General Immersion Day

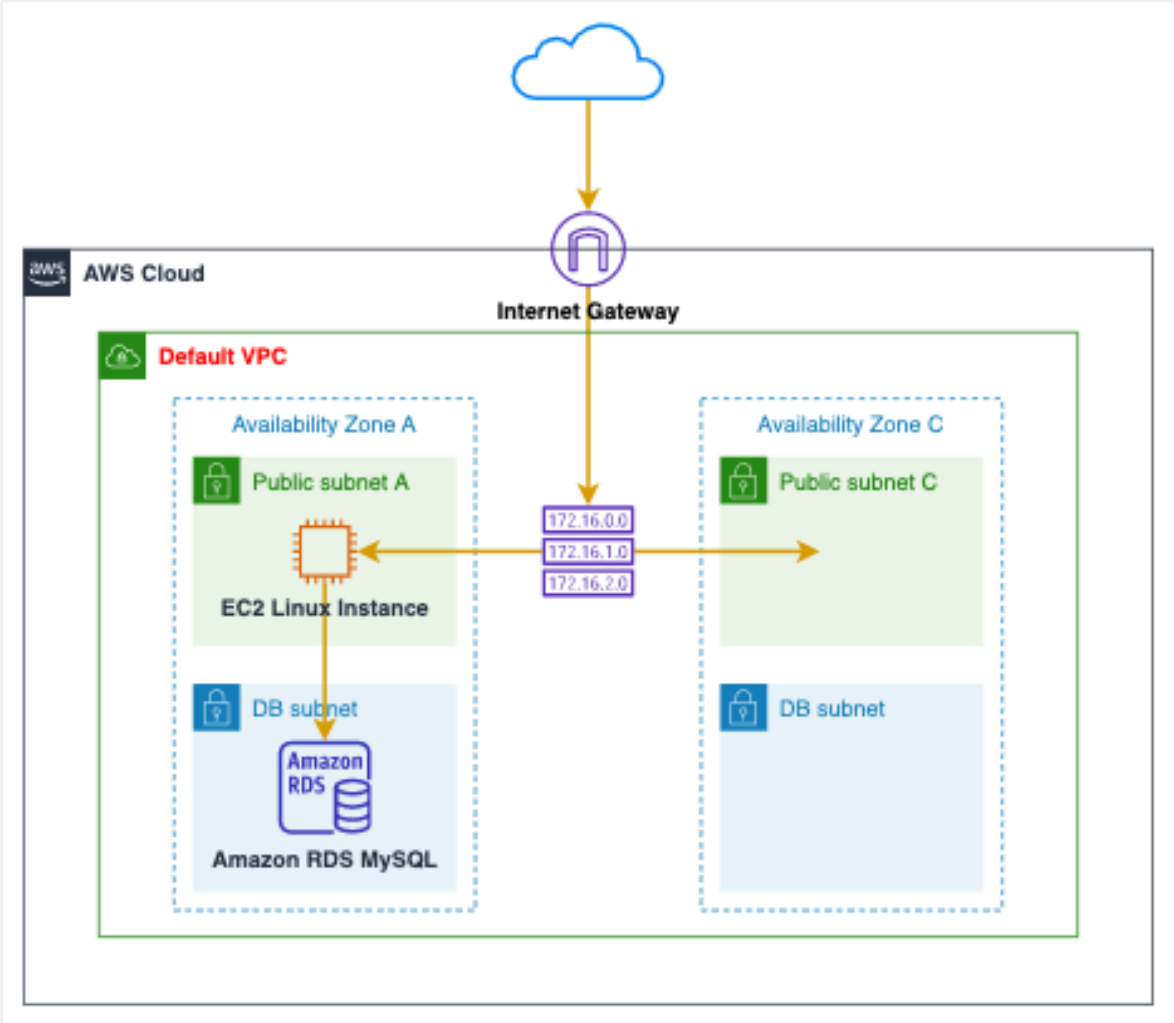
Lab 5

Amazon RDS MySQL Hands on Lab

Amazon RDS Overview

Amazon RDS is a web service that makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks, freeing you up to focus on your applications and business.

Note: This lab requires EC2 Linux Hands-On Lab in advance to complete. This lab will make use of the web server previously created in EC2 lab to connect RDS MySQL.



This lab will walk you through the following:

1. Create VPC Security Group
2. Launch an RDS Instance
3. Save RDS Credentials
4. Access RDS from EC2
5. Create an RDS Snapshot (Optional)
6. Modify RDS Instance Size (Optional)

