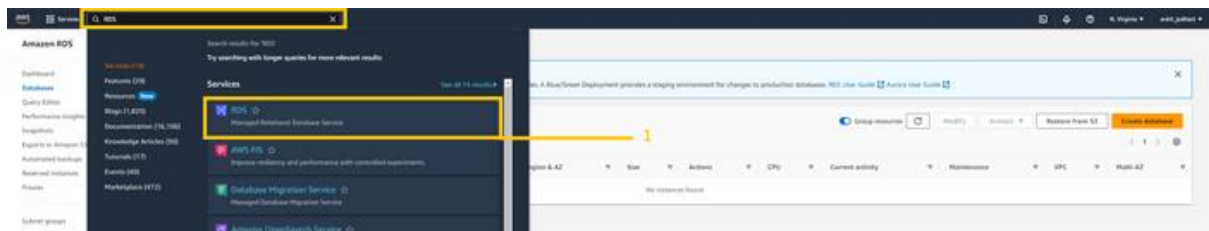
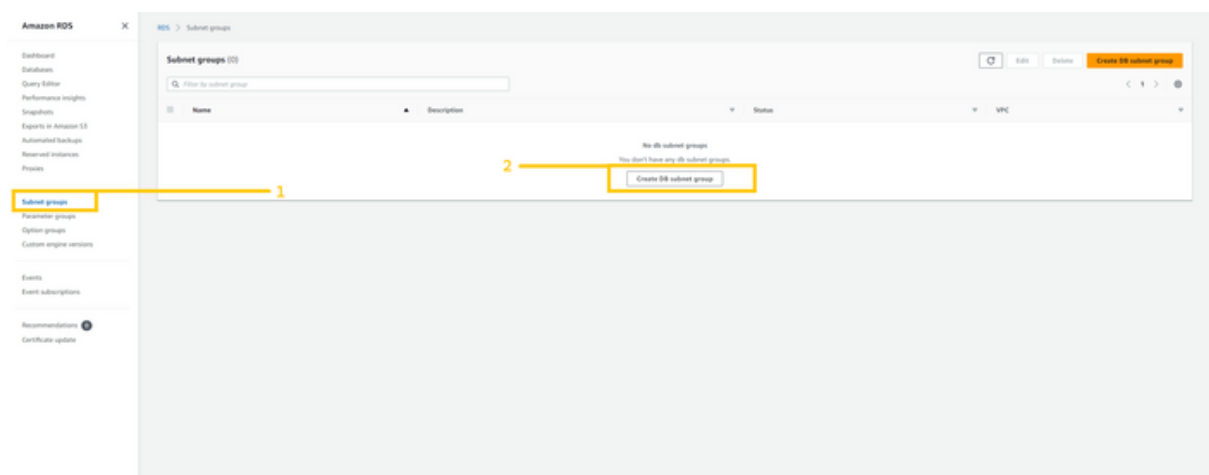


## ◆ RDS and Route 53

Now we are going to set up a database for our application. And for that, we are going to utilize the RDS service of AWS. So let's head over to the RDS dashboard. Just search RDS in the AWS console. And click on the service.



Now first we need to set up a subnet group. It specifies in which subnet and Availability zone our database instance will be created. So click on the subnet group button on the left panel. And click on the button Create database subnet group which is in the middle of the web page.



Here we can configure our VPC, subnet, and availability zone. Give any name to your subnet but make sure you select the correct VPC. and select Azs **us-east-1a** and **us-east-2b**. According to the architecture that I have shown you, our database will be in private subnet **pri-sub-7a** and **pri-sub-8b**. so please select as I have shown in the below figure. And then click on the create button.

**Amazon RDS** ×

**Create DB subnet group**

To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.

**Subnet group details**

Name: book-rds-db-subnet

Description: book-rds-db-subnet

VPC: Three-tier-app-VPC (sg-0442235d89779a0f)

**Add subnets**

Availability Zones: us-east-1a, us-east-1b

Subnets: subnet-023a241a2d0d76b99 (172.20.7.0/24), subnet-0279008b094d253a (172.20.8.0/24)

For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones.

**Subnets selected (2)**

Availability zone	Subnet ID	CIDR block
us-east-1a	subnet-023a241a2d0d76b99	172.20.7.0/24
us-east-1b	subnet-0279008b094d253a	172.20.8.0/24

Cancel Create

Now come to the **N. virginia** region and here we are going to create a database. So click on the database button on the left panel and then click on the created database button.

**Amazon 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](#) [Amazon User Guide](#)

**Databases (0)**

Filter by database

DB identifier	Status	Role	Engine	Region & AZ	Size	Actions	CPU	Current activity	Maintenance	VPC	Multi-AZ
No instances found											

Create Database

On this page, we can configure our database. Select stander create because I'm going to show you each and every step. select MySQL in the engine option because our application runs on MySQL database. If your app runs on other engines, you can select that one. Furthermore, you can select the engine version my application is compatible with MySQL version. But you can select according to the developer guild.

