

Cycle 08 AWS Homework

Task 1: Blue-Green Deployment Simulation

- **Objective:** Safely upgrade RDS with minimal downtime.

- **Steps:**

1. Create two RDS instances:

- **Blue:** MySQL 8.0 (current production).
- **Green:** MySQL 8.4 (new version).

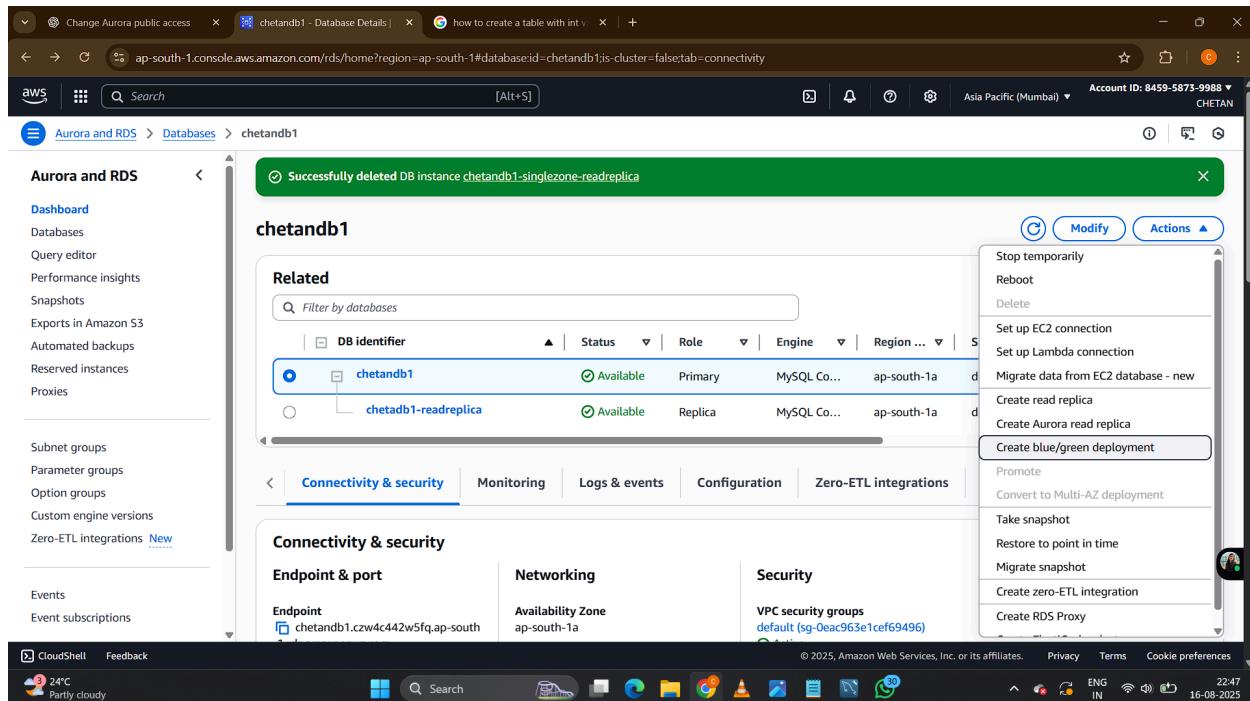
2. Set Green DB with **read-only permissions** to avoid accidental writes.

3. Use **Route53 Weighted Routing** or **ALB traffic splitting**:

- 80% traffic → Blue
- 20% traffic → Green

4. Test queries and app functionality on Green before shifting 100%.

⚡ **Key Point:** Zero-downtime upgrades; rollback is quick if issues occur.



Create blue/green deployment

Step 1
Create blue/green deployment

Step 2
Review and confirm

Blue/green deployment name

Blue database identifiers Blue

Selected database identifiers in the current production environment. The databases in the green environment are generated automatically when the blue/green deployment is created.

chetadb1
chetadb1-readreplica

Blue/green deployment name

Type a name for your blue/green deployment. The name must be unique across all blue/green deployments owned by your AWS account in the current AWS Region.

bg-deployment-1

The blue/green deployment name is case-insensitive, but is stored as all lowercase (as in "mybgdeployment"). 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.

Engine configurations

Blue instance configuration Blue

Blue engine version 8.0.42

Blue DB parameter group default.mysql8.0

Green instance storage configuration Green

Storage type Info

General Purpose SSD (gp2)
Baseline performance determined by volume size

Allocated storage Info

100 GiB

Allocated storage value must be 20 GiB to 65,536 GiB

Green replicas inherit the allocated storage of the green primary DB instance, which is the same as the blue primary DB instance.