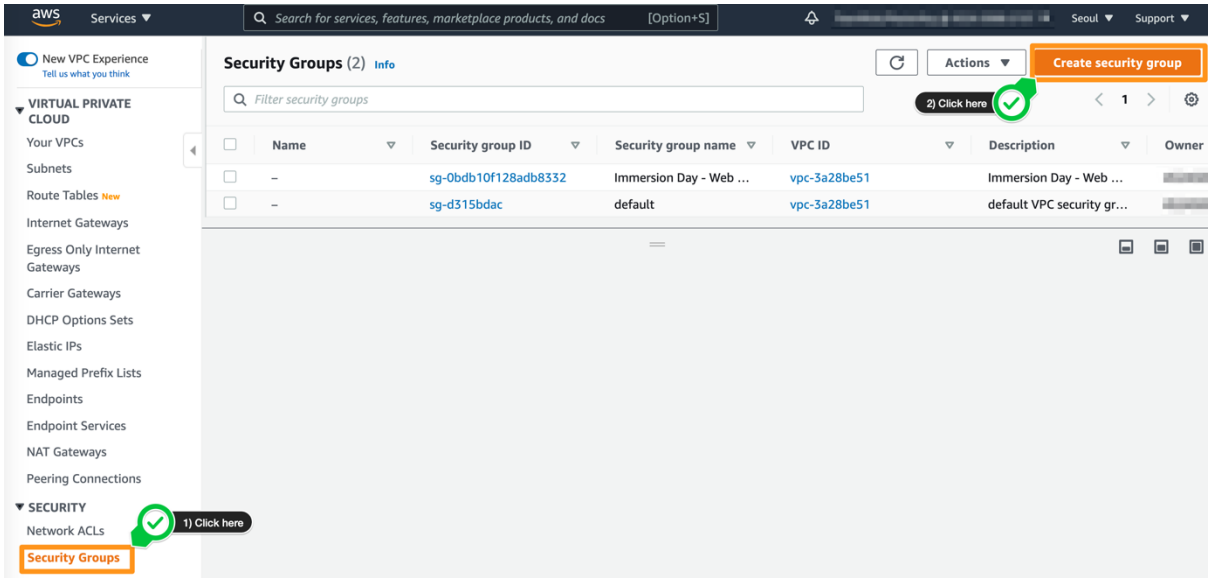
5-1 Create VPC Security Group

Prerequisite: EC2 Linux Hands-On Lab

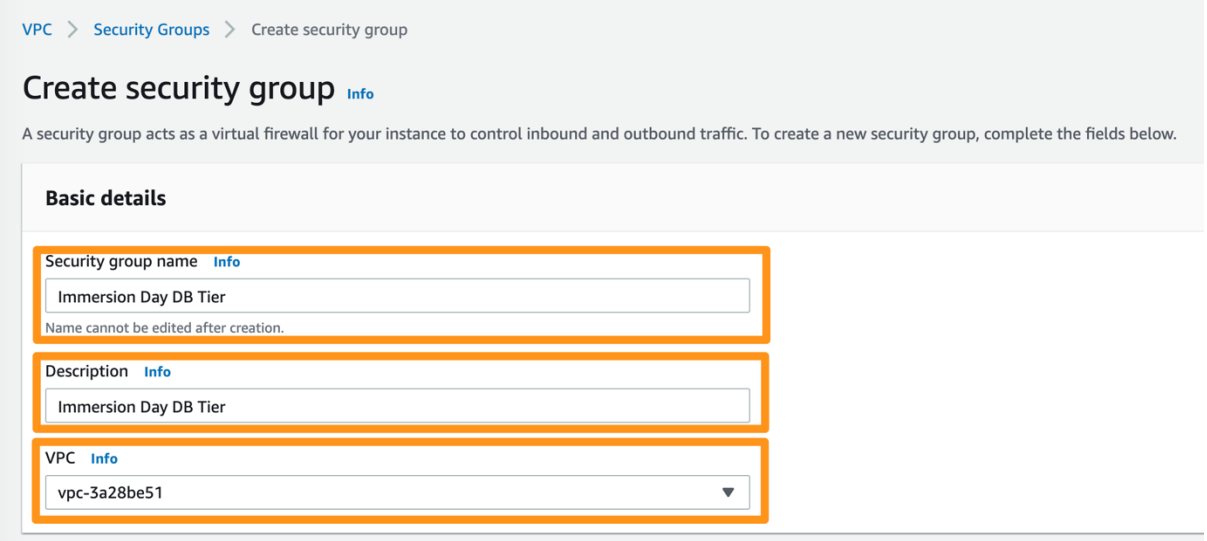
In EC2 Linux Hands-On Lab, we launched a web server EC2 instance with the security group, **Immersion Day - Web Server**, that allows TCP 80 for the web server.

First, we will create a new VPC security group, Immersion Day - DB Tier, for our database tier that only allows traffic from our web tier.

1. In the VPC dashboard, click **Security Groups**, then the **Create Security Group** button.



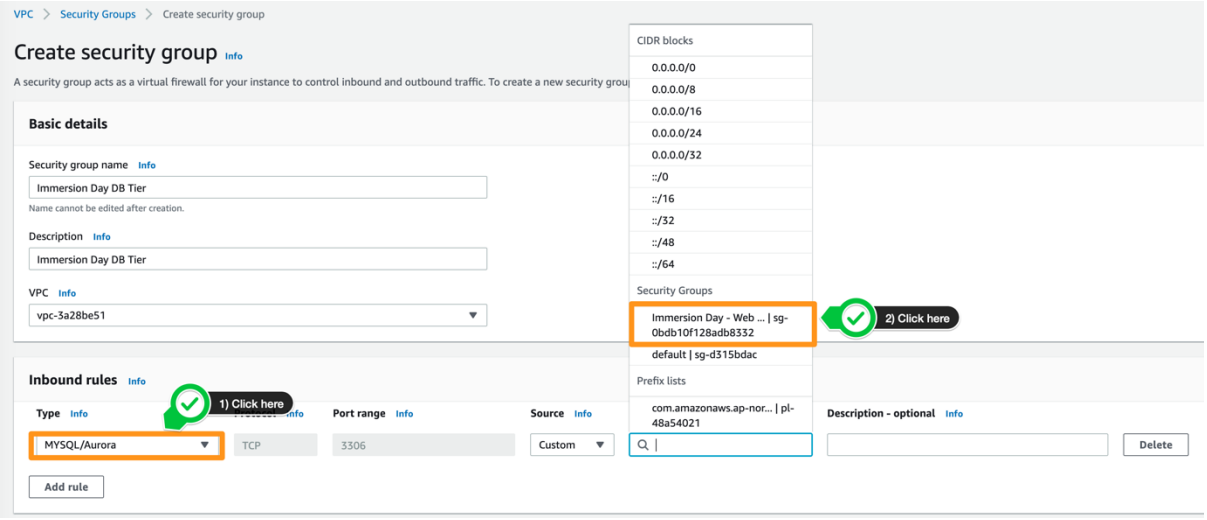
2. Type Security group name and Description as below and keep the VPC setting to the same VPC you've launched your EC2 instance in.