The screenshot displays the AWS RDS 'Create database' console. The 'Choose a database creation method' section has 'Standard create' selected. Under 'Engine options', 'MySQL' is chosen from a grid of options including Aurora (MySQL, PostgreSQL, Comprehend), MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server. The 'Edition' section shows 'MySQL Community' selected. A 'Known issues/limitations' link is present. At the bottom, the 'Engine Version' is set to 'MySQL 5.7.32'. A sidebar on the right provides additional information about MySQL on RDS, including its popularity, features, and supported instance classes.

**Create database**

Choose a database creation method [info](#)

- Standard create**  
You set all of the configuration options, including size for capacity, storage, 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, Comprehend)
- ☐ Aurora (PostgreSQL, Comprehend)
- ☒ MySQL
- ☐ MariaDB
- ☐ PostgreSQL
- ☐ Oracle
- ☐ Microsoft SQL Server

**Edition**

- ☒ MySQL Community

[Known issues/limitations](#)  
Review the [Known issues/limitations](#) to learn about potential compatibility issues with specific database versions.

**Instance filters**

- ☒ Show versions that support the Multi-AZ DB cluster [info](#)  
Create a Multi-AZ DB cluster with one primary DB instance and one readable standby DB instance. Multi-AZ DB clusters provide up to 3x faster transaction commit latency and automatic failover in typically under 30 seconds.
- ☒ Show versions that support the Amazon RDS Optimized Writes [info](#)  
Amazon RDS Optimized Writes improves write throughput by up to 3x at no additional cost.

**Engine Version**

MySQL 5.7.32

**MySQL**

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 54 TB.
- Supports General Purpose, Memory Optimized, and Burstable Performance Instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 3 read replica cross-region.

Scroll down, and select Dev/test as template. If you select the free tier then you won't be able to deploy RDS in a multi-availability zone. Select Multi-AZ DB instance from availability and durability option. In settings give any name to your database. In the credential setting give the username of the database in the Master username field and give the password in the Master password field. And then confirm the password below. Please do remember your username and password.

**Templates**

Choose a sample template to meet your use case.

☒ Production: The template is intended for production use, with high availability and performance.

☒ Dev/Test: The template is intended for development and testing environments.

☐ Free tier: For RDS Free Tier. This is available for certain instance classes and engine versions.

**Availability and durability**

Deployment options: [info](#)

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

☐ Multi-AZ DB Cluster: new

Creates a DB cluster with a primary DB instance and two standby-availability DB instances, with each DB instance in a different Availability Zone. Availability is increased by distributing data across multiple Availability Zones.

☒ Multi-AZ DB instance

Creates a primary DB instance and a standby DB instance in a different AZ. Provides high availability and data redundancy, but the standby DB instance doesn't support automatic failover.

☐ Single DB instance

Creates a single DB instance with no standby DB instances.

**Settings**

**DB instance identifier** [info](#)

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

book-db-01

The DB instance identifier is case insensitive, but is stored as all lowercase (as in "lowercase"). Consistently 1 to 63 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens, can't end with a hyphen.

**Master username** [info](#)

Type a login ID for the master user of your DB instance.

admin

You're responsible for password rotation. Your identifier must be a string.

☐ Manage master credentials in AWS Secrets Manager

Manage master user credentials in Secrets Manager. AWS can generate a password for you and manage it throughout its lifecycle.

☒ If you manage the master user credentials in Secrets Manager, some RDS features aren't supported. [Learn more](#)

☐ Auto generate a password

Amazon RDS can generate a password for you, or you can specify your own password.

**Master password** [info](#)

password

Constraints: At least 8 complex ASCII characters. Can't contain any of the following: spaces, single quotes, double quotes and @ or # signs.

Confirm master password [info](#)

password

Again scroll down, select `Brustable` class in the instance setting and select the instance type. Actually, it depends on your application uses. But for learning purposes, I am selecting `t3.micro`. now in storage type select `General purpose (GP2)` and allocate `22 GiB` for database. Please uncheck the auto-scaling option to keep our costs low. And In the connectivity option please select the option according below screenshot.

**Instance configuration**

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

**Amazon RDS Optimized Windows** [info](#)

Show instance classes that support Amazon RDS Optimized Windows.

**DB instance class** [info](#)

☐ Standard classes (includes all classes)

☒ Amazon RDS Optimized Windows (includes all classes)

db.t3.micro

2 vCPUs, 2 GB RAM, Network: 100 Mbps

☒ Include previous generation classes

**Storage**

**Storage type** [info](#)

General Purpose (SSD)

**Allocated storage** [info](#)

22 GiB

Processing less than 100 GiB of General Purpose (SSD) storage for high-throughput workloads could result in higher latencies upon exhaustion of the initial General Purpose (SSD) IO credit balance. [Learn more](#)

**Storage auto-scaling** [info](#)

Amazon RDS can automatically scale your database storage based on your application needs.

☐ Enable storage auto-scaling

Enabling this feature will allow the storage to increase after the specified threshold is reached.

**Connectivity** [info](#)

**Compute resource**

Choose whether to set up a compute resource to connect to the database. Setting up a compute resource with automatically managed credentials can make it easier to connect to the database.

☒ Don't connect to an EC2 compute resource

Don't set up a compute resource to connect to the database. You can manually set up a connection to a compute resource or use the Amazon RDS console to set up a connection.

☐ Connect to an EC2 compute resource

Set up a compute resource to connect to the database. You can manually set up a connection to a compute resource or use the Amazon RDS console to set up a connection.

