.
d. Glue Service Role Policy - For testing and editing the connections.

Also, you can assign roles like:

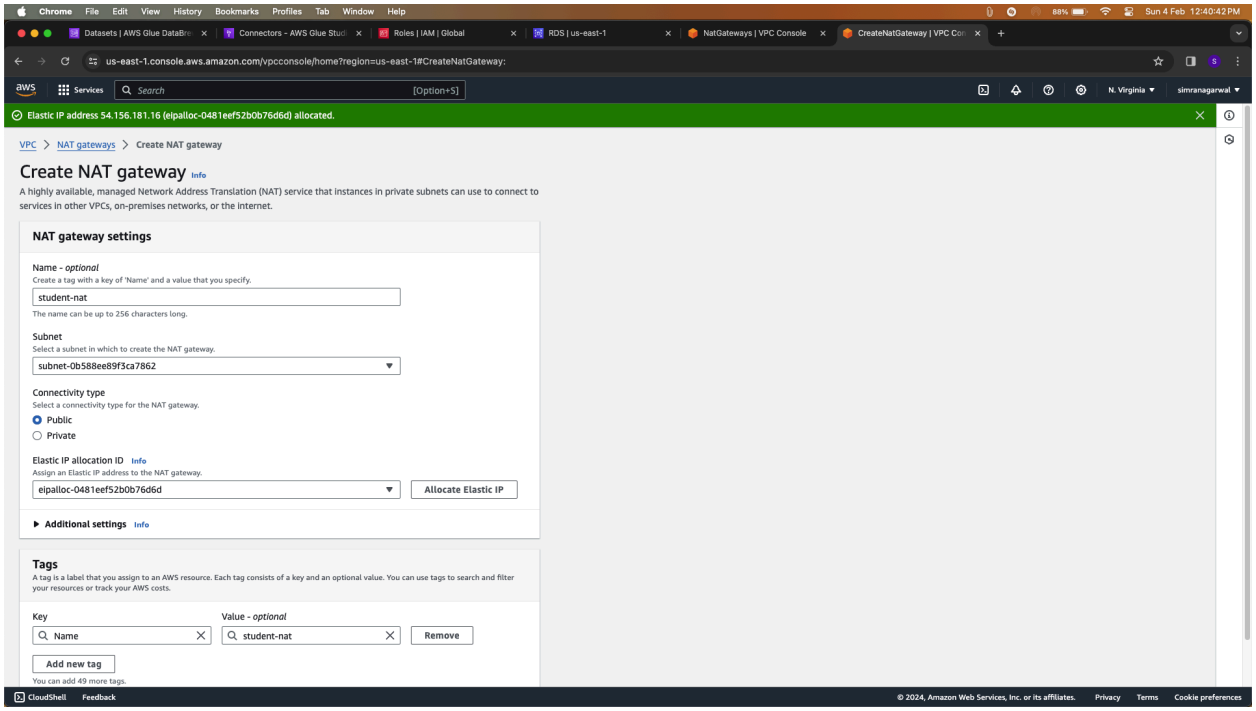
e. Athena Full Access Policy - for accessing Athena and Athena to access other resources as well.
f. Quick Sight Access Policy - for accessing the QuickSight.

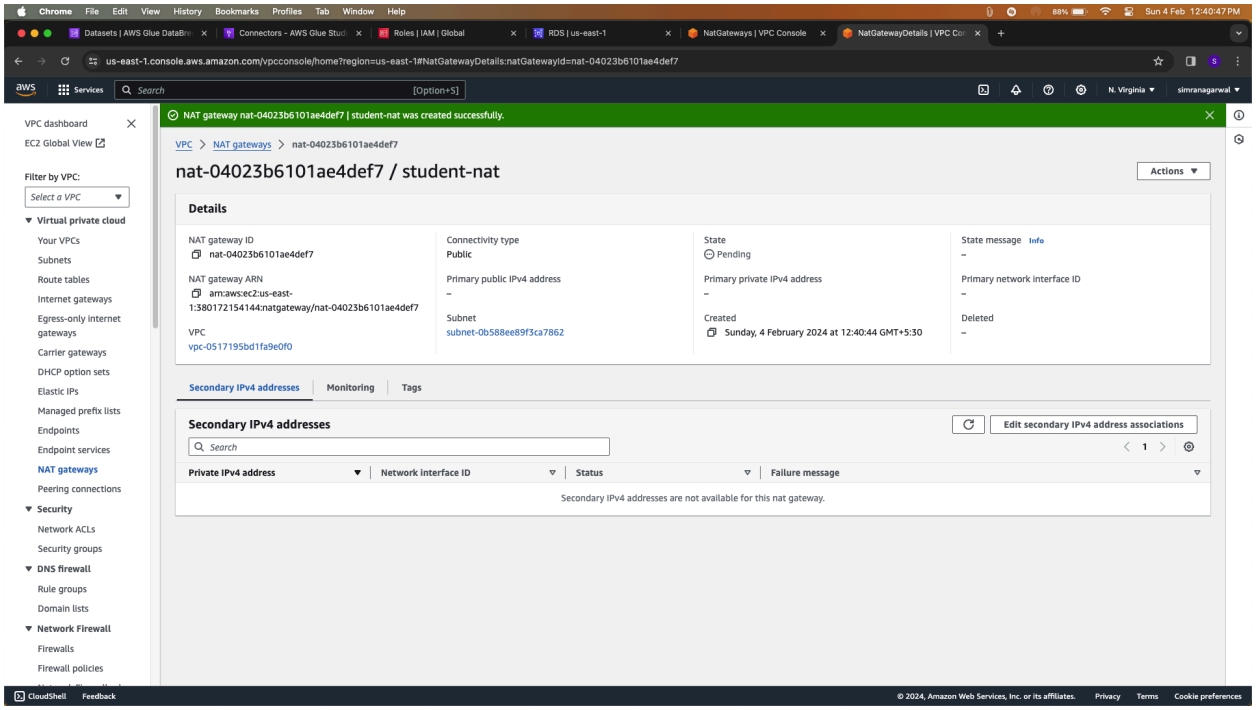
25. Now go to the Security Group sections under the VPC Console and click on edit the inbound rules (outbound rules remain the default, there is no change).





26. Now, go to the NAT Gateway section and create a NAT Gateway which will later be linked to our VPC.

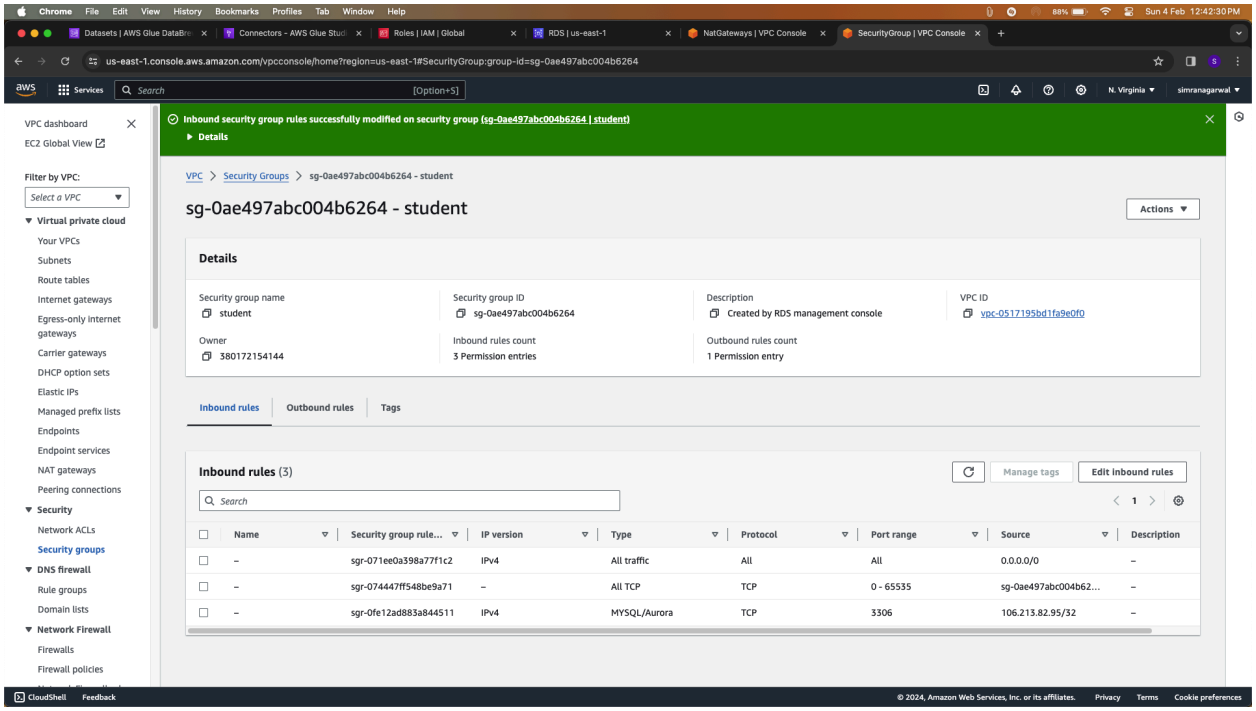




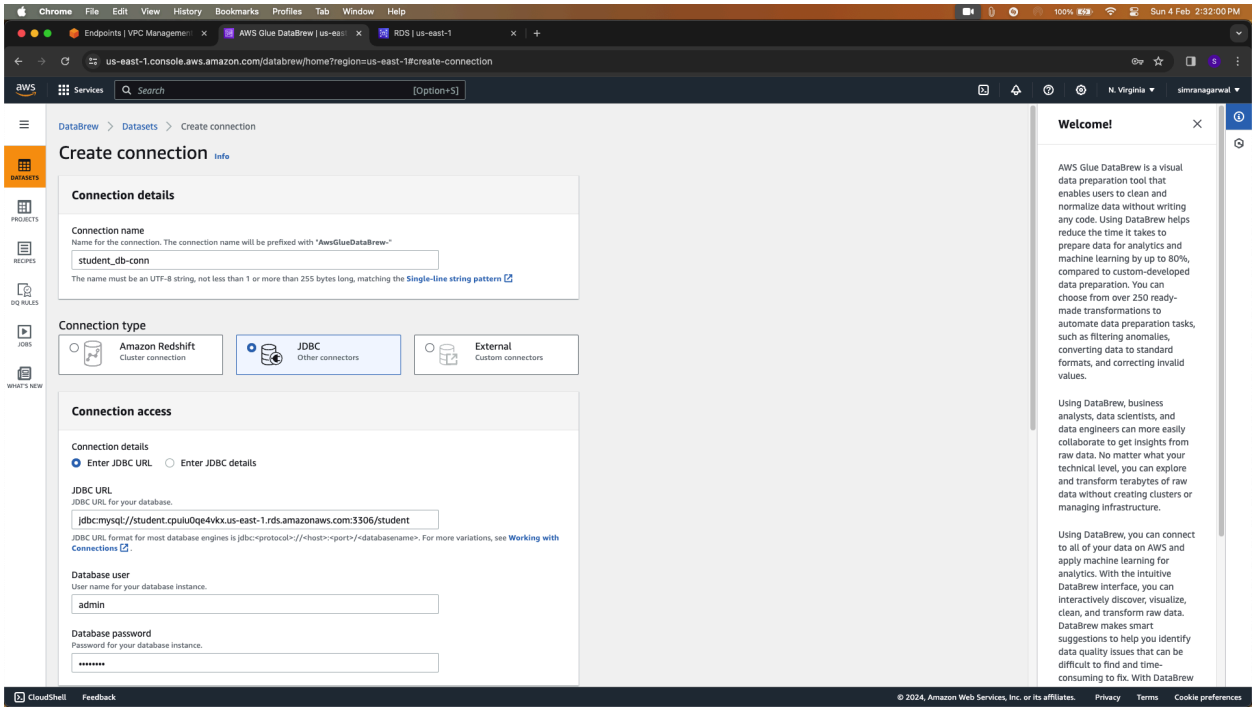
27. In the security group, you need to add all the details that are mentioned in the image below:

Inbound Rules (Must have)

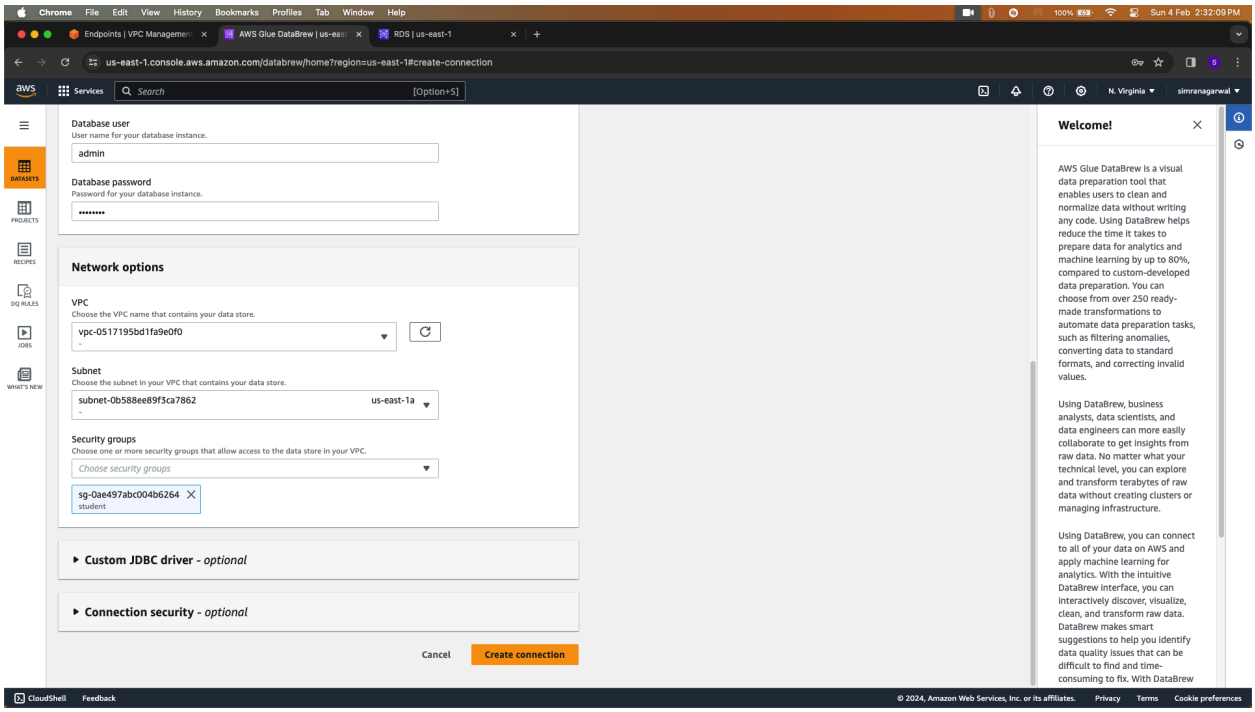
a. MySQL
b. All TCP

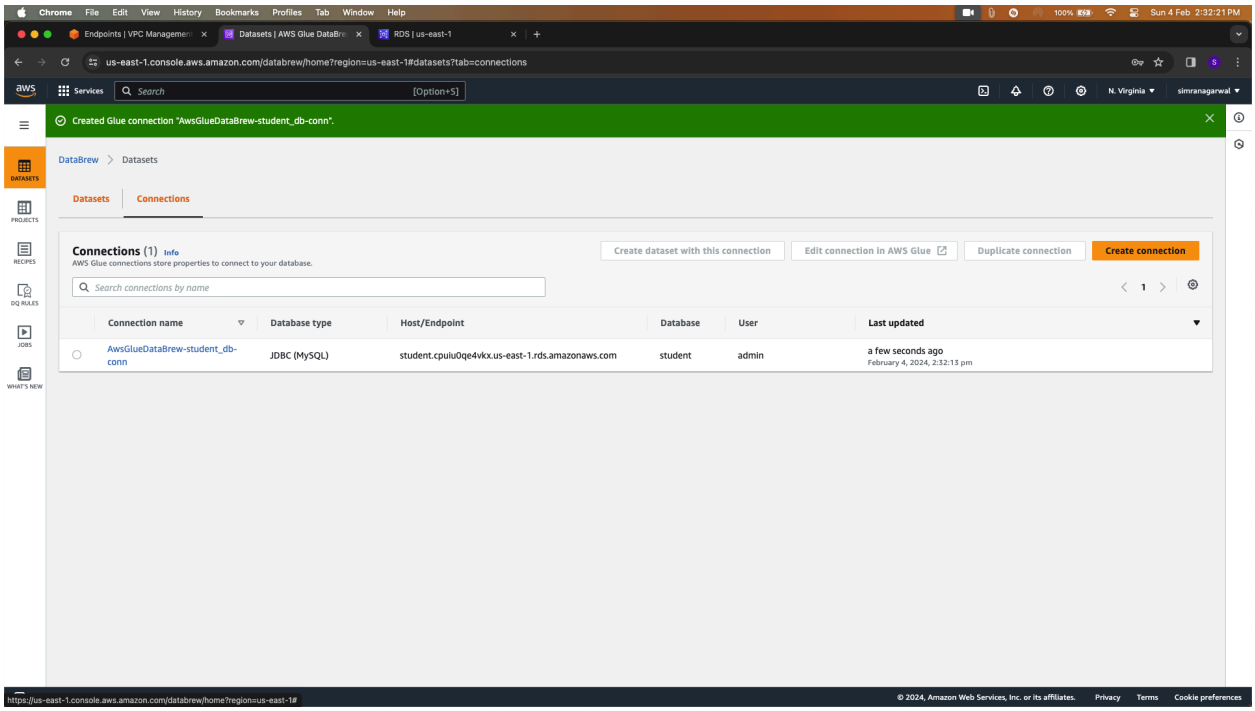


28. Go to Glue Databrew again, and create a connection, for this, you need to first click on Dataset then click on Connection and then Create Connection.

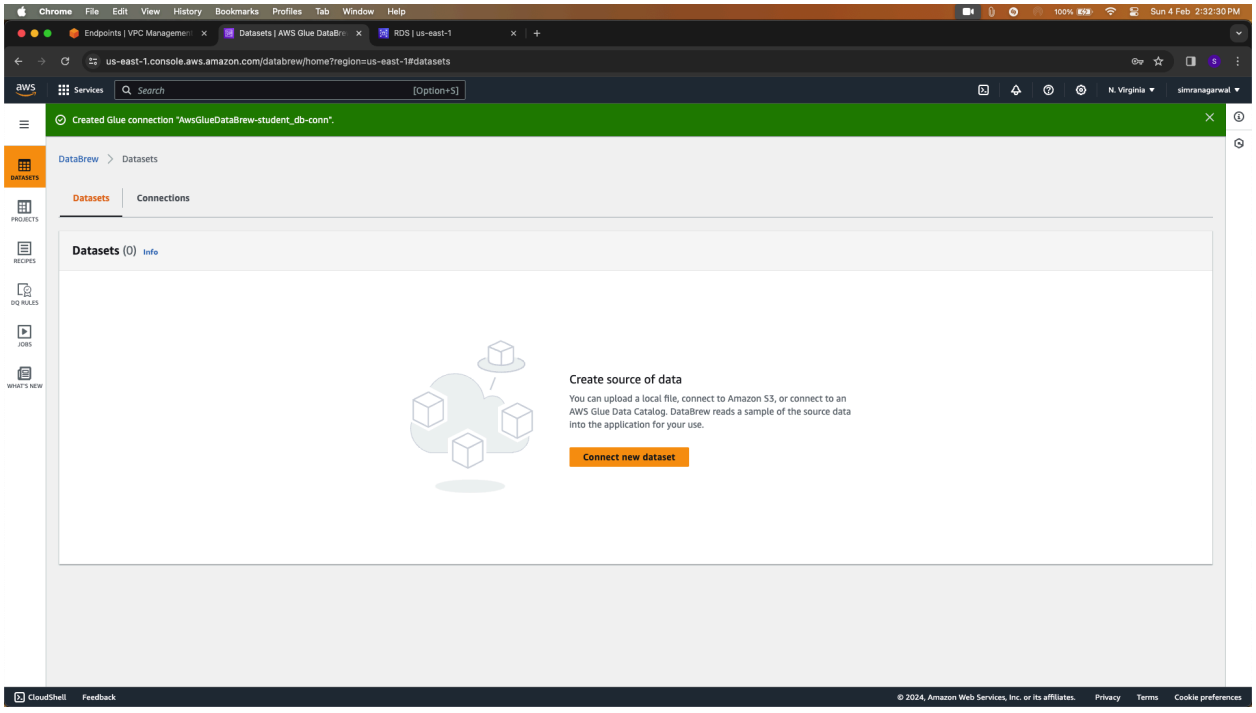


29. Proceed with JDBC Connection and fill out all the details asked in the form fields there. (Please be cautious or a bit careful while you enter the JDBC URL, username and password as these are the three things which will help you connect to RDS Database from Glue DataBrew)



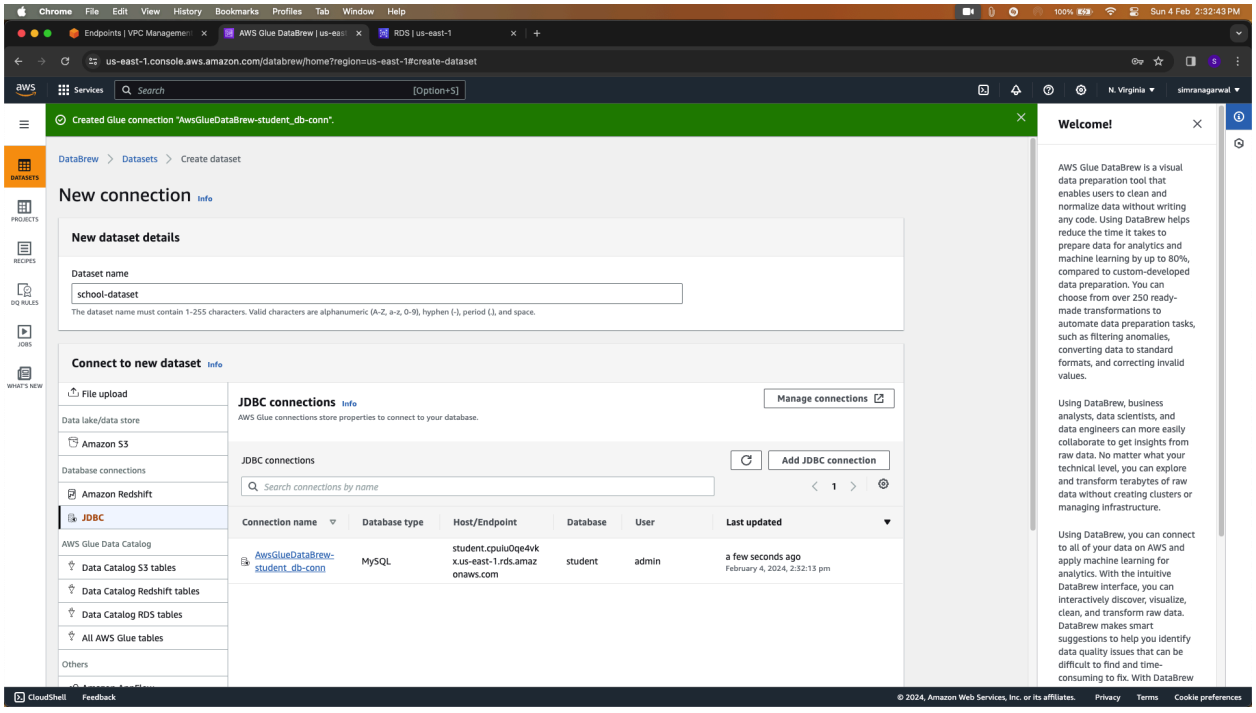


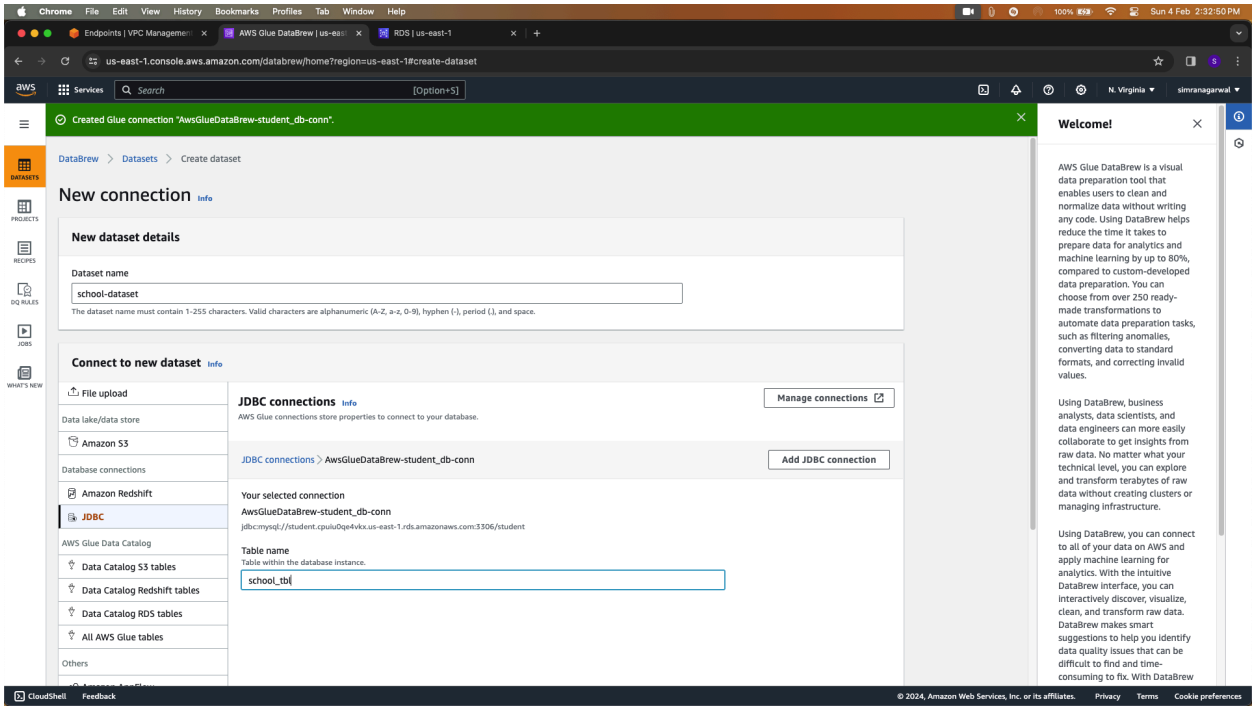
30. As the connection is successfully created, it's time to create Datasets that connects or corresponds to the tables in RDS MySQL Database.

