

# **Migrate from PostgreSQL to Amazon RDS with AWS DMS**

## Introduction

We migrate a self-managed PostgreSQL database instance to a fully managed database on [Amazon Relational Database Service](#) (Amazon RDS). First, you learn why you would want to use Amazon RDS to manage your relational database. Then, you work through the steps to migrate an existing PostgreSQL database to Amazon RDS. At the end of this lesson, you should feel confident in your ability to migrate an existing database to Amazon RDS.

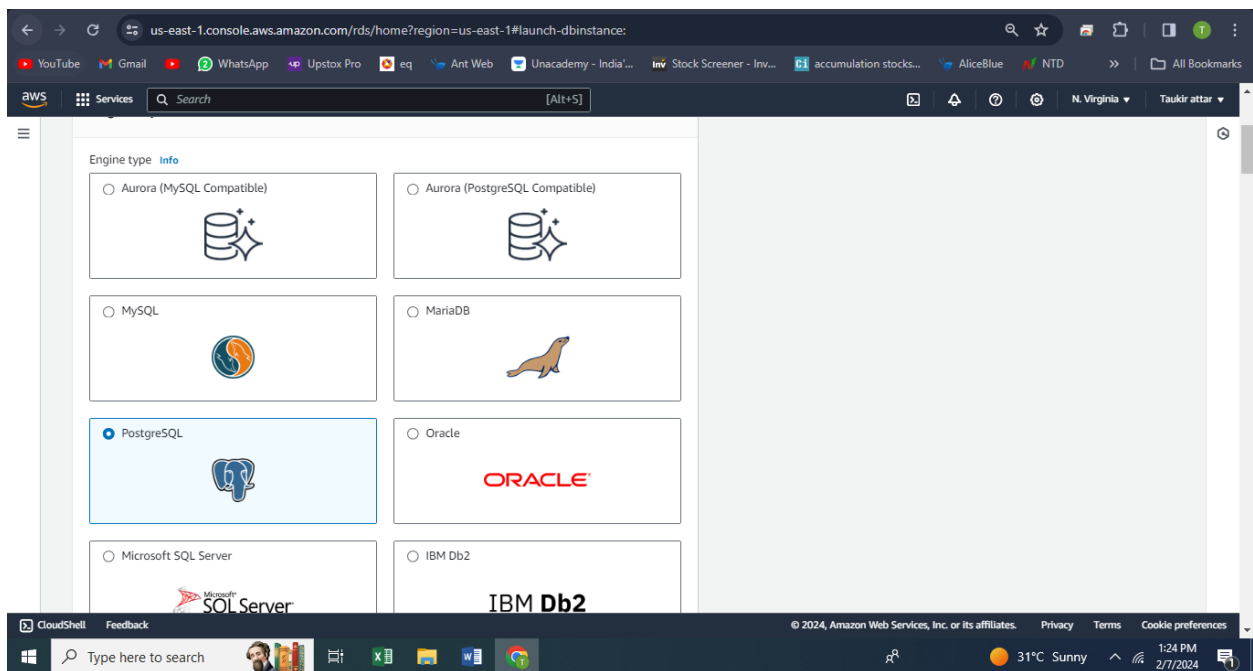
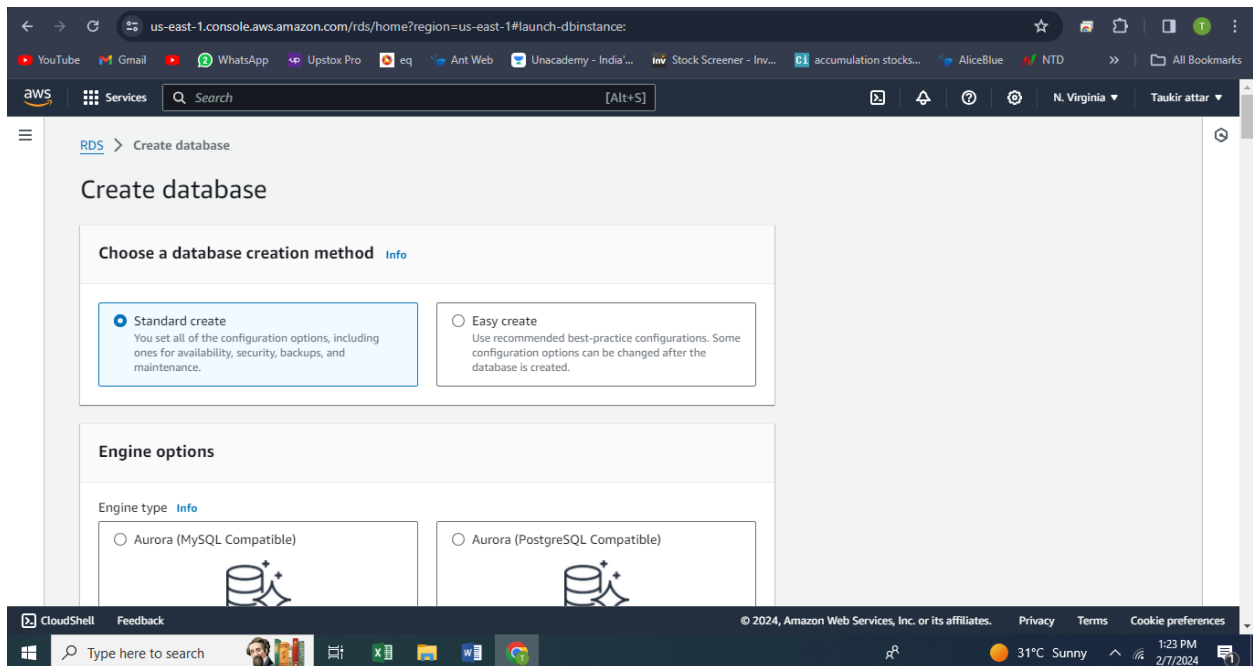
## Why use Amazon RDS?

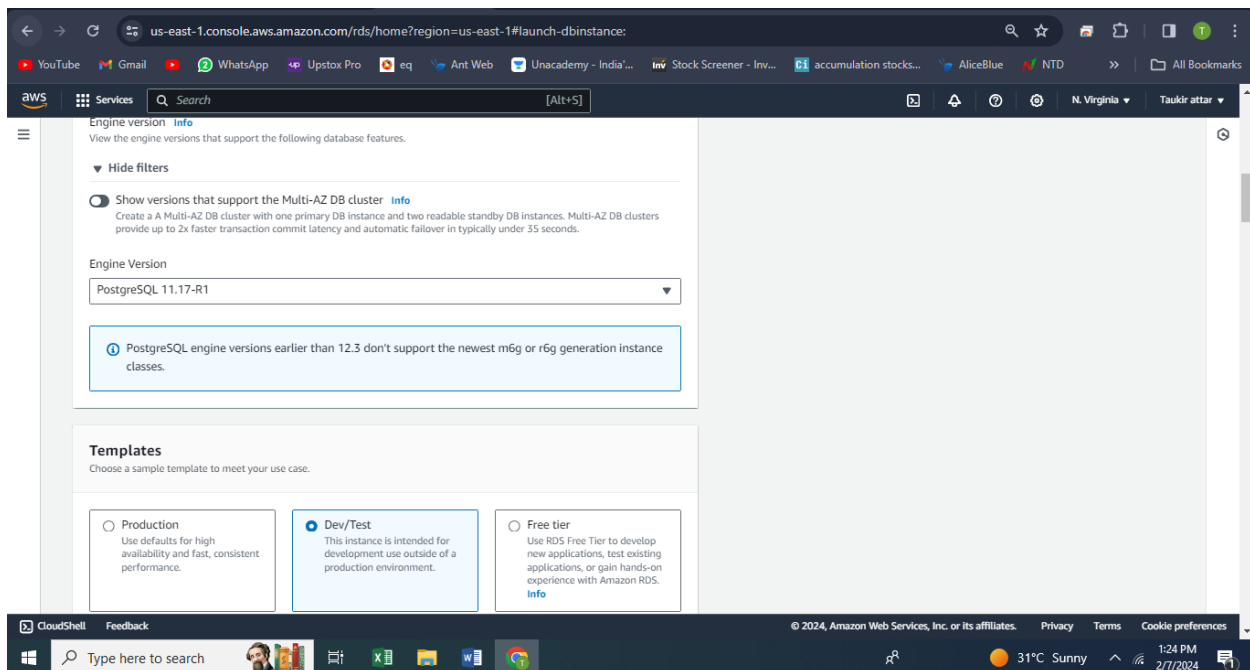
Amazon RDS is a fully managed service for running relational databases with AWS. It has support for six different database engines, including open-source options such as MySQL, PostgreSQL, and MariaDB.

With Amazon RDS, your database operations are managed by AWS, leaving your team free to focus on innovation. Amazon RDS handles instance failover, data backups, and software updates. Rely on the efficiencies of the AWS Cloud to use a faster, cost-efficient, and more reliable database option.