KEY	VALUE
Security group name	Immersion Day DB Tier
Description	Immersion Day DB Tier
VPC	VPC-xxxxxx (default)

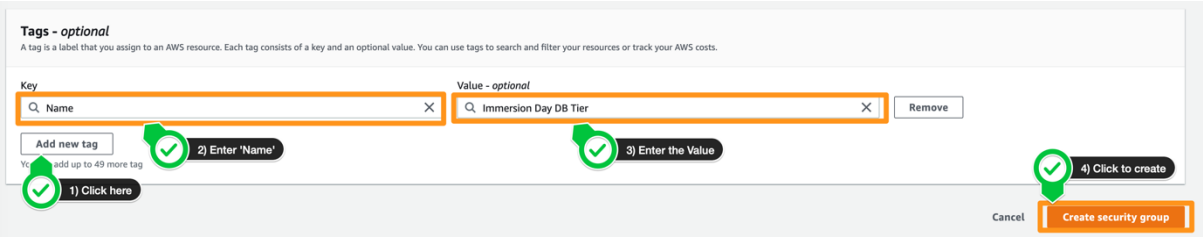
3. Under **Inbound Rules**, click **Add rule** button.

4. Add a new inbound rule for the EC2 server(s) in our web tier. The type should be **MySQL/Aurora (3306)**, the protocol **TCP (6)**, and in the source box, type the name of the security group to which your EC2 instance belongs. While you're typing, a list of security group(s) that match that name should be presented to you. Select your security group.



5. Set Name tag and group name to Immersion Day DB Tier

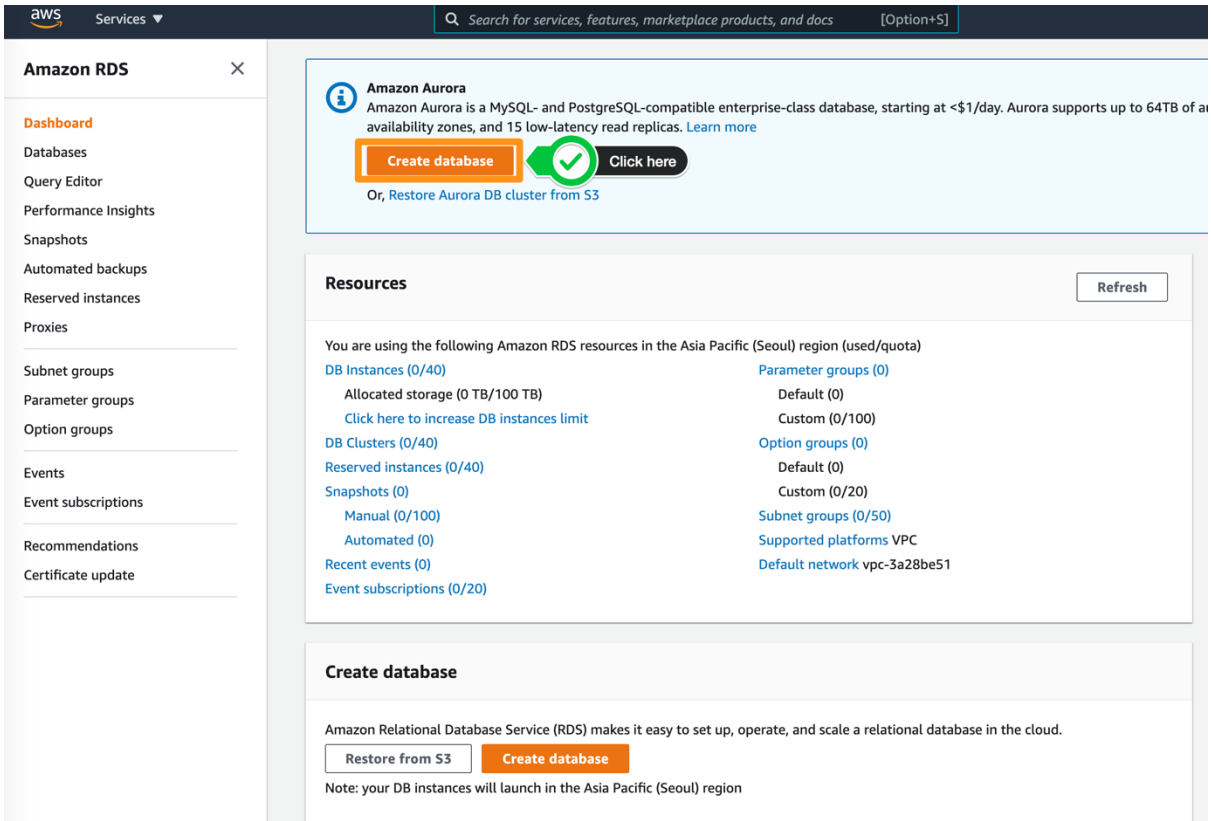
6. Then, scroll down and click on Click **Create security group** button. This will create the Security group for your RDS instance.



5-2 Launch an RDS Instance

Now that our VPC security group for Database is ready, let's configure and launch a MySQL RDS Instance.

