

Name - Manav Pahilwani

Roll No - 37

Class - D11AD

Aim: To study and Implement Database as a Service on SQL/NoSQL databases like AWS RDS, AZURE SQL/ MongoDB Lab/ Firebase.

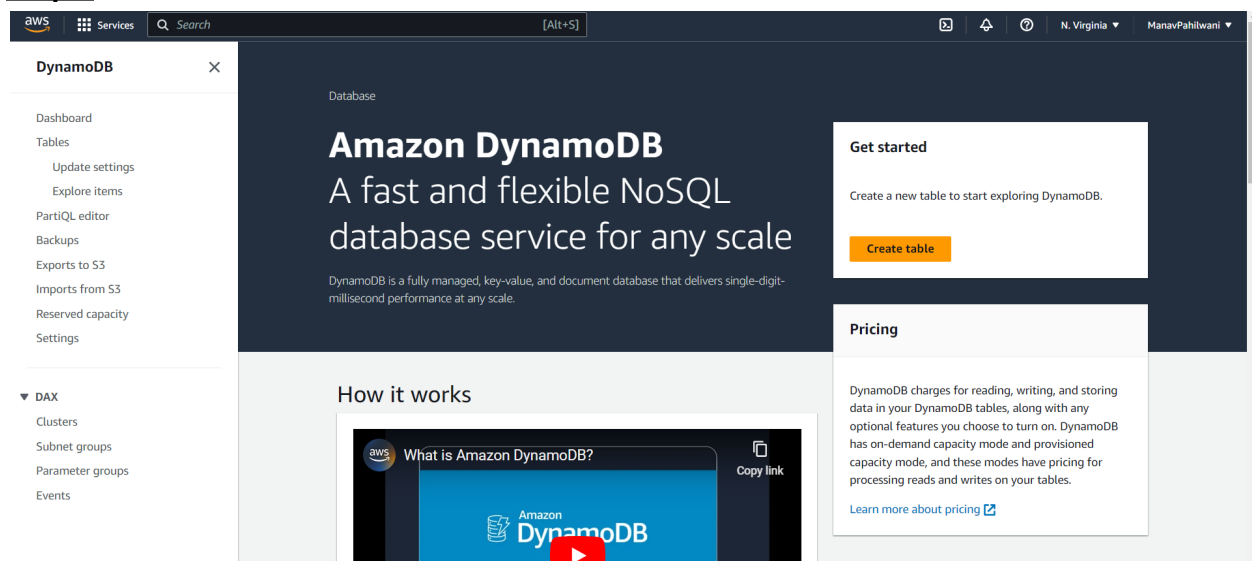
Theory :- Database As A Service:

Database as a service (DBaaS) is a cloud computing managed service offering that provides access to a database without requiring the setup of physical hardware, the installation of software, or the need to configure the database. Most maintenance and administrative tasks are handled by the service provider, freeing up users to quickly benefit from using the database. DBaaS uses The DBaaS model is ideal for small to medium-sized businesses that do not have well-staffed IT departments. Offloading the service and maintenance of the database to the DBaaS provider enables small to medium-sized businesses to implement applications and systems that they otherwise could not afford to build and support on-premises. Workloads involving data with stringent regulatory requirements may not be suitable for a DBaaS model. Furthermore, mission-critical applications that require optimal performance and 99.999% of uptime may be better suited for on-premises implementation. This is not to say that mission-critical workloads cannot run on cloud services, but much of the DBaaS adoption to date has been for less crucial applications, such as development and pilot programs.