▼ **Storage configuration upgrade** Info

Storage file system configuration upgrade
RDS recommends a storage file system configuration upgrade for your selected database instance.

You are on the latest storage configuration.

Cancel **Next**

DB identifier	Status	Role	Engine	Region	Size	Recom...	CPU
bg-deployment-1	Provisioning	Blue/gree...	-	-	-	-	-
chetandb1	Available	Primary	MySQL Comm.	ap-south-1a	db.t3.micro	4.0 GiB	4.0
chetandb1-readreplica	Available	Replica	MySQL Comm.	ap-south-1a	db.t3.micro	3.5 GiB	3.5
chetandb1-green-pet0j	Creating	Primary	MySQL Comm.	-	db.t3.micro	-	-

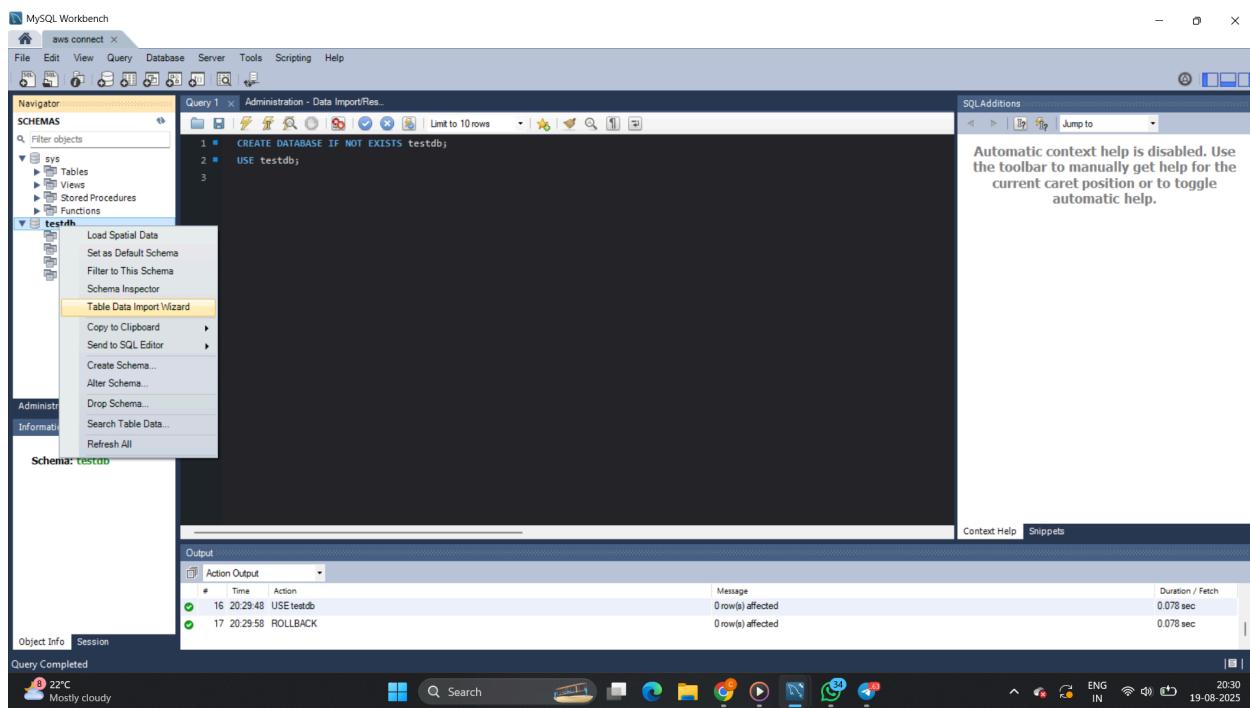
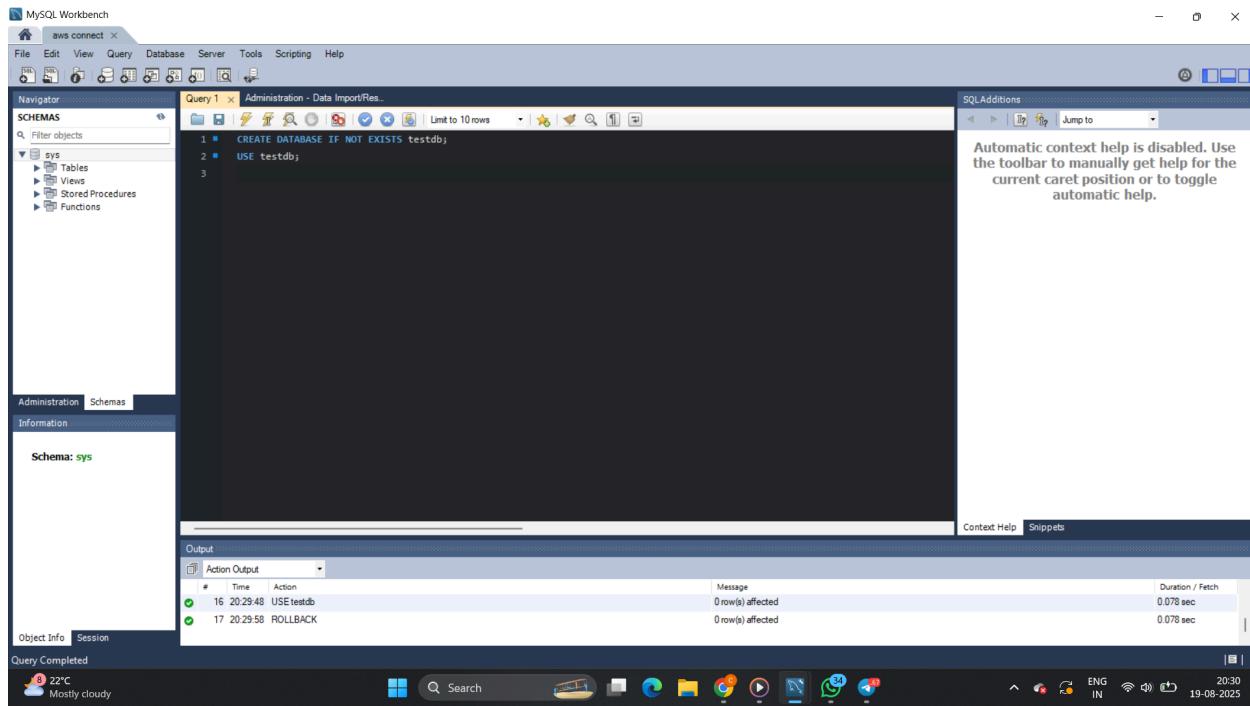
Task 2: RDS Snapshot & Point-in-Time Recovery