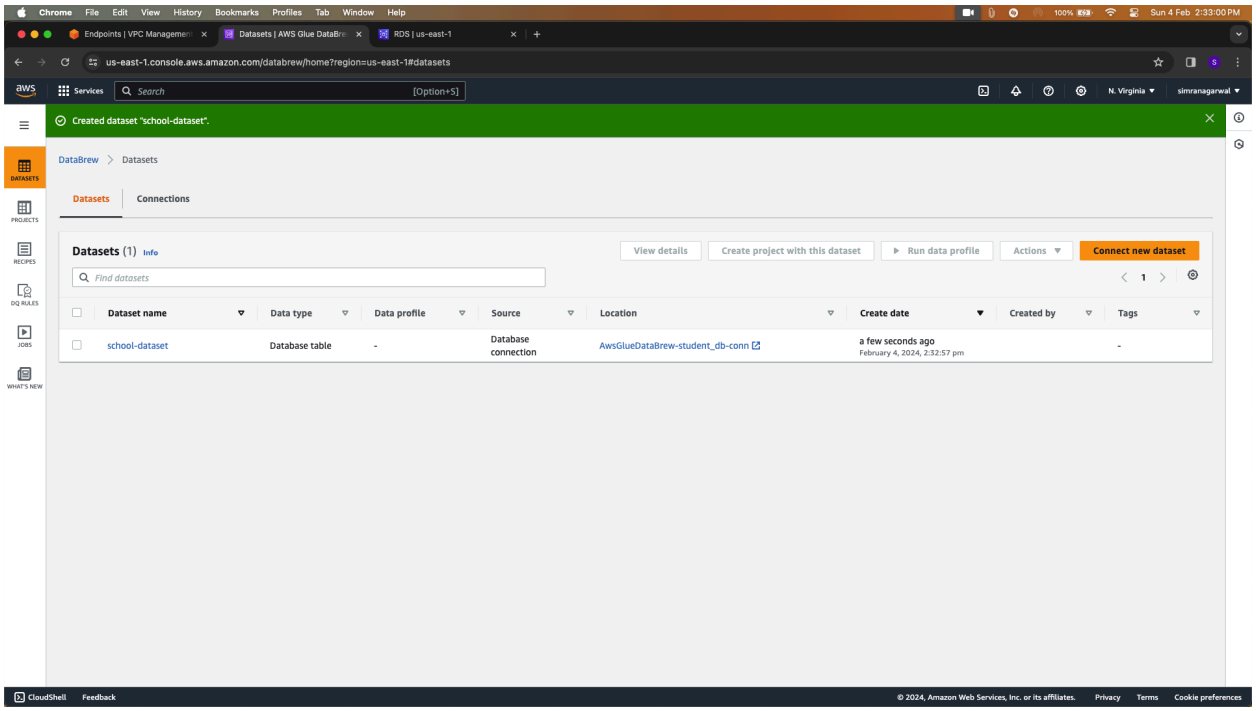


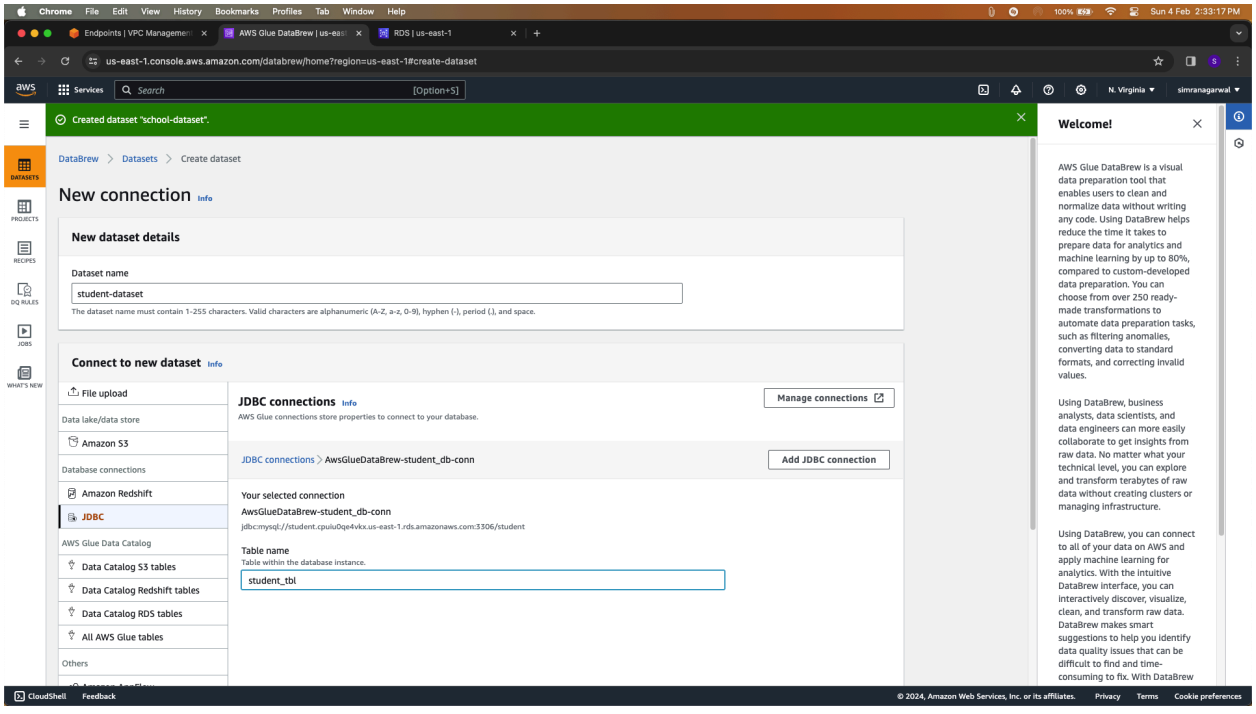
31. Now, you can create all three Datasets (corresponding to the three tables of the RDS MySQL “student” Database created by us.

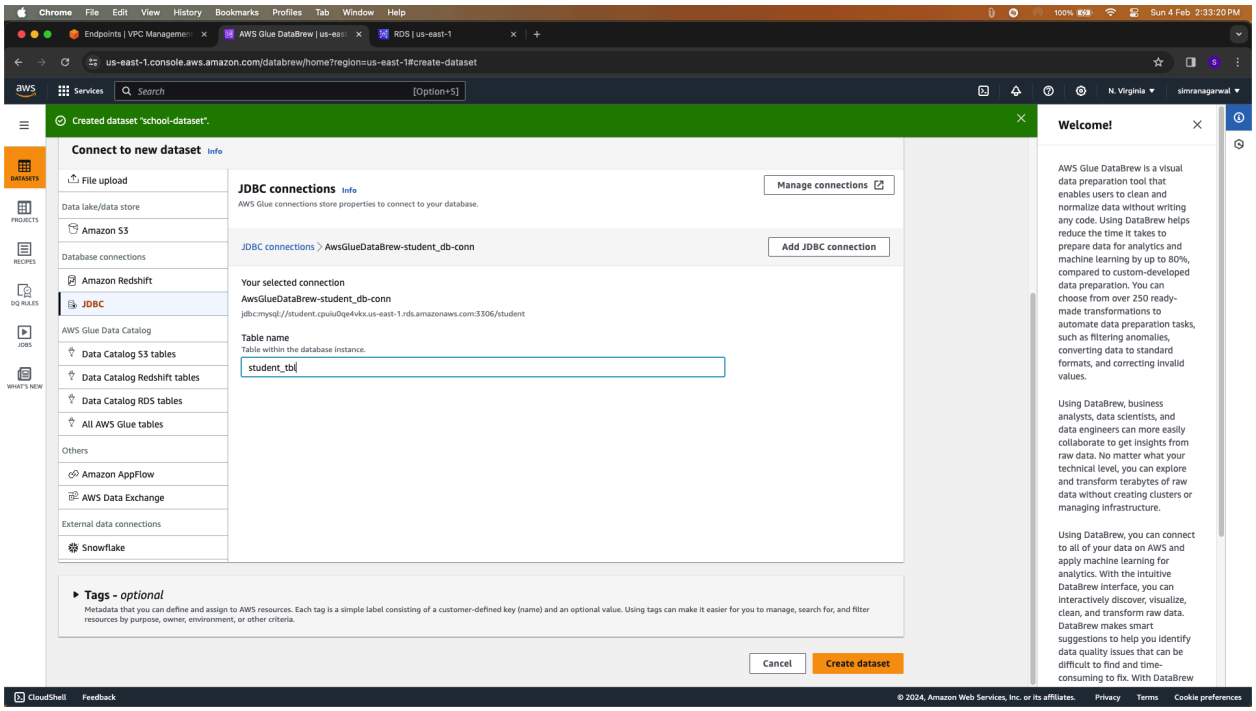
a. school-dataset for school_tbl.
b. student-dataset for student_tbl.
c. study-details-dataset for study_details_tbl.

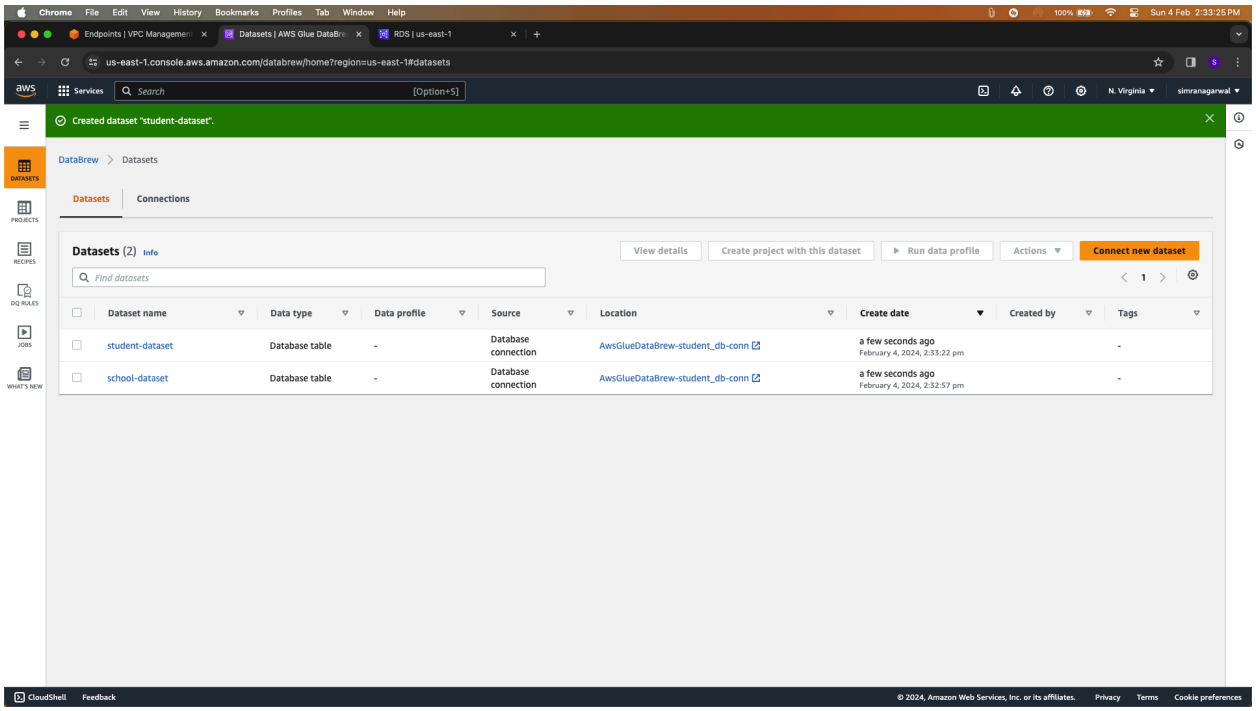

Created Glue connection "AwsGlueDataBrew-student_db-conn".
DataBrew > Datasets > Create dataset
New connection Info
New dataset details
Dataset name
school-dataset
The dataset name must contain 1-255 characters. Valid characters are alphanumeric (A-Z, a-z, 0-9), hyphen (-), period (.), and space.
Connect to new dataset Info
File upload
Data lake/data store
Amazon S3
Database connections
Amazon Redshift
JDBC
AWS Glue Data Catalog
Data Catalog S3 tables
Data Catalog Redshift tables
Data Catalog RDS tables
All AWS Glue tables
Others
JDBC connections Info
AWS Glue connections store properties to connect to your database.
Manage connections
JDBC connections
Add JDBC connection
Search connections by name
Connection name
Database type
Host/Endpoint
Database
User
Last updated
AwsGlueDataBrew-student_db-conn
MySQL
student.cpuiu0qe4vkx.us-east-1.rds.amazonaws.com
student
admin
a few seconds ago
February 4, 2024, 2:32:13 pm
Welcome!
AWS Glue DataBrew is a visual data preparation tool that enables users to clean and normalize data without writing any code. Using DataBrew helps reduce the time it takes to prepare data for analytics and machine learning by up to 80%, compared to custom-developed data preparation. You can choose from over 250 ready-made transformations to automate data preparation tasks, such as filtering anomalies, converting data to standard formats, and correcting invalid values.
Using DataBrew, business analysts, data scientists, and data engineers can more easily collaborate to get insights from raw data. No matter what your technical level, you can explore and transform terabytes of raw data without creating clusters or managing infrastructure.
Using DataBrew, you can connect to all of your data on AWS and apply machine learning for analytics. With the intuitive DataBrew interface, you can interactively discover, visualize, clean, and transform raw data. DataBrew makes smart suggestions to help you identify data quality issues that can be difficult to find and time-consuming to fix. With DataBrew
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

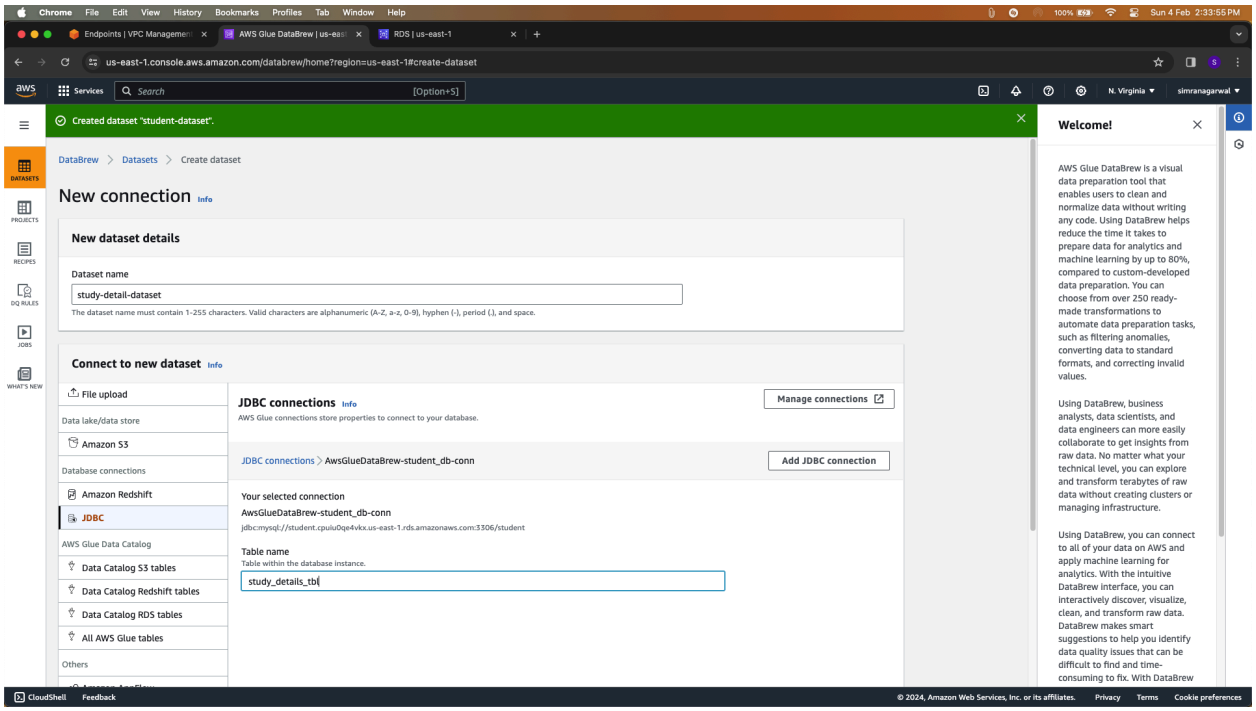

Created Glue connection "AwsGlueDataBrew-student_db-conn".
DataBrew > Datasets > Create dataset
New connection Info
New dataset details
Dataset name
school-dataset
The dataset name must contain 1-255 characters. Valid characters are alphanumeric (A-Z, a-z, 0-9), hyphen (-), period (.), and space.
Connect to new dataset Info
File upload
Data lake/data store
Amazon S3
Database connections
Amazon Redshift
JDBC
AWS Glue Data Catalog
Data Catalog S3 tables
Data Catalog Redshift tables
Data Catalog RDS tables
All AWS Glue tables
Others
JDBC connections Info
AWS Glue connections store properties to connect to your database.
Manage connections
JDBC connections > AwsGlueDataBrew-student_db-conn
Add JDBC connection
Your selected connection
AwsGlueDataBrew-student_db-conn
jdbc:mysql://student.cpuiu0qe4vkx.us-east-1.rds.amazonaws.com:3306/student
Table name
Table within the database instance.
school_tbl
Welcome!
AWS Glue DataBrew is a visual data preparation tool that enables users to clean and normalize data without writing any code. Using DataBrew helps reduce the time it takes to prepare data for analytics and machine learning by up to 80%, compared to custom-developed data preparation. You can choose from over 250 ready-made transformations to automate data preparation tasks, such as filtering anomalies, converting data to standard formats, and correcting invalid values.
Using DataBrew, business analysts, data scientists, and data engineers can more easily collaborate to get insights from raw data. No matter what your technical level, you can explore and transform terabytes of raw data without creating clusters or managing infrastructure.
Using DataBrew, you can connect to all of your data on AWS and apply machine learning for analytics. With the intuitive DataBrew interface, you can interactively discover, visualize, clean, and transform raw data. DataBrew makes smart suggestions to help you identify data quality issues that can be difficult to find and time-consuming to fix. With DataBrew
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

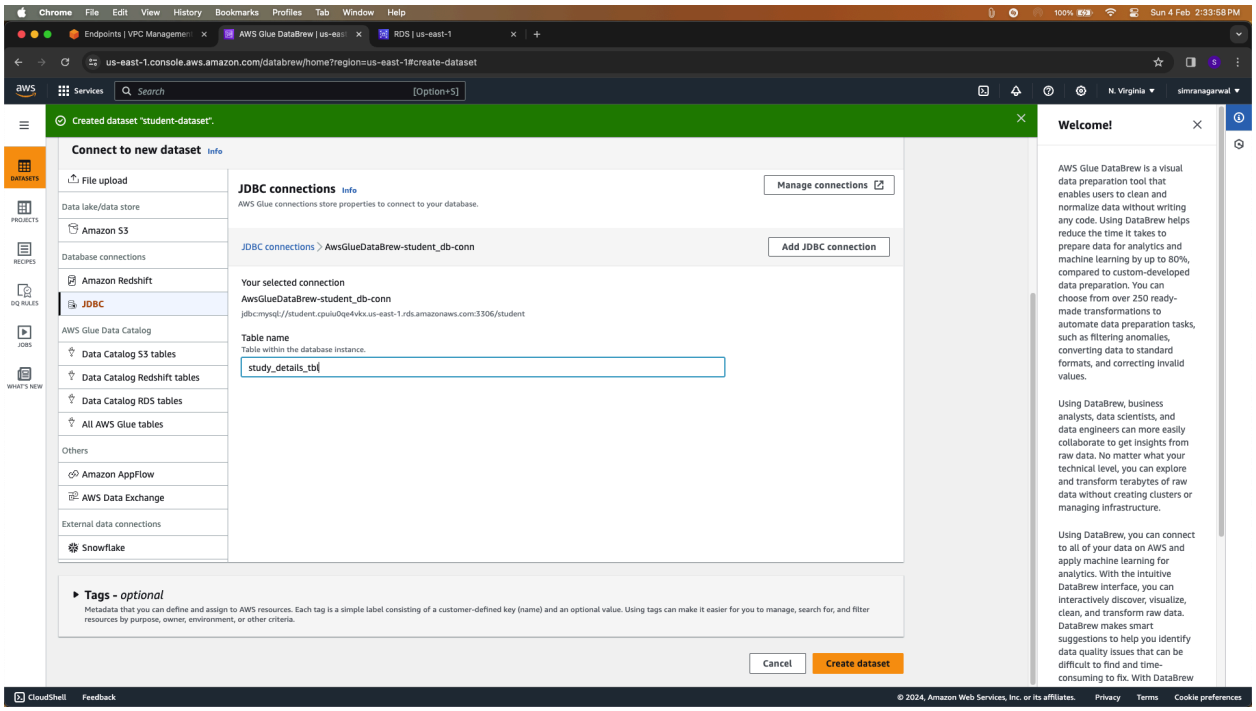

Created dataset "school-dataset".
DataBrew > Datasets
Datasets
Connections
Datasets (1) Info
View details
Create project with this dataset
Run data profile
Actions
Connect new dataset
Dataset name
Data type
Data profile
Source
Location
Create date
Created by
Tags
school-dataset
Database table
-
Database connection
AwsGlueDataBrew-student_db-conn
a few seconds ago
February 4, 2024, 2:32:57 pm
-

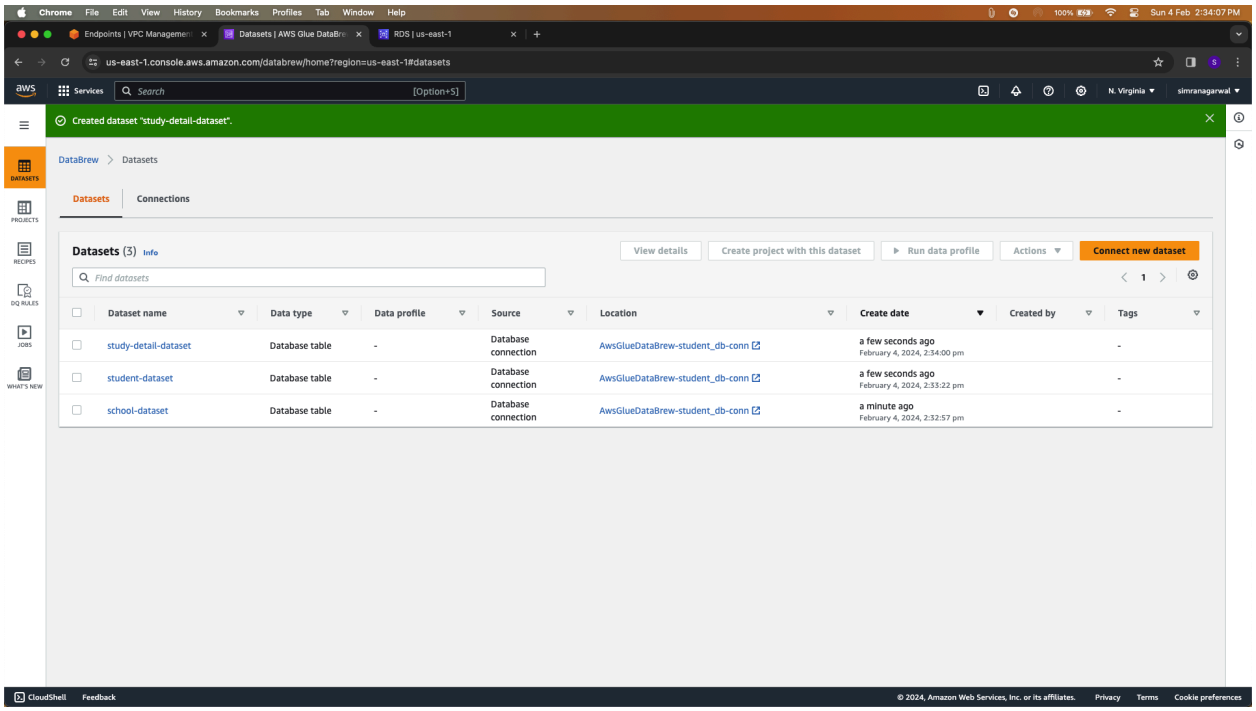

Created dataset "school-dataset".
DataBrew > Datasets > Create dataset
New connection Info
New dataset details
Dataset name
student-dataset
The dataset name must contain 1-255 characters. Valid characters are alphanumeric (A-Z, a-z, 0-9), hyphen (-), period (.), and space.
Connect to new dataset Info
File upload
Data lake/data store
Amazon S3
Database connections
Amazon Redshift
JDBC
AWS Glue Data Catalog
Data Catalog S3 tables
Data Catalog Redshift tables
Data Catalog RDS tables
All AWS Glue tables
Others
JDBC connections Info
AWS Glue connections store properties to connect to your database.
Manage connections
JDBC connections > AwsGlueDataBrew-student_db-conn
Add JDBC connection
Your selected connection
AwsGlueDataBrew-student_db-conn
jdbc:mysql://student.cpuiu0qe4vkx.us-east-1.rds.amazonaws.com:3306/student
Table name
Table within the database instance.
student_tbl
Welcome!
AWS Glue DataBrew is a visual data preparation tool that enables users to clean and normalize data without writing any code. Using DataBrew helps reduce the time it takes to prepare data for analytics and machine learning by up to 80%, compared to custom-developed data preparation. You can choose from over 250 ready-made transformations to automate data preparation tasks, such as filtering anomalies, converting data to standard formats, and correcting invalid values.
Using DataBrew, business analysts, data scientists, and data engineers can more easily collaborate to get insights from raw data. No matter what your technical level, you can explore and transform terabytes of raw data without creating clusters or managing infrastructure.
Using DataBrew, you can connect to all of your data on AWS and apply machine learning for analytics. With the intuitive DataBrew interface, you can interactively discover, visualize, clean, and transform raw data. DataBrew makes smart suggestions to help you identify data quality issues that can be difficult to find and time-consuming to fix. With DataBrew


Created dataset "school-dataset".
Connect to new dataset
JDBC connections
Manage connections
Add JDBC connection
Your selected connection
AwsGlueDataBrew-student_db-conn
Table name
student_tbl
Tags - optional
Cancel
Create dataset
Welcome!