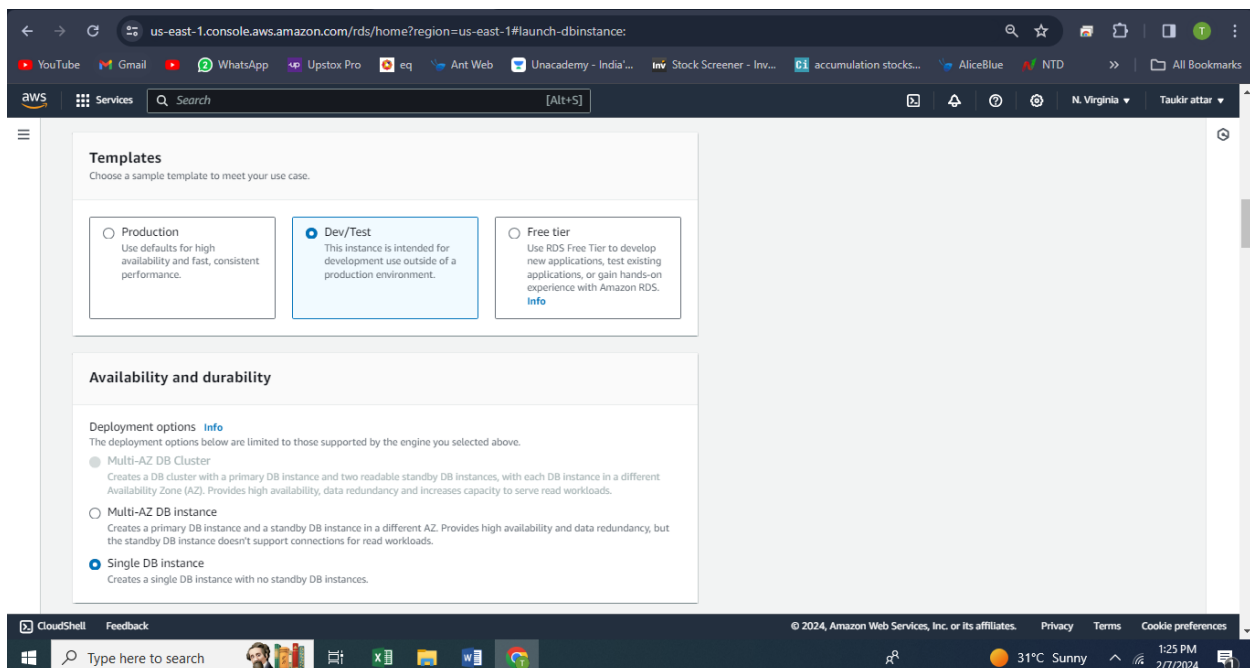
create a PostgreSQL database instance in Amazon RDS

- choose Create database to create a new database.
- This initiates the database creation wizard. In the Engine options section, choose PostgreSQL as your Engine type. Then, choose the version of PostgreSQL you want to use. Note that AWS DMS supports PostgreSQL versions 9.4 and later (9.x), 10.x, and 11.x.
- The database creation wizard includes templates to make it easier to configure the settings for your Amazon RDS database. If you are creating this database to be used in production, you should choose the production template.





Next, decide whether to create a standby instance. A standby instance is a replica of your data that is available in the event of failure. It is located in a different Availability Zone in the same AWS Region as your Amazon RDS database to limit the impact of failures within a single data center. If you are running a production database where uptime is essential, a standby instance is recommended.



–In the Settings section, give your database a name and set the master username and password. Do not autogenerate your password for this lab, and make sure you write down your password. You need the password to connect to your database and create additional users.

The screenshot displays the AWS Management Console for an Amazon RDS database instance. The browser address bar shows the URL: `us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance:`. The console interface includes a top navigation bar with the AWS logo, a search bar, and a list of services. The main content area is divided into two sections: 'Settings' and 'Instance configuration'.

**Settings section:**

- DB instance identifier:** A text input field containing 'my-database-1'. Below the field, a note states: 'The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.'
- Credentials Settings:**
  - Master username:** A text input field containing 'postgres'. Below the field, a note states: '1 to 16 alphanumeric characters. The first character must be a letter.'
  - ☐ **Manage master credentials in AWS Secrets Manager**  
Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.
- A blue information box at the bottom of the Credentials Settings section states: 'If you manage the master user credentials in Secrets Manager, some RDS features aren't supported. [Learn more](#)'

**Instance configuration section:**

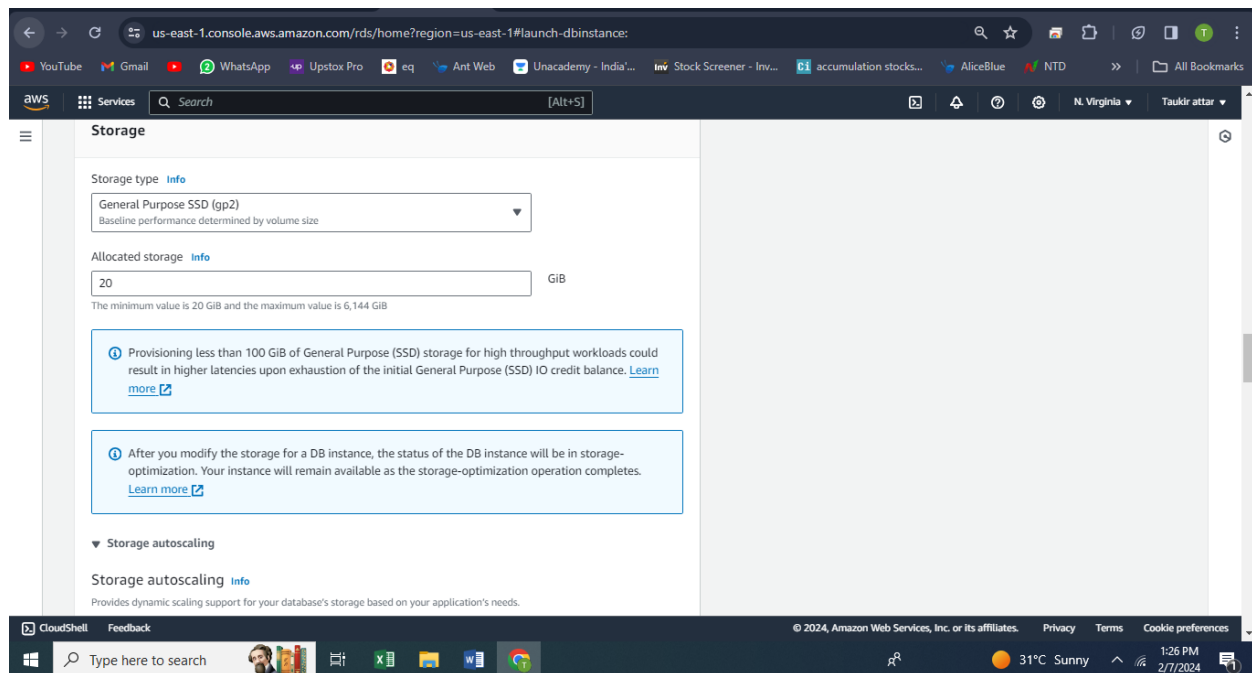
The DB instance configuration options below are limited to those supported by the engine that you selected above.

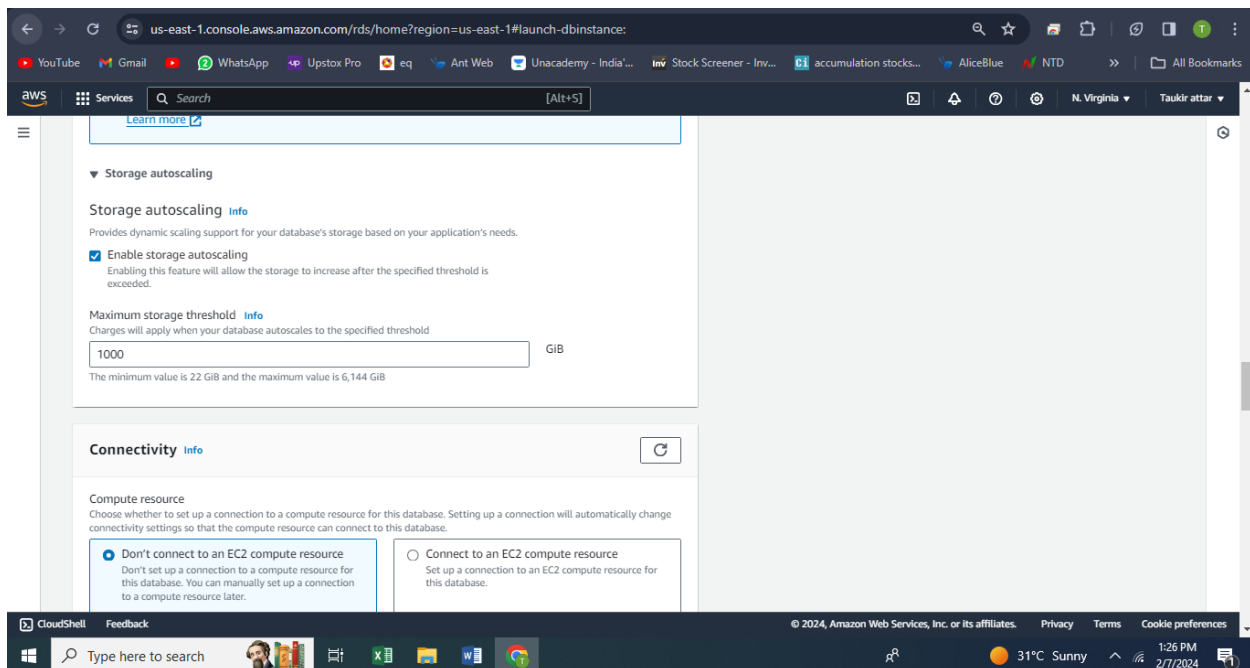
- DB instance class:**
  - A 'Hide filters' button is present.
  - A radio button labeled 'Include previous generation classes' is selected.
  - Three radio buttons are listed:
    - ☐ Standard classes (includes m classes)
    - ☐ Memory optimized classes (includes r and x classes)
    - ☒ Burstable classes (includes t classes)
  - A dropdown menu shows 'db.t3.micro' with specifications: '2 vCPUs 1 GiB RAM Network: 2,085 Mbps'.