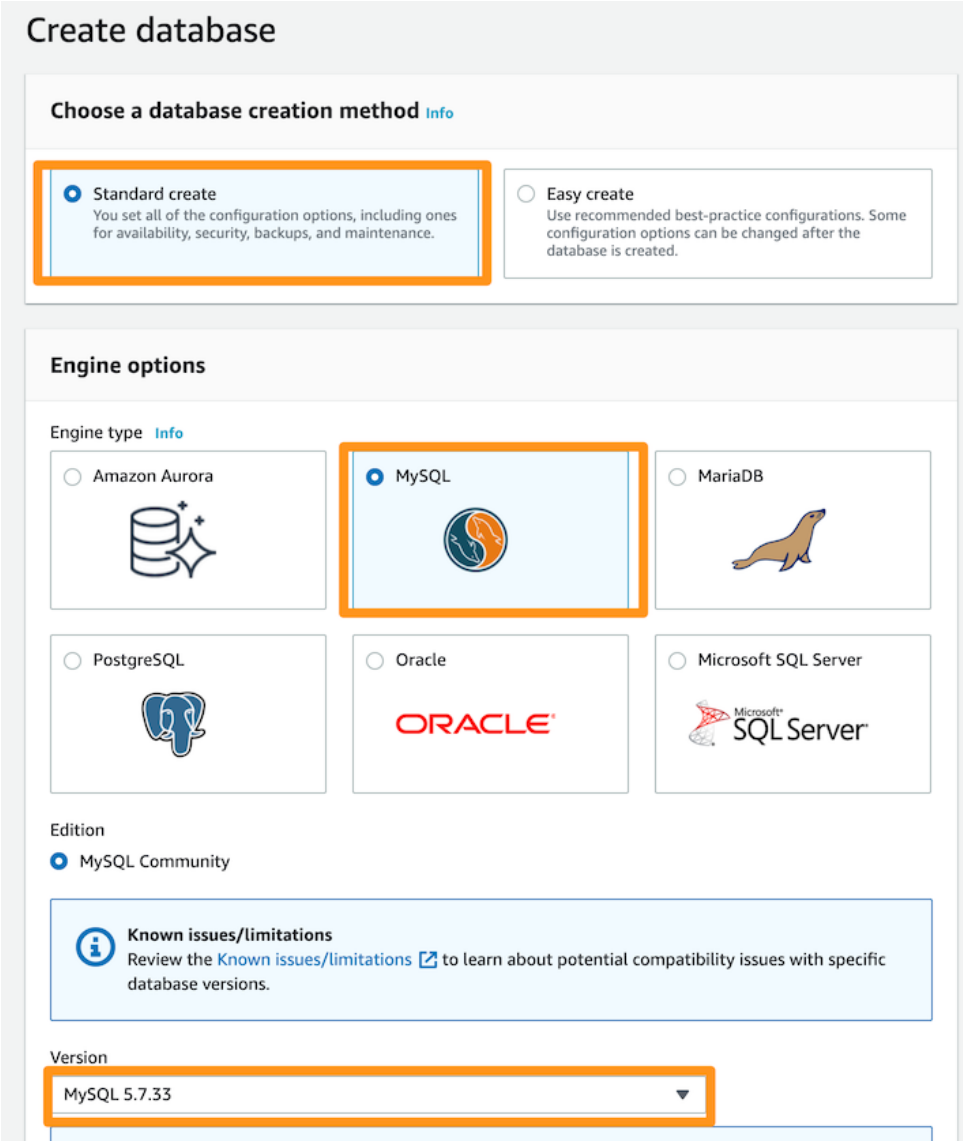
1. Sign into the AWS Management Console and open the [Amazon RDS console](#)
2. Click on **Create database**



3. For **Choose a database creation method**, select Standard option. With Standard Create, you setup the configurations for your database.

Note: Easy Create option provides recommended best-practices configurations to get started with deploying databases.

4. Select **MySQL** in **Engine Options**.

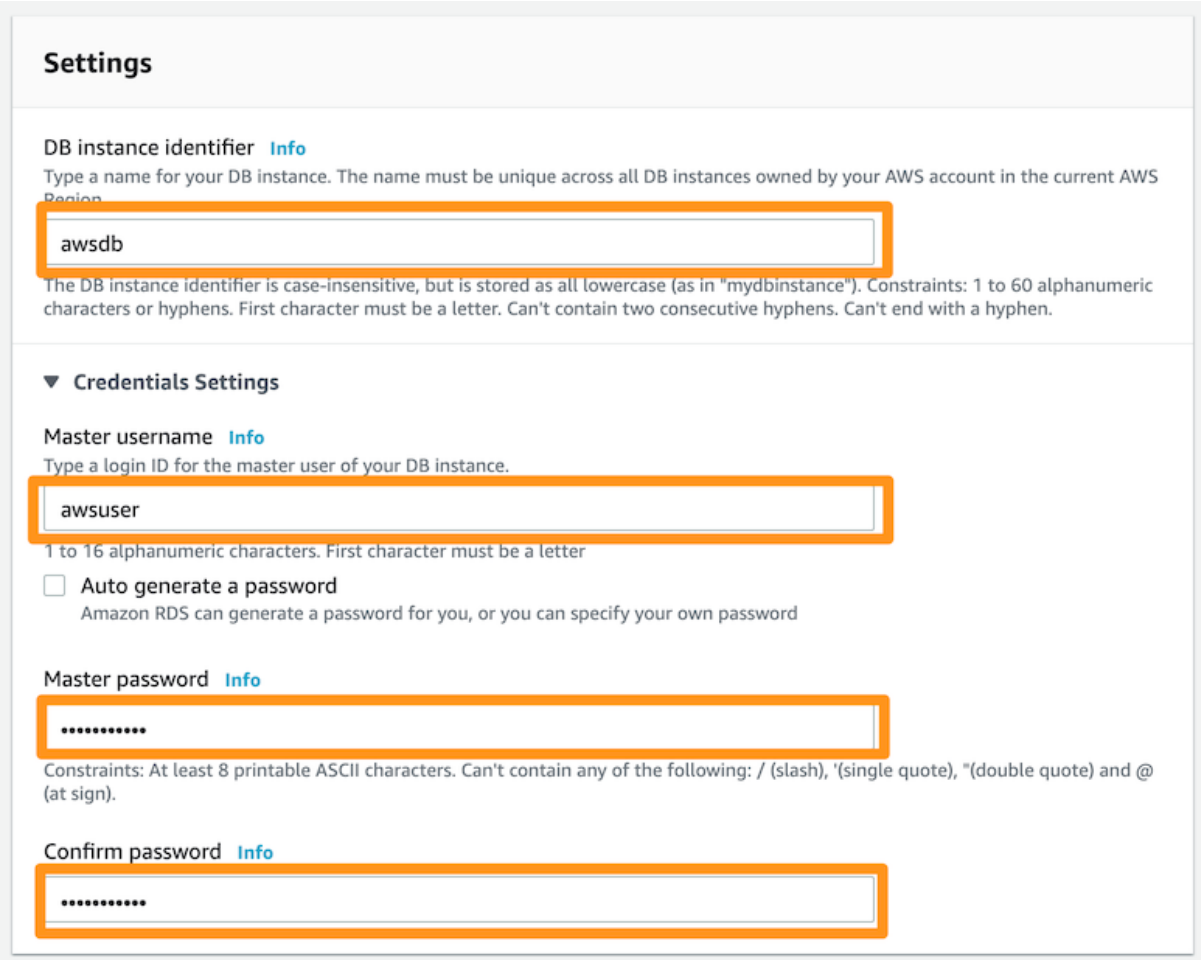


5. When you select MySQL as your database engine, the latest version will be automatically selected for you. For this lab, select **MySQL version 5.7.X.**