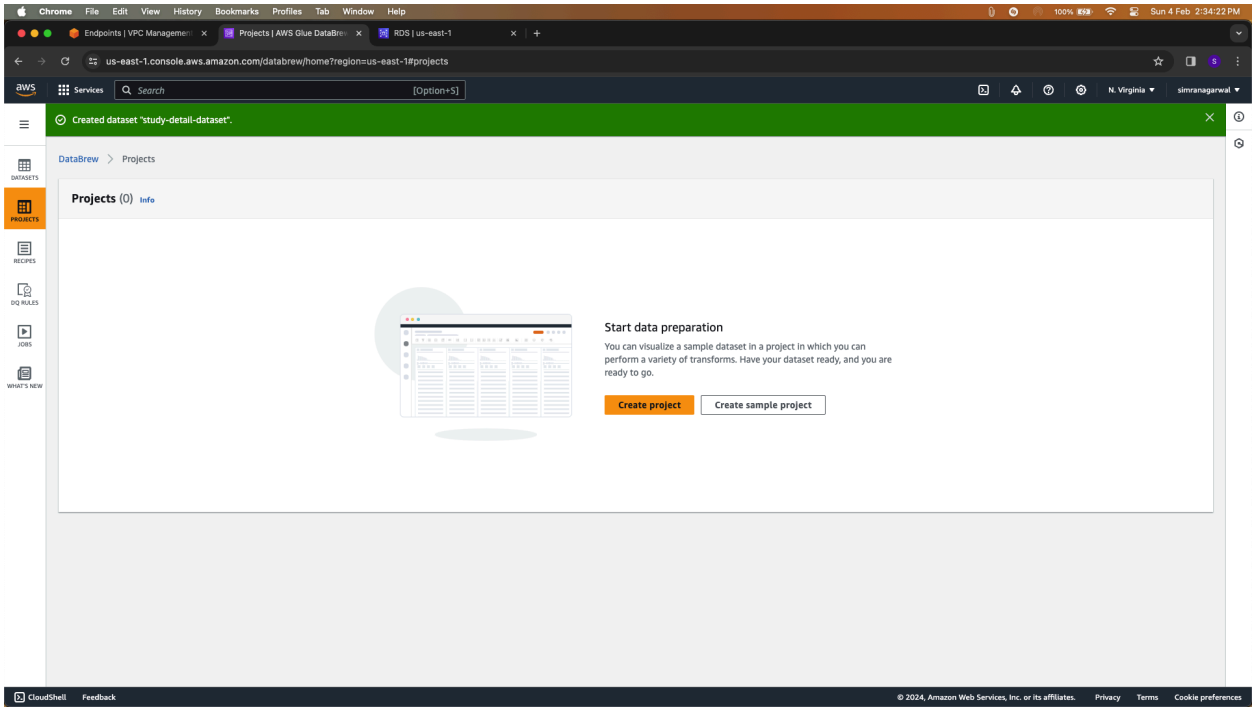

Created dataset "student-dataset".
DataBrew
Datasets
Connections
Datasets (2)
View details
Create project with this dataset
Run data profile
Actions
Connect new dataset
Dataset name
Data type
Data profile
Source
Location
Create date
Created by
Tags
student-dataset
school-dataset
Database table
Database connection
AwsGlueDataBrew-student_db-conn
a few seconds ago

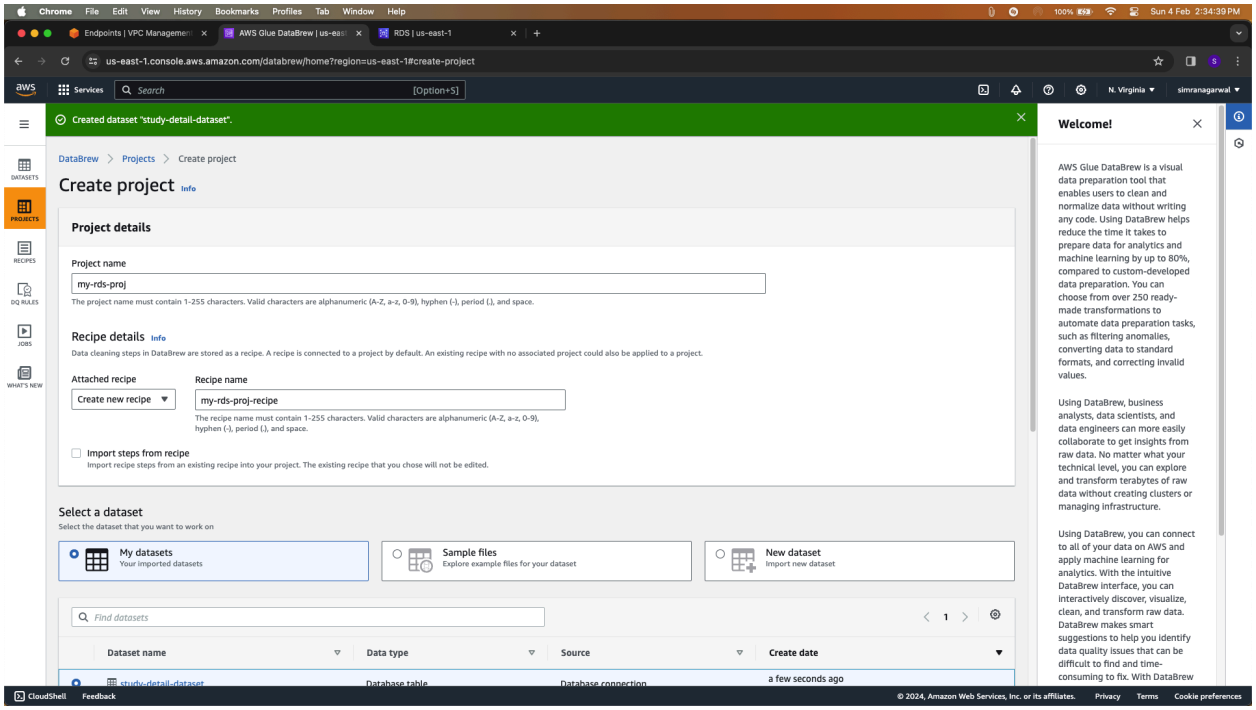

Created dataset "student-dataset".
DataBrew > Datasets > Create dataset
New connection Info
New dataset details
Dataset name
study-detail-dataset
The dataset name must contain 1-255 characters. Valid characters are alphanumeric (A-Z, a-z, 0-9), hyphen (-), period (.), and space.
Connect to new dataset Info
File upload
Data lake/data store
Amazon S3
Database connections
Amazon Redshift
JDBC
AWS Glue Data Catalog
Data Catalog S3 tables
Data Catalog Redshift tables
Data Catalog RDS tables
All AWS Glue tables
Others
JDBC connections Info
AWS Glue connections store properties to connect to your database.
Manage connections
JDBC connections > AwsGlueDataBrew-student_db-conn
Add JDBC connection
Your selected connection
AwsGlueDataBrew-student_db-conn
jdbc:mysql://student.cpuiu0qe4vkx.us-east-1.rds.amazonaws.com:3306/student
Table name
Table within the database instance.
study_details_tbl
Welcome!
AWS Glue DataBrew is a visual data preparation tool that enables users to clean and normalize data without writing any code. Using DataBrew helps reduce the time it takes to prepare data for analytics and machine learning by up to 80%, compared to custom-developed data preparation. You can choose from over 250 ready-made transformations to automate data preparation tasks, such as filtering anomalies, converting data to standard formats, and correcting invalid values.
Using DataBrew, business analysts, data scientists, and data engineers can more easily collaborate to get insights from raw data. No matter what your technical level, you can explore and transform terabytes of raw data without creating clusters or managing infrastructure.
Using DataBrew, you can connect to all of your data on AWS and apply machine learning for analytics. With the intuitive DataBrew interface, you can interactively discover, visualize, clean, and transform raw data. DataBrew makes smart suggestions to help you identify data quality issues that can be difficult to find and time-consuming to fix. With DataBrew
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Created dataset "student-dataset".
Connect to new dataset Info
File upload
Amazon S3
Amazon Redshift
JDBC
Data Catalog S3 tables
Data Catalog Redshift tables
Data Catalog RDS tables
All AWS Glue tables
Amazon AppFlow
AWS Data Exchange
Snowflake
JDBC connections Info
AWS Glue connections store properties to connect to your database.
Manage connections
JDBC connections > AwsGlueDataBrew-student_db-conn
Add JDBC connection
Your selected connection
AwsGlueDataBrew-student_db-conn
jdbc:mysql://student.cpuiu0qe4vkx.us-east-1.rds.amazonaws.com:3306/student
Table name
Table within the database instance.
study_details_tbl
Tags - optional
Cancel
Create dataset
Welcome!
AWS Glue DataBrew is a visual data preparation tool that enables users to clean and normalize data without writing any code. Using DataBrew helps reduce the time it takes to prepare data for analytics and machine learning by up to 80%, compared to custom-developed data preparation. You can choose from over 250 ready-made transformations to automate data preparation tasks, such as filtering anomalies, converting data to standard formats, and correcting invalid values.
Using DataBrew, business analysts, data scientists, and data engineers can more easily collaborate to get insights from raw data. No matter what your technical level, you can explore and transform terabytes of raw data without creating clusters or managing infrastructure.
Using DataBrew, you can connect to all of your data on AWS and apply machine learning for analytics. With the intuitive DataBrew interface, you can interactively discover, visualize, clean, and transform raw data. DataBrew makes smart suggestions to help you identify data quality issues that can be difficult to find and time-consuming to fix. With DataBrew

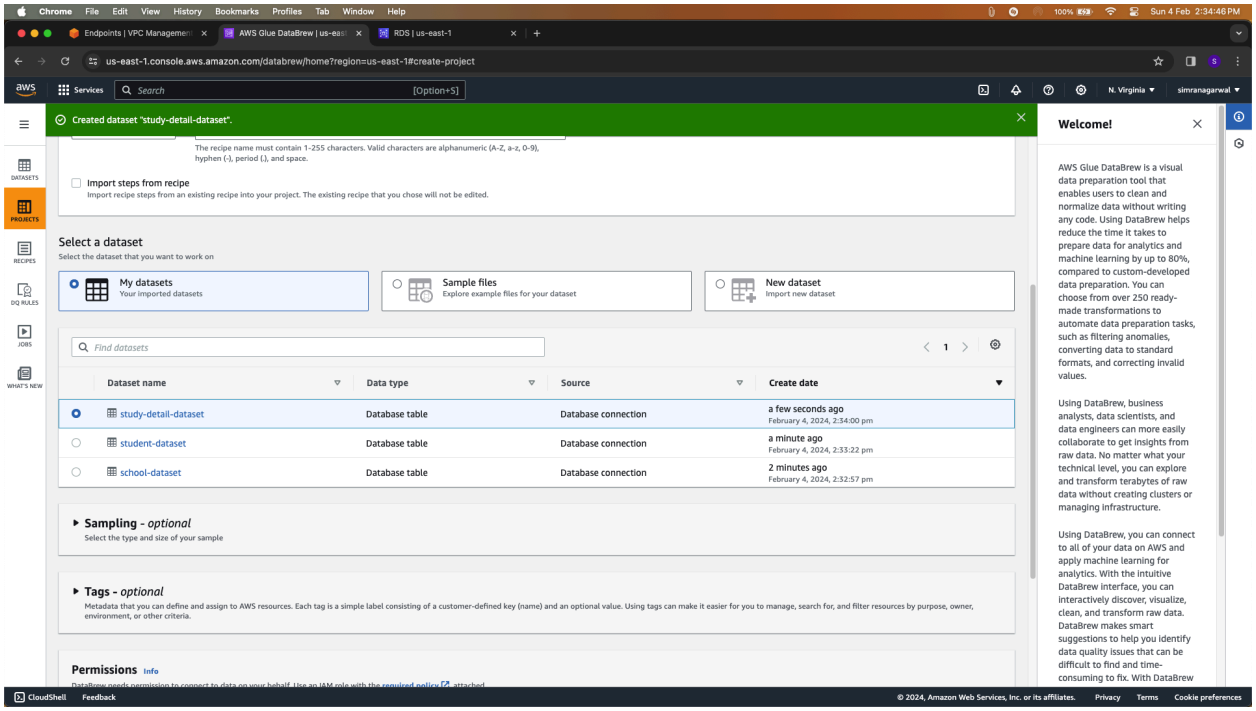

Created dataset "study-detail-dataset".
DataBrew > Datasets
Datasets
Connections
Datasets (3) Info
View details
Create project with this dataset
Run data profile
Actions
Connect new dataset
Find datasets
Dataset name
Data type
Data profile
Source
Location
Create date
Created by
Tags
study-detail-dataset
Database table
Database connection
AwsGlueDataBrew-student_db-conn
a few seconds ago
February 4, 2024, 2:34:00 pm
student-dataset
Database table
Database connection
AwsGlueDataBrew-student_db-conn
a few seconds ago
February 4, 2024, 2:33:22 pm
school-dataset
Database table
Database connection
AwsGlueDataBrew-student_db-conn
a minute ago
February 4, 2024, 2:32:57 pm

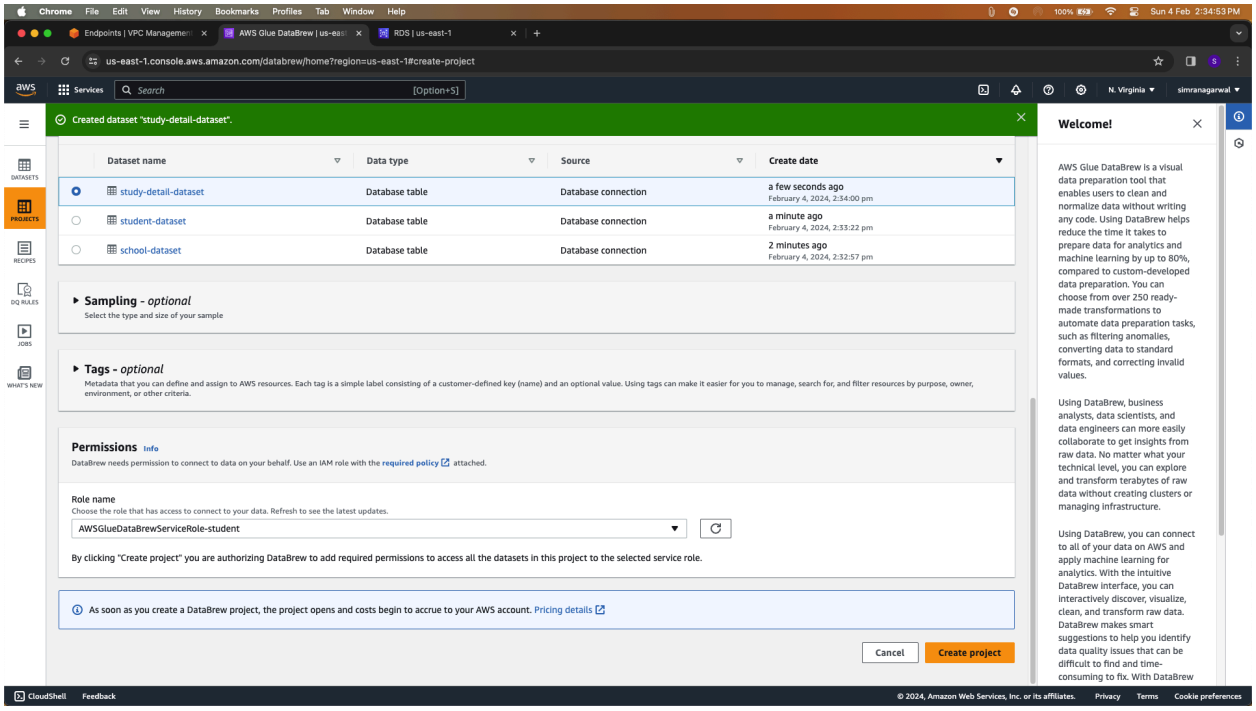
32. Now when all three datasets are created in Glue Databrew, it’s time to create a project and name it “my-rds-proj”.



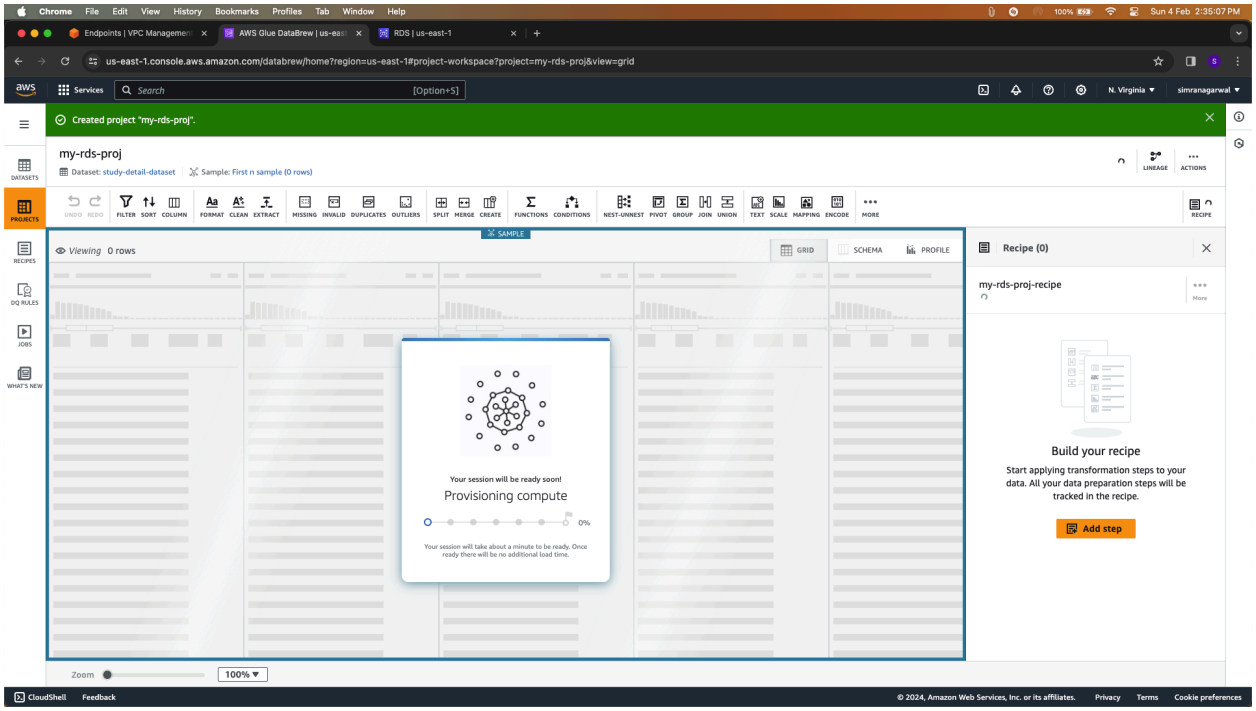


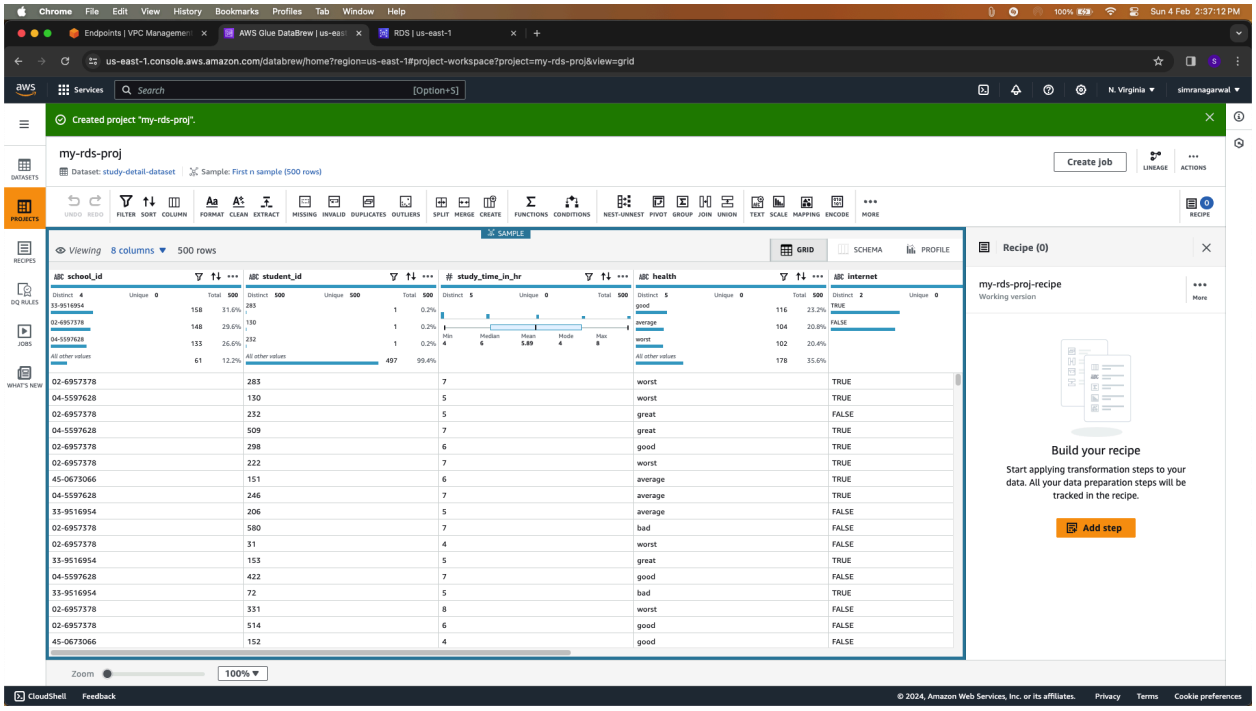
33. The form to create a project asks us to select a dataset.



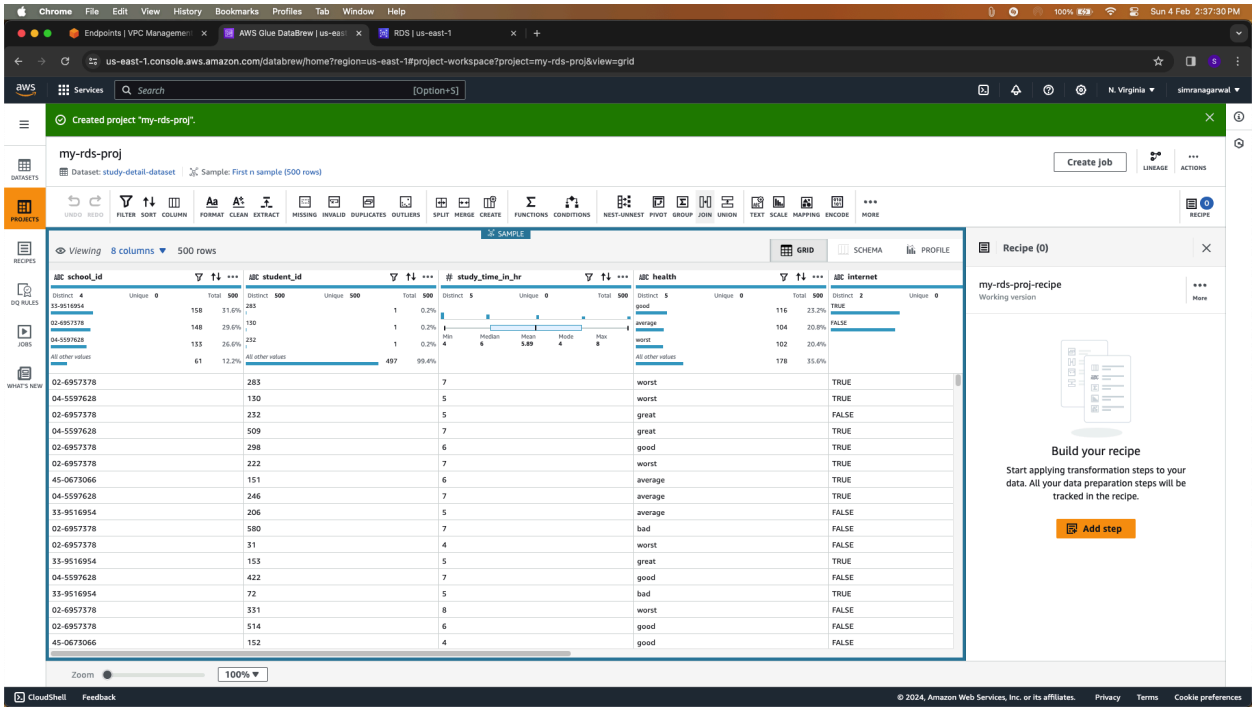


34. Wait for the project report to open and then we can perform operations like join and filter.

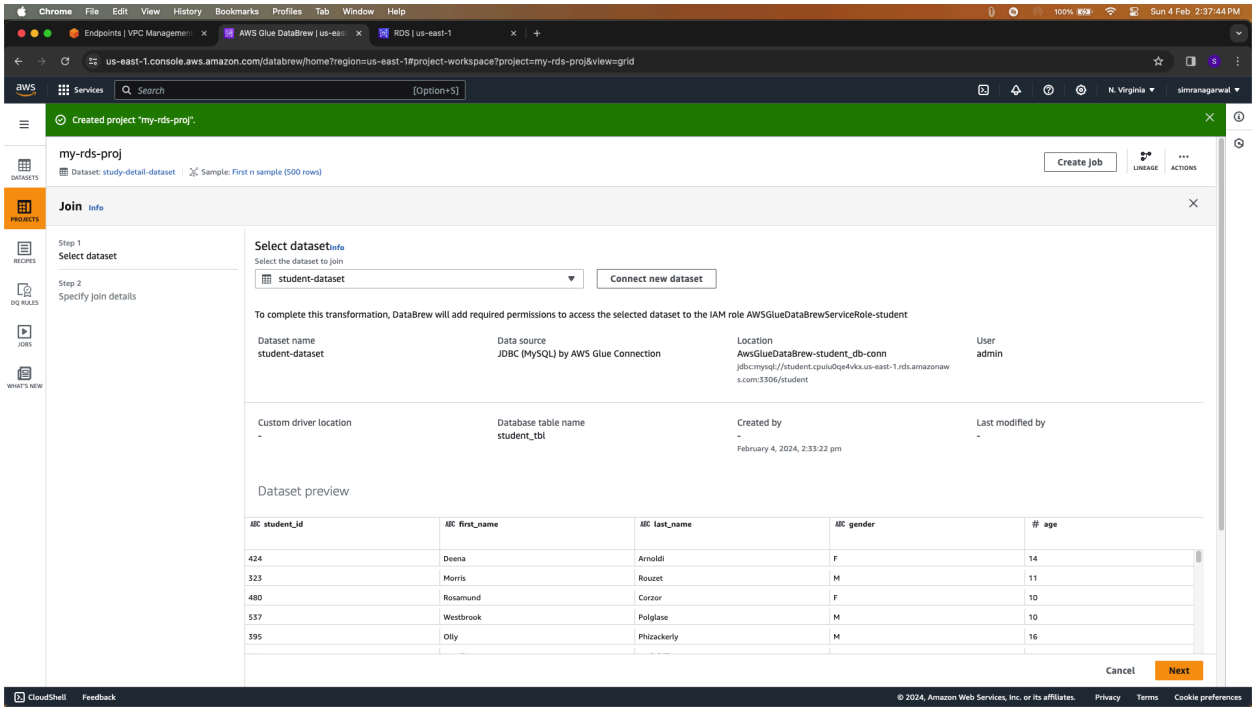




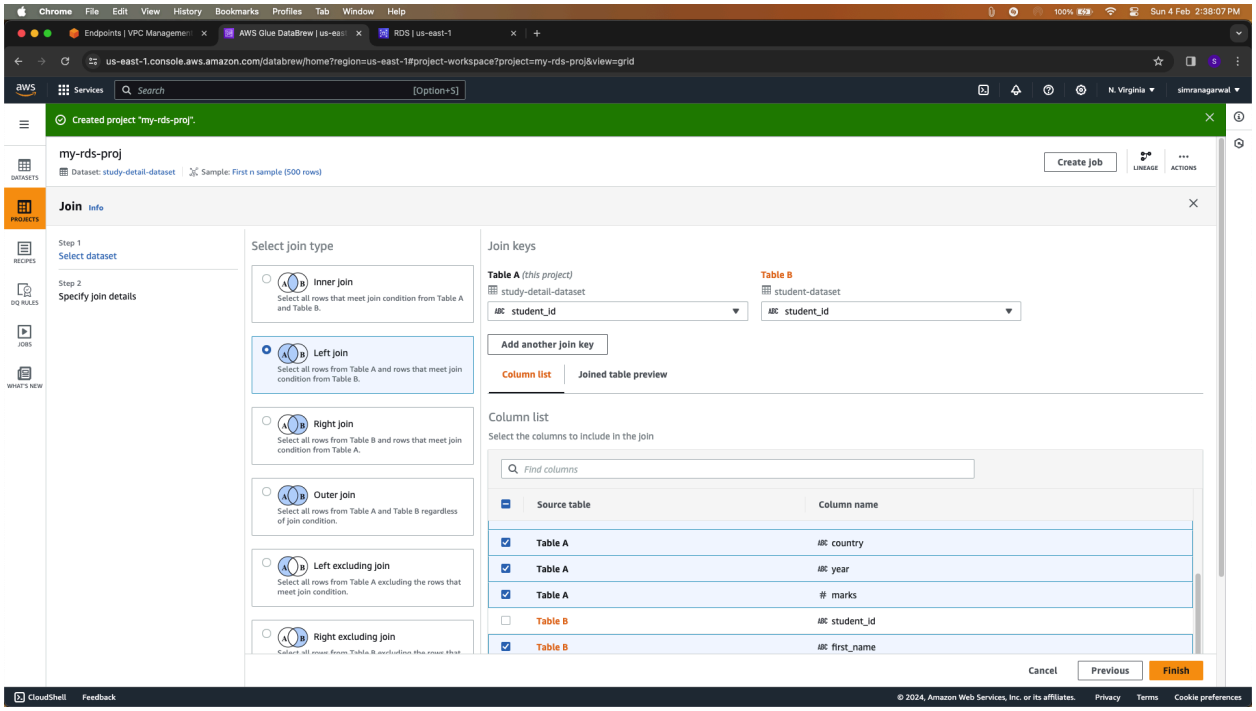
35. As the report have opened and we can start performing our operations, click on Join in the bar above the report.