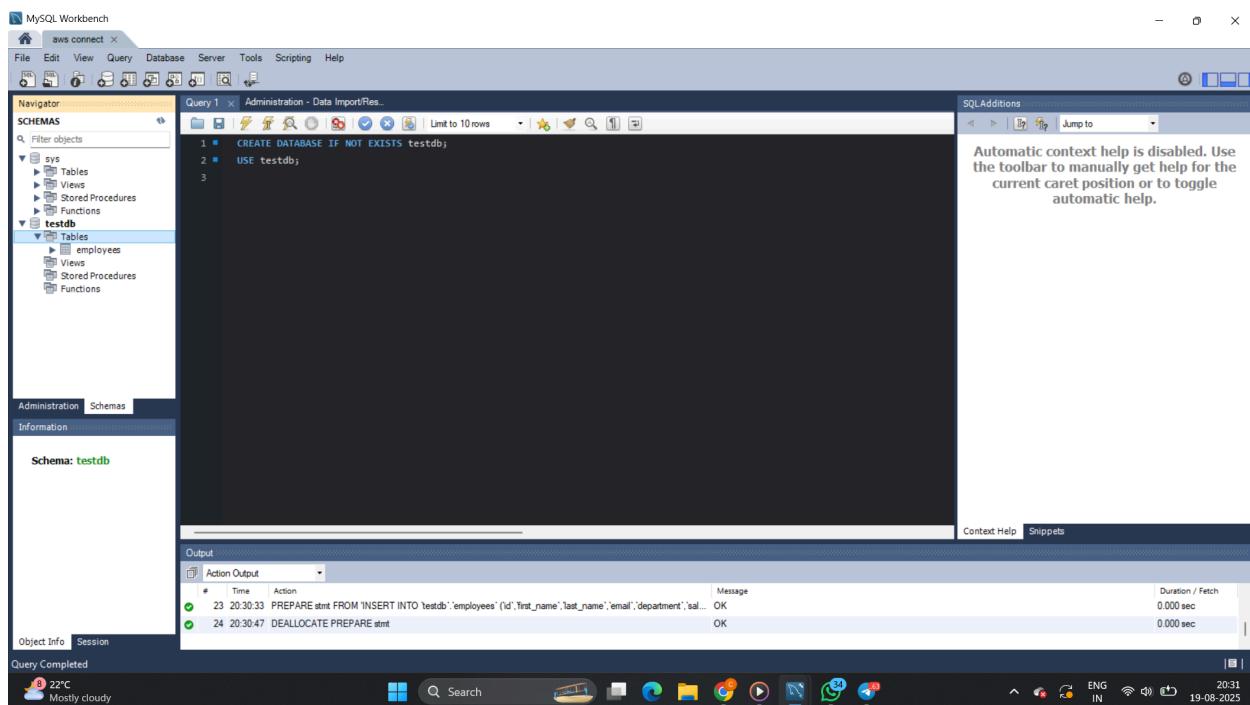
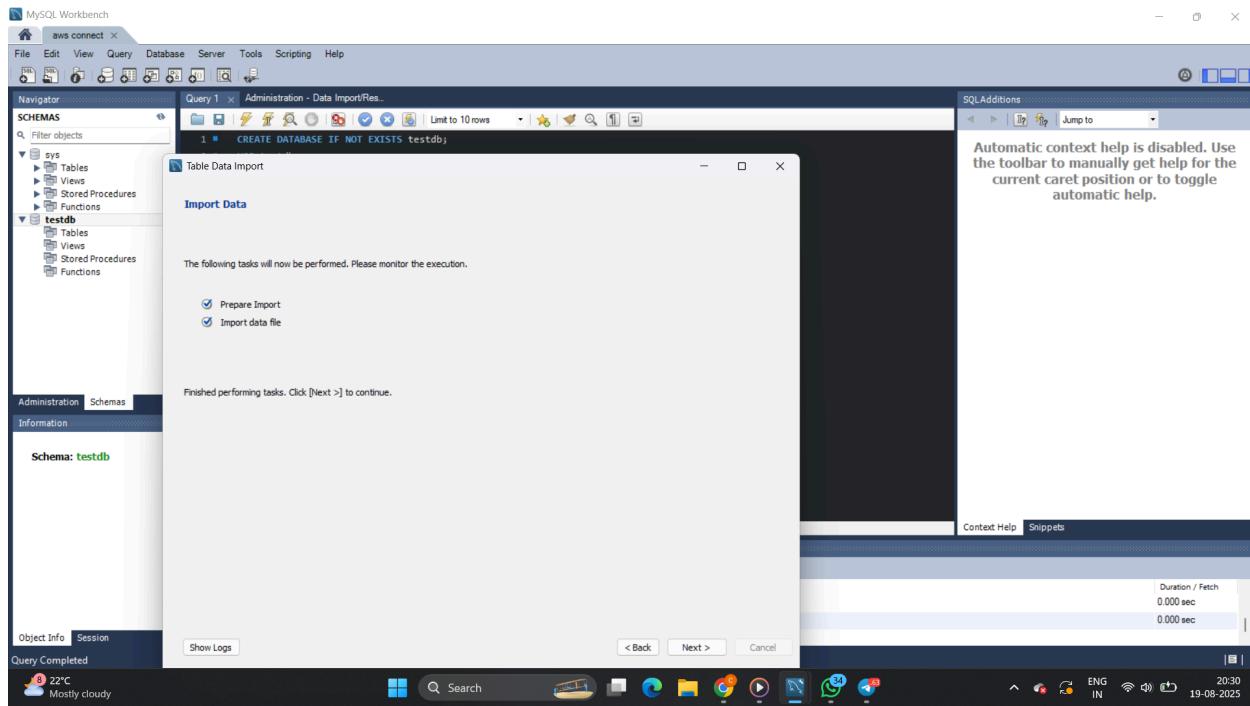
- Objective:** Practice backup and restore workflows.
- Steps:**
 - Create a **manual snapshot** of the database.
 - Delete a table intentionally.
 - Recovery methods:
 - Snapshot restore** → Launch a new DB instance from snapshot.
 - Point-in-time recovery** → Restore DB to a timestamp just before table deletion.
 - Export restored data using:
 - mysqldump** for SQL format.
 - AWS DMS** for JSON/other formats.