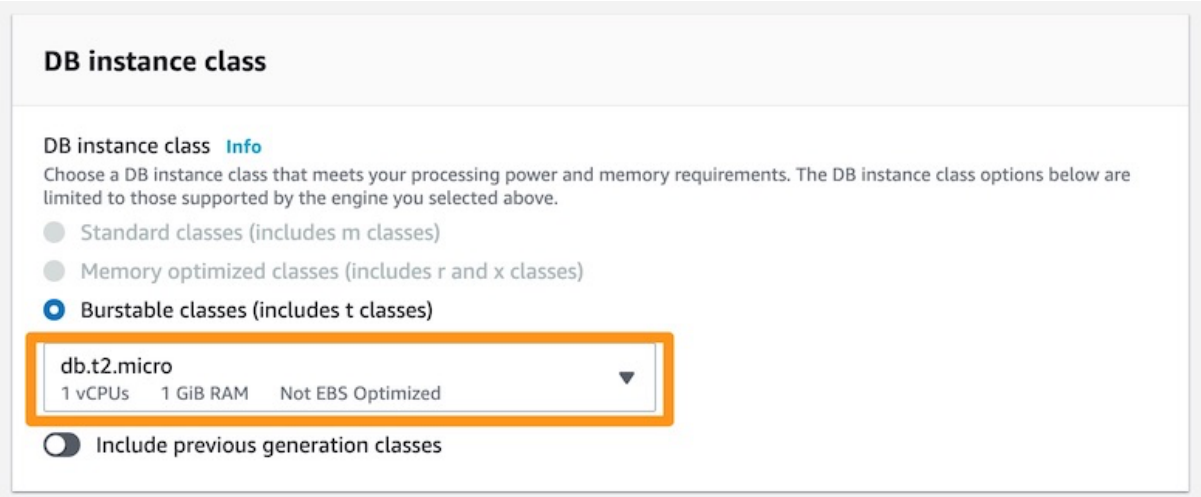
6. For Template, there are three options available: Production, Dev/Test and Free Tier. For the lab purpose, we will select **Free Tier**.

7. In **Settings** section, fill in the following for each field

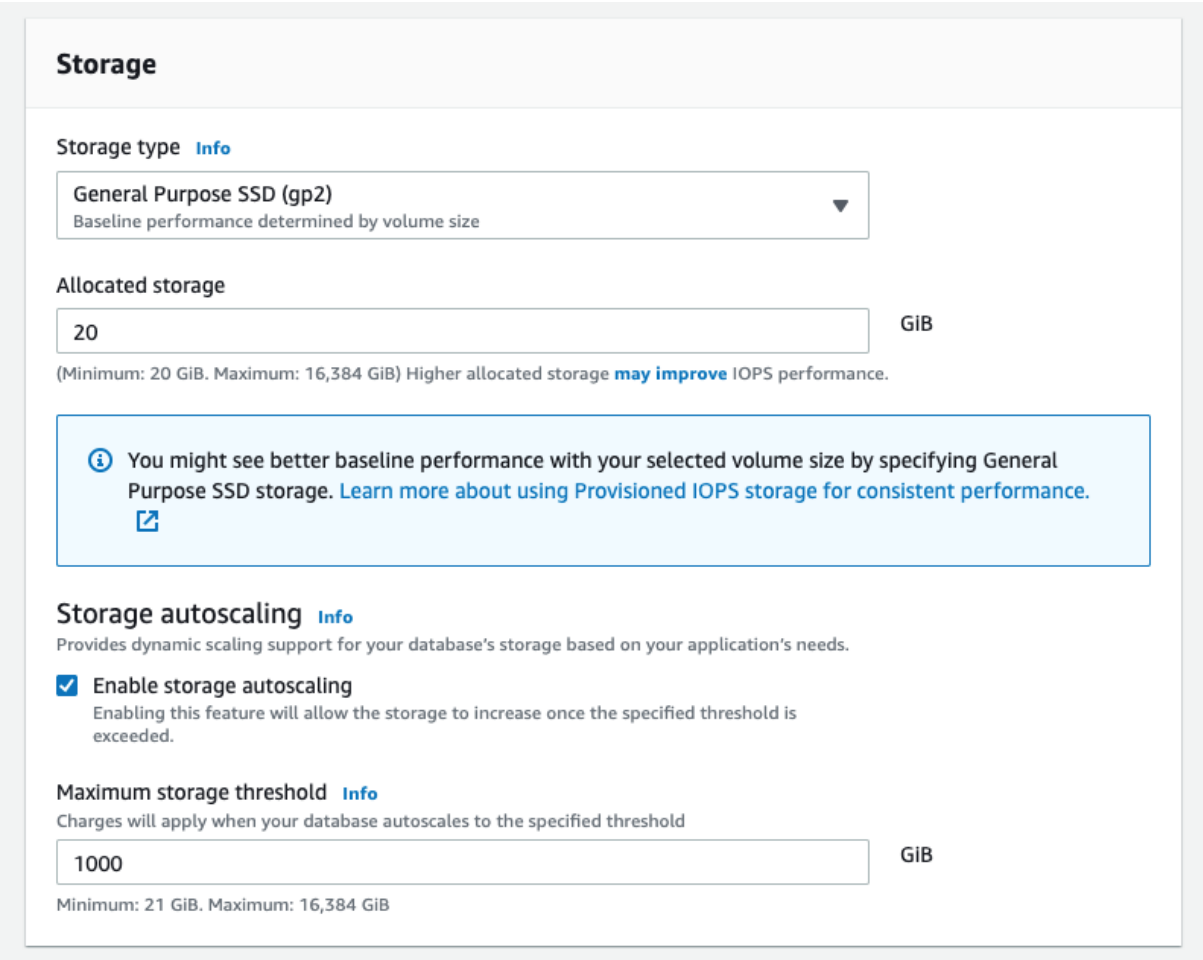
Parameter	Value
DB Instance Identifier	awsdb
Master Username	awsuser
Master Password	awspassword