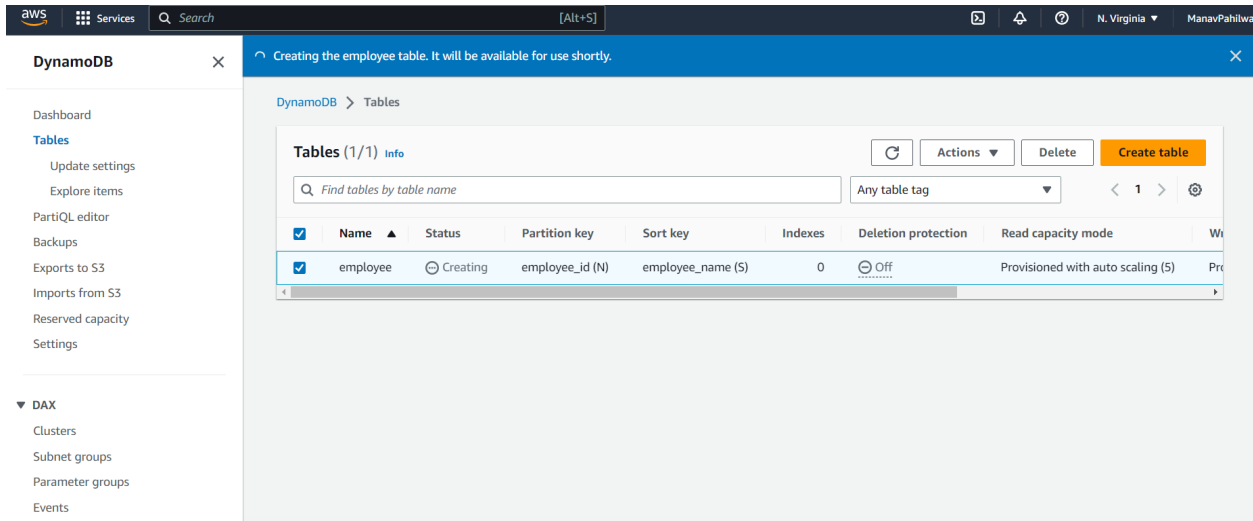
Implementation -

DynamoDB

Step 1: Select Database in Services



Step 2: Go to DynamoDB



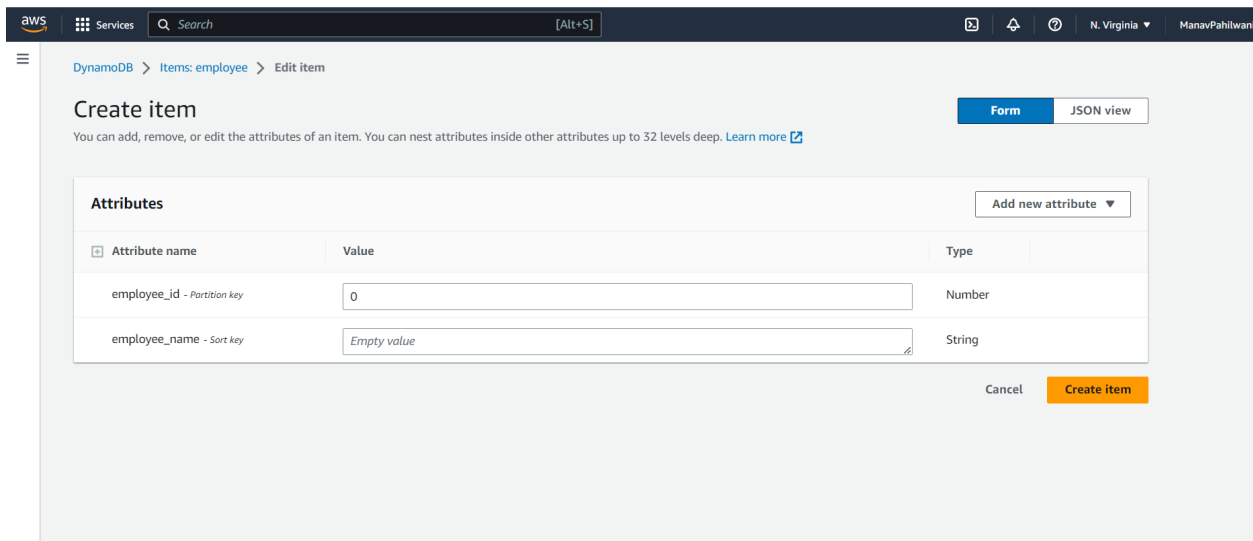
The screenshot shows the AWS Management Console for DynamoDB. The left sidebar contains navigation links for Dashboard, Tables, Update settings, Explore items, PartiQL editor, Backups, Exports to S3, Imports from S3, Reserved capacity, Settings, DAX, Clusters, Subnet groups, Parameter groups, and Events. The main content area shows the 'Tables' page with a table named 'employee' in the 'Creating' state. The table has a partition key 'employee_id (N)' and a sort key 'employee_name (S)'. The read capacity mode is 'Provisioned with auto scaling (5)'. A blue banner at the top of the console indicates 'Creating the employee table. It will be available for use shortly.'