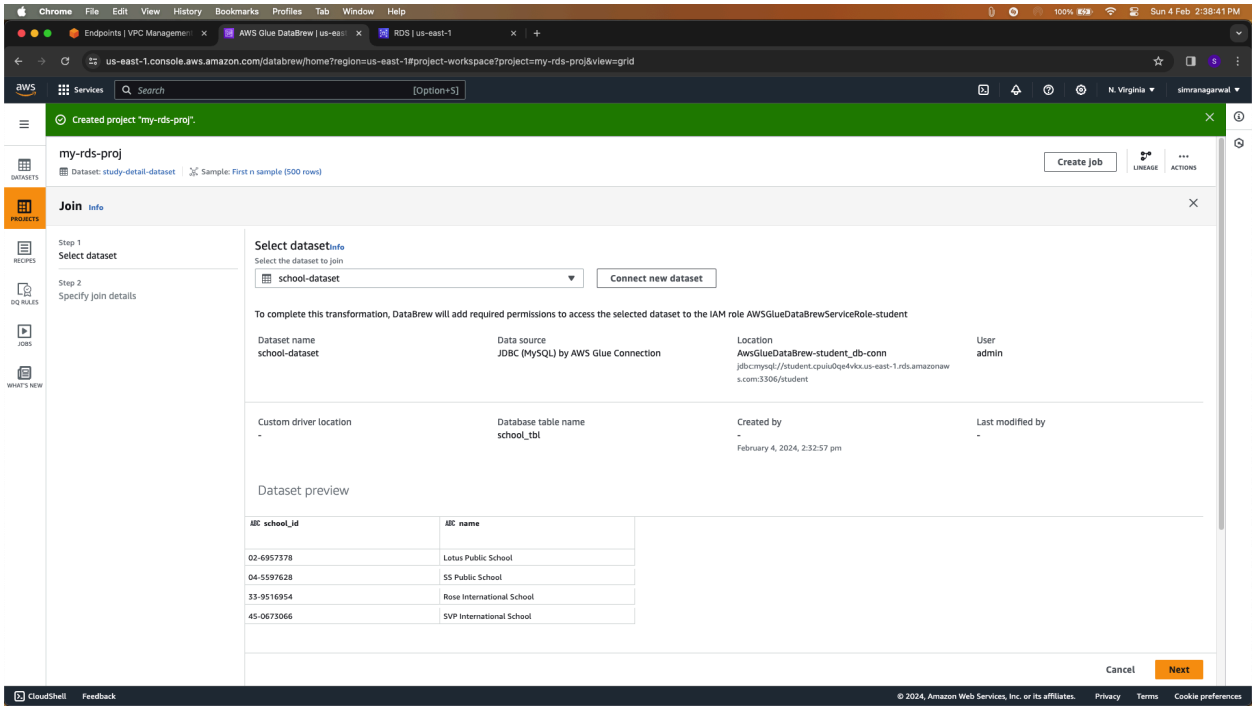


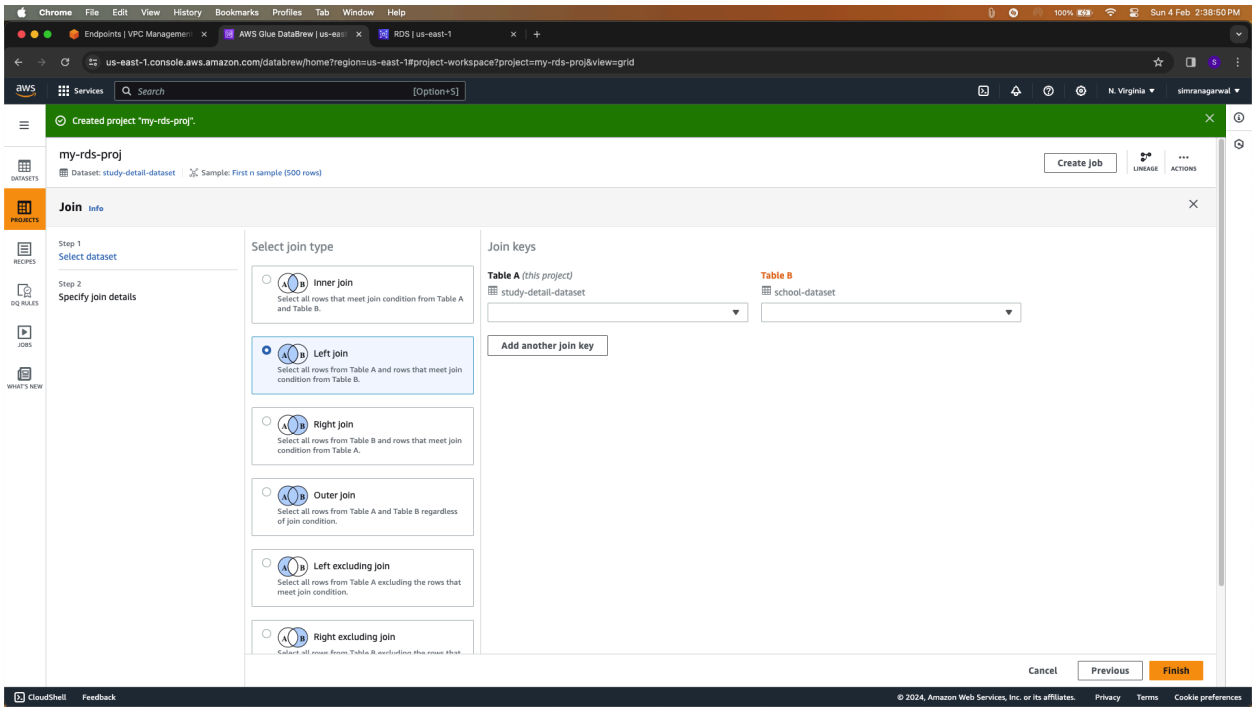
36. Select the student-dataset and click on Next.

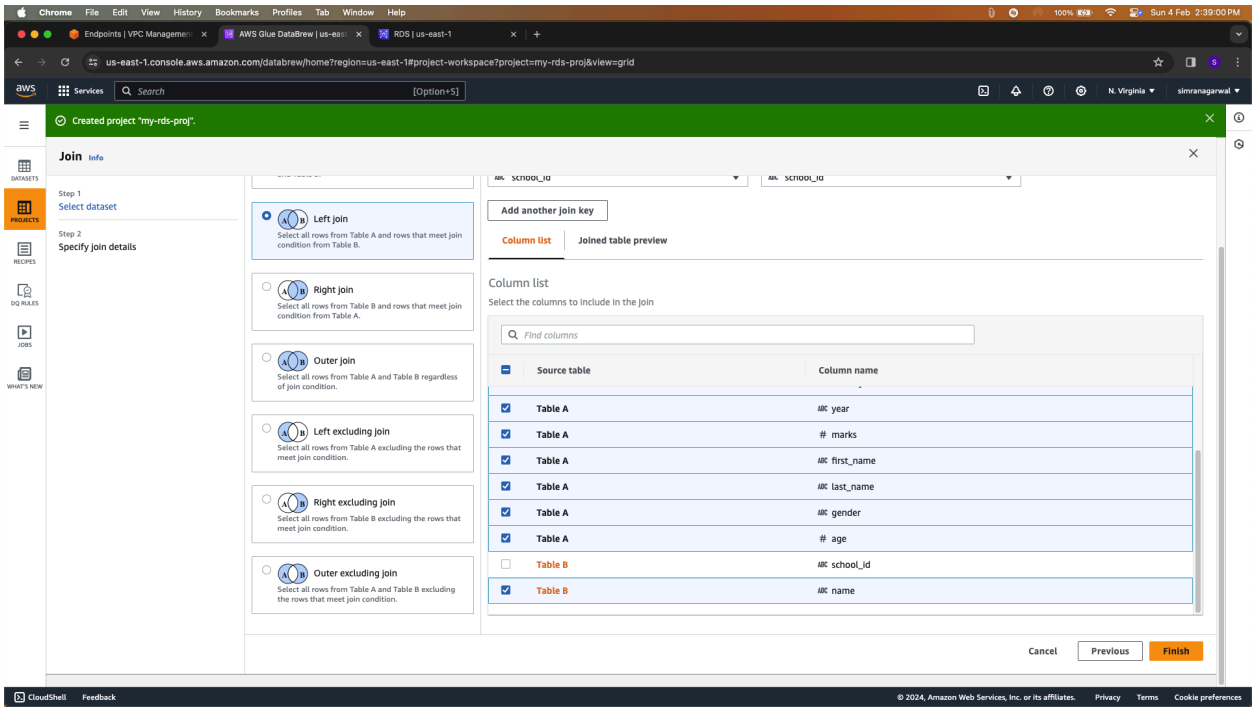


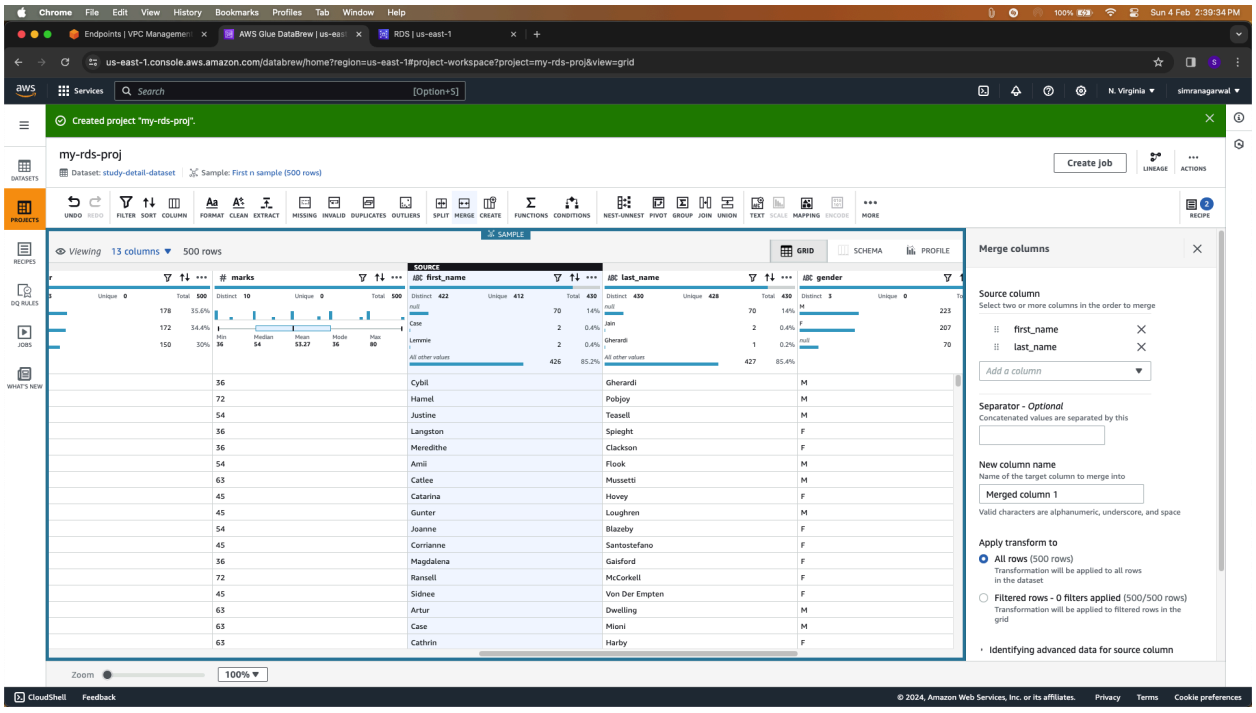
37. Select Left Join and the remove the duplicate (or the common) field from table B. Similary join the third table with the left join and excluding the duplicate / common field from table.



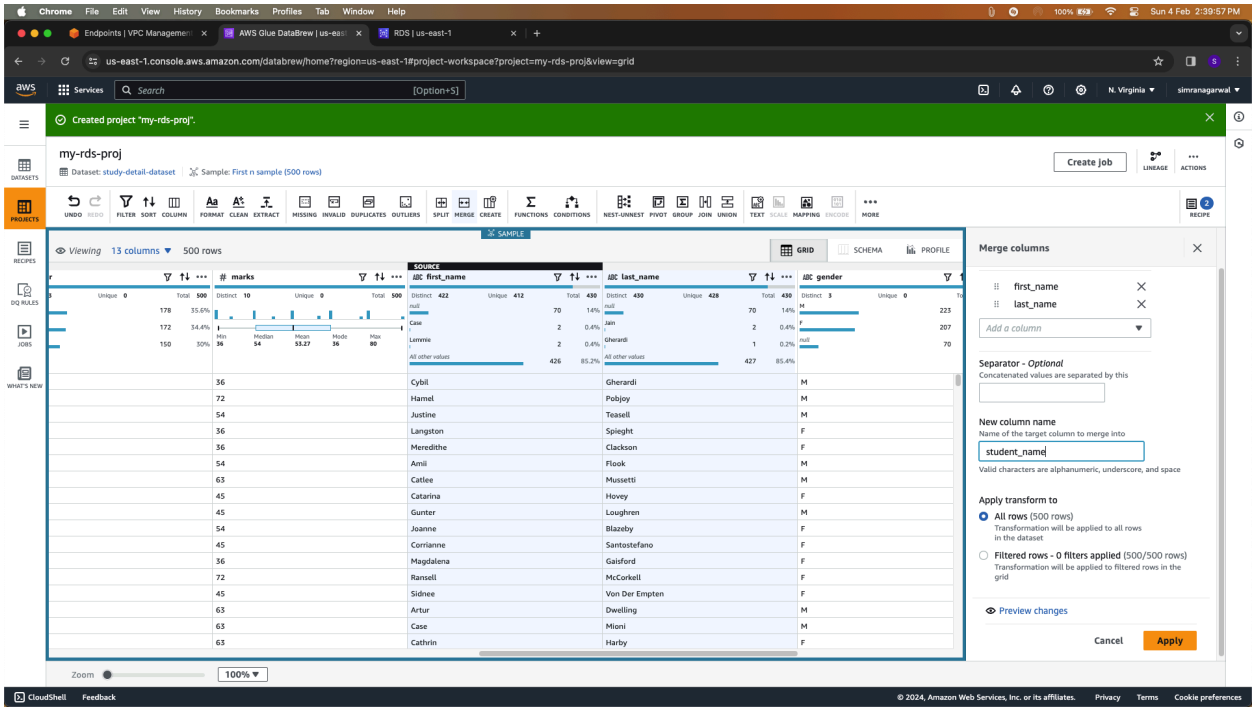



Created project "my-rds-proj".
my-rds-proj
Dataset: study-detail-dataset
Sample: First n sample (500 rows)
Create job
Join Info
Step 1
Select dataset
Step 2
Specify join details
Select join type
Inner join
Select all rows that meet join condition from Table A and Table B.
Left join
Select all rows from Table A and rows that meet join condition from Table B.
Right join
Select all rows from Table B and rows that meet join condition from Table A.
Outer join
Select all rows from Table A and Table B regardless of join condition.
Left excluding join
Select all rows from Table A excluding the rows that meet join condition.
Right excluding join
Join keys
Table A (this project)
study-detail-dataset
Table B
school-dataset
Add another join key
Cancel
Previous
Finish
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Created project "my-rds-proj".
Join
Step 1
Select dataset
Step 2
Specify join details
Left join
Right join
Outer join
Left excluding join
Right excluding join
Outer excluding join
Add another join key
Column list
Joined table preview
Column list
Select the columns to include in the join
Find columns
Source table
Column name
Table A
year
marks
first_name
last_name
gender
age
Table B
school_id
name
Cancel
Previous
Finish

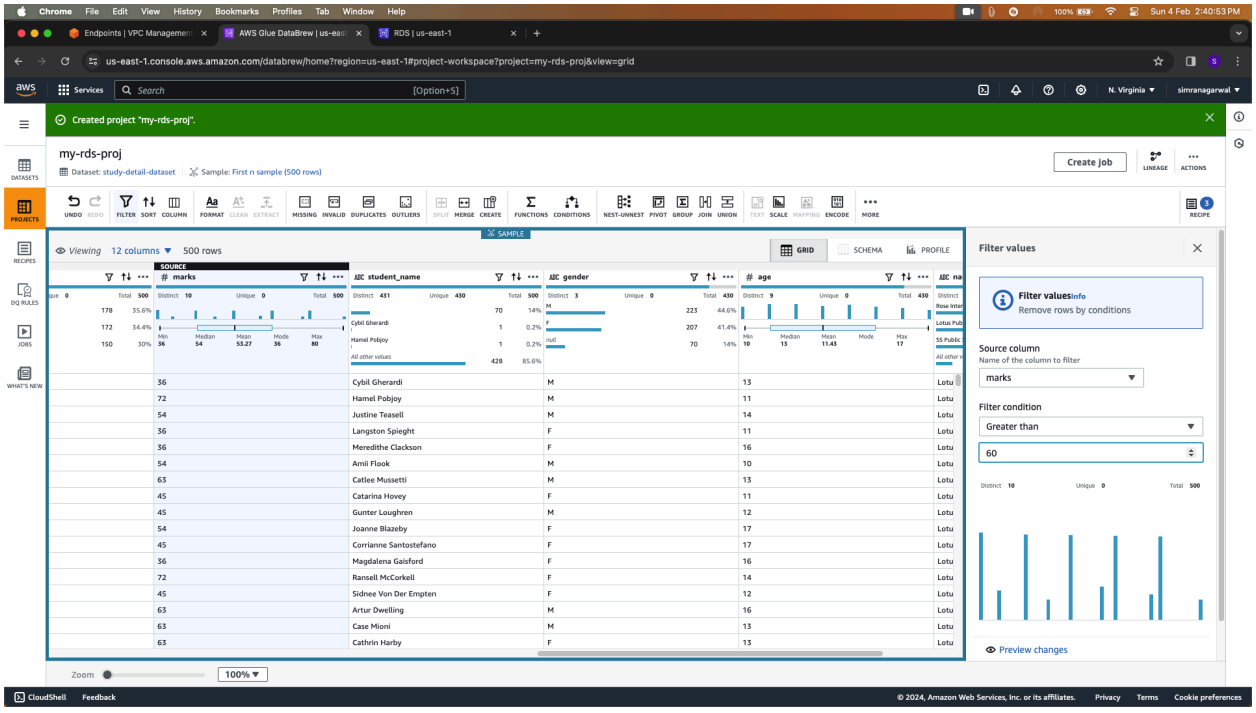

Created project "my-rds-proj".
my-rds-proj
Create job
Viewing 13 columns 500 rows
GRID
SCHEMA
PROFILE
Merge columns
Source column
Select two or more columns in the order to merge
first_name
last_name
Separator - Optional
Concatenated values are separated by this
New column name
Merged column 1
Apply transform to
All rows (500 rows)
Filtered rows - 0 filters applied (500/500 rows)
Identifying advanced data for source column

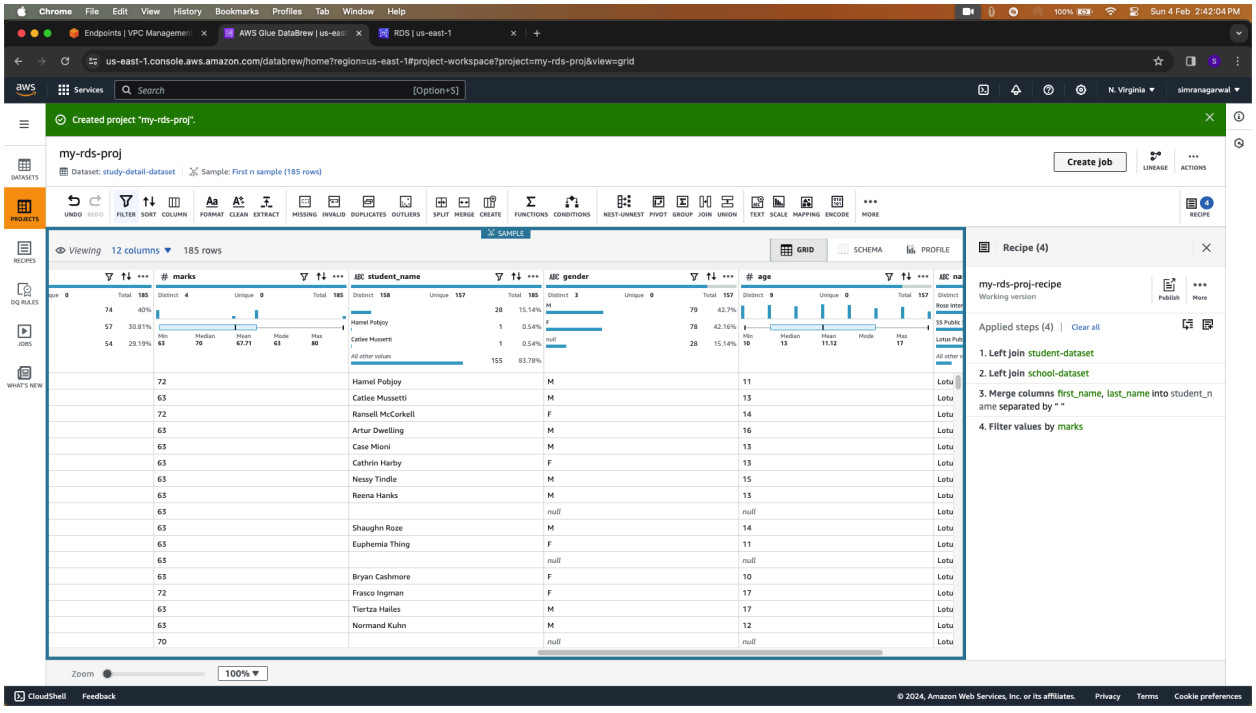
38. Now we will click on first name field and then click on merge with the last name field and rename the resulting field as student_name with a seperator as a “ “ (space bar).



39. We need to filter out the high performing students, so for that we will choose the marks field and choose the filter option and within that we will choose the greater than filter and enter the value as “60”.