Next, configure the storage options for your Amazon RDS database. There are two storage options in Amazon RDS: general purpose and provisioned

IOPS, or I/O operations per second. With general-purpose storage, you receive 3 IOPS per GiB of storage allocated. Thus, 20 GiB of storage would have 300 IOPS. Additionally, you receive burst capacity up to 3,000 IOPS.

With provisioned storage, you provision IOPS separately from your storage capacity. This allows you to tune your storage and operations settings to fit your needs. Additionally, you can enable storage autoscaling. With storage autoscaling, Amazon RDS automatically increases your storage capacity as your database is close to running out of free disk space.

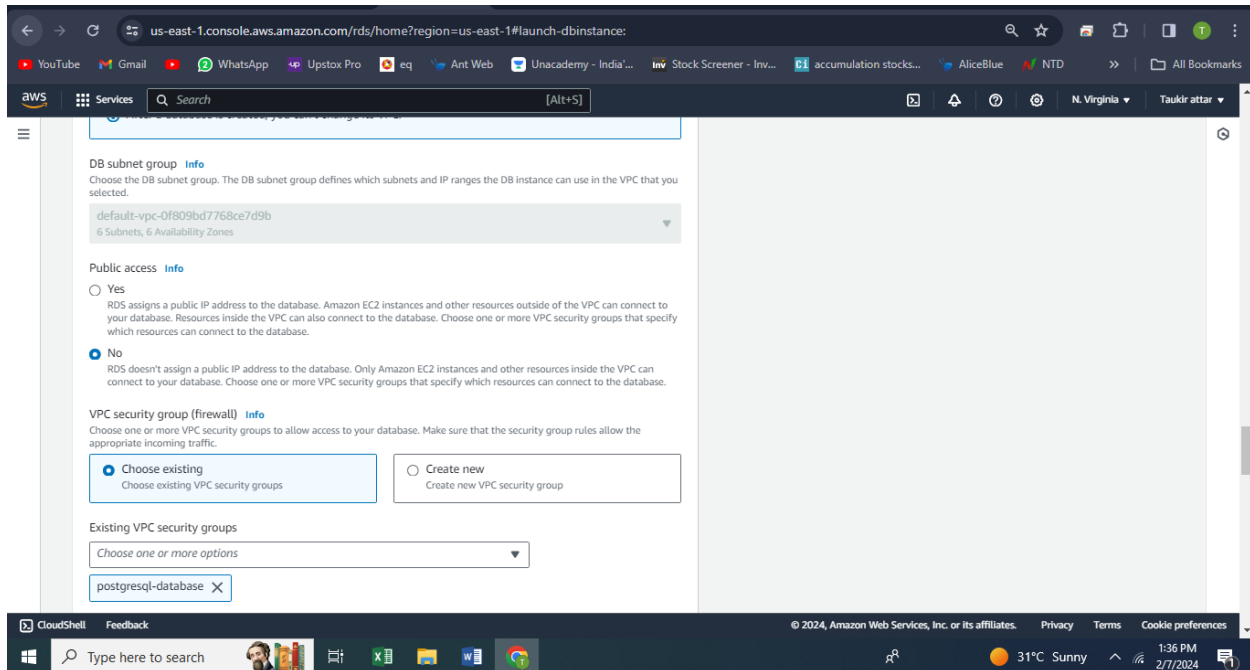




The next section in the Amazon RDS database creation wizard is about connectivity. You must specify the Amazon Virtual Private Cloud (Amazon VPC) in which your database resides, as well as the network subnet and security groups for your database instance.

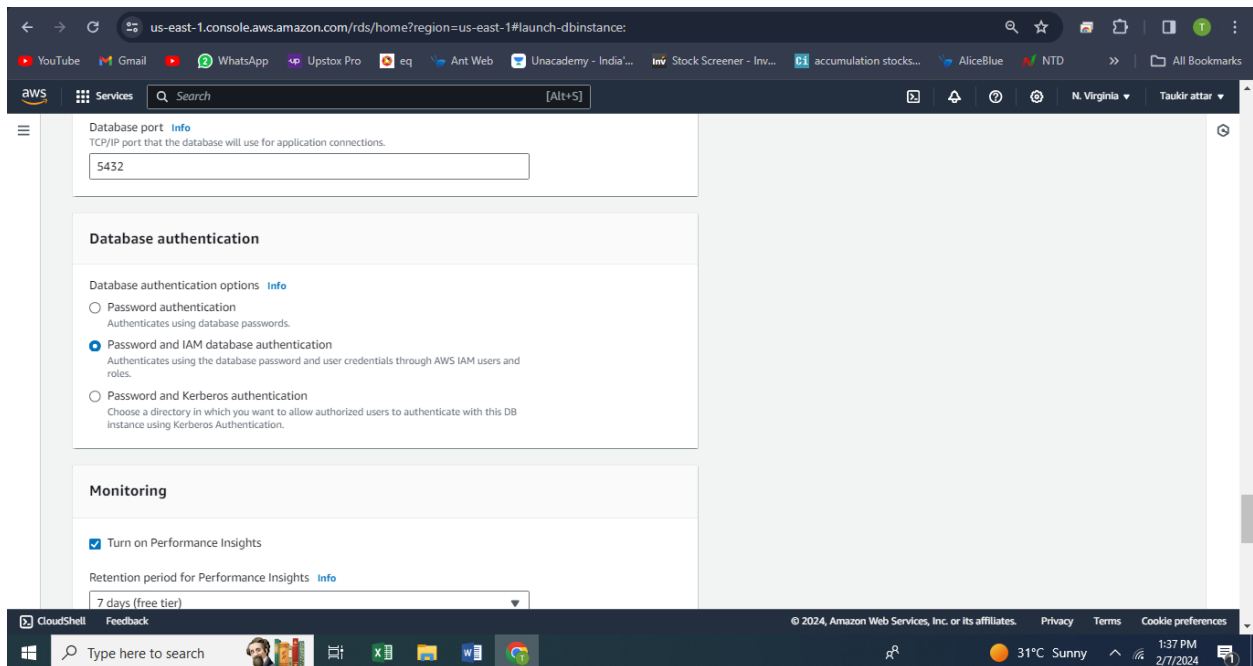
If you are migrating from a self-managed database instance on Amazon EC2, you can use the same Amazon VPC and security groups as your existing database.

If you are migrating from a database that is not hosted on AWS but your application is hosted on AWS, choose the same Amazon VPC that is used for your application. Then create a new security group for your database instance.