**Network type** [info](#)

To use dual-stack mode, make sure that your instance is in a VPC with a private IP address.

☒ Public

Then resources are accessible only over the Internet.

☐ Dual-stack mode

Then resources are accessible over both IPv4 and IPv6.

In VPC, select VPC that we created earlier and in DB subnet group select the group that we just created, In the public access option please select No, choose existing security, and select security group **book-rds-db**.

The screenshot shows the 'Create database' page in the AWS RDS console. The 'VPC' section is highlighted with yellow boxes and numbered 1, 2, and 3. 1 points to the 'Virtual private cloud (VPC)' dropdown, 2 points to the 'DB subnet group' dropdown, and 3 points to the 'VPC security group (AWS)' dropdown. The 'Public access' section is set to 'No'. The 'Database authentication' section is partially visible at the bottom.

Scroll down, click on Additional Configuration, and in the database option give the name **test** because we need a database with the name of the **test** in the application. Enable Automated Backup. Note: you have to enable automated backup otherwise you won't be able to create a read replica of the RDS instance

The screenshot shows the 'Additional configuration' section of the 'Create database' page. The 'Database options' section is highlighted with a yellow box and numbered 1. The 'Backup' section is highlighted with a yellow box and numbered 2. The 'Monitoring' section is also visible.

Scroll down, mark on enable encryption checkbox to make the database bit more secure, and click on Create database button below.

Please note that automated backups are currently supported for InnoDB storage engine only. If you are using MySQL, refer to [details here](#).

**Backup retention period** [help](#)  
The number of days (1-35) for which automated backups are kept.  
1 day

**Backup window** [help](#)  
The daily time range (in UTC) during which RDS takes automated backups.  
☐ Choose a window  
☒ No preference

☒ Copy logs to snapshots

**Backup replication** [help](#)  
☐ Enable replication in another AWS Region  
Enabling replication automatically creates backups of your DB instance in the selected Region, for disaster recovery in addition to the current Region.

**Encryption**  
☒ Enable encryption  
Enables the encryption of the database. Master key IDs and aliases appear in the list after they have been created using the AWS Key Management Service console. [help](#)  
AWS KMS key [help](#)  
default:aws/hls

**Account**  
136230758702

**KMS key ID**  
0fa76d1-0264-433b-b29a-c95d8ff6f9fa

**Log exports**  
Select the log types to publish to Amazon CloudWatch Logs.  
☐ Audit log  
☐ Error log  
☐ General log  
☐ Slow query log

**IAM role**  
The following service-linked role is used for publishing logs to CloudWatch Logs.  
RDS-service-linked-role

**Maintenance**  
[Auto minor version upgrade help](#)  
☒ Enable auto minor version upgrade  
Enabling auto minor version upgrade will automatically upgrade to new minor versions as they are released. The automatic upgrade occurs during the maintenance window for the database.

**Maintenance window** [help](#)  
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  
Prevents the database from being deleted accidentally. While this option is enabled, you can't delete the database.

**Estimated monthly costs**

DB instance	12.41 USD
Multi-AZ standby instance	12.41 USD
Storage	5.06 USD
<b>Total</b>	<b>29.88 USD</b>

This billing estimate is based on on-demand usage as described in [Amazon RDS Pricing](#). Estimate does not include costs for backup storage, I/O if applicable, or data transfer.  
[Estimate your monthly costs for the DB instance using the AWS Simple Monthly Calculator](#)

☒ 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.

[Go to console](#)

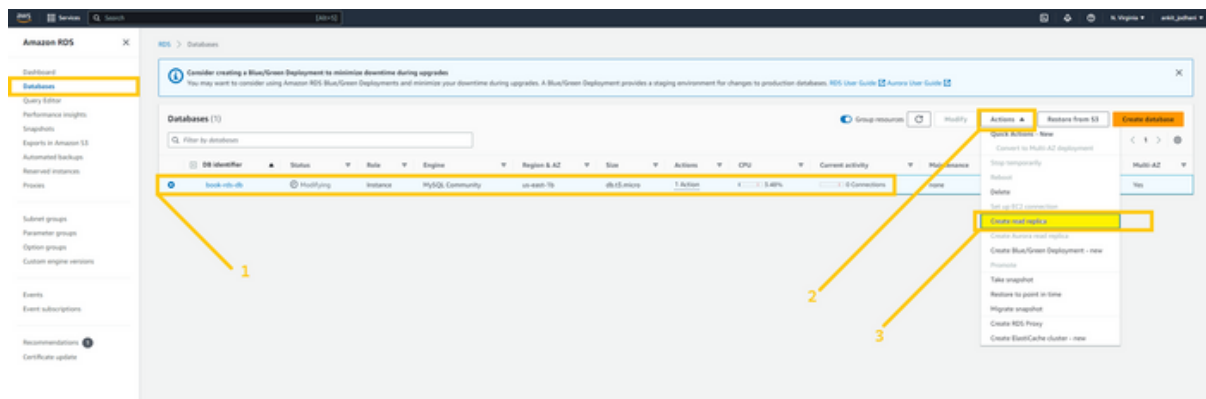
**MySQL** [X](#)