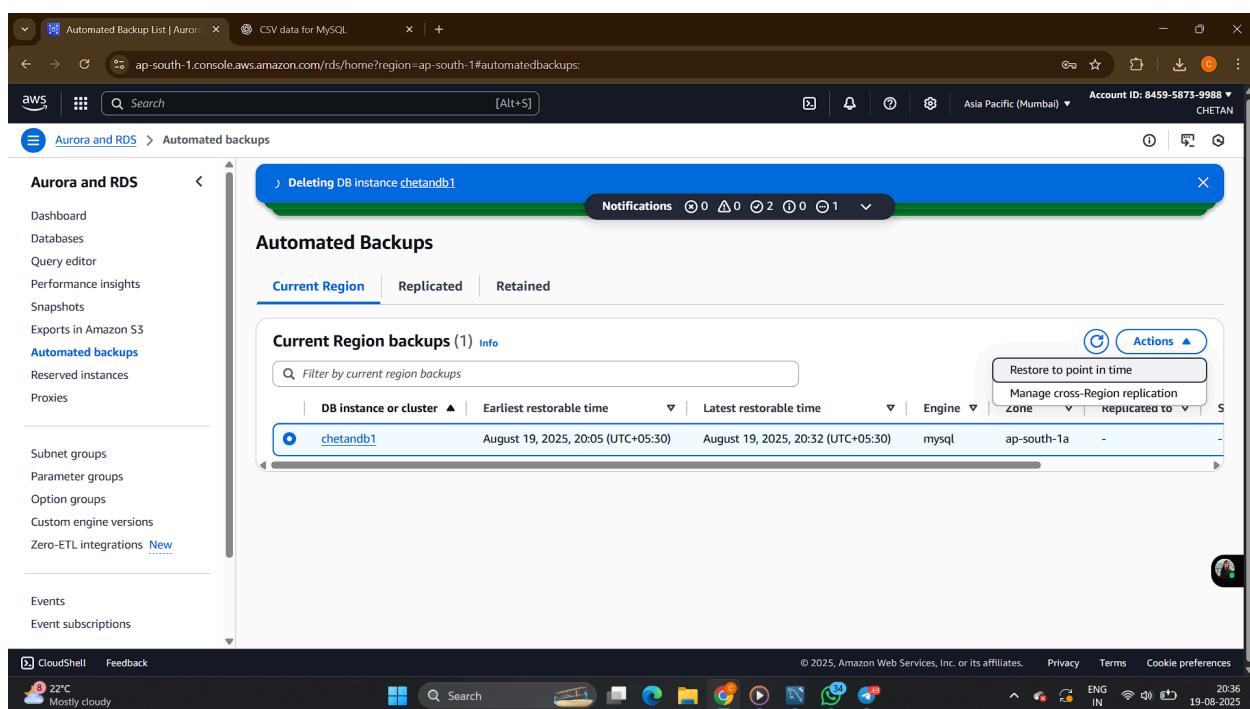
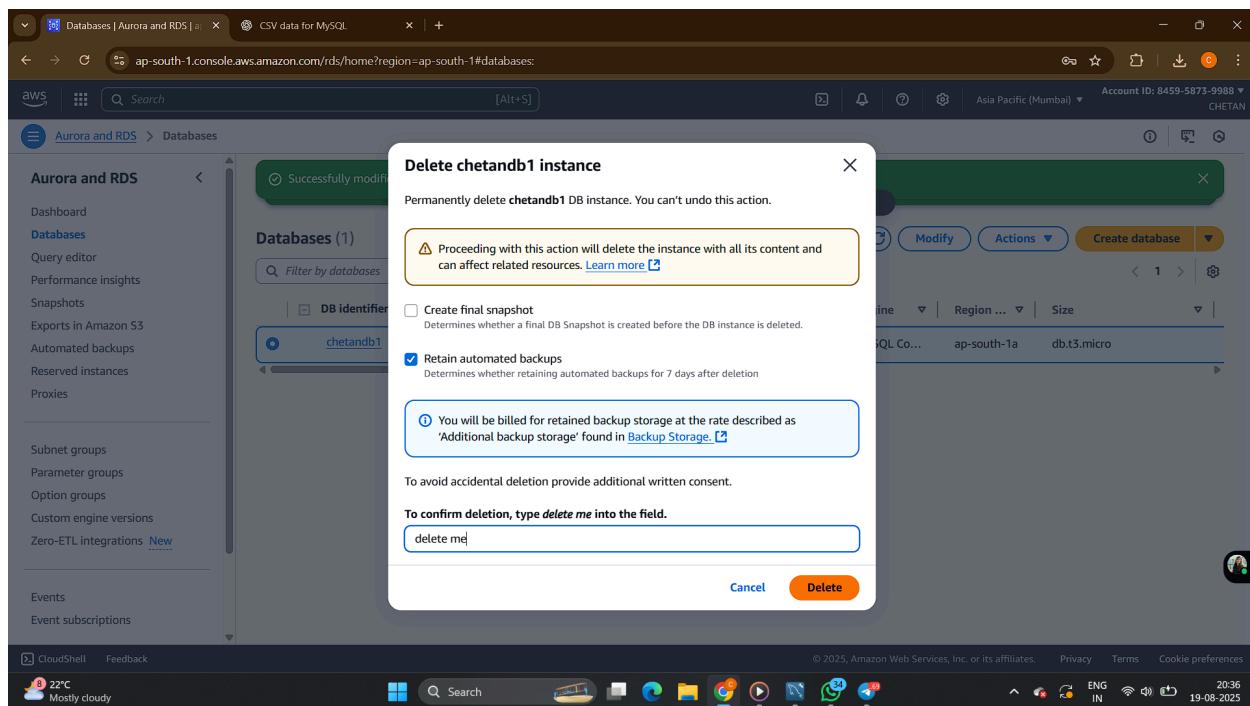
⚡ Key Point: Snapshot = full backup; PITR = precise recovery.