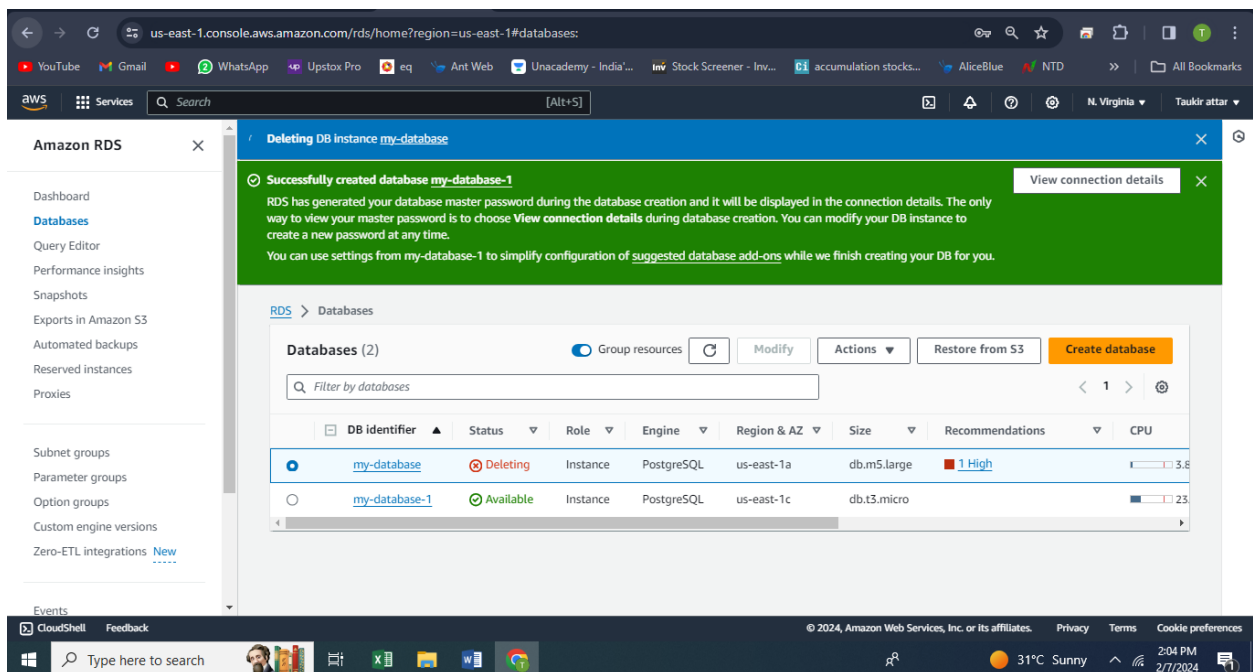


Finally, choose the database authentication methods to allow in your database. PostgreSQL databases traditionally allow for username and password authentication. With PostgreSQL on Amazon RDS, you also can choose to allow for authentication using AWS Identity and Access Management (IAM). This integrates easily with your application compute and removes the need for credential rotation.

We recommend that you allow for both password and IAM database authentication. If you only want to start with password authentication, you may add IAM database authentication later, but it will result in some downtime for your Amazon RDS database instance.



The end of the database creation wizard shows you the estimated monthly costs for your database instance. Choose Create database to create your database instance



created a fully managed, production-ready PostgreSQL database instance in Amazon RDS



We are creating a replication instance in AWS DMS.

- AWS DMS is a service that copies data from an existing database into a fully managed database on AWS. A replication instance is an Amazon EC2 instance that can host replication tasks within AWS DMS. In the next module, you will create endpoints.
  - Before using AWS DMS to migrate your existing PostgreSQL database to a PostgreSQL database in Amazon RDS, you should review the AWS DMS documentation about using a PostgreSQL database as a source and using a PostgreSQL database as a target. In certain cases, it may make sense to use native tools such as `pg_dump` and `pg_restore` to migrate your database.
1. To create a replication instance, go to the Replication Instances section of the AWS DMS console. Choose Create replication instance to begin the replication instance creation wizard.
  2. In the Replication instance configuration section, give your replication instance a name and description. Then choose your instance class. The instance class you use depends on the size of your existing database and the amount of data flowing through it.
  3. Then choose an engine version for AWS DMS. Finally, choose the amount of allocated storage for your replication instance

As you continue in the Replication instance configuration section, you need to choose a VPC for your replication instance. Choose the same VPC in which you provisioned your Amazon RDS database to ease network access for the replication instance.

us-east-1.console.aws.amazon.com/dms/v2/home?region=us-east-1#createReplicationInstance

YouTube Gmail WhatsApp Upstox Pro eq Ant Web Unacademy - India... Stock Screener - Inv... accumulation stocks... AliceBlue NTD All Bookmarks

aws Services Search [Alt+S]

DMS > Replication Instances > Create replication instance

## Create replication instance [info](#)

### Settings

**Name**  
The name must be unique among all of your replication instances in the current AWS region.

Replication instance name must not start with a numeric value

**Descriptive Amazon Resource Name (ARN) - optional**  
A friendly name to override the default DMS ARN. You cannot modify it after creation.

**Description - optional**

The description must only have unicode letters, digits, whitespace, or one of these symbols: \_/+=+@. 1000 maximum character.

### Instance configuration [info](#)

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search

Near record 2:13 PM 2/7/2024

us-east-1.console.aws.amazon.com/dms/v2/home?region=us-east-1#createReplicationInstance

YouTube Gmail WhatsApp Upstox Pro eq Ant Web Unacademy - India... Stock Screener - Inv... accumulation stocks... AliceBlue NTD All Bookmarks

aws Services Search [Alt+S]

DMS > Replication Instances > Create replication instance

## Create replication instance [info](#)

### Settings

**Name**  
The name must be unique among all of your replication instances in the current AWS region.

Replication instance name must not start with a numeric value

**Descriptive Amazon Resource Name (ARN) - optional**  
A friendly name to override the default DMS ARN. You cannot modify it after creation.

**Description - optional**

The description must only have unicode letters, digits, whitespace, or one of these symbols: \_/+=+@. 1000 maximum character.

### Instance configuration [info](#)

**Instance class [info](#)**

dms.t3.medium  
2 vCPUs 4 GiB Memory

☐ Include previous-generation instance classes

**Engine version**  
Choose an AWS DMS version to run on your replication instance. For more details, see the [AWS DMS release notes](#). For information about DMS version support, see [AWS DMS support lifecycle policy](#).

3.4.6

☐ Include Beta DMS versions

**High Availability [info](#)**  
The Multi-AZ option deploys a primary replication instance in one Availability Zone (AZ) and a standby in another AZ. The Single-AZ option deploys a single replication instance in one AZ. Billing is based on DMS pricing.

Dev or test workload (Single-AZ)

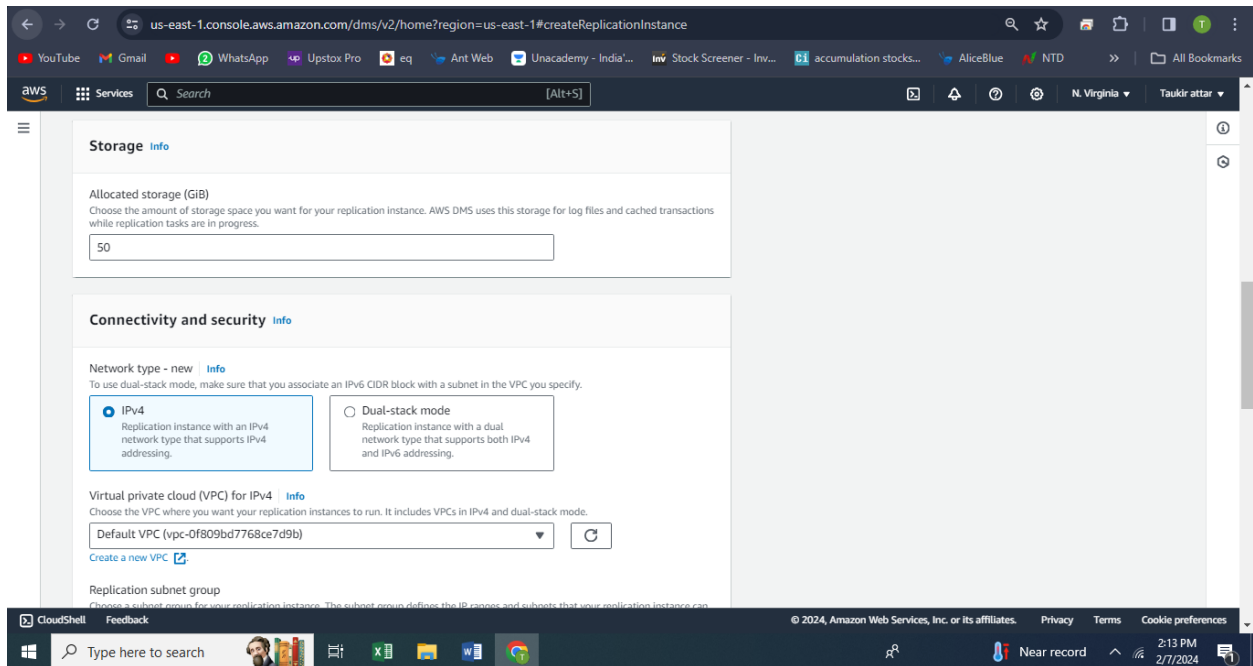
### Storage [info](#)