Name	Status	Partition key	Sort key	Indexes	Deletion protection	Read capacity mode
employee	Creating	employee_id (N)	employee_name (S)	0	Off	Provisioned with auto scaling (5)

Step 3: Click on create table

Give a name for the table

Select partition key and sort key



The screenshot shows the AWS Management Console for DynamoDB, specifically the 'Create item' form for the 'employee' table. The form has two tabs: 'Form' and 'JSON view'. The 'Form' tab is active, showing a table with two attributes: 'employee_id' (Partition key, Number) and 'employee_name' (Sort key, String). The 'employee_id' attribute has a value of '0' and the 'employee_name' attribute has an 'Empty value'.

Attribute name	Value	Type
employee_id - Partition key	0	Number
employee_name - Sort key	Empty value	String

Step 4: Click on create table

The screenshot shows the 'Create item' page in the AWS Management Console. The breadcrumb navigation is 'DynamoDB > Items: employee > Edit item'. The page title is 'Create item'. Below the title, there is a note: 'You can add, remove, or edit the attributes of an item. You can nest attributes inside other attributes up to 32 levels deep. [Learn more](#)'. There are two tabs: 'Form' (selected) and 'JSON view'. The 'Attributes' section contains a table with two rows:

Attribute name	Value	Type
employee_id - Partition key	12	Number
employee_name - Sort key	Manav Pahlwani	String

At the bottom right, there are 'Cancel' and 'Create item' buttons.

Step 5: Go to actions and click on create an item

The screenshot shows the 'employee' table page in the AWS Management Console. The breadcrumb navigation is 'DynamoDB > Items > employee'. The page title is 'employee'. There is a 'Scan or query items' section with a button 'Scan or query items'. Below this, there is a green status bar: 'Completed. Read capacity units consumed: 0.5'. The 'Items returned (3)' section shows a table with three rows:

employee_id	employee_name
603	Nikita
129	Neha
12	Manav Pahlwani

At the bottom right, there are 'Actions' and 'Create item' buttons.

Step 6: Go to scan and then filter

The screenshot shows the AWS Management Console interface for the DynamoDB console. The left sidebar contains navigation options like Dashboard, Tables, Update settings, Explore items, PartiQL editor, Backups, Exports to S3, Imports from S3, Reserved capacity, Settings, DAX, Clusters, Subnet groups, Parameter groups, and Events. The main content area is titled 'employee' and shows the 'Scan or query items' section. The 'Scan' radio button is selected, and the 'Table - employee' is chosen from the 'Select a table or index' dropdown. The 'Select attribute projection' dropdown is set to 'All attributes'. The 'Filters' section is empty, with a placeholder 'Enter attribute name'. The 'Run' button is highlighted.

The screenshot shows the AWS Management Console interface for the DynamoDB console. The left sidebar contains navigation options like Dashboard, Tables, Update settings, Explore items, PartiQL editor, Backups, Exports to S3, Imports from S3, Reserved capacity, Settings, DAX, Clusters, Subnet groups, Parameter groups, and Events. The main content area is titled 'employee' and shows the 'Scan or query items' section. The 'Scan' radio button is selected, and the 'Table - employee' is chosen from the 'Select a table or index' dropdown. The 'Select attribute projection' dropdown is set to 'All attributes'. A filter is added: 'employee_name' is equal to 'Nikita'. The 'Run' button is highlighted. Below the scan section, a green notification bar indicates 'Completed. Read capacity units consumed: 0.5'. The 'Items returned (1)' section shows one item with 'employee_id' 603 and 'employee_name' Nikita.

DynamoDB [Alt+S] employee

Tables (1) Any table tag Find tables by table name employee

Scan or query items

☒ Scan ☐ Query

Select a table or index: Table - employee Select attribute projection: All attributes

Filters

Attribute name	Type	Condition	Value	
employee_id	Number	Equal to	12	Remove

Add filter

Run Reset

Completed. Read capacity units consumed: 0.5

Items returned (1)

employee_id	employee_name
12	Manav Pahlwani

Step 7: Delete the table

DynamoDB > Tables

Tables (1/1) info

Name	Status	Partition
employee	Active	employee

Delete table

Delete table **employee** in **US East (N. Virginia)** permanently? This action cannot be undone.

Warning: Proceeding with this action will delete the table and you won't be able to retrieve this data.

☒ Delete all CloudWatch alarms for **employee**.