The screenshot shows the 'Create database' page in the AWS RDS console. At the top, there's a navigation bar with tabs like 'Aurora and RDS' and 'Create database'. Below it, the main title is 'Create database' with an 'Info' link. A section titled 'Choose a database creation method' contains two options: 'Standard create' (selected) and 'Easy create'. Under 'Engine options', several engine types are listed with icons: Aurora (MySQL Compatible), Aurora (PostgreSQL Compatible), MySQL (selected), PostgreSQL, MariaDB, Oracle, Microsoft SQL Server, and IBM Db2. The MySQL option is highlighted with a blue border. At the bottom of the page, there are links for 'CloudShell', 'Feedback', and social media sharing. The status bar at the bottom right shows the date and time: 19-08-2025.

The screenshot shows the MySQL Workbench application window. The title bar says 'MySQL Workbench'. The main area features a large 'Welcome to MySQL Workbench' heading. Below it is a descriptive paragraph about the tool's features: 'MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.' There are three links at the bottom: 'Browse Documentation >', 'Read the Blog >', and 'Discuss on the Forums >'. On the left, there's a sidebar with icons for database connections, users, and privileges. The central panel shows 'MySQL Connections' with two entries: 'local host' (root, 127.0.0.1:3306) and 'aws connect' (chetandb1, chetandb1.czww4c442w5fq.ap-south-1). The bottom status bar shows system information like battery level (-0.70%) and network connection.







Aurora and RDS > Databases

Databases (2)

DB identifier	Status	Role	Engine	Region ...	Size
chetandb1	Deleting	Instance	MySQL Co...	ap-south-1a	db.t3.micro
chetandb2	Creating	Instance	MySQL Co...	-	db.m7g.large

chetandb2 - Database Details

chetandb2

Summary

DB identifier	Status	Role	Engine	Recommendations
chetandb2	Available	Instance	MySQL Community	
CPU	Class	Current activity	Region & AZ	
2.94%	db.m7g.large	0 Connections	ap-south-1b	

Connectivity & security

Endpoint copied

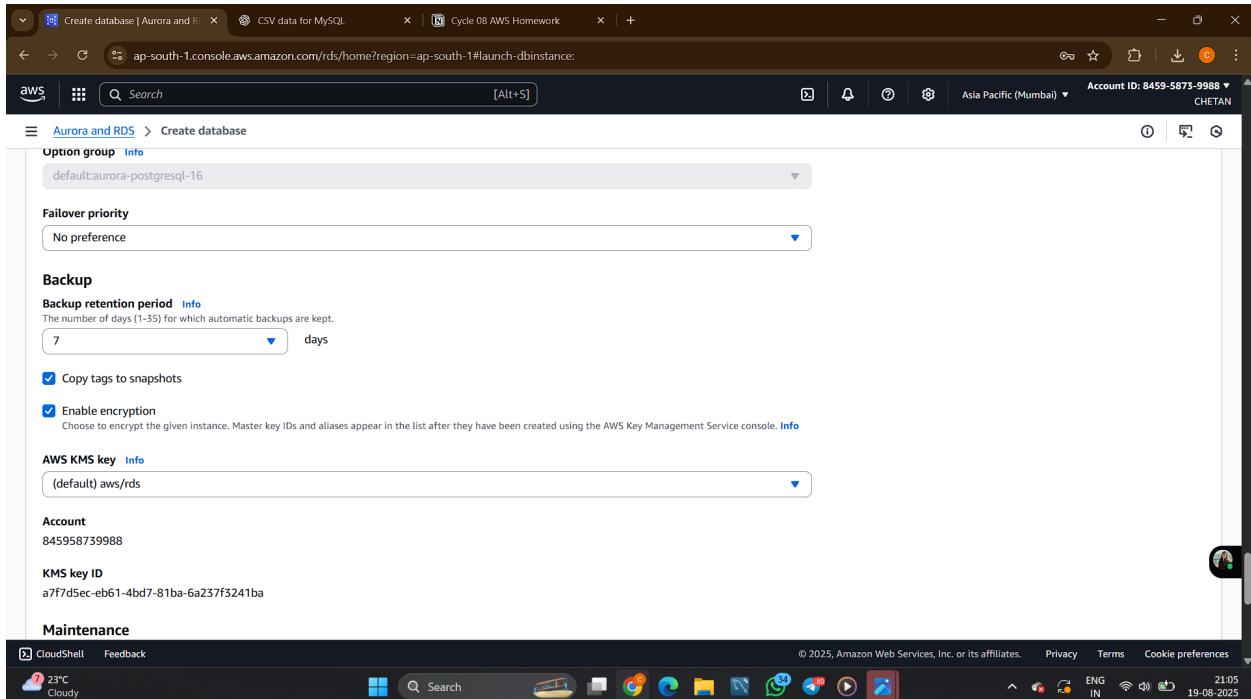
Endpoint & port	Networking	Security
chetandb2.czww4c442w5fq.ap-south-1.rds.amazonaws.com	Availability Zone ap-south-1b	VPC security groups default sg-0eac963e1cef69496 Active
Port 3306	VPC vpc-00dcef4d69d6b4f5d	Publicly accessible No