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

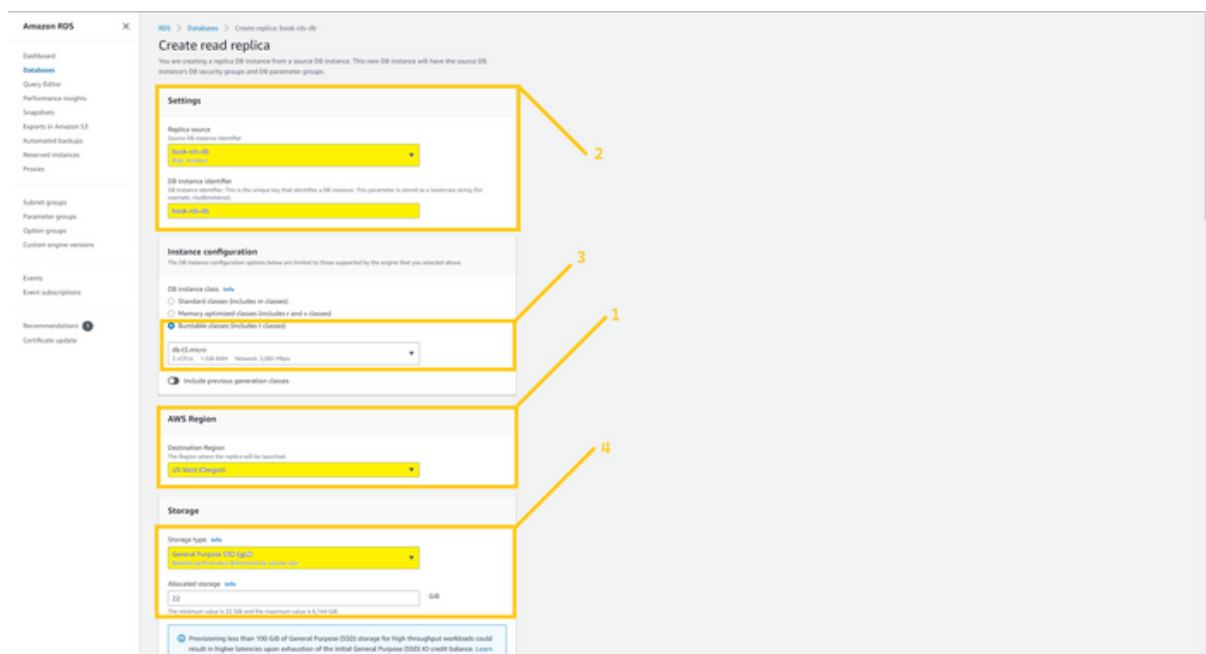
- Supports database size up to 64 TB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replica cross-region.

**Note:** RDS take 15-20 minute because it creates a database and then take a snapshot. So please have patience and wait for it to be ready.

After your database is completely ready and you see the status Available then select the database and click on the Action button. There you can see the drop-down list. Please click on created read-replica.

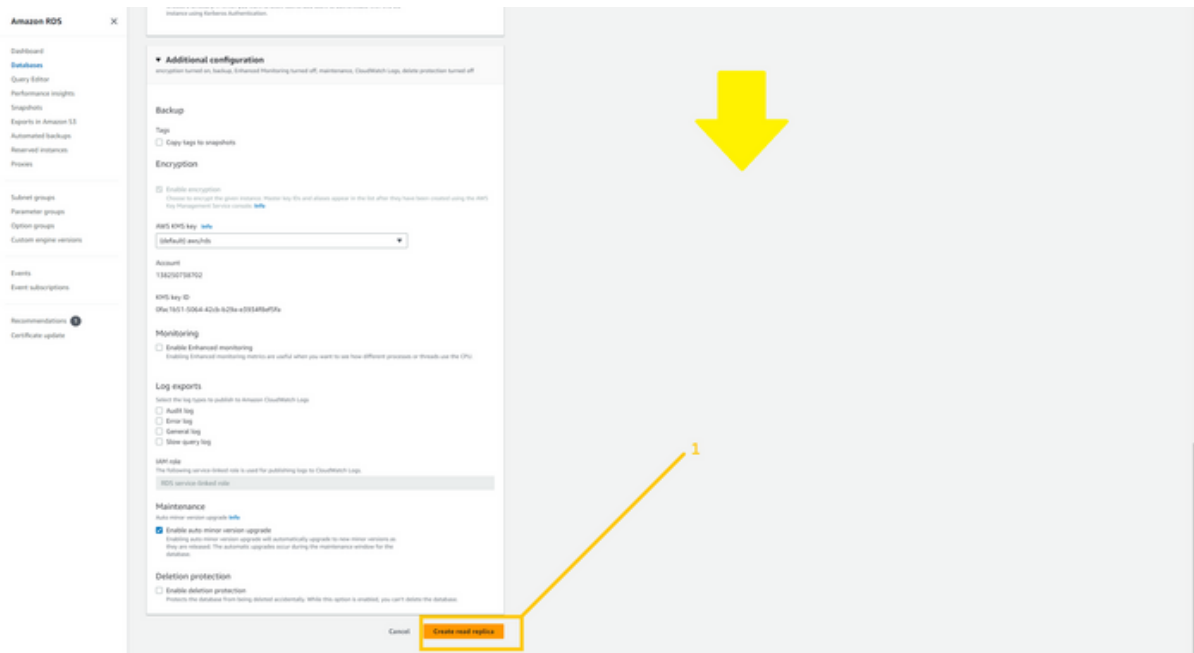


This page is similar to creating a database. In the AWS region select the region where you want to create the read replica. In my case, It is **Oregon (us-west-2)**. Give a name to your read replica, and select all the necessary configurations that we did before while creating the database. For your reference, I have shown everything in the below images.