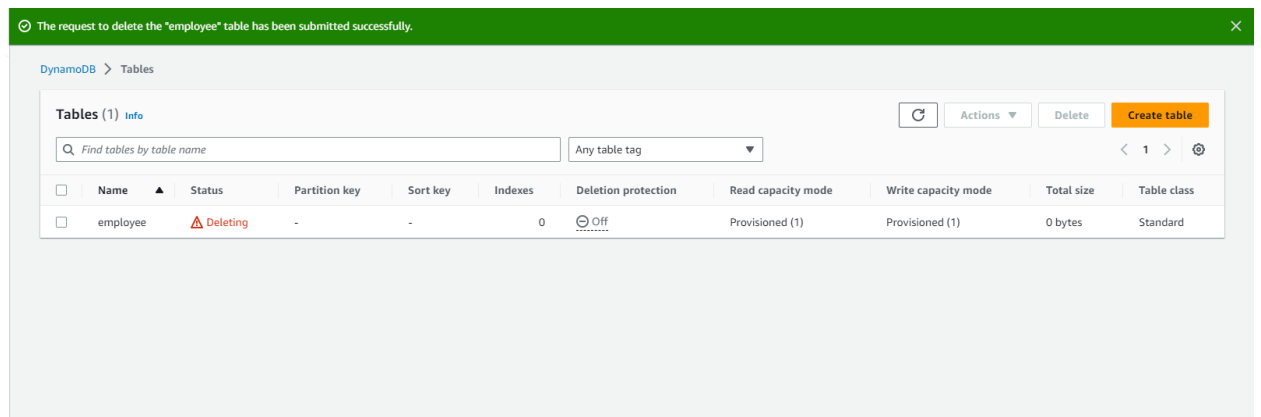
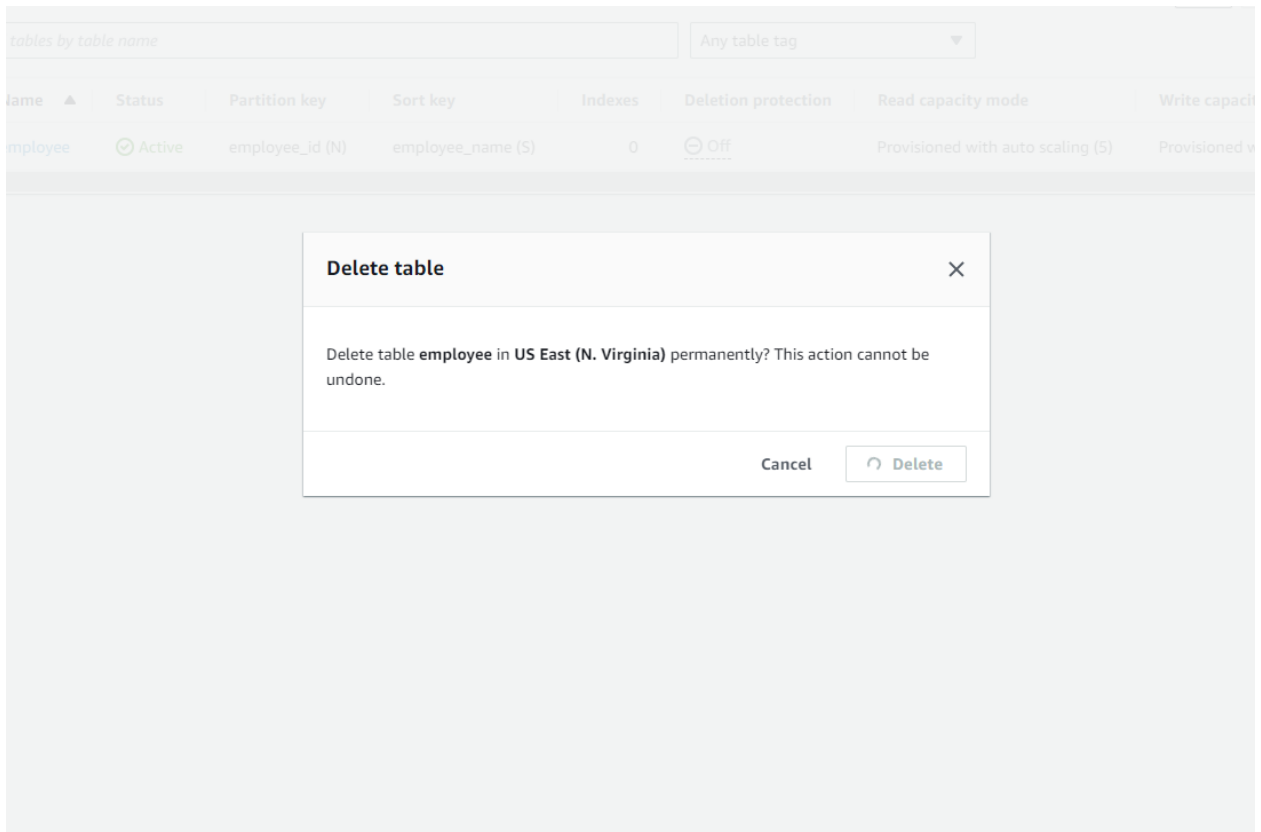
☐ Create an on-demand backup of **employee** before deletion.
You can create an on-demand backup of your table for long-term retention and data archiving. You can then use this backup to restore your data to its exact state before table deletion. Additional charges apply for on-demand backup and restore. For more information see [Pricing](#).