8. In **DB Instance size** section, for **DB instance class**, select **burstable classes-db.t2.micro**. This option will be automatically selected for you.

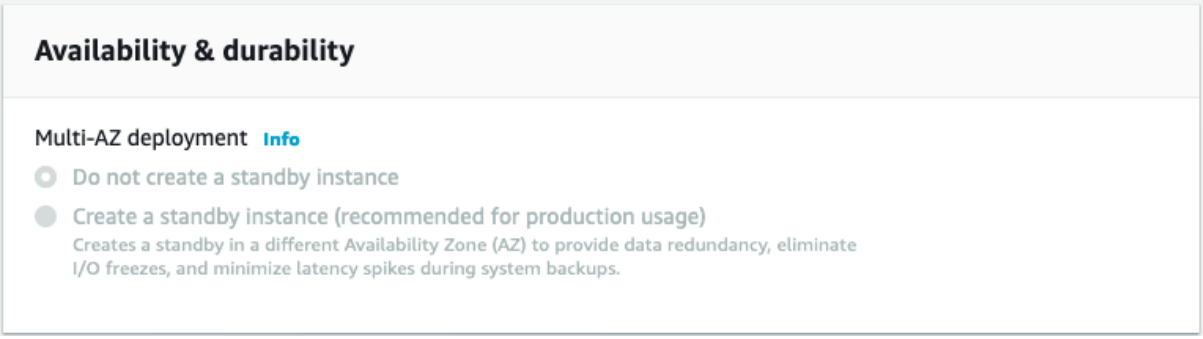


9. In the **Storage** section, select the **Storage Type** as **General Purpose SSD**. You can select or deselect the option for Auto Scaling for the lab purposes.



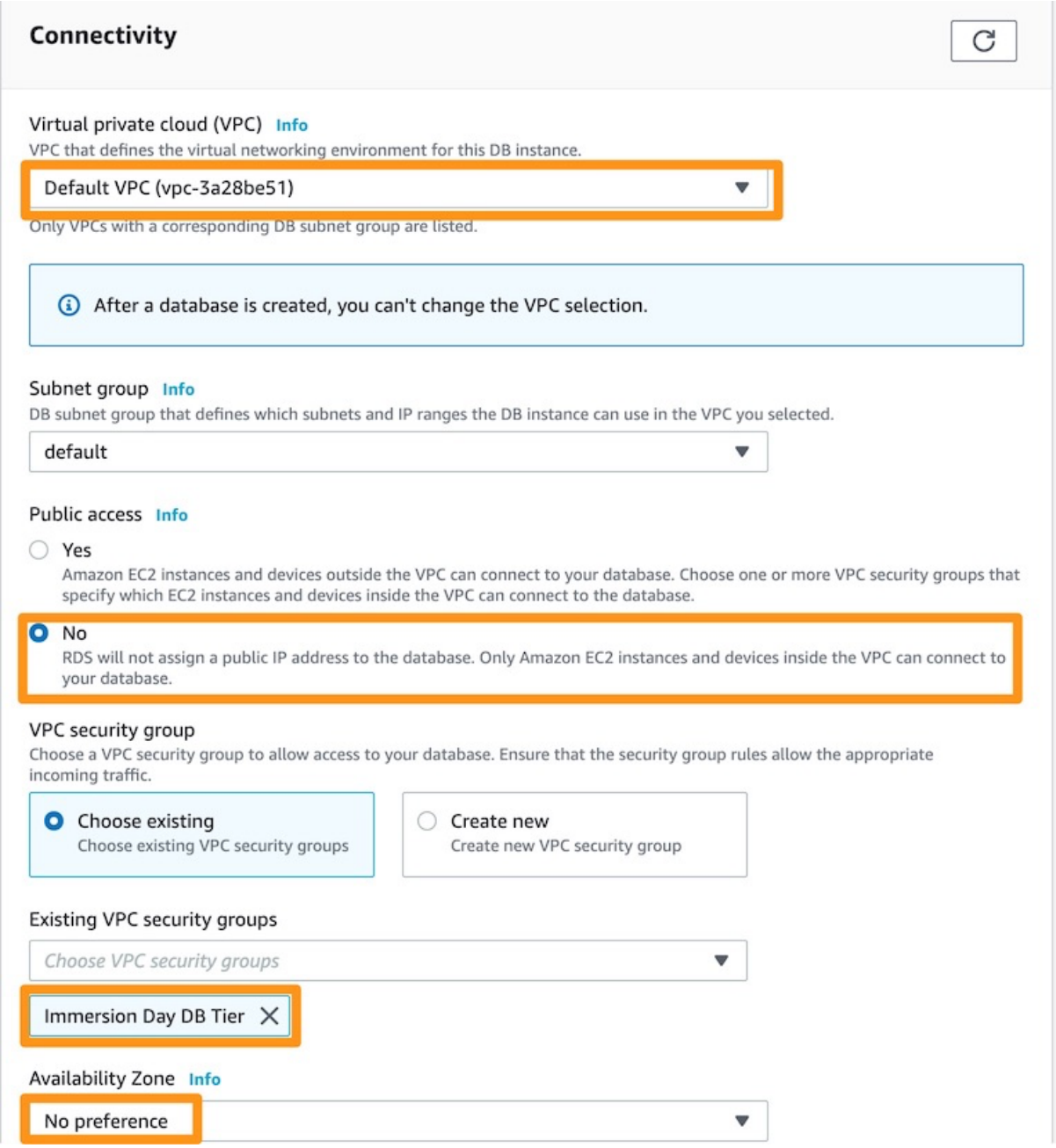
10. Since we selected the Template option as Free Tier-used only for doing hands-on or testing the applications, Multi-AZ deployment is not required and hence, the Availability and Durability section will be disabled for you.