Task 3: Automated Backups Configuration

- Objective:** Set up automatic backups.
- Steps:**

1. Enable **automated backups** with **7-day retention**.
2. Configure **backup window** (e.g., daily at 2 AM).
3. Simulate deletion and test restoring from automated backup.

⚡ Key Point: Automated backups ensure point-in-time recovery (default: up to 35 days).



Task 4: Multi-AZ Deployment Testing

- **Objective:** Validate high availability.
- **Steps:**
 1. Launch an RDS instance with **Multi-AZ enabled**.
 2. Simulate a failure: **Reboot instance with failover**.
 3. Observe automatic promotion of standby to primary.

⚡ Key Point: Multi-AZ = High Availability (not scaling).

Screenshot of the AWS RDS 'Create database' wizard, Step 1: Choose a database creation method.

Choose a database creation method

- Standard create: You set all of the configuration options, including ones for availability, security, backups, and maintenance.
- Easy create: Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type: [Info](#)

- Aurora (MySQL Compatible)
- Aurora (PostgreSQL Compatible)
- MySQL
- PostgreSQL
- MariaDB
- Oracle
- Microsoft SQL Server
- IBM Db2

CloudShell Feedback Rain warning In effect

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 23:11 16-08-2025

Screenshot of the AWS RDS 'Create database' wizard, Step 2: Availability and durability.

Deployment options: [Info](#)

Choose the deployment option that provides the availability and durability needed for your use case. AWS is committed to a certain level of uptime depending on the deployment option you choose. Learn more in the [Amazon RDS service level agreement \(SLA\)](#).

- Multi-AZ DB cluster deployment (3 instances): Creates a primary DB instance with two readable standbys in separate Availability Zones. This setup provides:
 - 99.95% uptime
 - Redundancy across Availability Zones
 - Increased read capacity
 - Reduced write latency
- Multi-AZ DB instance deployment (2 instances): Creates a primary DB instance with a non-readable standby instance in a separate Availability Zone. This setup provides:
 - 99.95% uptime
 - Redundancy across Availability Zones
- Single-AZ DB instance deployment (1 instance): Creates a single DB instance without standby instances. This setup provides:
 - 99.5% uptime
 - No data redundancy

CloudShell Feedback Rain warning In effect

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 23:11 16-08-2025

The screenshot shows the 'Create database' step in the AWS RDS console. The 'DB cluster identifier' is set to 'chetandb1'. Under 'Master username', 'chetandb1' is selected. In the 'Credentials management' section, 'Self managed' is chosen. A 'Master password' is entered, and its strength is marked as 'Neutral'. The 'Confirm master password' field is also present.

The screenshot continues the 'Create database' process. In the 'Instance configuration' section, the 'DB instance class' is set to 'db.c6gd.medium'. In the 'Storage' section, the 'Storage type' is 'General Purpose SSD (gp3)', and the 'Allocated storage' is set to 100 GiB.

The screenshot shows the 'Create database' wizard in the AWS RDS console. The 'DB subnet group' section is selected, showing the 'default-vpc-00dcf4d69d6b4f5d' subnet group which includes 2 subnets across 2 Availability Zones. A note indicates that 3 AZs are required for a Multi-AZ DB cluster, and a link to 'Edit new subnet' is provided. The 'Public access' section shows the 'Yes' option selected, allowing public IP access. The 'VPC security group (firewall)' section shows the 'Choose existing' option selected, with 'default' chosen from the dropdown. The 'Existing VPC security groups' dropdown also lists 'default'. The 'RDS Proxy' section shows 'CloudShell' and 'Feedback' buttons.

The screenshot shows the 'Databases' page in the AWS RDS console. A blue banner at the top indicates that the database 'chetandb1' is being created, with a note about using settings from the suggested database add-ons. The main table displays four database instances under the cluster 'chetandb1': 'chetandb1-instance-1', 'chetandb1-instance-2', and 'chetandb1-instance-3' are Reader instances, while 'chetandb1' is the Multi-AZ DB cluster. All instances are currently 'Creating'. The left sidebar shows navigation links for Aurora and RDS, including 'Dashboard', 'Databases', 'Query editor', and 'Subnet groups'. The bottom status bar shows 'CloudShell' and 'Feedback' buttons, along with system information like '24°C Partly cloudy'.