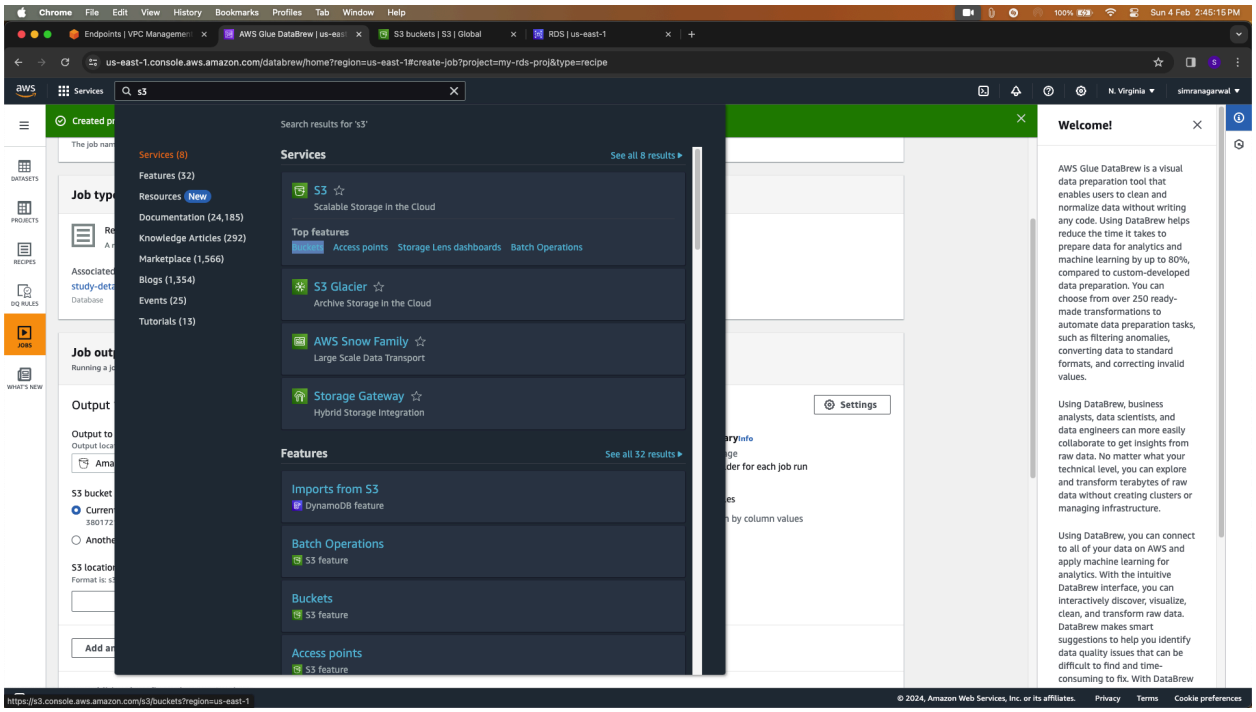
Resulting Report will have the details of all the high performing students who scored more than 60.


Created project "my-rds-proj".
my-rds-proj
Dataset: study-detail-dataset
Sample: First n sample (500 rows)
Create Job
Viewing 12 columns 500 rows
Filter values
Filter values
Remove rows by conditions
Source column
Name of the column to filter
marks
Filter condition
Greater than
60
Preview changes

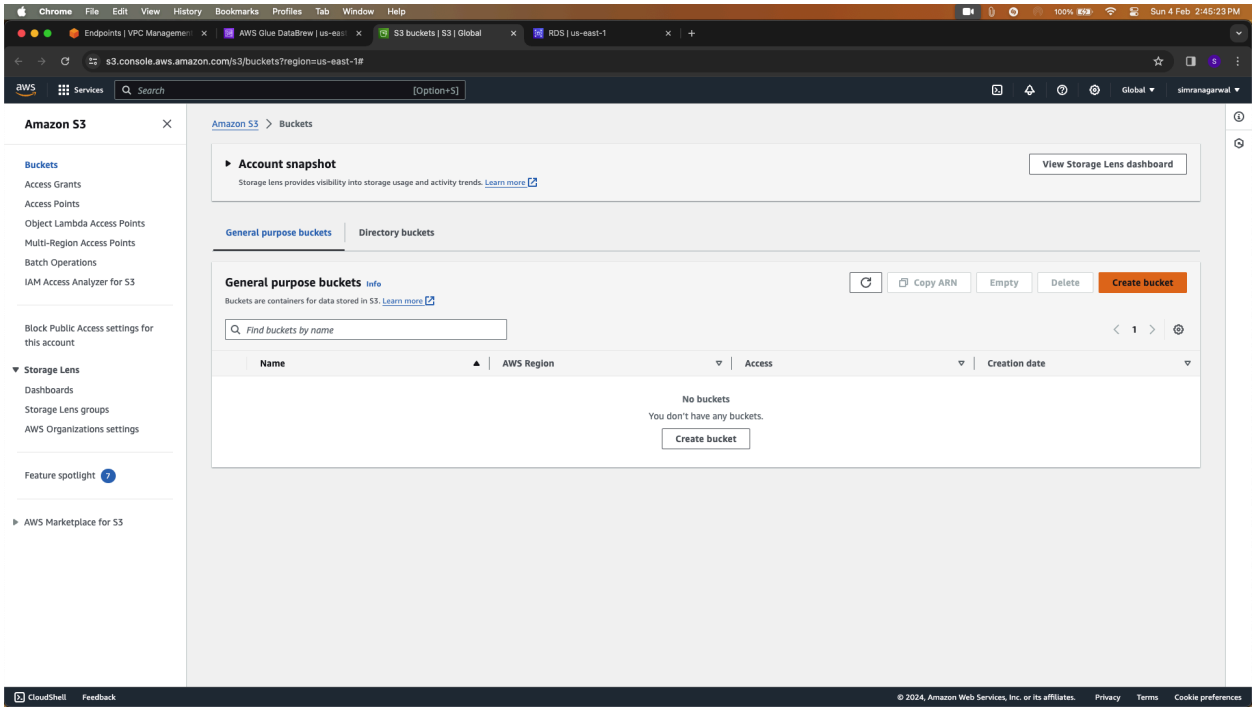

Created project "my-rds-proj".
my-rds-proj
Dataset: study-detail-dataset
Sample: First n sample (185 rows)
Create Job
Viewing 12 columns 185 rows
Recipe (4)
my-rds-proj-recipe
Working version
Applied steps (4)
Clear all
1. Left join student-dataset
2. Left join school-dataset
3. Merge columns first_name, last_name into student_name separated by " "
4. Filter values by marks

40. Now that you are ready to run the job you first need ro create an S3 bucket for the job to store the resulting dataset.

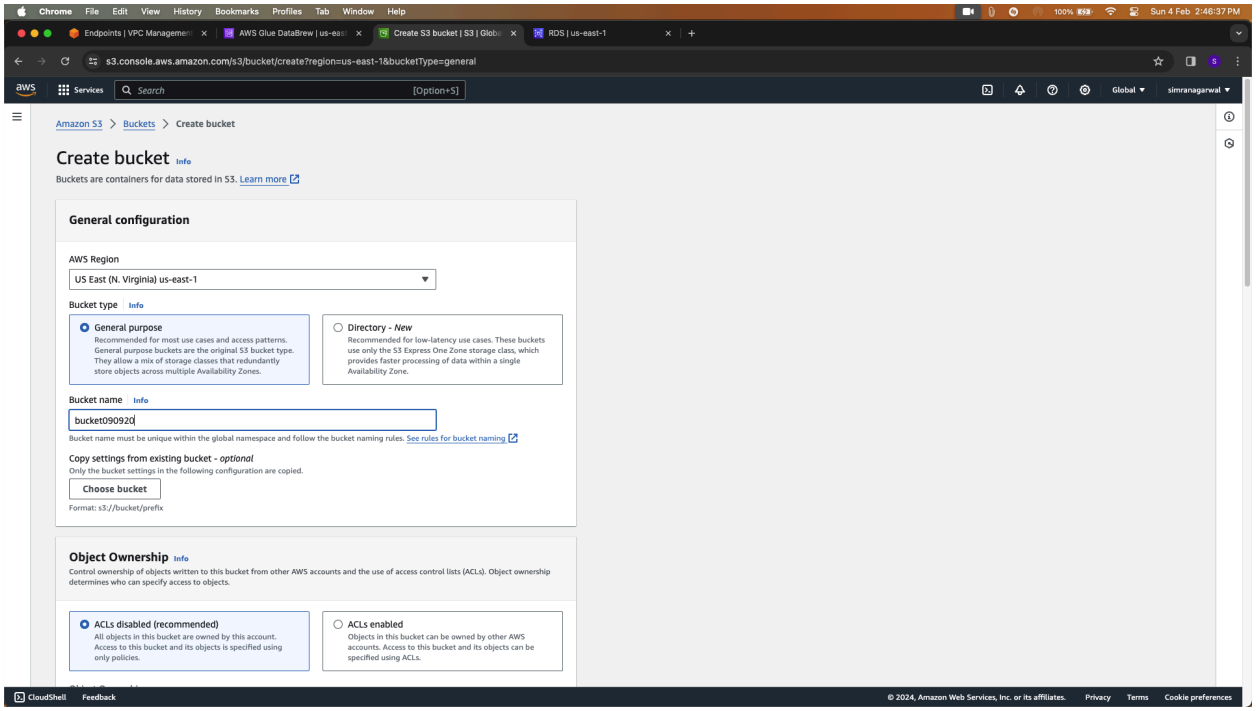
Search for S3 in AWS Console Search Bar and click on S3 Buckets.

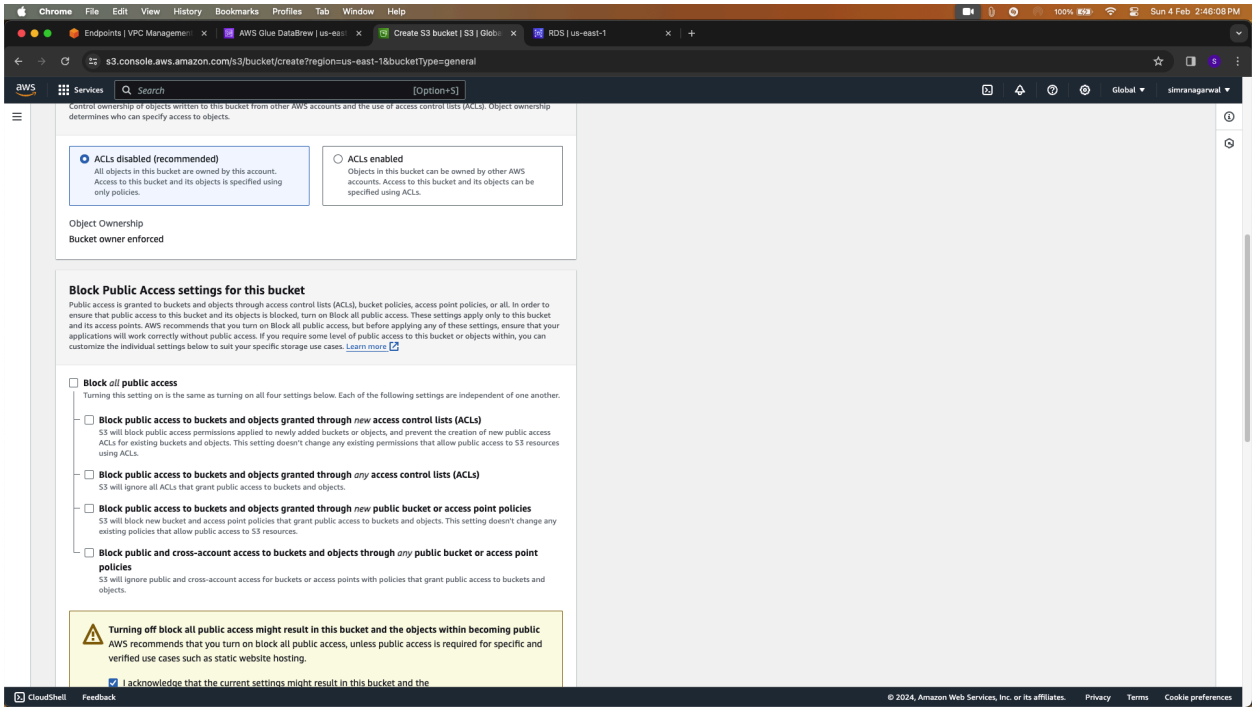


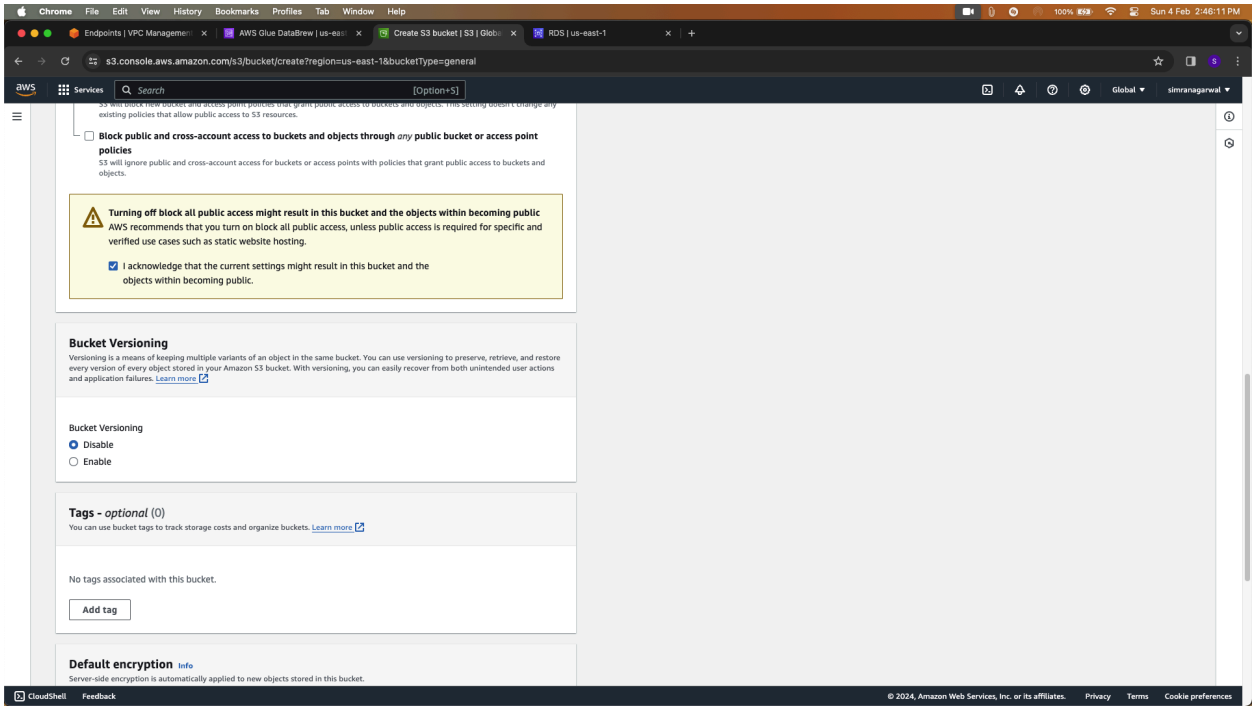
41. Click on create a bucket.

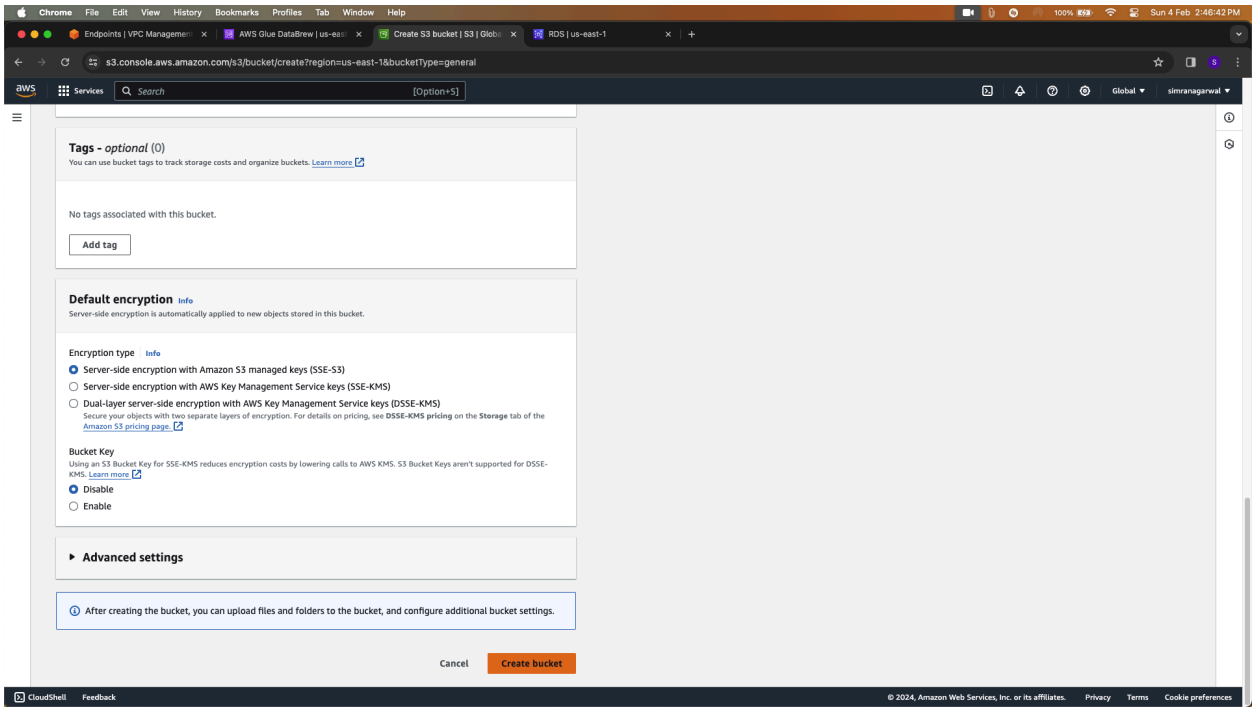


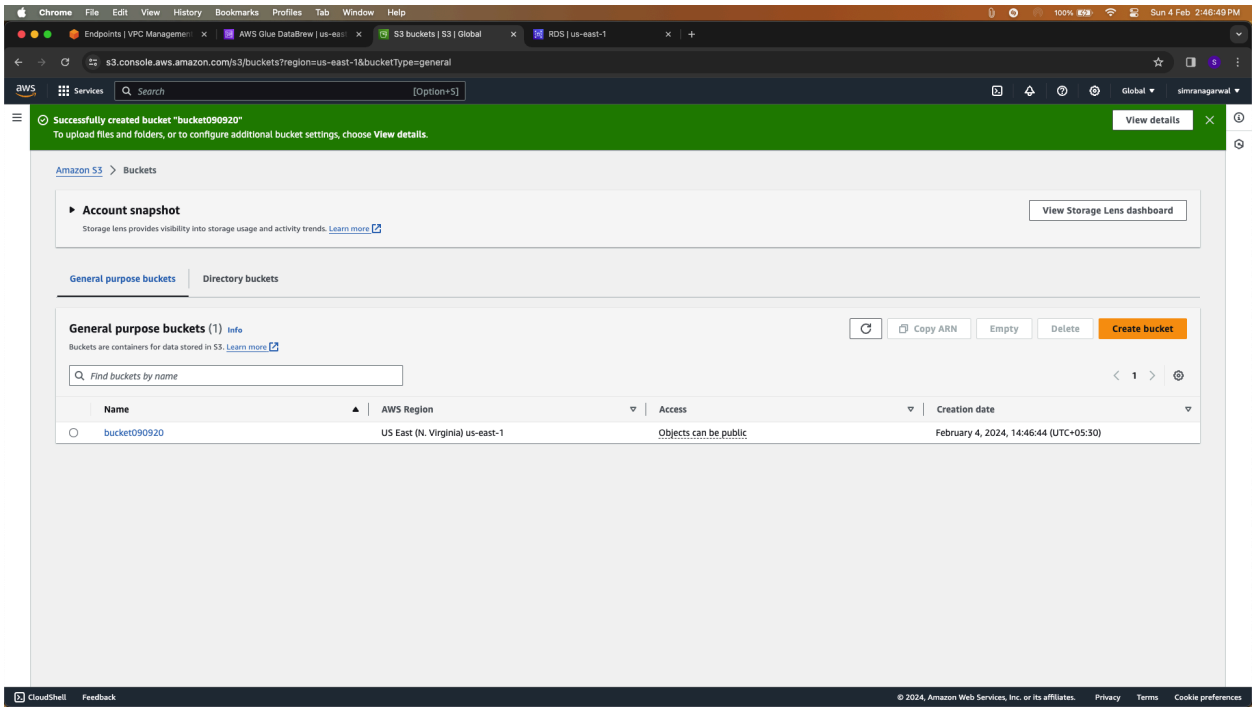
42. Choose a desired name for the bucket and fill out the required the details. Remember not to block the public access.



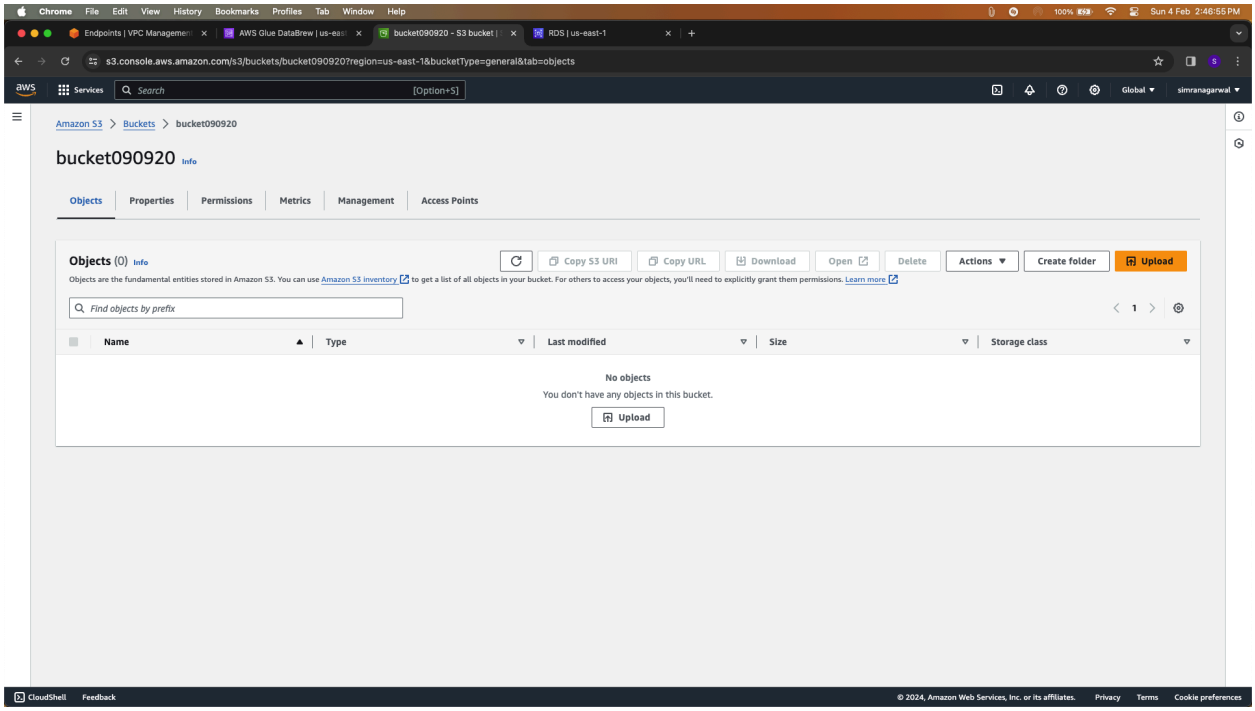

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sun 4 Feb 2:46:08 PM
s3.console.aws.amazon.com/s3/bucket/create?region=us-east-1&bucketType=general
Services Search [Option+S]
Global simranagarwal
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.
ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.
ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.
Object Ownership
Bucket owner enforced
Block Public Access settings for this bucket
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more
Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
Block public access to buckets and objects granted through new access control lists (ACLs)
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
Block public access to buckets and objects granted through any access control lists (ACLs)
S3 will ignore all ACLs that grant public access to buckets and objects.
Block public access to buckets and objects granted through new public bucket or access point policies
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
Block public and cross-account access to buckets and objects through any public bucket or access point policies
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.
Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.
I acknowledge that the current settings might result in this bucket and the
CloudShell Feedback
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sun 4 Feb 2:46:11 PM
s3.console.aws.amazon.com/s3/bucket/create?region=us-east-1&bucketType=general
Services Search [Option+S]
Global simranagarwal
Block public and cross-account access to buckets and objects through any public bucket or access point policies
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.
Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.
I acknowledge that the current settings might result in this bucket and the objects within becoming public.
Bucket Versioning
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more
Bucket Versioning
Disable
Enable
Tags - optional (0)
You can use bucket tags to track storage costs and organize buckets. Learn more
No tags associated with this bucket.
Add tag
Default encryption Info
Server-side encryption is automatically applied to new objects stored in this bucket.
CloudShell Feedback
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

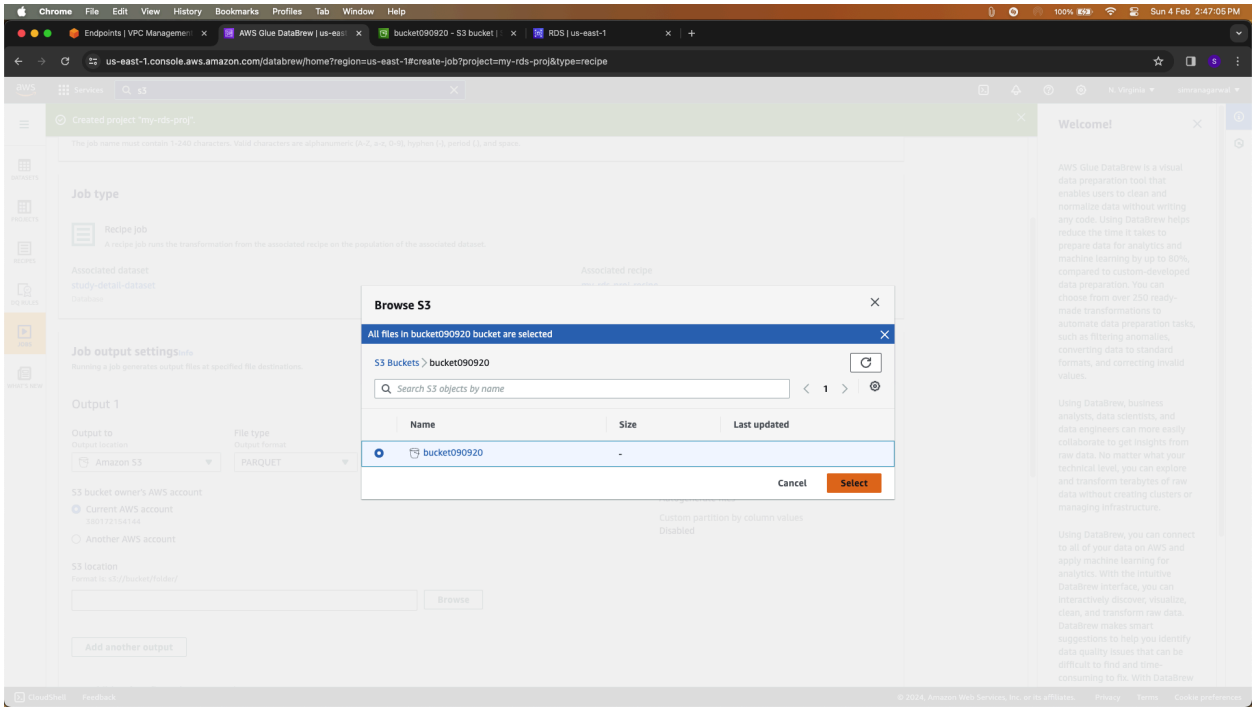

Tags - optional (0)
You can use bucket tags to track storage costs and organize buckets. Learn more
No tags associated with this bucket.
Add tag
Default encryption Info
Server-side encryption is automatically applied to new objects stored in this bucket.
Encryption type Info
Server-side encryption with Amazon S3 managed keys (SSE-S3)
Server-side encryption with AWS Key Management Service keys (SSE-KMS)
Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the Amazon S3 pricing page.
Bucket Key
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. Learn more
Disable
Enable
Advanced settings
After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.
Cancel
Create bucket