To avoid unintentional deletions, we ask you to provide additional confirmation.

Enter "confirm" to agree.

confirm

Cancel Delete



RDS:

Step 8: Now again go to services. Click on database and go to RDS

The screenshot shows the Amazon RDS console dashboard. On the left is a navigation menu with options like 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, Events, Event subscriptions, Recommendations, and Certificate update. The main content area features a top banner with a message about the new Multi-AZ deployment option for MySQL and PostgreSQL, a 'Create database' button, and a link to 'Restore Multi-AZ DB Cluster from Snapshot'. Below this is a 'Resources' section with a 'Refresh' button, listing various RDS resources in the US East (N. Virginia) region, such as DB Instances, Allocated storage, DB Clusters, Reserved instances, Snapshots, Manual DB Cluster, DB Instance, Automated DB Cluster, DB Instance, Recent events, and Event subscriptions. To the right of the Resources section is a 'Recommended for you' section with links to 'Build RDS Operational Tasks', 'Implementing Cross-Region DR', 'Time-Series Tables in PostgreSQL', and 'Migrate SSRS to RDS for SQL Server'. At the bottom right is an 'Additional information' section with links to 'Getting started with RDS', 'Overview and features', and 'Documentation'.

The screenshot shows the 'Choose a database creation method' and 'Engine options' sections of the Amazon RDS console. The 'Choose a database creation method' section has two radio buttons: 'Standard create' (selected) and 'Easy create'. The 'Engine options' section has a sub-section 'Engine type' with a grid of database engine options. The options are: Aurora (MySQL Compatible), Aurora (PostgreSQL Compatible), MySQL (selected), MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server. Each option has a corresponding icon or logo.

Database authentication

Database authentication options [Info](#)

- ☒ **Password authentication**
Authenticates using database passwords.
- ☐ **Password and IAM database authentication**
Authenticates using the database password and user credentials through AWS IAM users and roles.
- ☐ **Password and Kerberos authentication**
Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

Step 9: Click on create a database. It may take some time to create a database. After it gets created click on it to see details.

☒ **Enable auto minor version upgrade**

Enabling auto minor version upgrade will automatically upgrade to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the database.

Maintenance window [Info](#)

Select the period you want pending modifications or maintenance applied to the database by Amazon RDS.

- ☐ Choose a window
- ☒ No preference

Deletion protection

☐ **Enable deletion protection**

Protects the database from being deleted accidentally. While this option is enabled, you can't delete the database.

Estimated monthly costs

The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:

- 750 hrs of Amazon RDS in a Single-AZ db.t2.micro, db.t3.micro or db.t4g.micro Instance.
- 20 GB of General Purpose Storage (SSD).
- 20 GB for automated backup storage and any user-initiated DB Snapshots.