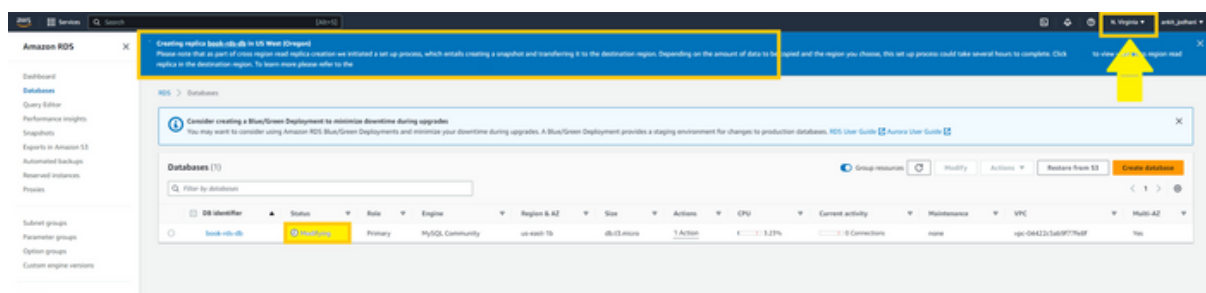




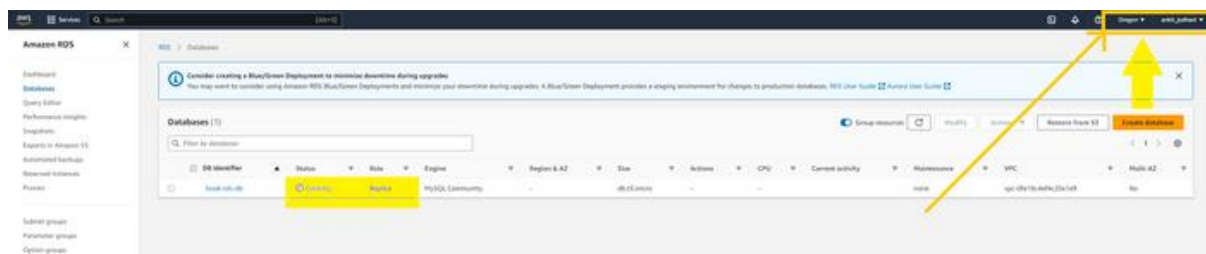




Once you click on the button **create replica**. It will start creating that

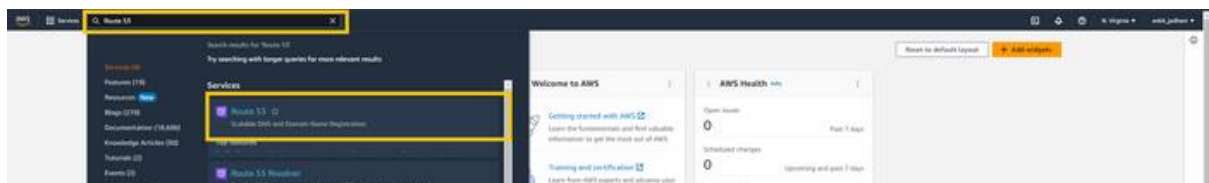


You can check your read replica on the specified region's RDS dashboard. So let me head over to **Oregon** and show you the read replica.

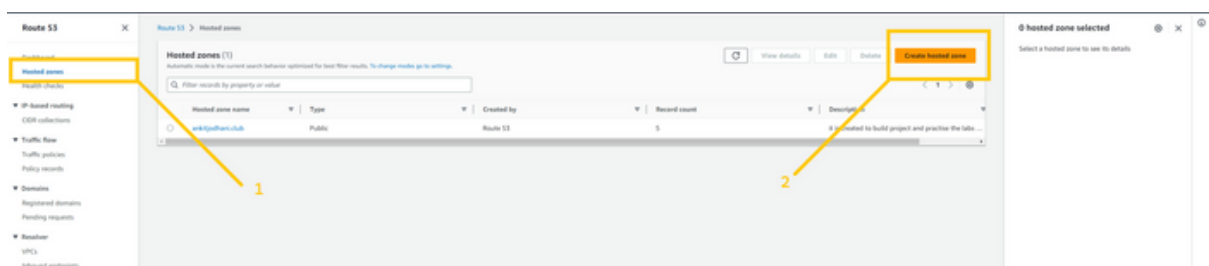


**Note: we can't write anything into a read replica. It is just read-only database. So when a disaster happens we just have to promote read replica so that it becomes the primary database in that region.**