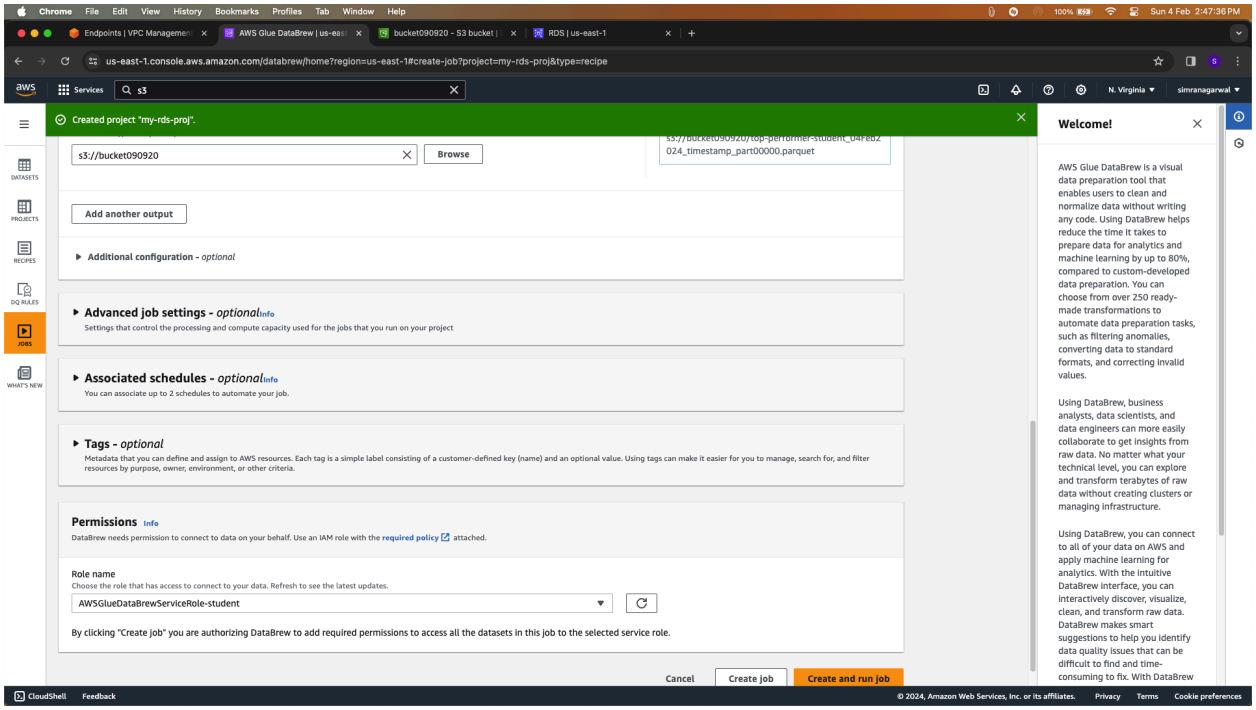

Successfully created bucket "bucket090920"
To upload files and folders, or to configure additional bucket settings, choose View details.
View details
Amazon S3 > Buckets
Account snapshot
Storage lens provides visibility into storage usage and activity trends. Learn more
View Storage Lens dashboard
General purpose buckets
Directory buckets
General purpose buckets (1) Info
Buckets are containers for data stored in S3. Learn more
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
Name
AWS Region
Access
Creation date
bucket090920
US East (N. Virginia) us-east-1
Objects can be public
February 4, 2024, 14:46:44 (UTC+05:30)

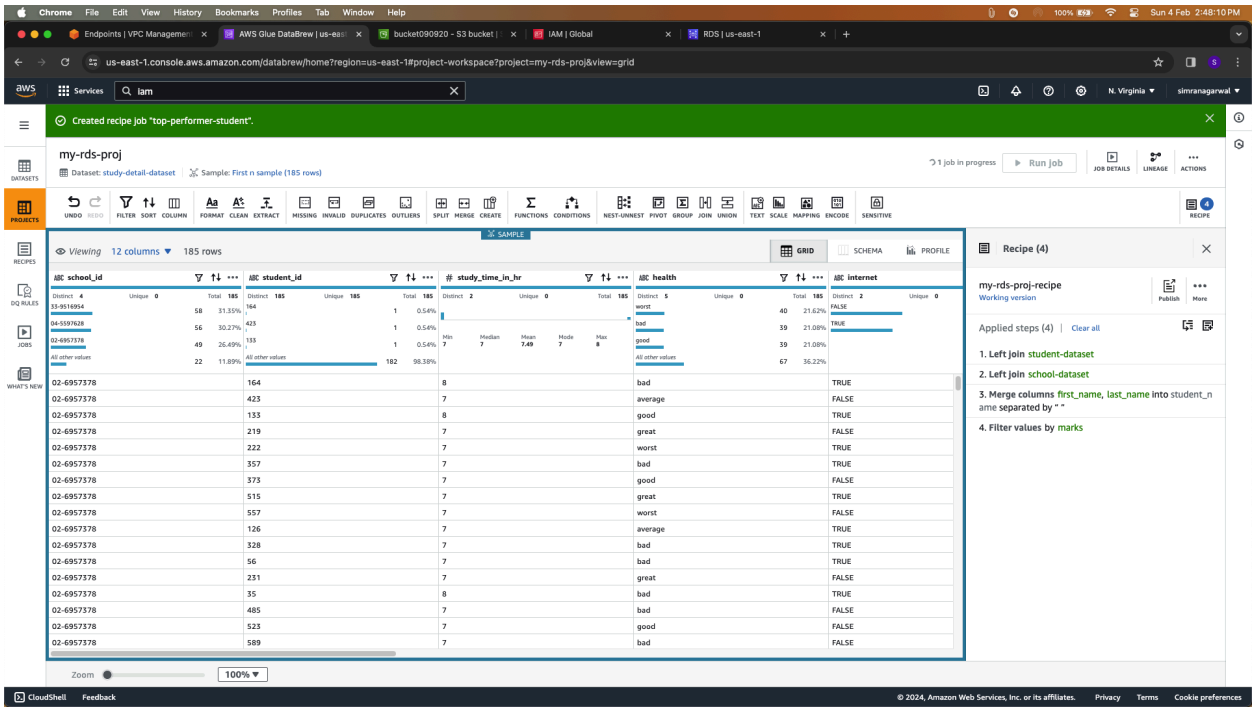
43. We have successfully created a bucket here.

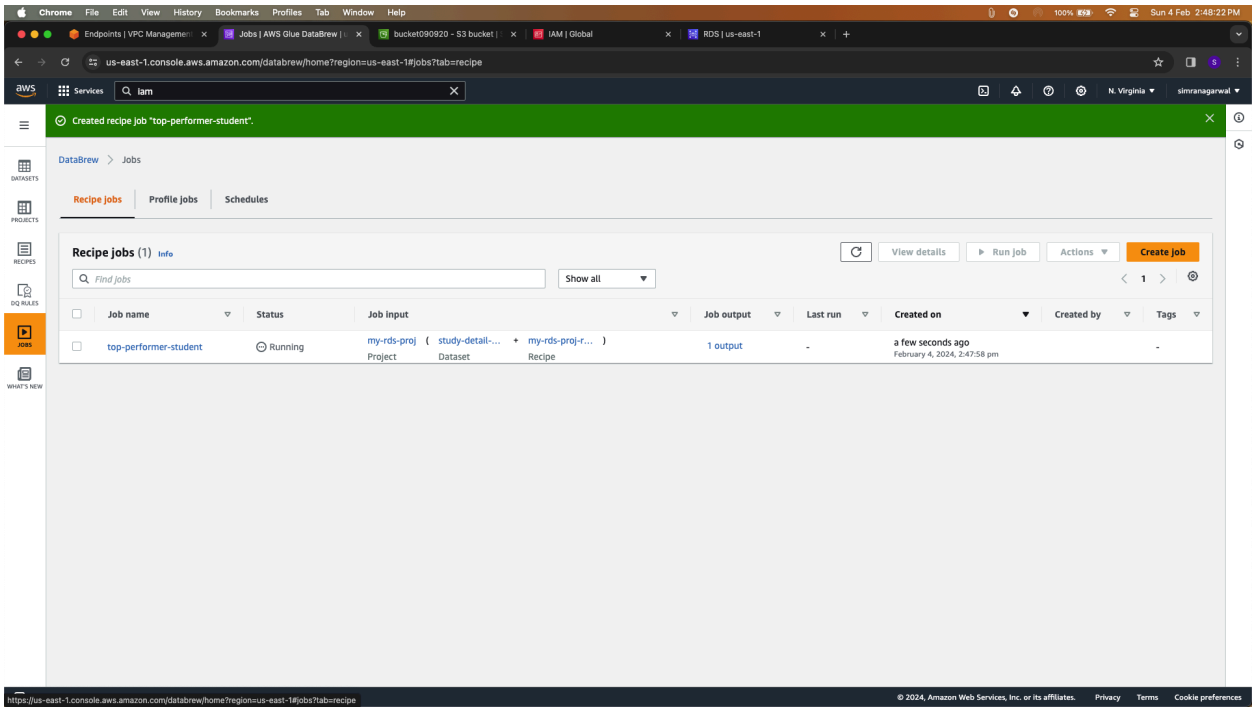


44. Now click on create a job and then select the output location as the location of the S3 bucket. After configuring all the settings for the job click on “**Create and run the job”**.

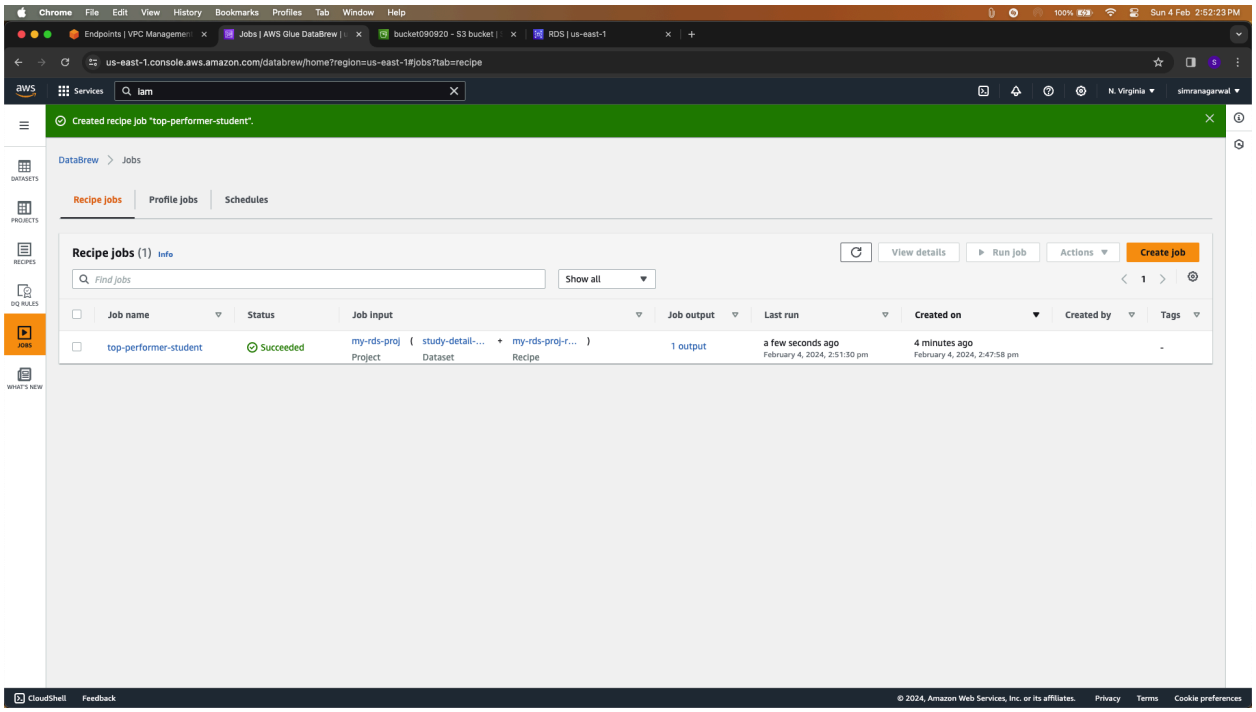


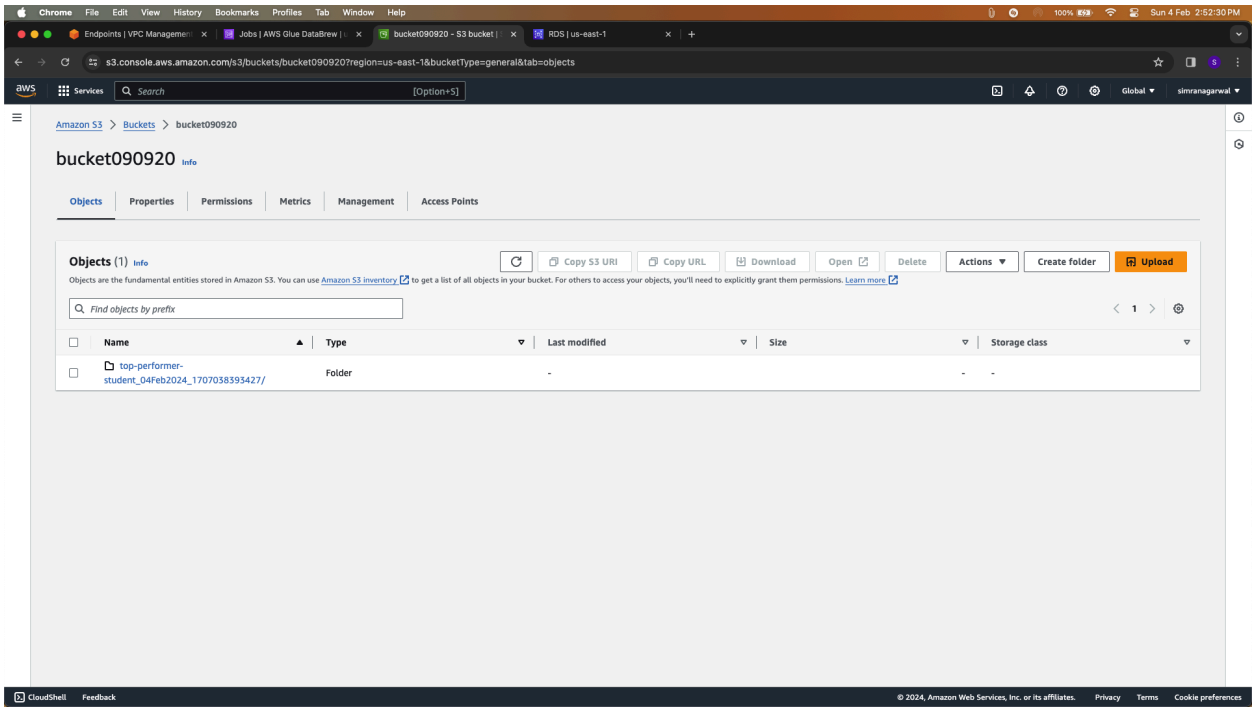



Created recipe job "top-performer-student".
my-rds-proj
Dataset: study-detail-dataset
Sample: First n sample (185 rows)
Run Job
Viewing 12 columns 185 rows
Recipe (4)
my-rds-proj-recipe
Working version
Applied steps (4)
1. Left join student-dataset
2. Left join school-dataset
3. Merge columns first_name, last_name into student_name separated by " "
4. Filter values by marks


Created recipe job "top-performer-student".
DataBrew > Jobs
Recipe jobs
Profile jobs
Schedules
Recipe jobs (1) Info
Create job
top-performer-student
Running
1 output
a few seconds ago

45. After you see the job running status as succeed, you should check the S3 Bucket whether the job hai created the resulting dataset in the output location (i.e. S3 Bucket) mentioned in the earlier steps.



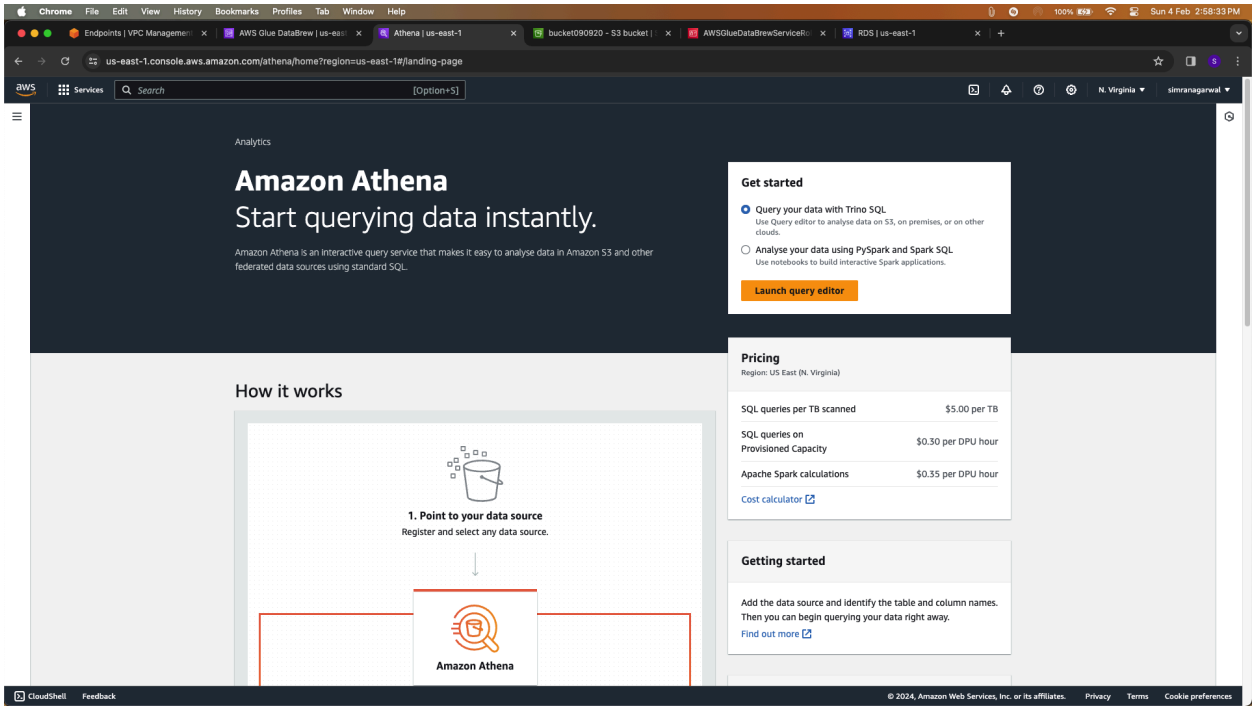


We have successfully created a resulting dataset in the S3 Bucket, now this S3 bucket will act as

an input source for the Athena to read and map data into its table and then use it for querying and analysis and provide the resulting data in another S3 Bucket.

46. Before starting with Amazon Athena we will be requiring an S3 Bucket for storing the output, so for that you need to create another S3 Bucket (follow the process from step number 41.)

47. Open Amazon Athena and click on Launch query editor.



First of all, in the settings click on settings and choose the output location as the newly created S3 Bucket.
So now, we have two S3 Buckets - The first one which was the output bucket for DataBrew will act as the input bucket for Athena and the freshly created S3 Bucket will act as output bucket for Athena.

48. Now click on Create and select S3 Bucket Data and then start populating the fields in the form starting with table and database name..

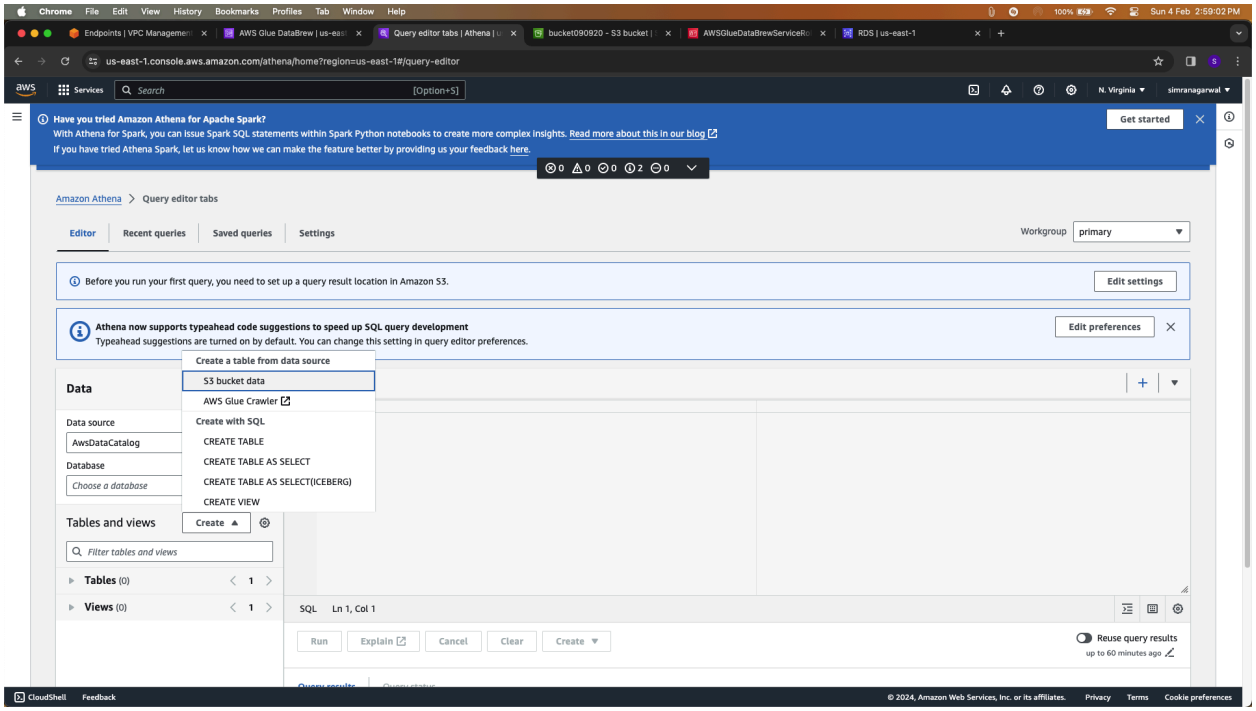
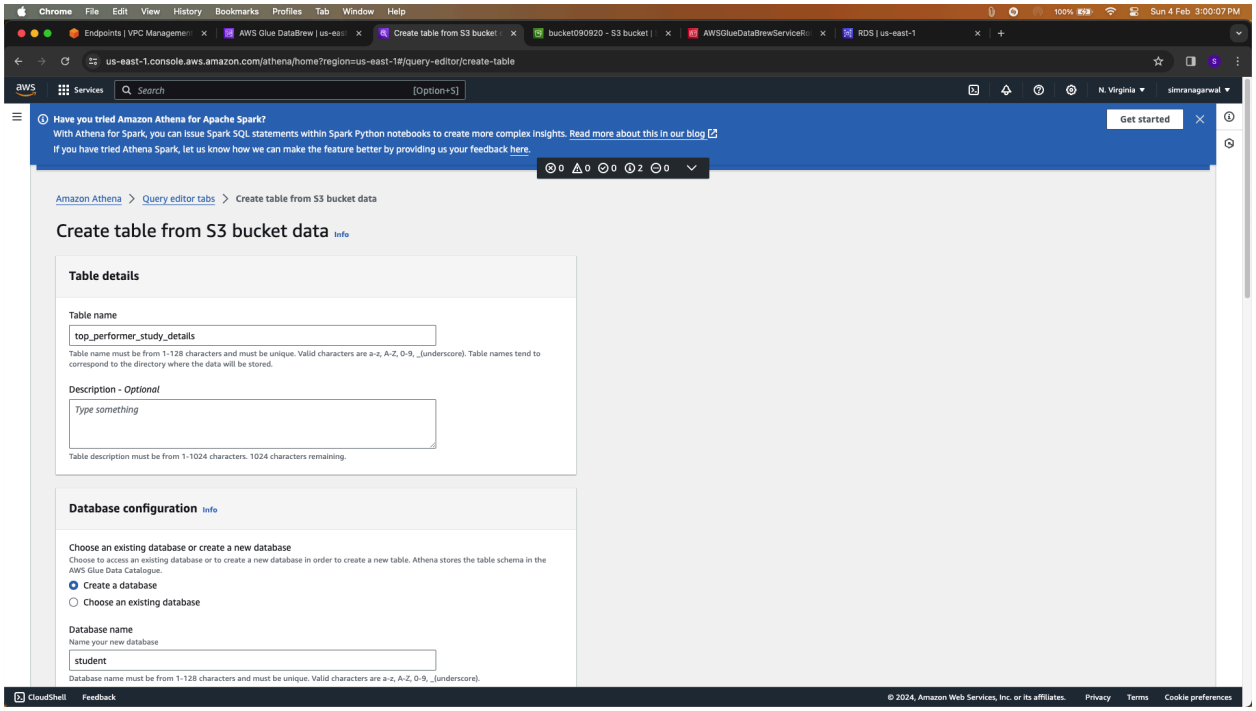
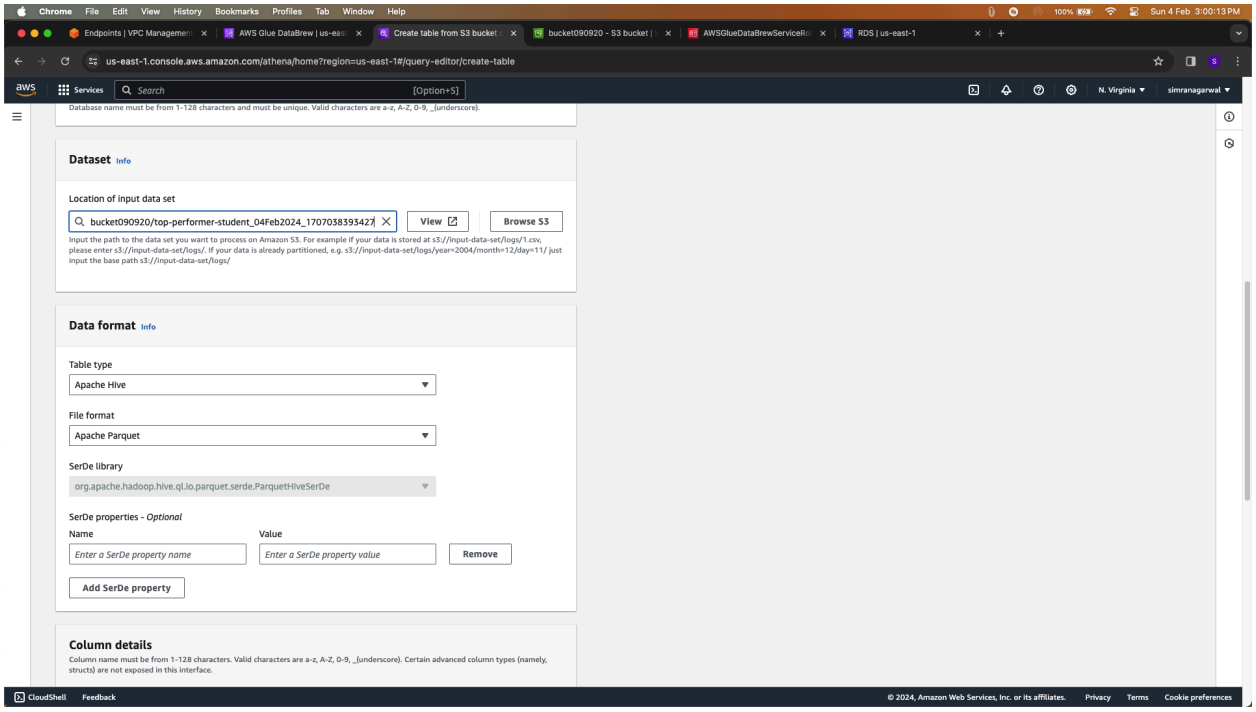