Note: For a database used in Production and Dev/Test, we recommend using a **Multi-AZ Deployment**.



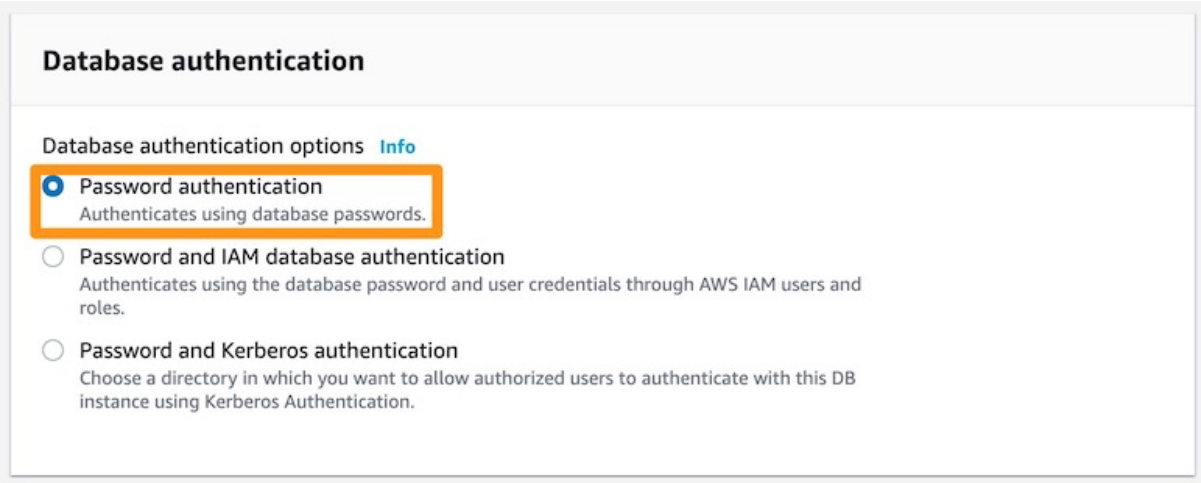
11. In the **Connectivity** section:

Parameter	Value
VPC	Default VPC
Additional connectivity configuration	
Subnet Group	default
Publicly accessible	No
VPC Security Group(s)	Select Choose existing VPC security groups, then pick Immersion Day DB Tier
Availability Zone	No preference
Database port	3306



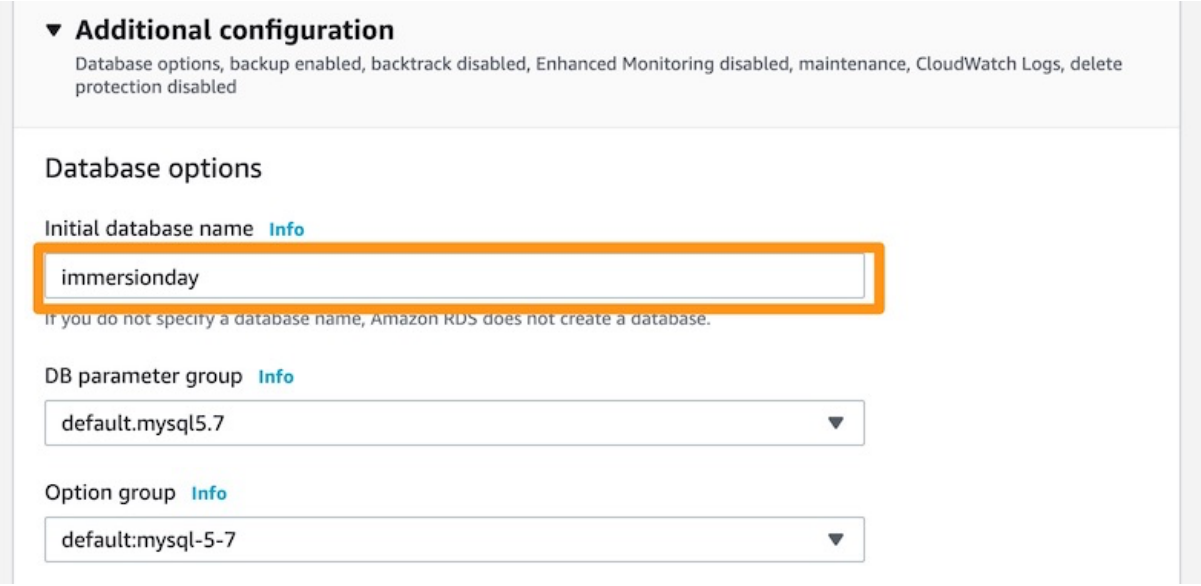
12. For **Database authentication**, there are two options to select from. **Password Authentication** will authenticate the user only with the database password. With **Password and IAM Database authentication**, the user will be authenticated with the database password and also with the user credentials through IAM roles and policies.

For this lab, we will select: Password Authentication.