Task 5: Compare Blue-Green vs. Rolling Deployments

- **Blue-Green:**

- Zero downtime, easy rollback.

 Costlier (duplicate infra).

- **Rolling Deployments:**

 Saves cost, gradual update.

 Risk of mixed versions during rollout.

- **AWS Services supporting Blue-Green:** RDS, Elastic Beanstalk, CodeDeploy, ECS.

Task 6: Encryption & Compliance

- **Scenario:** Client-side encryption required.

- **Steps:**

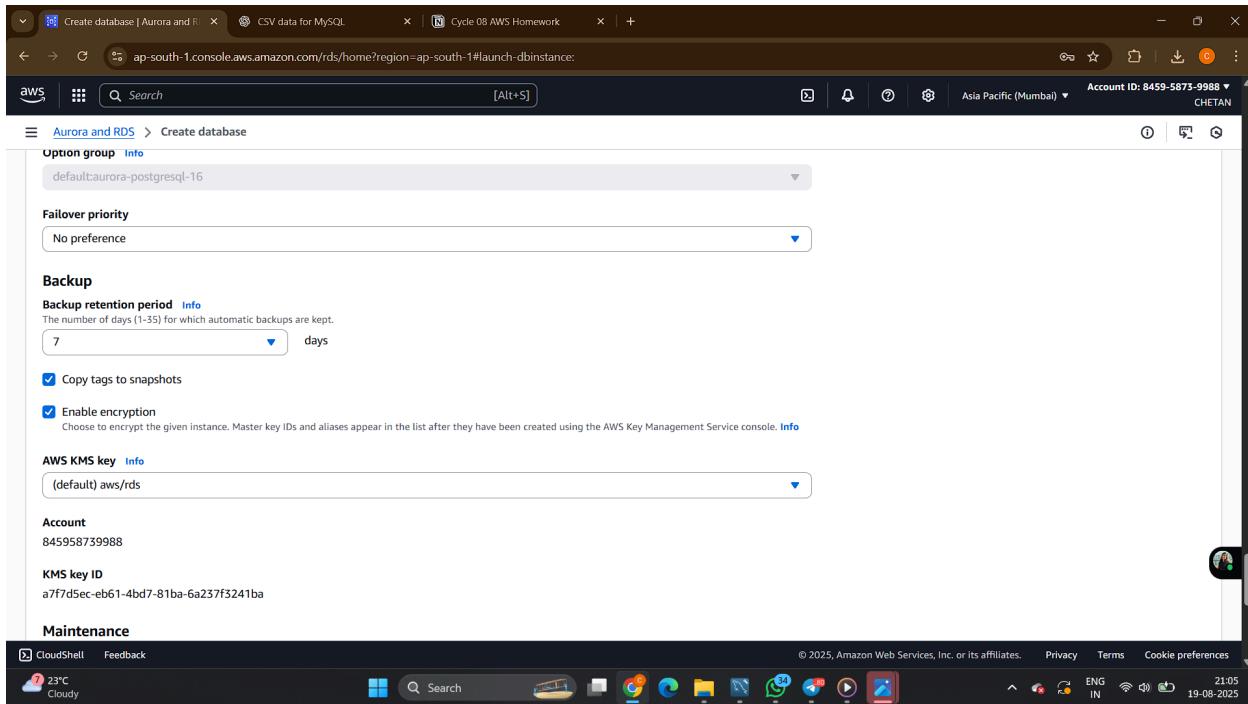
1. Use **AWS KMS** to manage keys.

2. Enable **RDS encryption at rest** (handled by AWS).

3. Compare:

- **RDS encryption (PaaS):** Easy, automatic, integrated with KMS.
- **Self-managed (on EC2):** More control, but manual setup and compliance overhead.

 **Key Point:** For compliance (HIPAA, PCI-DSS), always enable encryption.



Task 7: Cost Analysis for Backups

- **Objective:** Estimate costs.
- **Steps:**
 1. Use AWS Pricing Calculator.
 2. Calculate for:
 - 1TB automated backup retained for **35 days**.
 - Manual snapshot retained for **6 months**.

⚡ **Key Point:** Backup storage is charged separately from DB instance.

Task 8: Disaster Recovery Drill

- **Scenario:** AZ outage.
- **Steps:**
 1. Create a **snapshot** in Region A.
 2. **Copy snapshot to Region B.**

3. Restore DB in Region B.
4. Update **Route53** to point traffic to Region B DB.

 **Key Point:** Cross-region replication = disaster recovery & business continuity.