CloudShell Feedback

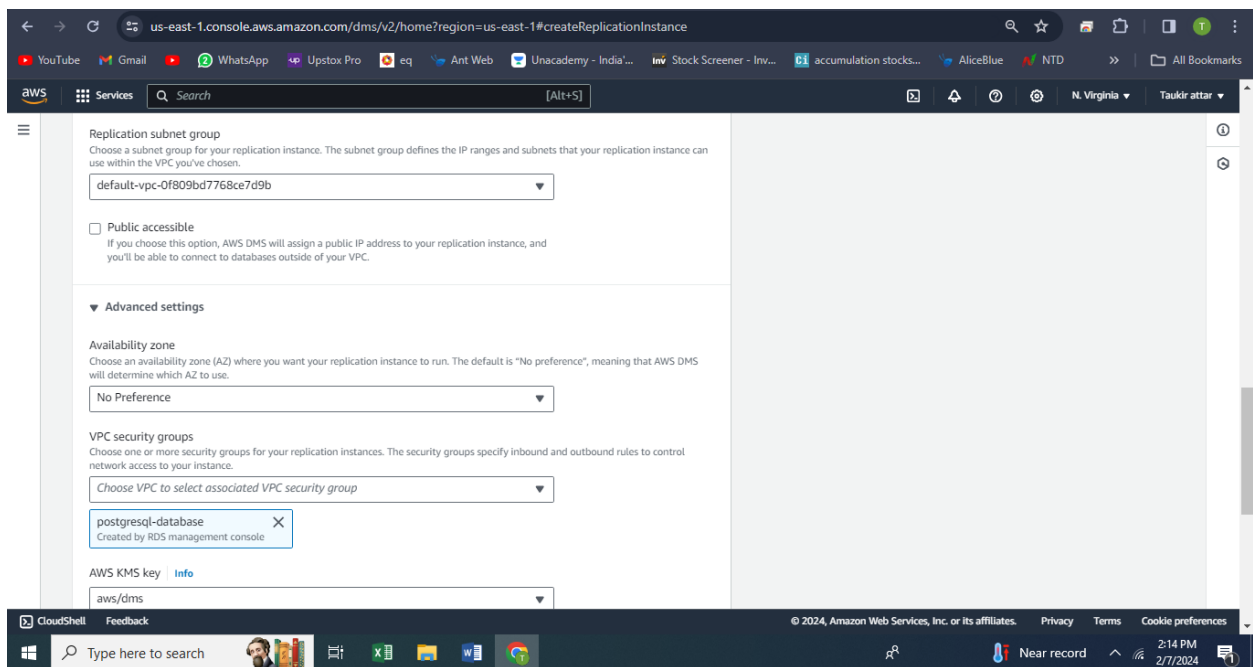
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search

Near record 2:13 PM 2/7/2024

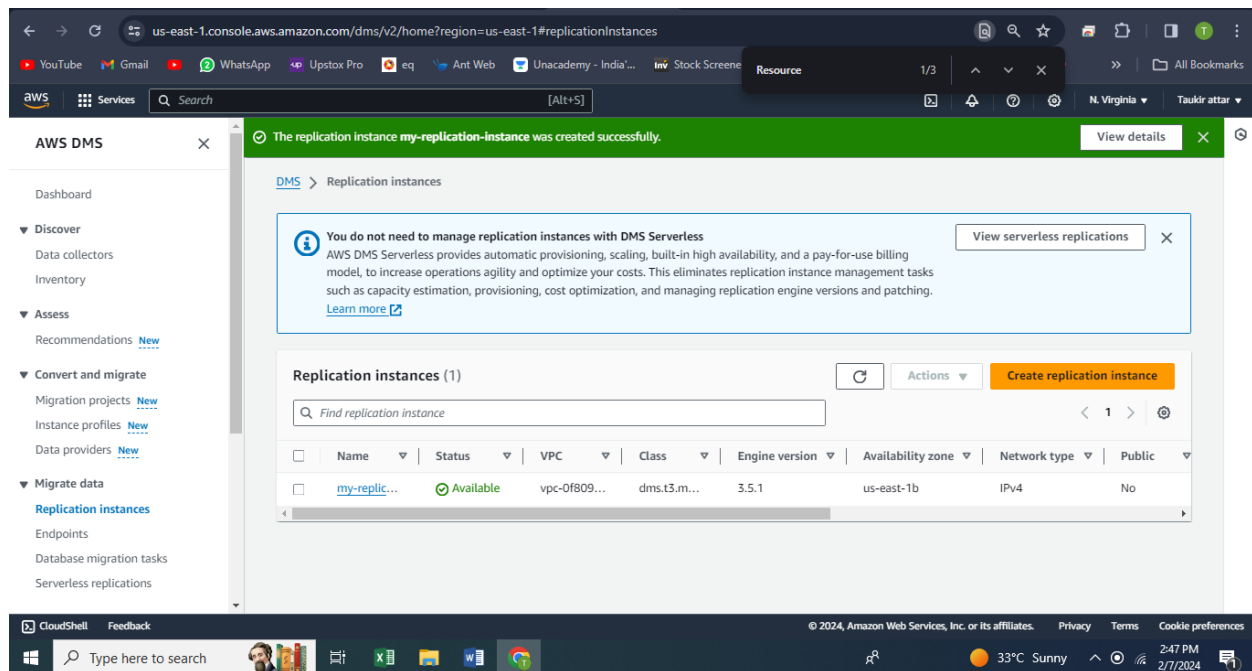


- Next, open the Advanced security and network configuration section. For the VPC security group(s) configuration, choose the same security group that you attached to your Amazon RDS database. This allows your replication instance to access your Amazon RDS database.



You also may edit the maintenance and tags settings.

When you're ready, choose Create to create your replication instance in AWS DMS.

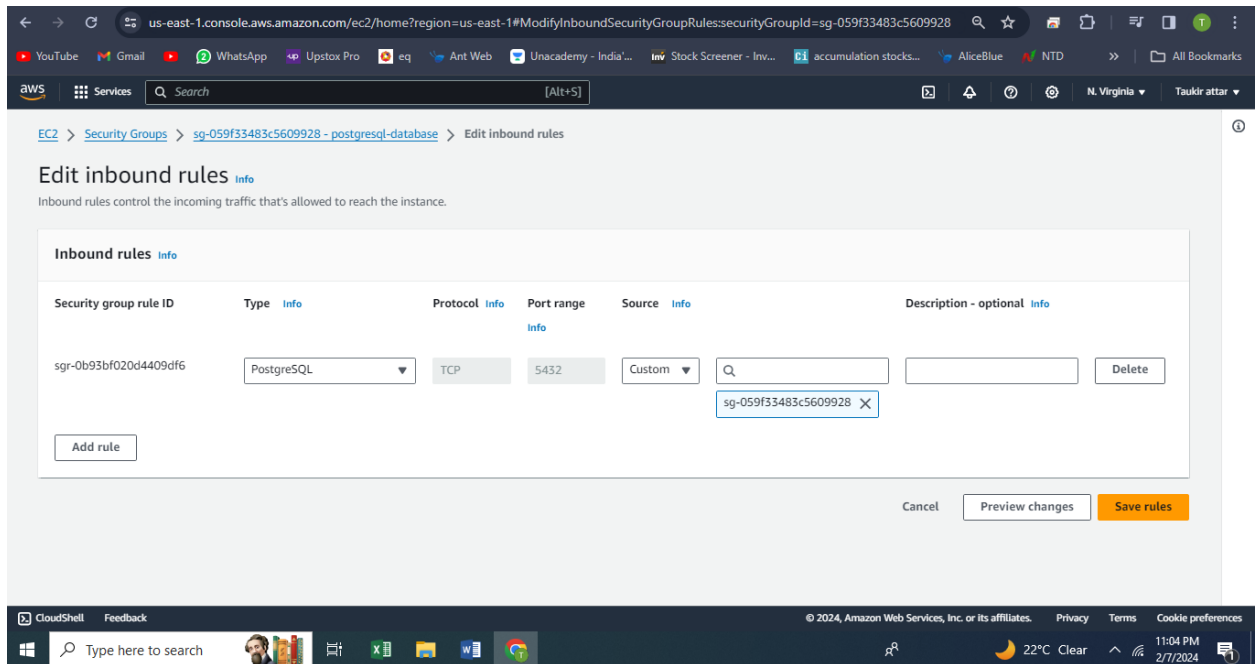


Added the postgresql new rule unable to take the screen shot

When our replication instance is ready to go, its Status is Available.

- Go to the Security Groups section in the Amazon EC2 console. You need to add a rule to your security group to allow your replication instance to access your database.
- In the Security Groups section, find the security group you attached to your PostgreSQL database instance and your replication instance, and choose it.
- Your security group has an existing rule that allows for access to your PostgreSQL instance from the IP address you used to create the database. Remove the existing IP address and enter the name of the security group used for your Amazon RDS database instance and replication instance.
- Choose Save rules to save the updated rules for your security group.

- When your replication instance is available and you have updated the rules for your security group, you may move on to the next module.