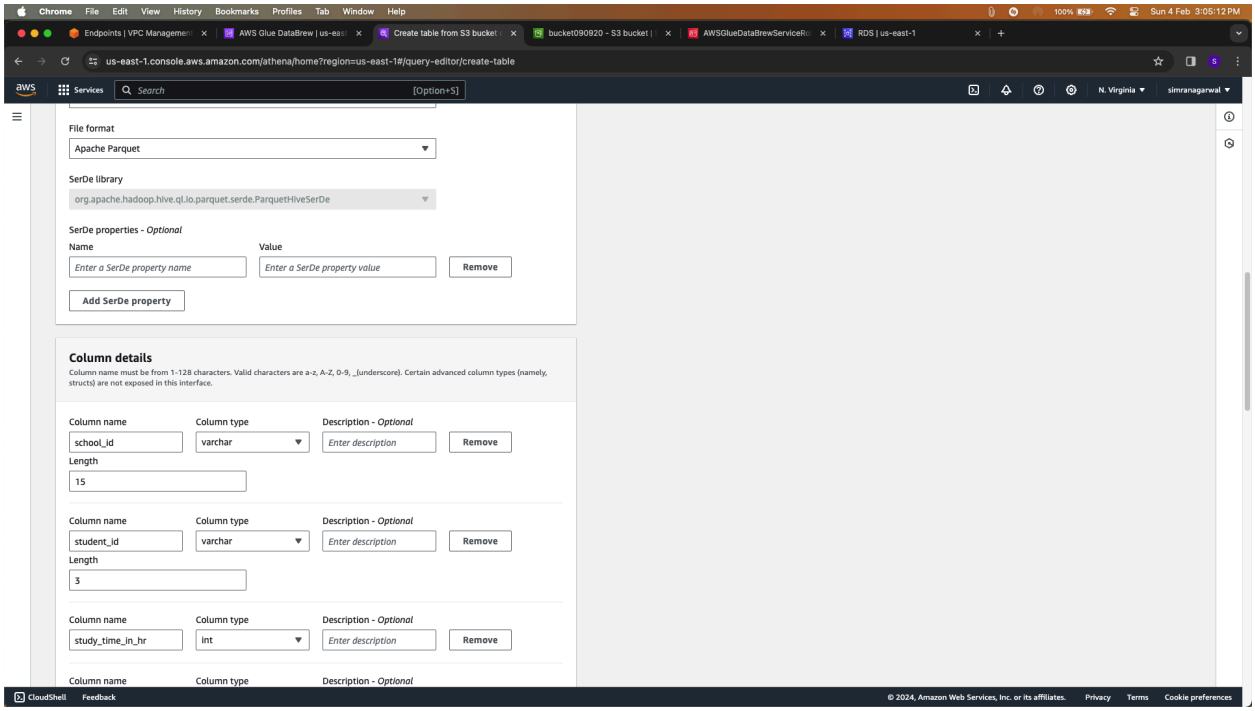
Table Name: top_performer_study_details, Database Name: student.

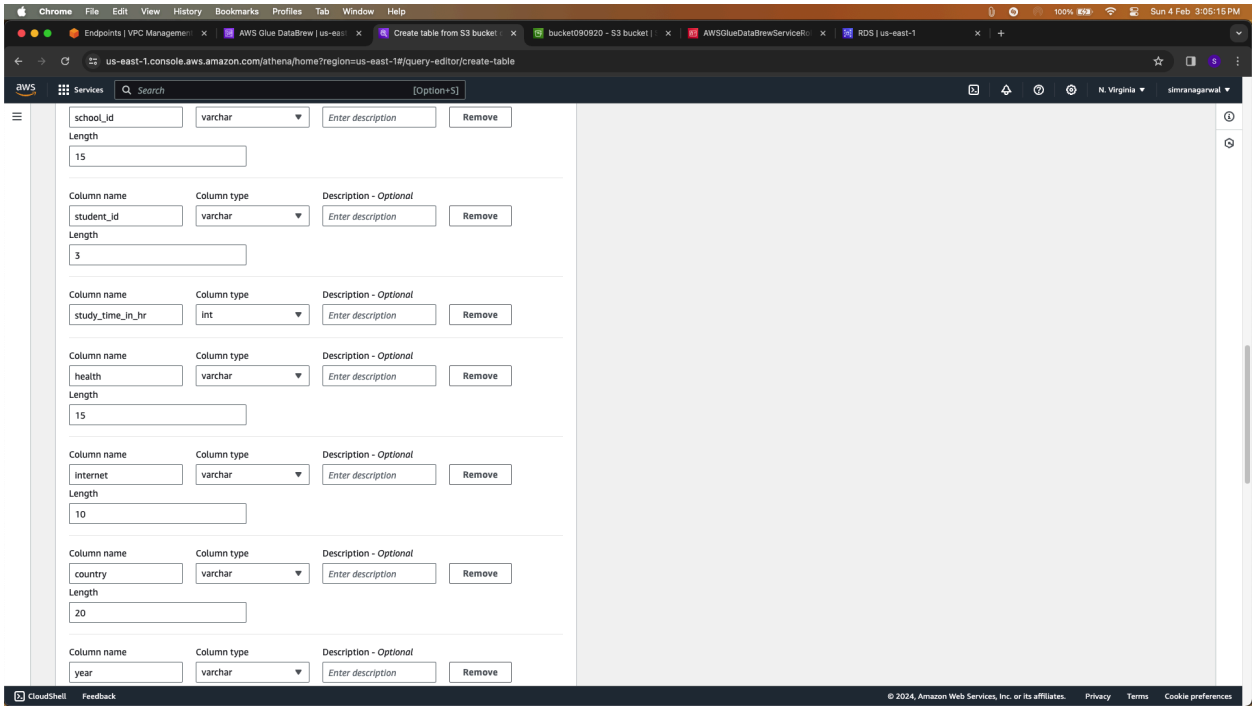


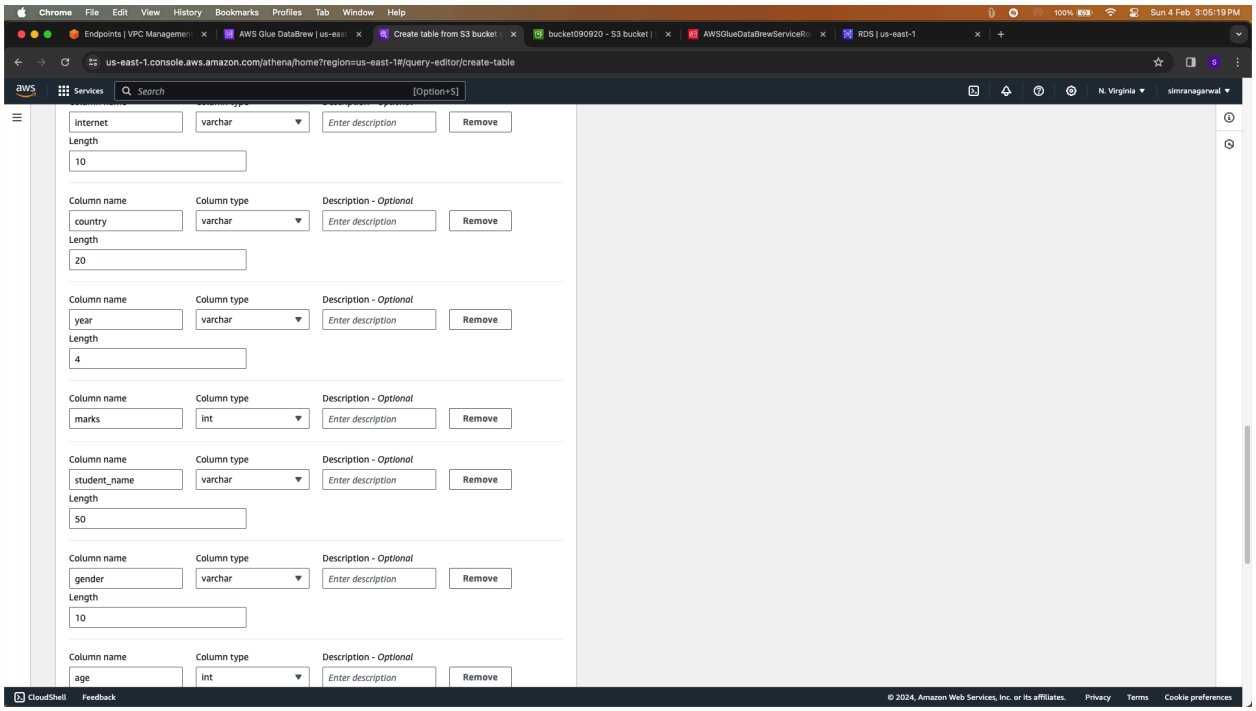
49. Choose the Dataset as the path for the first bucket which is now an input source for Athena.

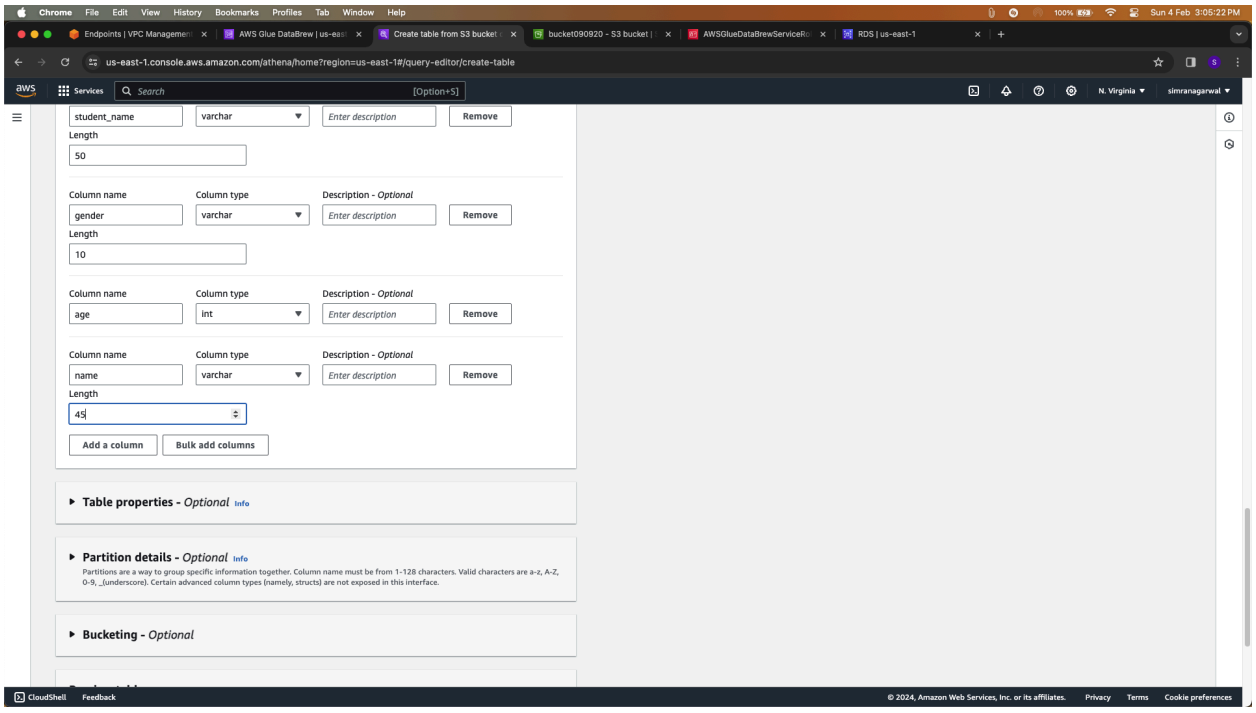


50. For creating the table, you need to add all the column with their datatypes and lengths.

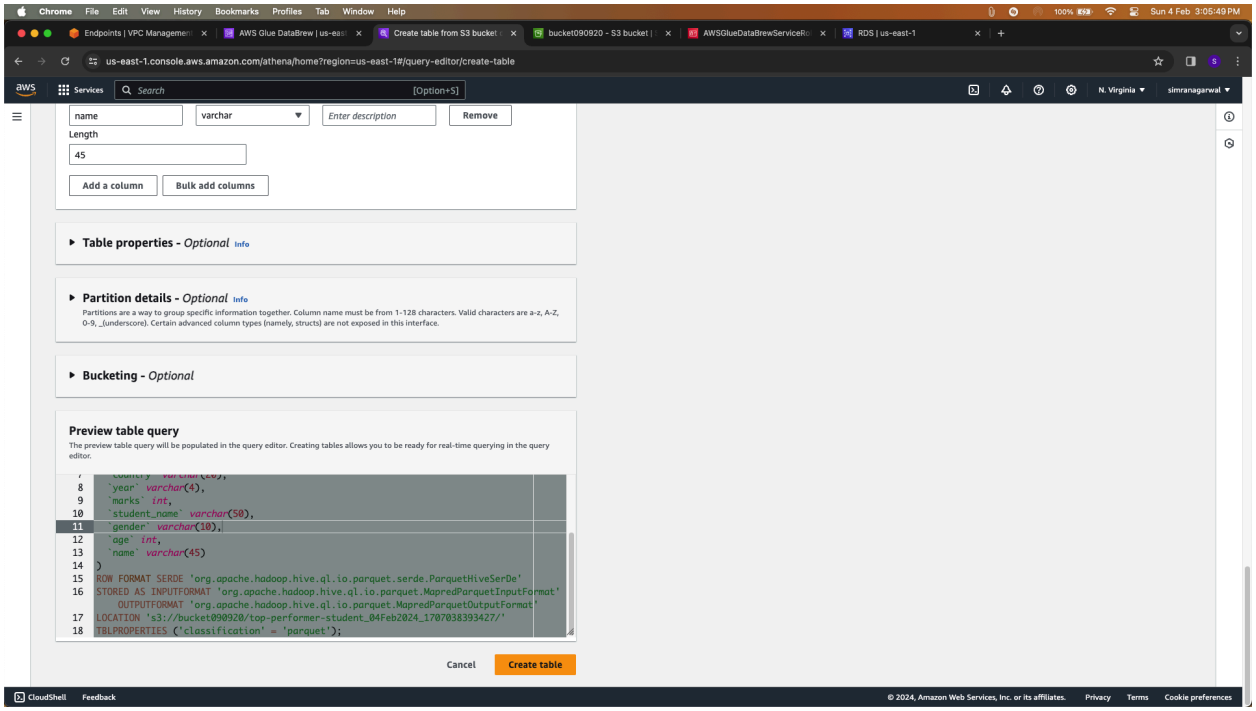






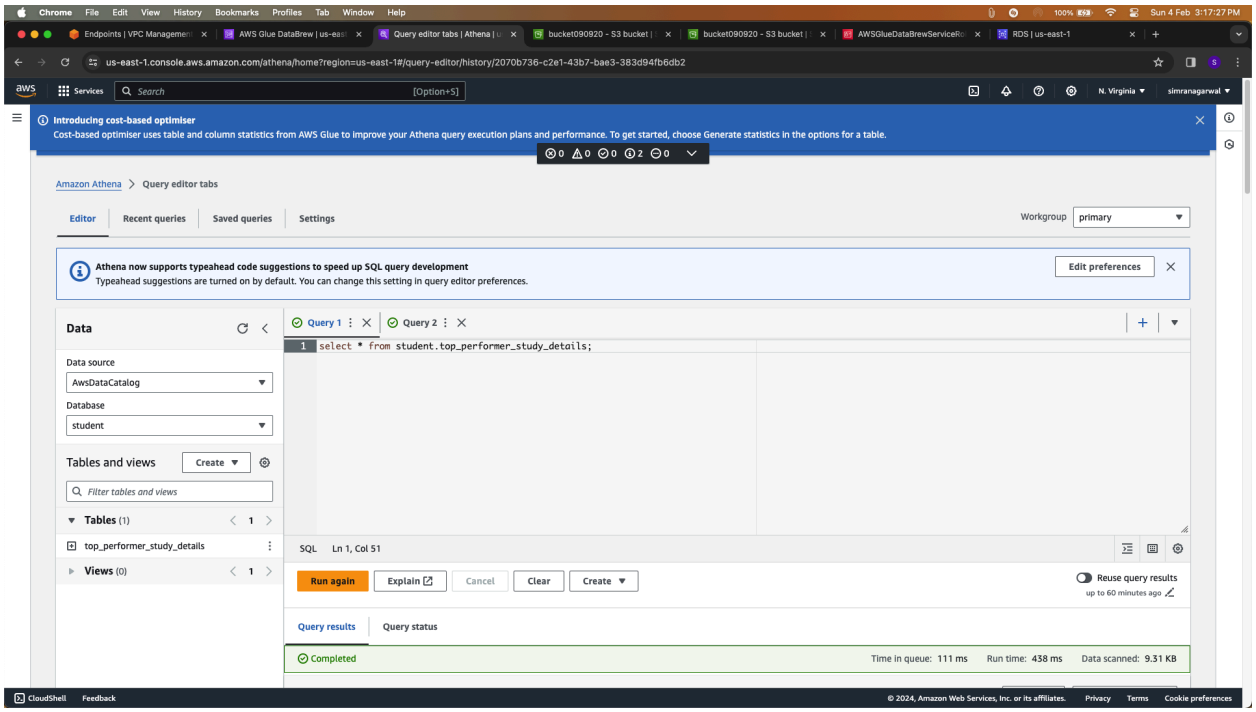


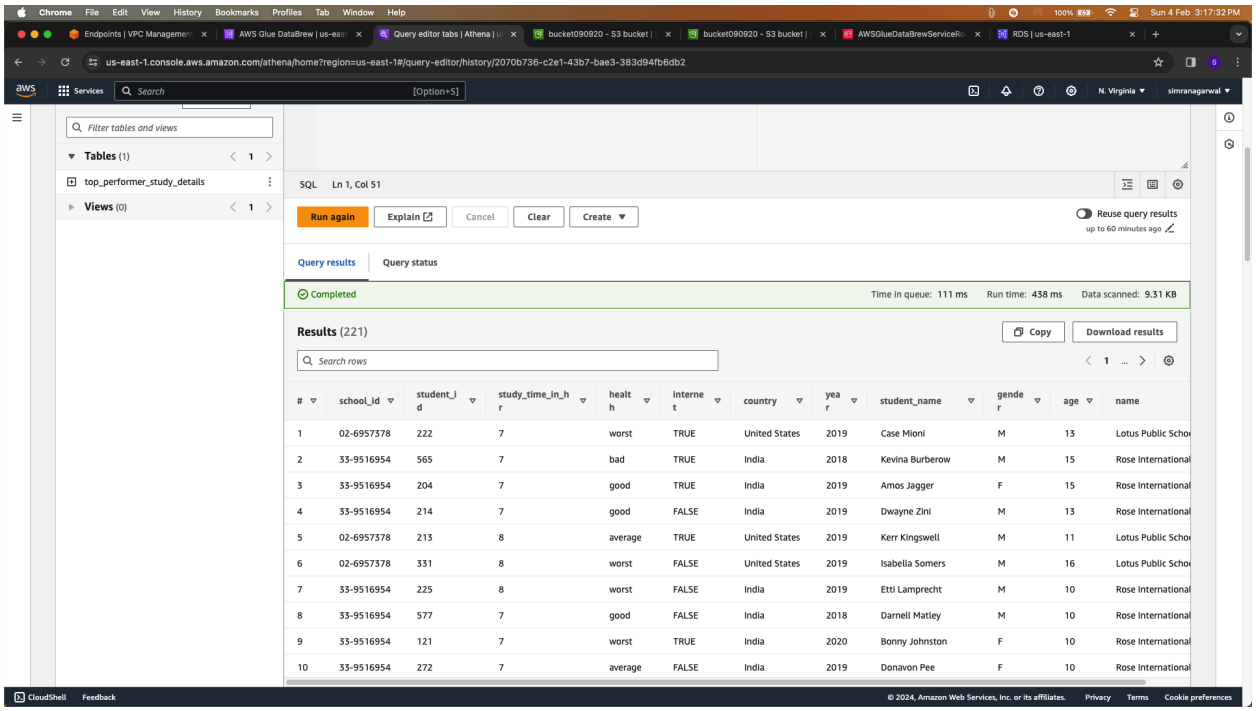
After adding all the columns with their datatypes and lengths click on create table.


name
varchar
Enter description
Remove
Length
45
Add a column
Bulk add columns
Table properties - Optional Info
Partition details - Optional Info
Partitions are a way to group specific information together. Column name must be from 1-128 characters. Valid characters are a-z, A-Z, 0-9, _(underscore). Certain advanced column types (namely, structs) are not exposed in this interface.
Bucketing - Optional
Preview table query
The preview table query will be populated in the query editor. Creating tables allows you to be ready for real-time querying in the query editor.
`year` varchar(4),
`marks` int,
`student_name` varchar(50),
`gender` varchar(10),
`age` int,
`name` varchar(45)
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION 's3://bucket090920/top-performer-student_04Feb2024_1707038393427/'
TBLPROPERTIES ('classification' = 'parquet');
Cancel
Create table

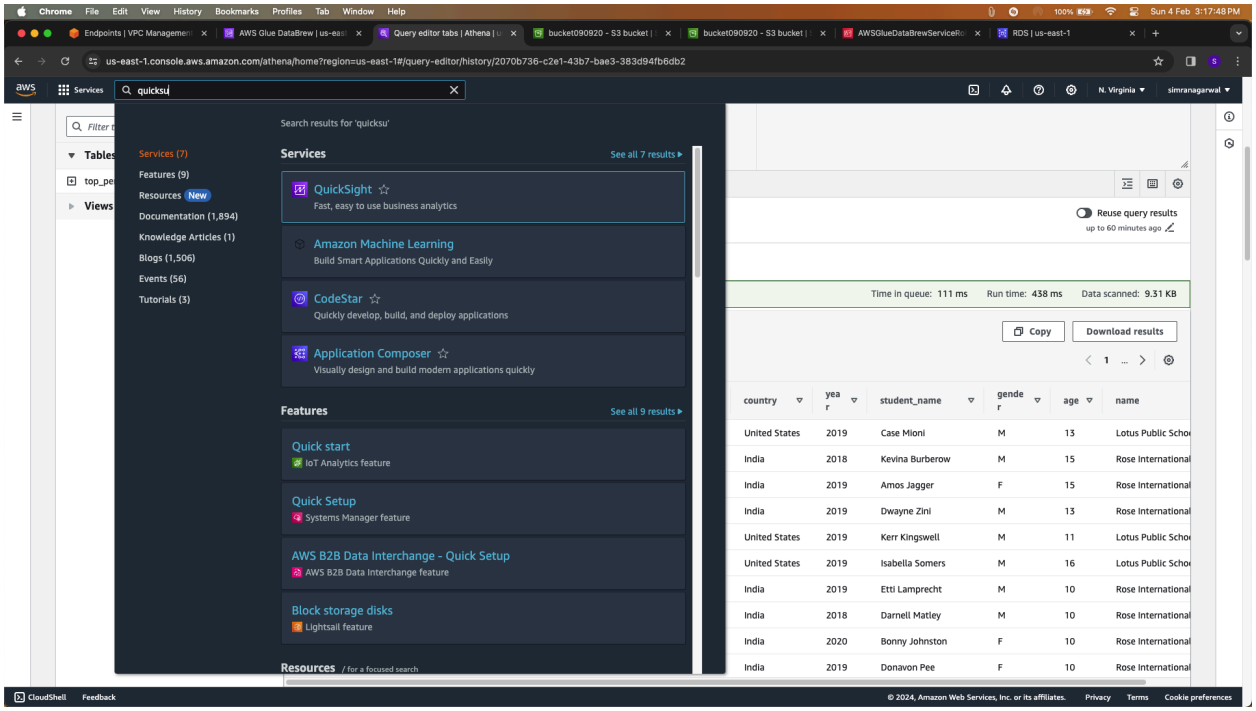
51. Now we can check whether we created the table correctly or not, this can be verified by running the following query:

“ select * from student.top_performer_study_details;”

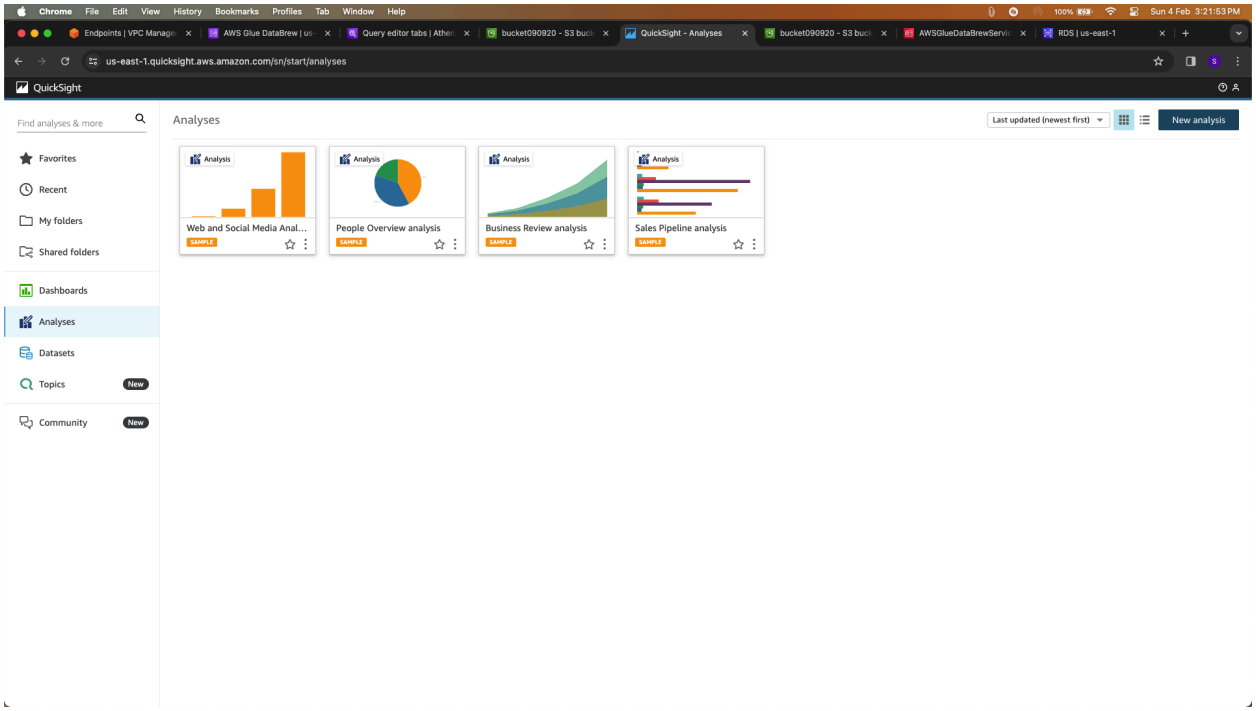


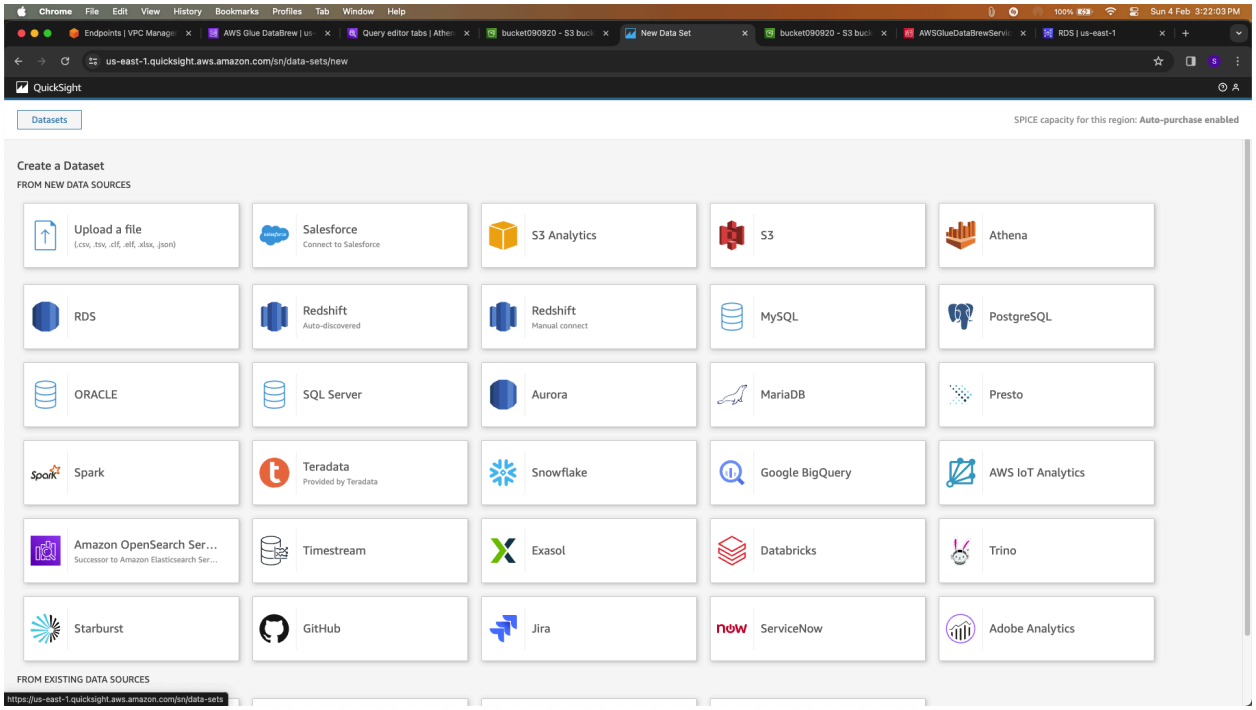


52. Now search for QuickSight in AWS Console, and then click on QuickSight.

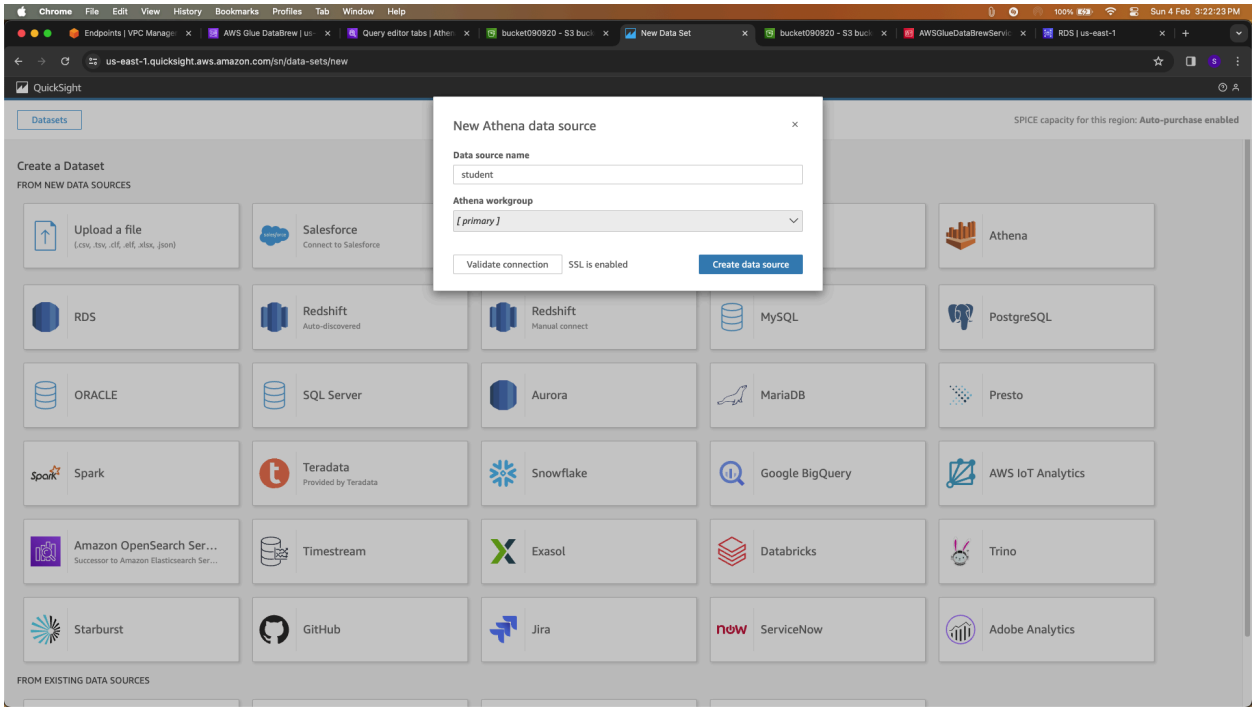


53. Now click on New Analysis, then click on Athena from the list of Datasets

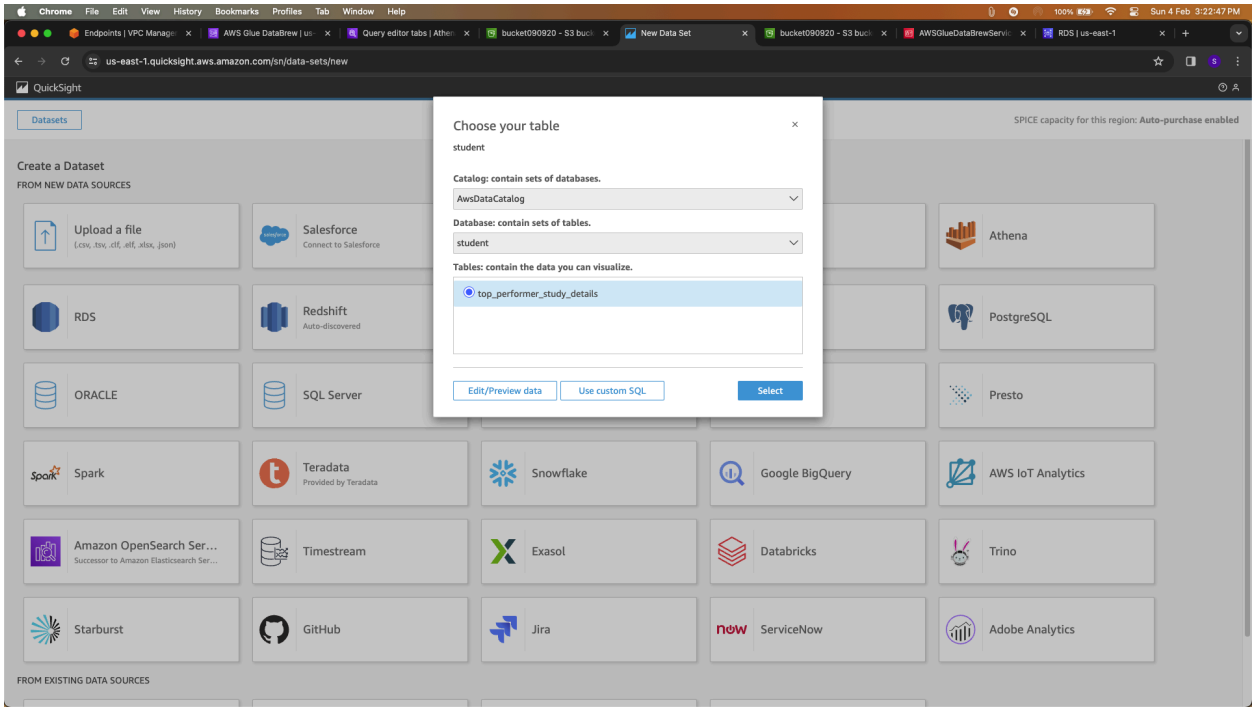


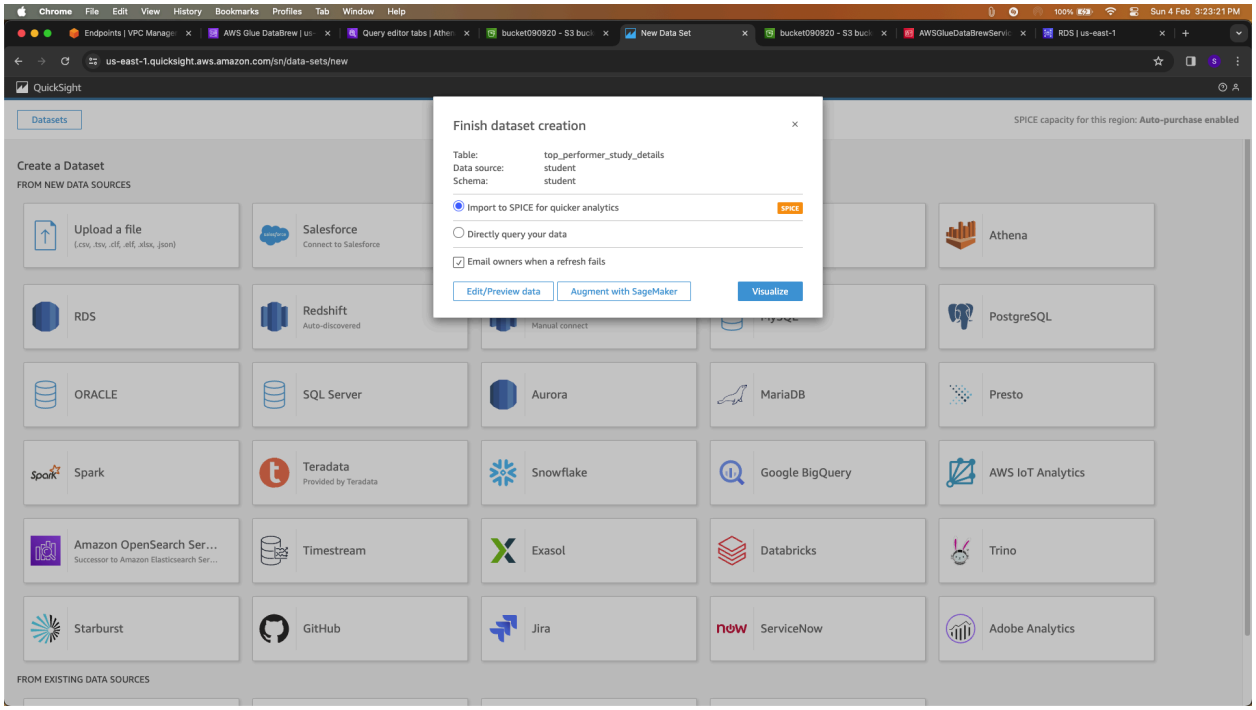


54. Enter the details like Data source name as “student” and Athena Workgroup as “primary”.



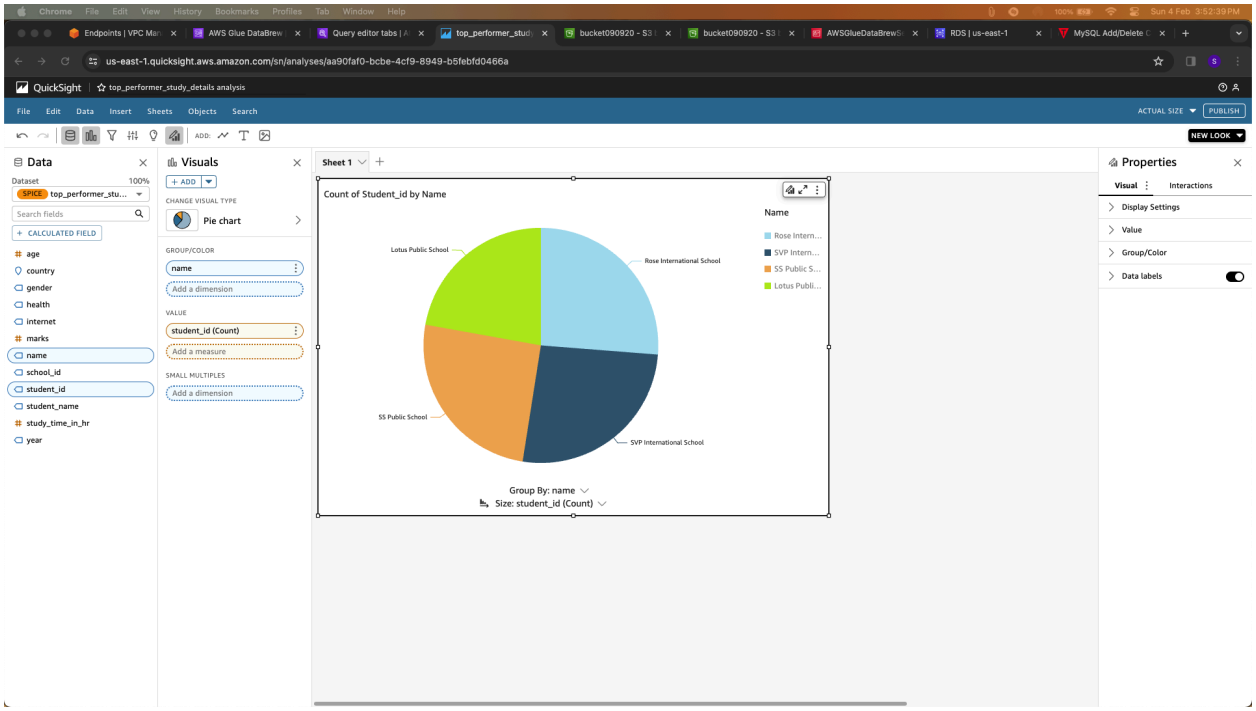
55. If we are doing it right we should see the table that we created in Athena (i.e. top_performer_study_details).

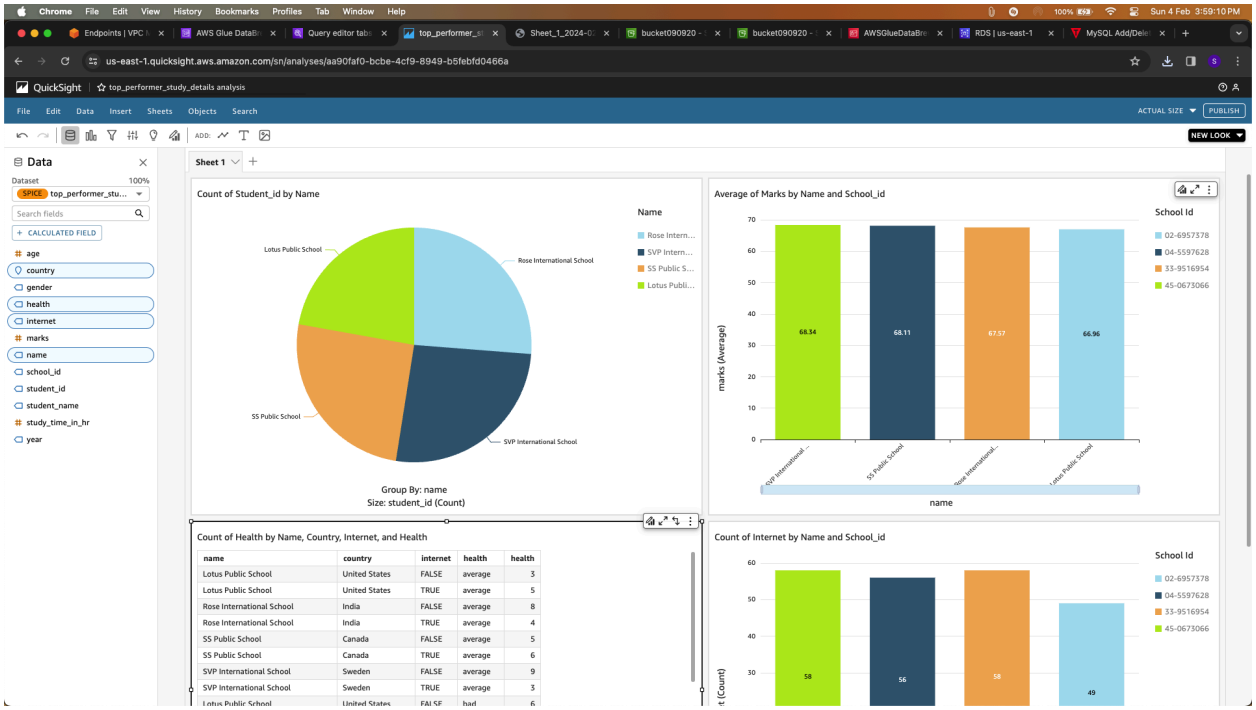




Now click on Visualize.

56. Now prepare the different charts considering different parameters for filtering, sorting and ordering.





Our Visualization is ready and now we can proceed towards cleanup of all the resources used for this project.

Chapter 6: Cost Analysis

To provide a monthly billing estimate for the implemented solution of data preparation using an Amazon RDS for MySQL database with AWS Glue DataBrew, we need to consider the costs associated with each component based on the assumed usage by 1000 users for a month.

1. **Amazon RDS for MySQL:**

Database Instance: The cost of running the RDS instance will depend on factors such as instance type, storage type, storage size, and region. Assuming a moderate instance type and storage size, the cost could range from approximately $100 to $500 per month.

Backup Storage: Backup storage costs may add an additional 10-20% to the RDS instance cost.

Data Transfer: Assuming moderate data transfer volumes, the cost could be around $10 to $50 per month.

2. **AWS Glue DataBrew:**

DataBrew Usage: The cost of using AWS Glue DataBrew is based on the amount of data processed and the duration of data preparation jobs. Assuming moderate usage by 1000 users, the monthly cost could range from approximately $500 to $1000.

Data Transfer Costs: Data transfer costs between Amazon RDS and AWS Glue DataBrew are minimal if both services are in the same AWS region. Assuming minimal cross-region data transfer, the cost could be negligible.

Optional Services: Additional costs may apply if intermediate data storage is utilized, such as storing data in Amazon S3 as an intermediate layer between RDS and Glue DataBrew. Costs for storing intermediate data could range from approximately $50 to $100 per month.

Monitoring and Logging: Costs for storing logs in CloudWatch Logs are typically minimal and may add around $10 to $20 per month.

Total Monthly Estimate:

Adding up the estimated costs for each component:

Amazon RDS for MySQL: $100 - $500

AWS Glue DataBrew: $500 - $1000

Data Transfer: $10 - $50

Optional Services (S3 storage): $50 - $100

Monitoring and Logging: $10 - $20

Total Monthly Estimate: $670 - $1670

Please note that these are rough estimates and actual costs may vary based on factors such as specific usage patterns, instance configurations, data volumes, and AWS pricing updates. It's recommended to use AWS Pricing Calculator or Cost Explorer to get more accurate estimates based on your specific requirements. Additionally, leveraging Reserved Instances or Savings Plans can help reduce costs for AWS services with predictable usage.

Chapter 7: Lessons and Observations

1. Data Pipeline Design: Designing a data pipeline that involves multiple AWS services (RDS, DataBrew, S3, Athena, QuickSight) requires careful planning to ensure smooth data flow and efficient processing. Understanding the capabilities and limitations of each service is crucial for designing an effective pipeline.

2. Resource Provisioning and Configuration: Provisioning and configuring resources such as RDS instances, VPC, subnets, NAT Gateway, and security groups need to be done thoughtfully to ensure security, scalability, and optimal performance. Proper configuration of access control and network settings is essential for protecting sensitive data and preventing unauthorized access.

3. Integration and Connectivity: Establishing connections between different services (e.g., RDS and DataBrew, DataBrew and S3) requires understanding of connectivity options (e.g., JDBC connection) and configuring appropriate access permissions. Ensuring seamless integration between services is critical for data transfer and processing.

4. Data Transformation and Analysis: Utilizing AWS DataBrew for data transformation tasks such as cleaning, filtering, and aggregating data simplifies the data preparation process. Leveraging tools like Athena for SQL-based analysis enables data analysts to perform complex queries and gain insights from large datasets efficiently.

5. Visualization and Reporting: QuickSight provides a user-friendly interface for creating interactive visualizations and dashboards based on data stored in Athena. Building insightful visualizations allows stakeholders to interpret data trends and make informed decisions.

6. Cost Management: Monitoring resource usage and optimizing costs is essential for managing AWS expenses effectively. Understanding the pricing models of different AWS services helps in estimating and controlling costs. Implementing cost-saving measures such as resource cleanup and rightsizing resources can lead to significant cost savings in the long run.

7. Resource Cleanup and Deletion: Proper cleanup of resources after completing a project is crucial for avoiding unnecessary charges and maintaining a clean AWS environment. Deleting unused resources such as RDS instances, S3 buckets, IAM roles, and QuickSight analyses helps in reducing costs and minimizing security risks.

8. Documentation and Best Practices: Documenting the entire process, including setup steps, configurations, and lessons learned, is valuable for future reference and knowledge sharing. Following AWS best practices and guidelines ensures that the solution is scalable, secure, and well-architected.

In conclusion, building a data pipeline on AWS involves various stages such as setup, configuration, integration, data processing, analysis, visualization, and cleanup. By following best practices and leveraging AWS services effectively, organizations can streamline data workflows, gain actionable insights, and drive business success.

Chapter 8: Conclusion

In conclusion, the article provides a structured approach to leveraging AWS services for student data analysis, encompassing data ingestion, transformation, analysis, and visualization. By following the outlined steps, users can effectively utilize AWS RDS, Glue Databrew, Athena, and QuickSight to gain insights from complex datasets. The emphasis on pre lab setup, including security considerations and resource cleanup, ensures a streamlined and cost-effective workflow. Overall, this use case demonstrates the power of cloud-based solutions in handling data-intensive tasks and driving informed decision-making in educational contexts.

Chapter 9: Summary

The provided article outlines a comprehensive use case involving the analysis of student data across schools using AWS services. Three datasets are utilized: school details, student details, and student study details. The workflow begins with setting up an RDS MySQL instance, establishing a JDBC connection, and creating a DataBrew project for data transformation. DataBrew is used to identify top-performing students, with the final output written to an S3 bucket. External tables are then created using Athena, enabling business users to perform BI reporting through QuickSight. The article also covers prerequisites, and pre lab setup steps including database creation, table setup, endpoint creation, IAM role assignment, and security group configuration. It further details the process of creating Glue Databrew connections, datasets, and projects, followed by data manipulation tasks such as joining, filtering, and creating resulting datasets. Additionally, S3 bucket creation, job execution, Athena configuration, and QuickSight setup are explained. The article concludes with a cleanup section, highlighting the importance of deleting all resources used in the project to avoid unnecessary costs.