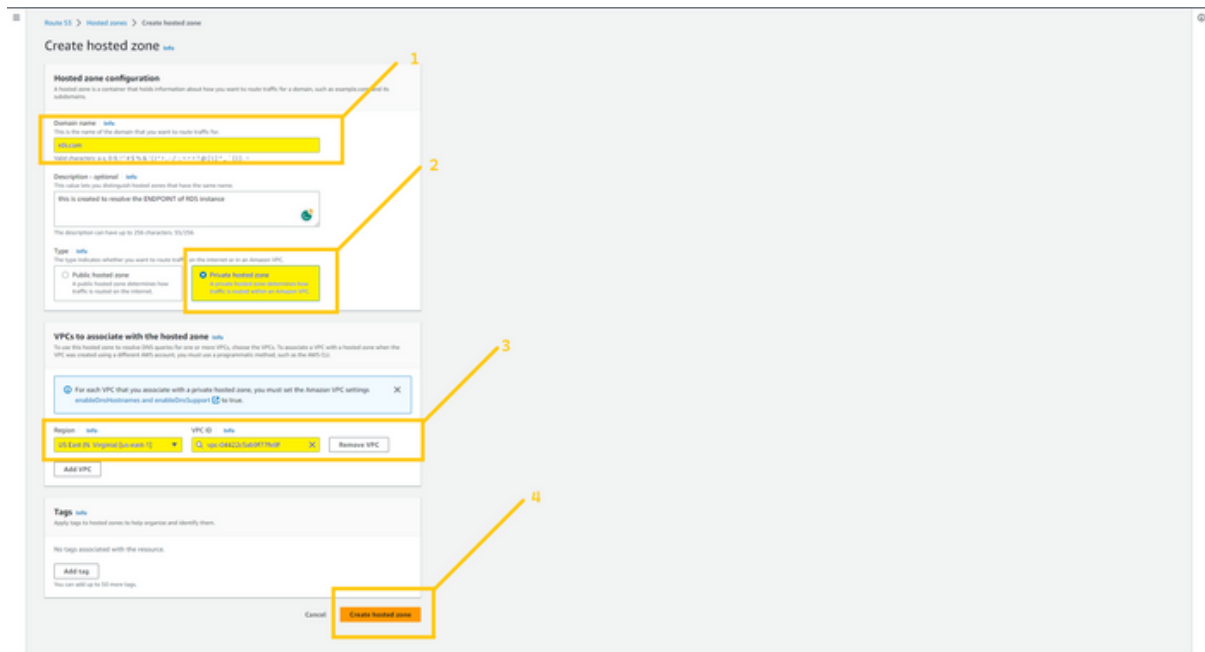
Now we are going to utilize route 53 service and create two private hosted zone. One for **north Virginia(us-east-1)** and another one for **Oregon region (us-west-2)** with the same name. you may think Why Two hosted zone with the same name? don't worry I will answer it later. So head over to Route 53. Type route 53 in the AWS console. And click on the service.



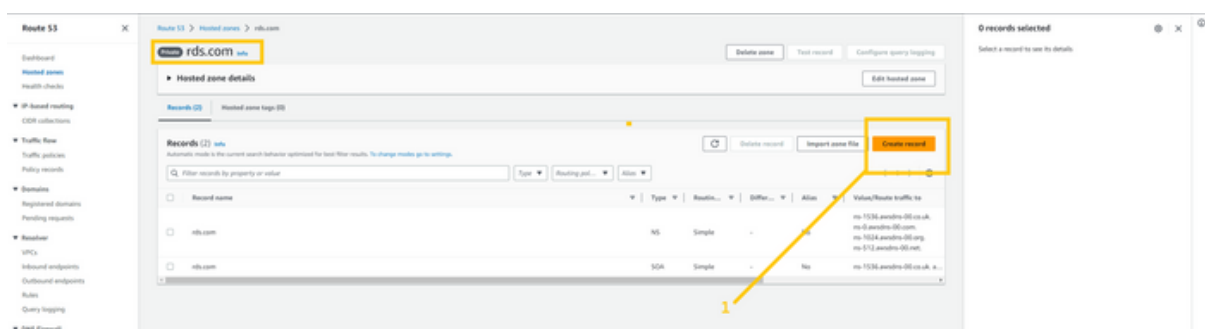
Firstly, we are gonna create a hosted zone for **us-east-1**. Click on the Hosted Zones button on the left panel and click on the created hosted zone button on the top right corner.



Give any domain name because anyhow it will be private hosted zone but it would be great if you give the name same as mine (**rds.com**). Please select the private hosted zone and Select the region. In my case, it is **us-east-1**. And then select VPC ID. Make sure you select VPC that we created earlier. Because this hosted zone will resolve the record only in specified VPC. and then click on the Create hosted zone.