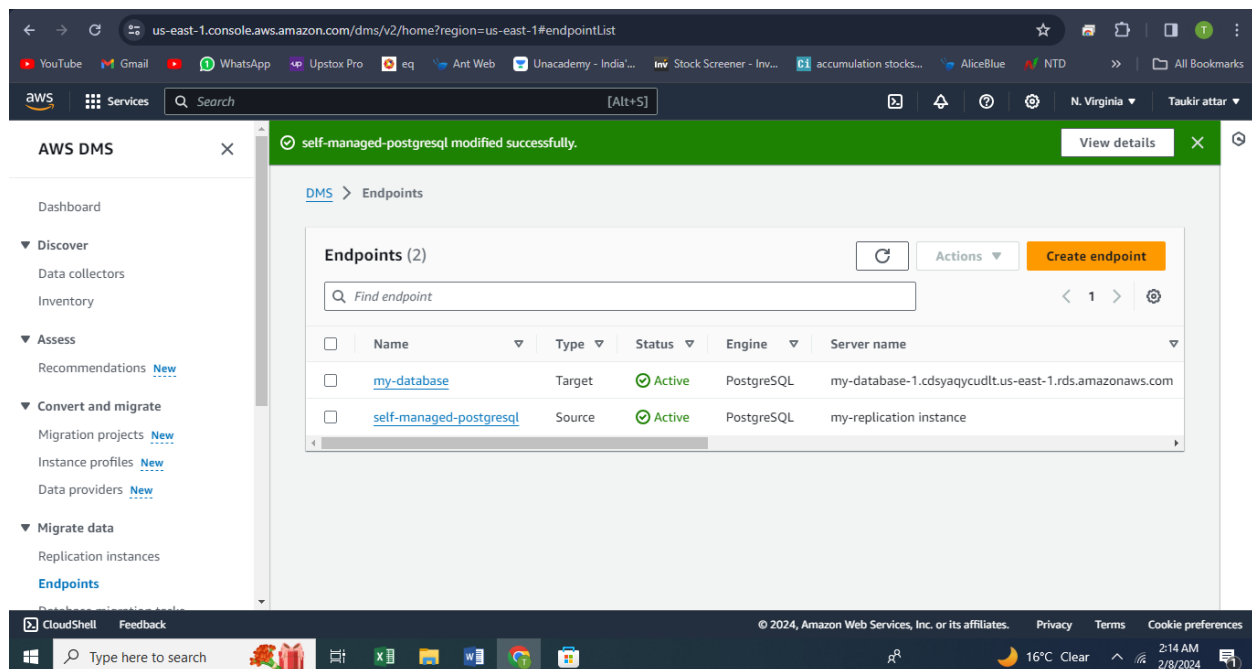
- We created a replication instance in AWS DMS. The replication instance is used to host the replication tasks that migrate data from an existing database to a fully managed database in Amazon RDS. You also updated a security group to allow access from your replication instance to your PostgreSQL database instance in Amazon RDS.

create source and target endpoints for a replication task in AWS DMS.

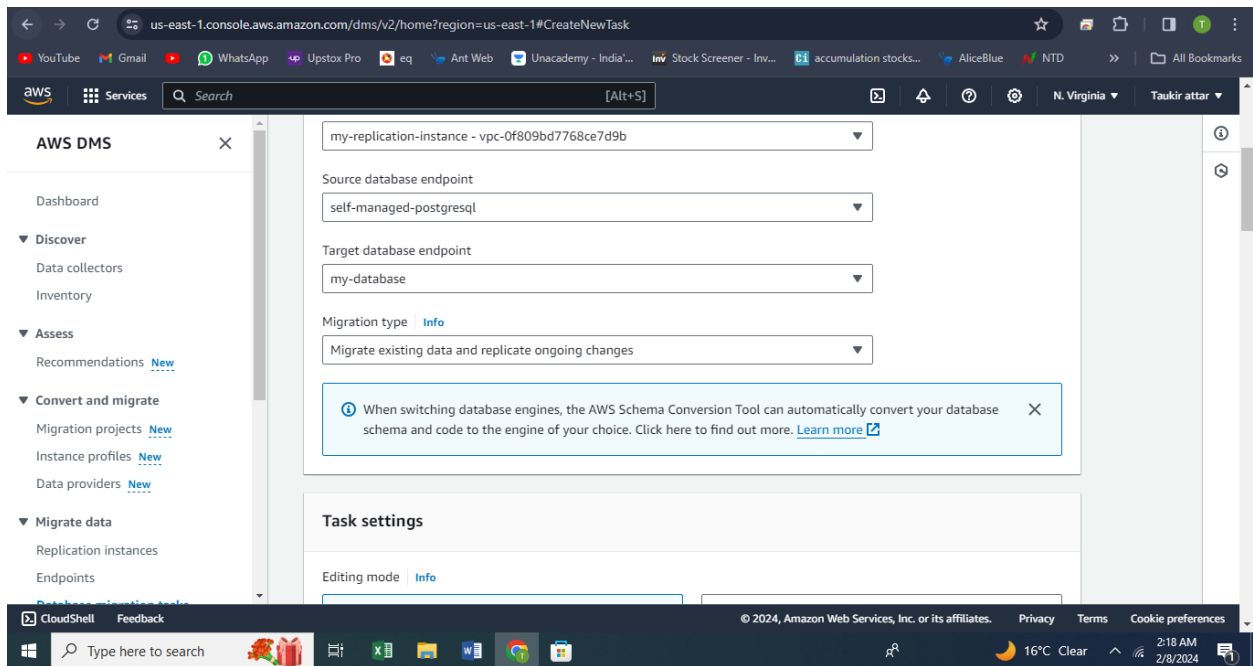
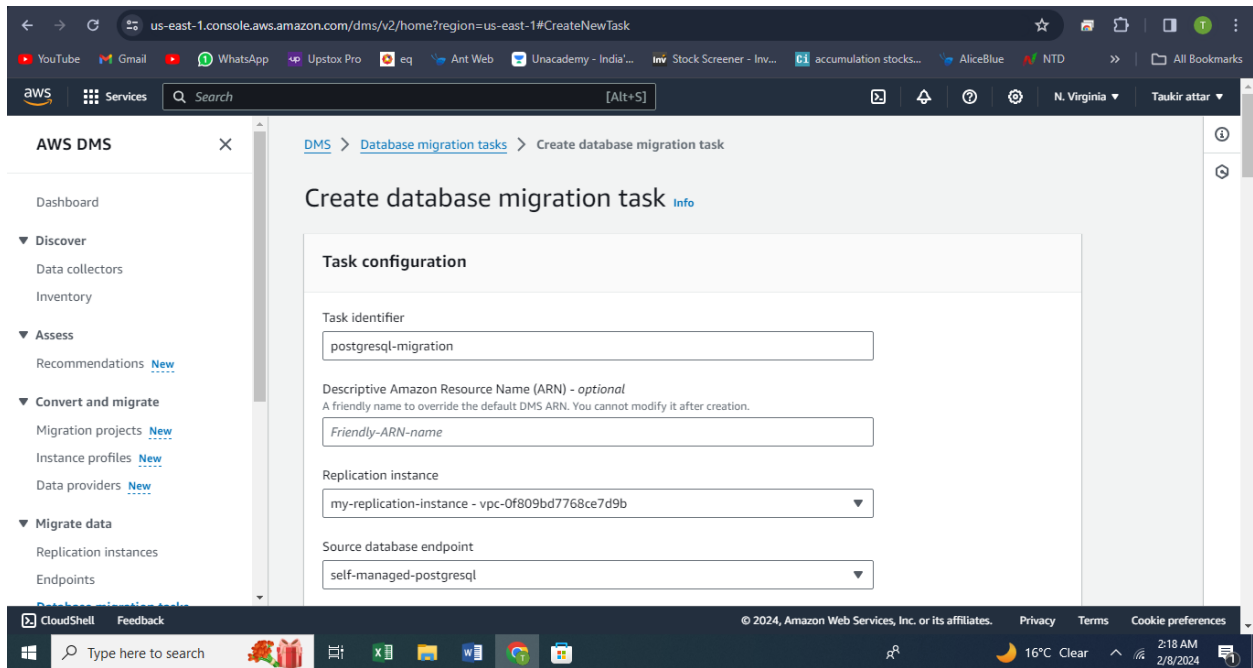
A replication task is a job to migrate data from one database to another by using AWS DMS. Before creating a replication task, you must register endpoints for your source and target databases. An endpoint describes the

connection address, credentials, and other information required to connect to a database.