[Learn more about AWS Free Tier.](#)

When your free usage expires or if your application use exceeds the free usage tiers, you simply pay standard, pay-as-you-go service rates as described in the [Amazon RDS Pricing page](#).

i You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

Cancel

Create database

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

Events
Event subscriptions

Recommendations 1
Certificate update

Creating database database-1
Your database might take a few minutes to launch.
How was your experience creating an Amazon RDS database? [Provide feedback](#)

[View credential details](#)

RDS > Databases

Databases Group resources Refresh Modify Actions Restore from S3 Create database

DB identifier	Role	Engine	Region & AZ	Size	Status	Actions	CPU	Current activity	Maintenance
database-1	Instance	MySQL Community	-	db.t3.micro	Creating	-	-	-	none

Step 10: Click on vpc

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

Events
Event subscriptions

Recommendations 1
Certificate update

Creating database database-1
Your database might take a few minutes to launch.
How was your experience creating an Amazon RDS database? [Provide feedback](#)

[View credential details](#)

RDS > Databases > database-1

database-1 Modify Actions

Summary

DB identifier database-1	CPU -	Status Backing-up	Class db.t3.micro
Role Instance	Current activity	Engine MySQL Community	Region & AZ us-east-1a

Connectivity & security | Monitoring | Logs & events | Configuration | Maintenance & backups | Tags

Connectivity & security

Endpoint and port Endpoint database-1.cpokuuutoj4c.us-east-1.rds.amazonaws.com Port 3306	Networking Availability Zone us-east-1a VPC vpc-0ba61bf50d5b0bf19 Subnet group	Security VPC security groups default (sg-01ed4c791ca5d8464) Active Publicly accessible No GetRouteTableInfo
---	--	---

Your VPCs (1) [Info](#) Refresh Actions Create VPC

Clear filters

VPC ID: vpc-0ba61bf50d5b0bf19

	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main route table
<input type="checkbox"/>	-	vpc-0ba61bf50d5b0bf19	Available	172.31.0.0/16	-	dopt-024972b4b922e...	rtb-0453356a9a51a0f15

Select a VPC above

Step 11: Delete the instance. It may take some time to delete the instance (database).



Try the new Amazon RDS Multi-AZ deployment option for MySQL and PostgreSQL

For your Amazon RDS for MySQL and PostgreSQL workloads, improve transactional commit latencies by 2x, experience faster failover typically less than 3 instances by deploying the Multi-AZ DB cluster [Learn more](#)

[Create database](#)

Or, Restore Multi-AZ DB Cluster from Snapshot

Resources

[Refresh](#)

You are using the following Amazon RDS resources in the US East (N. Virginia) region (used/quota)

[DB Instances](#) (1/40)

Allocated storage (0.2 TB/100 TB)

[Increase DB instances limit](#)

[DB Clusters](#) (0/40)

[Reserved instances](#) (0/40)

[Snapshots](#) (1)

Manual

[DB Cluster](#) (0/100)

[DB Instance](#) (0/100)

Automated

[DB Cluster](#) (0)

[DB Instance](#) (1)

[Recent events](#) (6)

[Event subscriptions](#) (0/20)

[Parameter groups](#) (1)

[Default](#) (1)

[Custom](#) (0/100)

[Option groups](#) (1)

[Default](#) (1)

[Custom](#) (0/20)

[Subnet groups](#) (1/50)

[Supported platforms](#) [VPC](#)

[Default network](#) vpc-0ba61bf50d5b0bf19

Deployment to minimize downtime during upgrades
g Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides

Delete database-1 instance?

Are you sure you want to Delete the **database-1** DB Instance?

☒ Create final snapshot
Determines whether a final DB Snapshot is created before the DB instance is deleted.

Final snapshot name
Determines whether to retain automated backups after deletion.

database-1-snapshot

☒ Retain automated backups
Determines whether retaining automated backups for 7 days after deletion

You will be billed for retained backup storage at the rate described as 'Additional backup storage' found in [Backup Storage](#).

To confirm deletion, type *delete me* into the field

delete me

Cancel Delete

Deleting DB instance database-1

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

Group resources

Modify

Actions

Restore from S3

Create database

Filter by databases

DB identifier	Role	Engine	Region & AZ	Size	Status	Actions	CPU	Current activity
database-1	Instance	MySQL Community	us-east-1a	db.t3.micro	Deleting	2 Actions	2.10%	0 Connections

Conclusion - Amazon Web Services (AWS) provides a wide range of Database as a Service (DBaaS) options to help organizations manage their database needs. AWS's managed database services include Amazon RDS (Relational Database Service) for relational databases such as MySQL, Oracle, and PostgreSQL; Amazon DynamoDB for NoSQL databases; and Amazon Redshift for data warehousing. These services are designed to be highly scalable, reliable, and cost-effective, and they offer a range of features for backup and recovery, monitoring, security, and performance optimization.