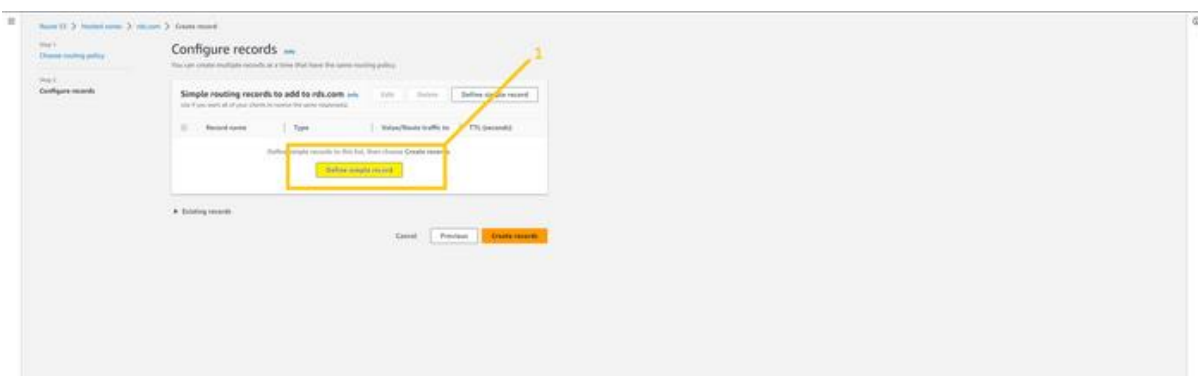
Now we are going to create a Record that points to our RDS instance which is in **us-east-1**. So click on create record button on the top right corner.



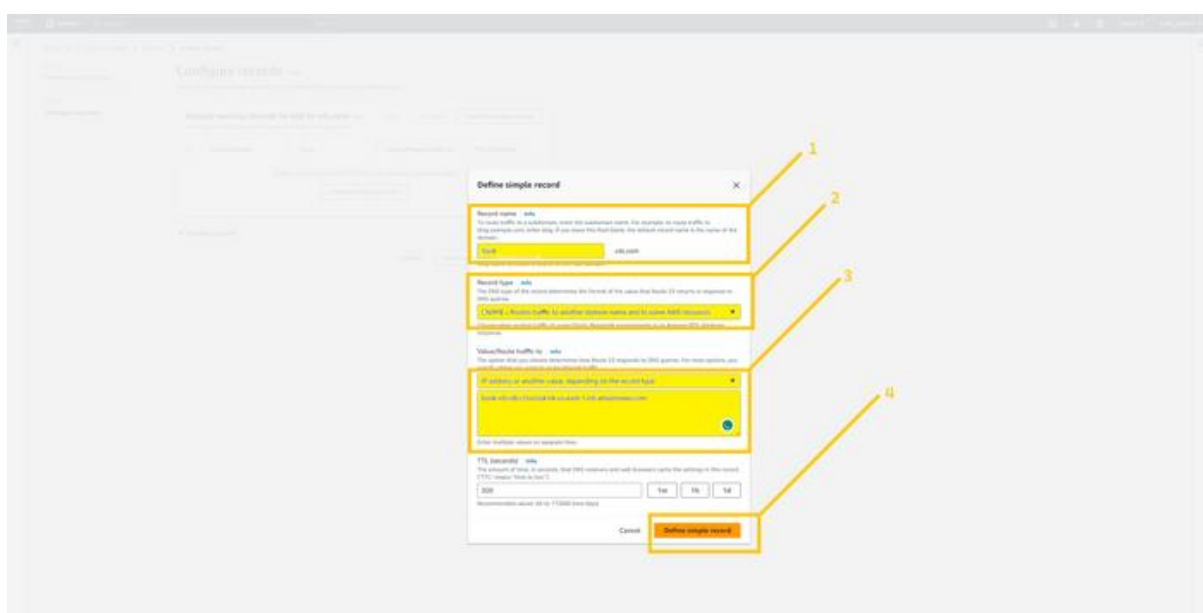
Select **simple routing**,



Click on the defined record button in the middle of the box.



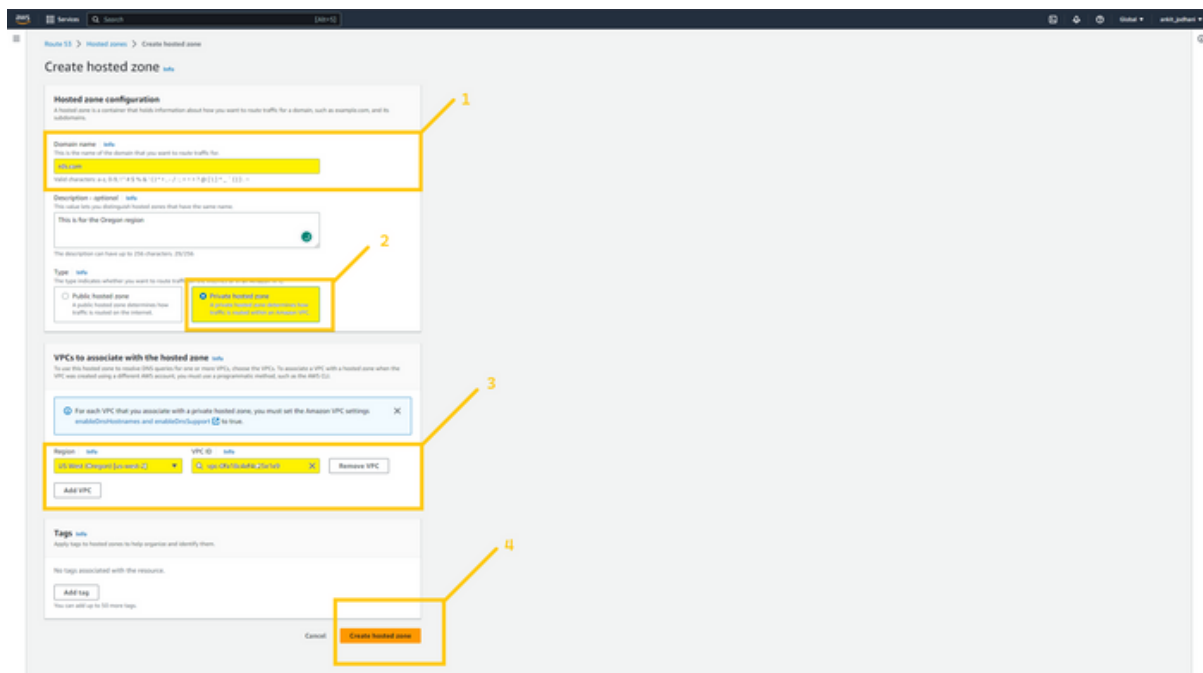
Here types book in the record name field. In the record type select CNAME. In the value field paste **endpoint of the RDS which is in us-east-1**. Then click on the defined record button.