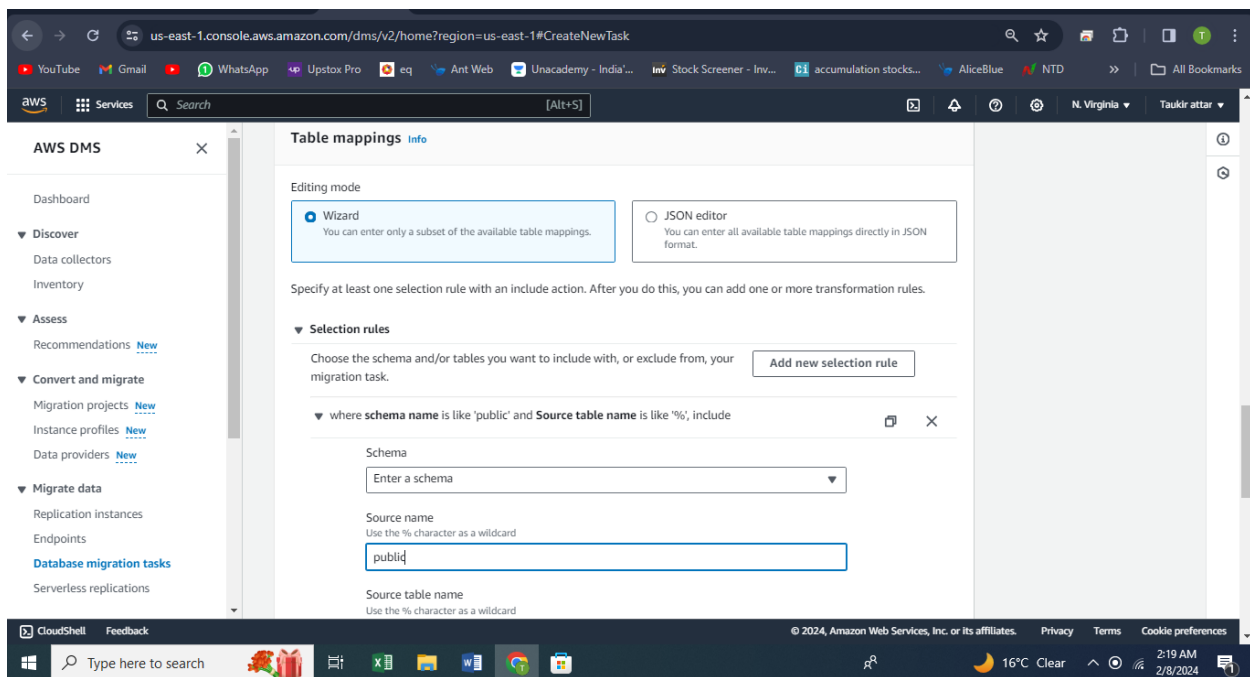
- First, create the endpoint for your target database. This is the database you created in Amazon RDS.
- Choose Create endpoint to create a new endpoint.



- In the endpoint creation wizard, choose to create a Target endpoint. Choose the check box to Select RDS DB instance, and choose your newly created Amazon RDS database in the dropdown.
- This fills in most of the Endpoint configuration details for you. You need to enter your password near the bottom of the section.
- Choose the replication instance you want to use, and then choose Run test. After a few seconds, you should see a Status of successful. This indicates that you configured your security group and endpoint correctly. To save your endpoint, choose Create endpoint.



- In the Table mappings section, choose the tables you want to copy over. Enter the name of the schemas and tables you want to copy. You can use % as a wildcard character to copy multiple tables or schemas.
- In this walkthrough, you migrate your tables exactly as they exist in the source database. If you have more complex requirements, AWS DMS supports filters and transformations of your source data before writing to your target database.
- When you are ready, choose Create task to start your replication task.





The screenshot shows the AWS DMS console in the 'Database migration tasks' section. A blue banner at the top indicates 'postgresql-migration creation in progress.' Below this, a 'New feature announcements' box lists updates like 'Fleet Advisor recommendations' and 'DMS schema conversion'. The main area displays 'Database migration tasks (1)' with a search bar and a table. The table has columns for Identifier, Status, Progress, Type, Source, Target, and Replication instances. One task, 'postgresql-migration', is listed with a status of 'Creating' and a progress bar.

Identifier	Status	Progress	Type	Source	Target	Replication instances
postgresql-migration	Creating		Full load, ongoing replication	self-managed-postgresql	my-database	my-replication-instances

- After you create your task, your task is shown in the Database migration tasks section with a Status of Creating.

This screenshot shows the same AWS DMS console interface, but the task 'postgresql-migration' now has a status of 'Ready' and a progress bar at 0%. The 'New feature announcements' box and the table structure remain the same.

Identifier	Status	Progress	Type	Source	Target	Replication instances
postgresql-migration	Ready	0%	Full load, ongoing replication	self-managed-postgresql	my-database	my-replication-instances

After the task is initialized, its Status is Starting or Ready.

The screenshot shows the AWS DMS console in the us-east-1 region. The left sidebar contains navigation links for Dashboard, Discover (Data collectors, Inventory), Assess (Recommendations), Convert and migrate (Migration projects, Instance profiles, Data providers), and Migrate data (Replication instances, Endpoints, Database migration tasks, Serverless replications). The main content area is titled 'Database migration tasks' and shows a table with one task: 'postgresql-migration'. The task's status is 'Load complete, replication ongoing', and its progress is 100%. The source is 'self-managed-postgresql' and the target is 'my-database'. The task type is 'Full load, ongoing replication'. The elapsed time is 0 minutes, and 3 tables have been loaded. The bottom of the screen shows the Windows taskbar with the time 2:23 AM on 2/8/2024.

Name	Status	Source	Target	Type	Progress	Elapsed time	Tables loaded
postgresql-migration	Load complete, replication ongoing	self-managed-postgresql	my-database	Full load, ongoing replication	100%	0 m	3

After the migration of existing data is complete, it shows a Status of Load complete, replication ongoing. Any updates to your source database at this point are copied to your target database

In this module, you created a replication task in AWS DMS to migrate your existing data and sync ongoing changes from your previous database to your new database in Amazon RDS.

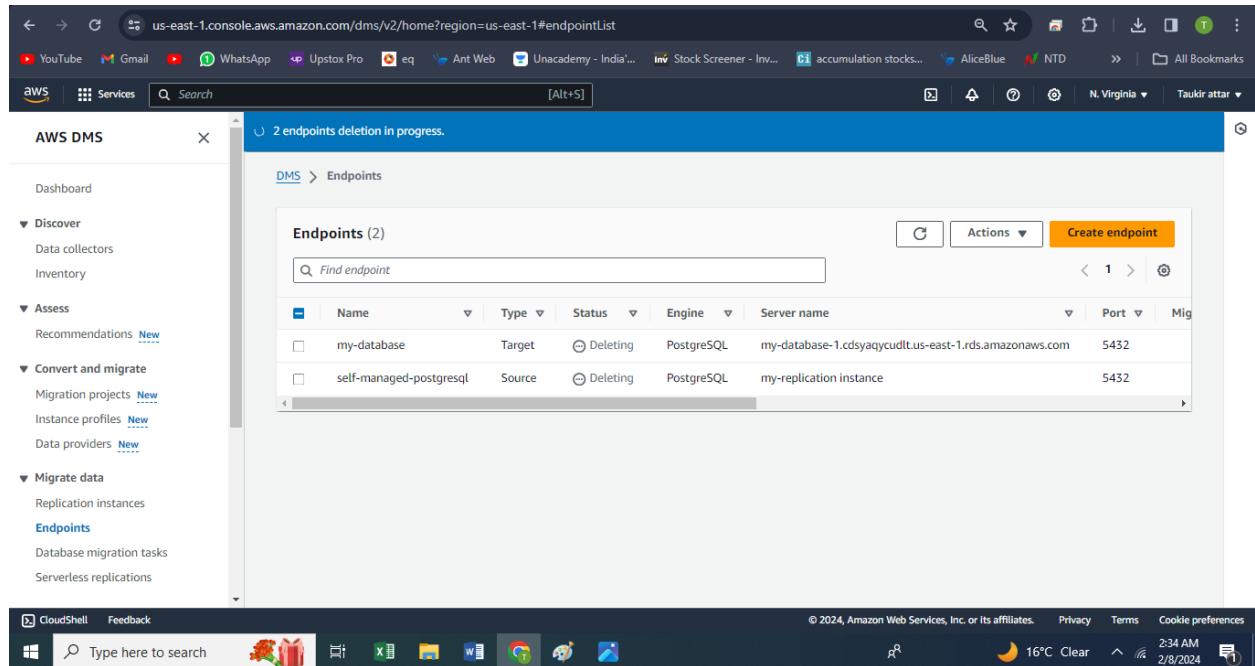
## Termination

The screenshot shows the AWS DMS console with a blue banner at the top indicating 'postgresql-migration deletion in progress'. The main content area shows the 'Database migration tasks' table with one task: 'postgresql-migration'. The task's status is 'Deleting', and its progress is 0%. The task type is 'Full load, ongoing replication', the source is 'self-managed-postgresql', the target is 'my-database', and the replication instance is 'my-replication'. The bottom of the screen shows the Windows taskbar with the time 2:33 AM on 2/8/2024.