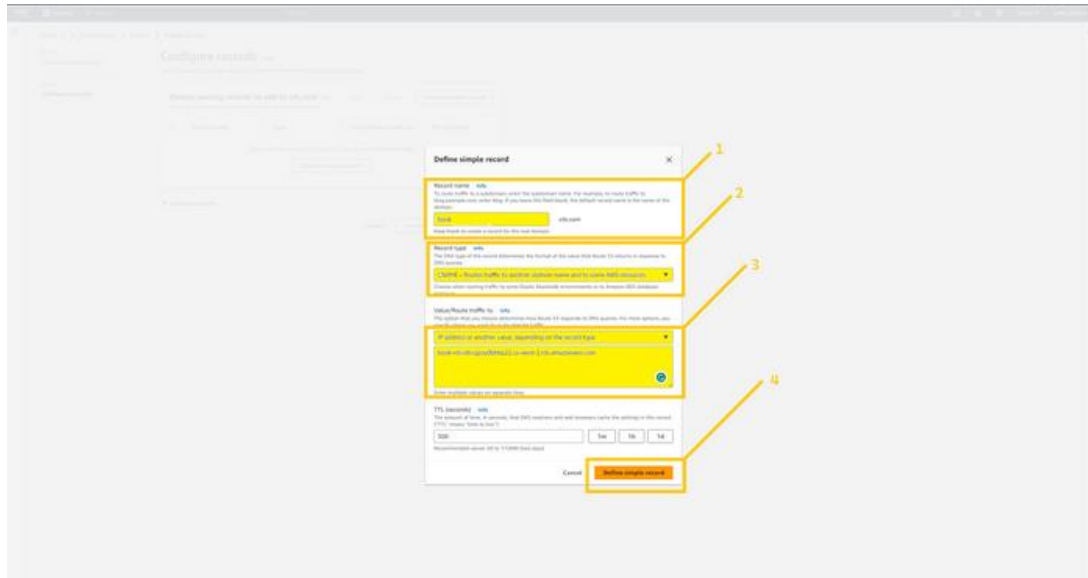
Click on **create record** button.



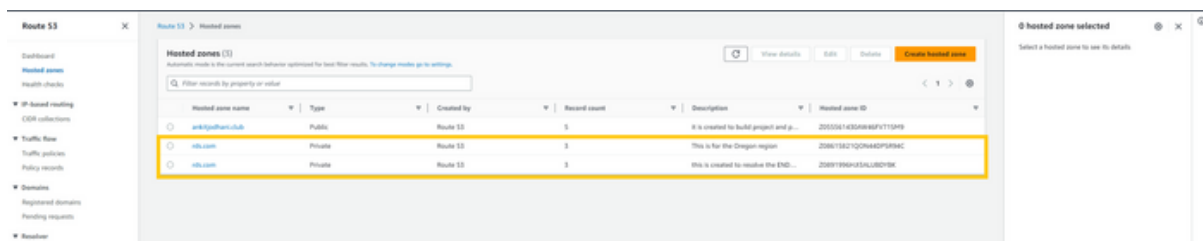
Now we are going to create a new hosted zone with the same name, but for **disaster recovery region** and that is **us-west-2 (Oregon)**. While creating hosted zone please keep in mind that you need to choose the **us-west-2** region and select VPC that you have created in the **the us-west-2** region. Again you can utilize the below image for reference.



Our next step is to set up a **simple record** that points to the **read replica** (database) which is in the **us-west-2 (Oregon)**. So select the hosted zone that was created for **us-west-2** and defined a simple record in that. Everything is the same as we defined the record in the us-east-1 hosted zone.



After successfully completing the above steps your Route 53 console look like this.



You may think that We can connect two regions and VPC in one hosted zone then why two private hosted zone with the same name? And the answer is Endpoint of both databases will be different and we can't implement a health check coz we attached a security group that allow traffic from **3306 port from only backend SG**. So route 53 can't check the health of databases and because of that we can't implement a Failover record here but we will do that with the application server (backend-server).

And here we successfully completed our RDS setup. Let's go ahead and explore more services