Identifier	Status	Progress	Type	Source	Target	Replication Instance
postgresql-migration	Deleting	0%	Full load, ongoing replication	self-managed-postgresql	my-database	my-replication

stop and delete the database migration task to replicate your data. Navigate to the Database migration tasks section of the AWS DMS console. Choose the task you want to remove, and then choose Stop.

end point deleting

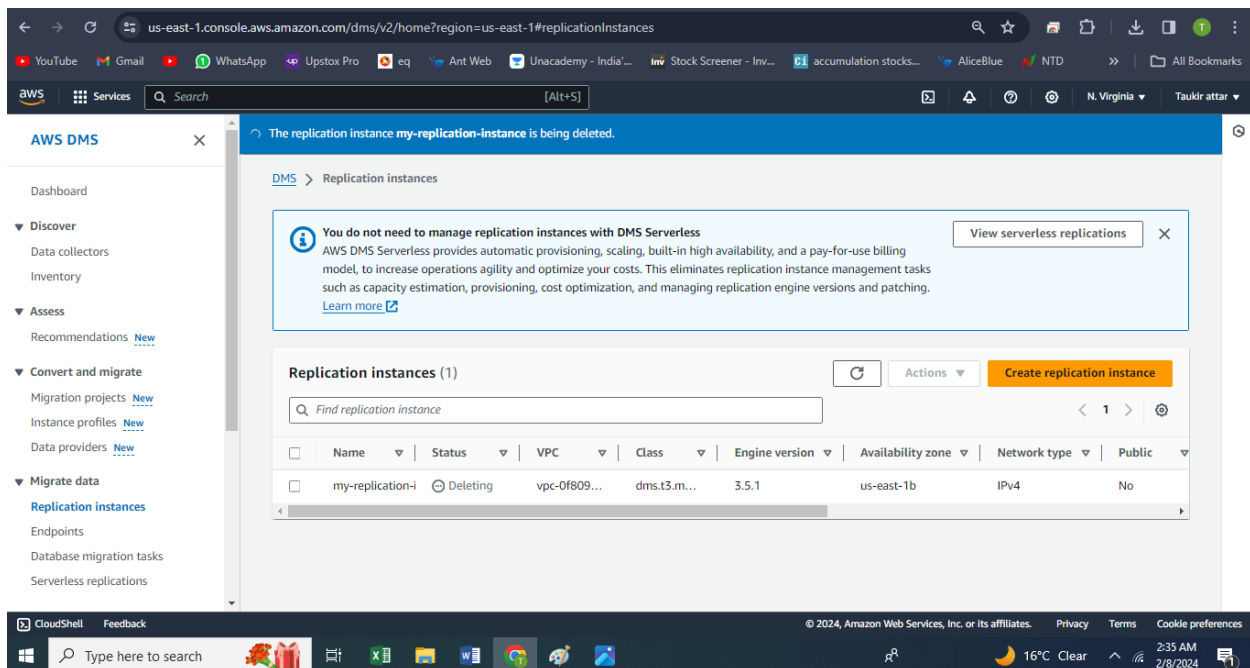


The screenshot shows the AWS DMS console interface. A blue banner at the top indicates "2 endpoints deletion in progress." The left sidebar contains navigation options: Dashboard, Discover (Data collectors, Inventory), Assess (Recommendations), Convert and migrate (Migration projects, Instance profiles, Data providers), and Migrate data (Replication instances, Endpoints, Database migration tasks, Serverless replications). The main content area is titled "Endpoints (2)" and features a search bar and a table of endpoints. The table has columns for Name, Type, Status, Engine, Server name, Port, and Migration. Two endpoints are listed: "my-database" (Target, Deleting, PostgreSQL, my-database-1.cdsyaqycudlt.us-east-1.rds.amazonaws.com, 5432) and "self-managed-postgresql" (Source, Deleting, PostgreSQL, my-replication instance, 5432). The bottom of the screen shows a Windows taskbar with the date 2/8/2024 and time 2:34 AM.

	Name	Type	Status	Engine	Server name	Port	Mig
<input type="checkbox"/>	my-database	Target	Deleting	PostgreSQL	my-database-1.cdsyaqycudlt.us-east-1.rds.amazonaws.com	5432	
<input type="checkbox"/>	self-managed-postgresql	Source	Deleting	PostgreSQL	my-replication instance	5432	

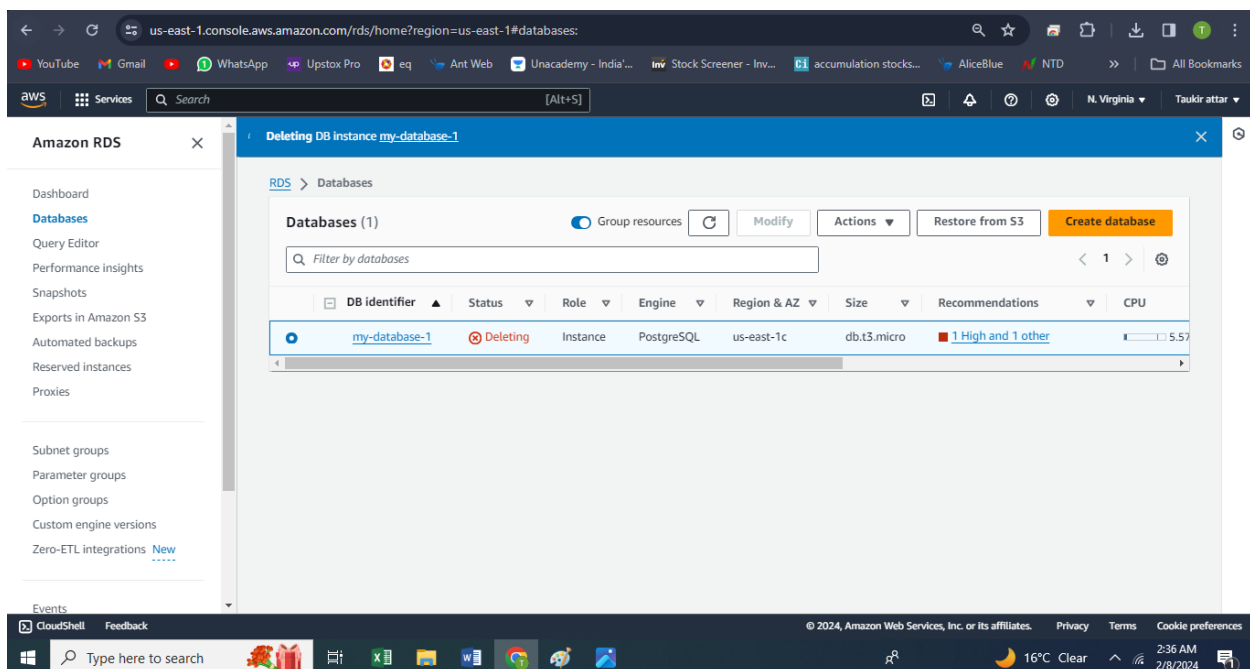
Next, navigate to the Endpoints section of the AWS DMS console. Choose both your source endpoint and your target endpoint, and then choose Delete.

Replication instance



Then go to the Replication instances section of the AWS DMS console. If your replication instance is not being used for any other replication tasks, choose it and then choose Delete.

## Rds



Navigate to the Amazon RDS console. Choose your database instance, and then choose Delete from the Actions dropdown.

## Lessons and Observations

- Planning and Preparation:

Lesson: Proper planning is crucial for a successful migration. Understand the source database schema, data types, and dependencies.

Observation: Conduct a thorough assessment of the database schema, size, and data transfer requirements to estimate the migration effort and identify potential challenges.

- AWS DMS Configuration:

Lesson: Configure AWS DMS to connect to the source PostgreSQL database and the target Amazon RDS instance.

Observation: AWS DMS provides a user-friendly interface for configuring and monitoring the migration process, making it relatively straightforward to set up.

- Schema Conversion:

Lesson: Convert the PostgreSQL schema to a format compatible with Amazon RDS (e.g., MySQL, Oracle, or SQL Server).

Observation: AWS Schema Conversion Tool (SCT) can automate much of the schema conversion process, but manual intervention may be required for complex schemas.

- Data Replication:

Lesson: Use AWS DMS to replicate data from PostgreSQL to Amazon RDS in near-real-time.

Observation: Monitor the replication process closely to ensure data consistency and minimize downtime during the migration.

- Testing and Validation:

Lesson: Perform thorough testing and validation of the migrated data to ensure that it is accurate and complete.

Observation: Use tools like AWS SCT and database query tools to compare data between the source and target databases and validate the migration.

- Performance Optimization:

Lesson: Optimize the performance of the Amazon RDS instance for the migrated database workload.

Observation: Adjust the instance size, storage configuration, and database parameters based on performance metrics and workload characteristics.

- Post-Migration Tasks:

Lesson: After the migration is complete, perform post-migration tasks such as updating application configurations and DNS settings.

Observation: Monitor the performance of the migrated database in the Amazon RDS environment and make any necessary adjustments.

- Backup and Disaster Recovery:

Lesson: Set up backup and disaster recovery strategies for the migrated database in Amazon RDS.

Observation: Use Amazon RDS automated backup and snapshot features to create regular backups and ensure data durability.