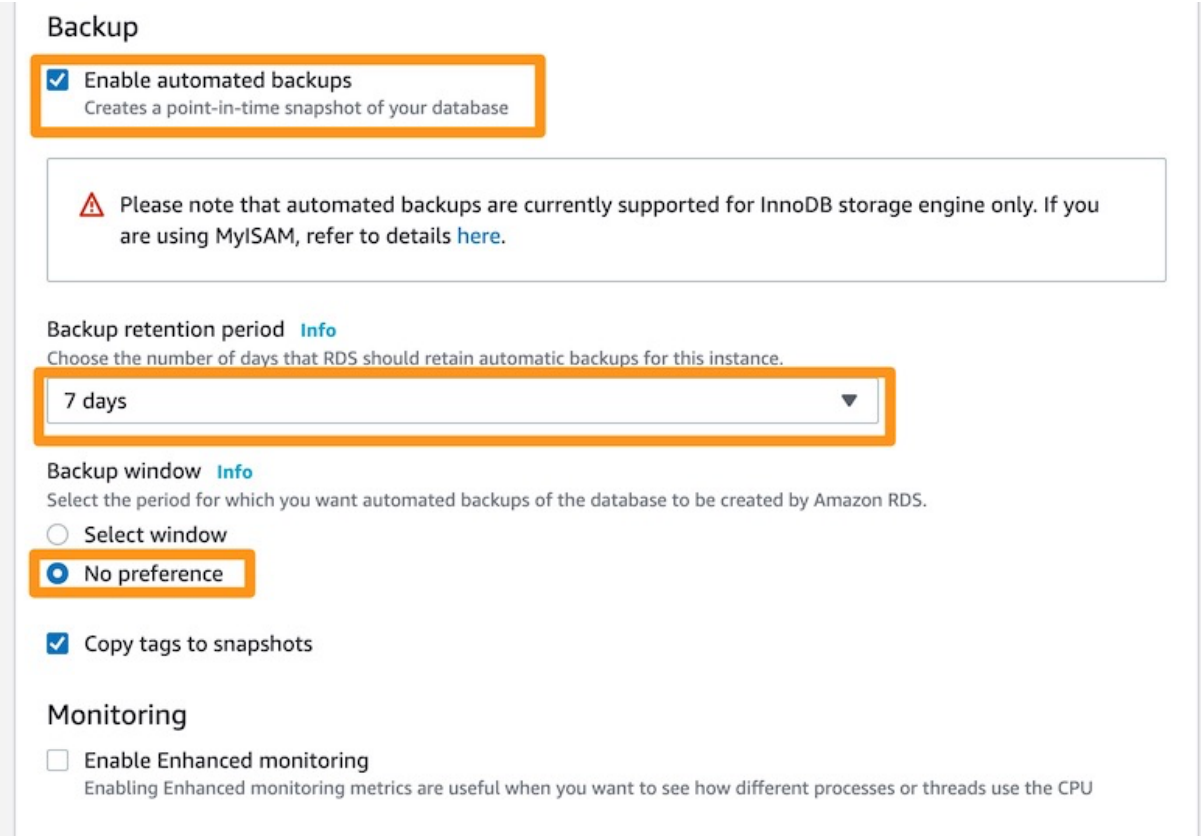


13. Expand on **Additional Configuration**.

- For the Database options, provide the following:
  - Initial Database name: **immersionday**
  - DB Parameter group and Option group: default.mysql5.7



- For Backup:
  - Check on **enable automatic backups**.
  - Provide **Backup retention period** as **7 days**.
  - **Backup Window: No preference**
  - Leave rest as defaults



- For **Log exports**, you can select from various options that which type of logs you would like to analyze in CloudWatch. **Leave as default**.
- For **Maintenance**, leave as defaults. The default options will be auto check on enable auto minor version upgrade and maintenance window will be selected as **No preference**.
- For **Deletion protection**, if checked, it protects your database from accidental deletion and your database cannot be deleted as long as this option is checked. **Leave as default**.

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

> Ensure that General, Slow Query, and Audit Logs are turned on. Error logs are enabled by default. Learn more

Maintenance

Auto minor version upgrade Info

- [x] 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.

- ( ) Select 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.

14. At last, it will give you estimated costs for your selected configurations:

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 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.

15. Review your settings and click **Create database**.
16. In the RDS Dashboard, monitor your new DB instance until the status changes from "**creating**" to "**backing up**" to "**available**".

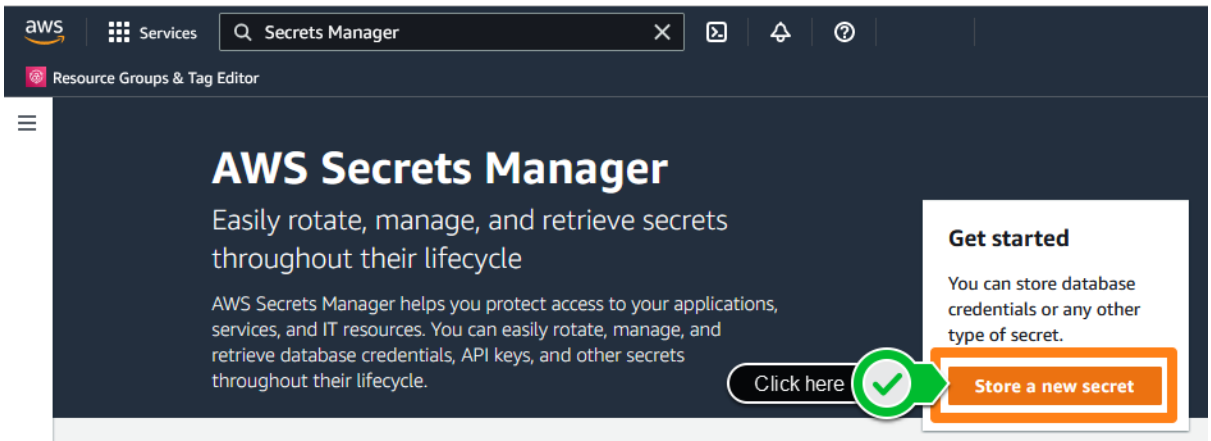
Note: This may take up to 5 minutes as the database is being created and backed up.

5-3 Save RDS Credentials

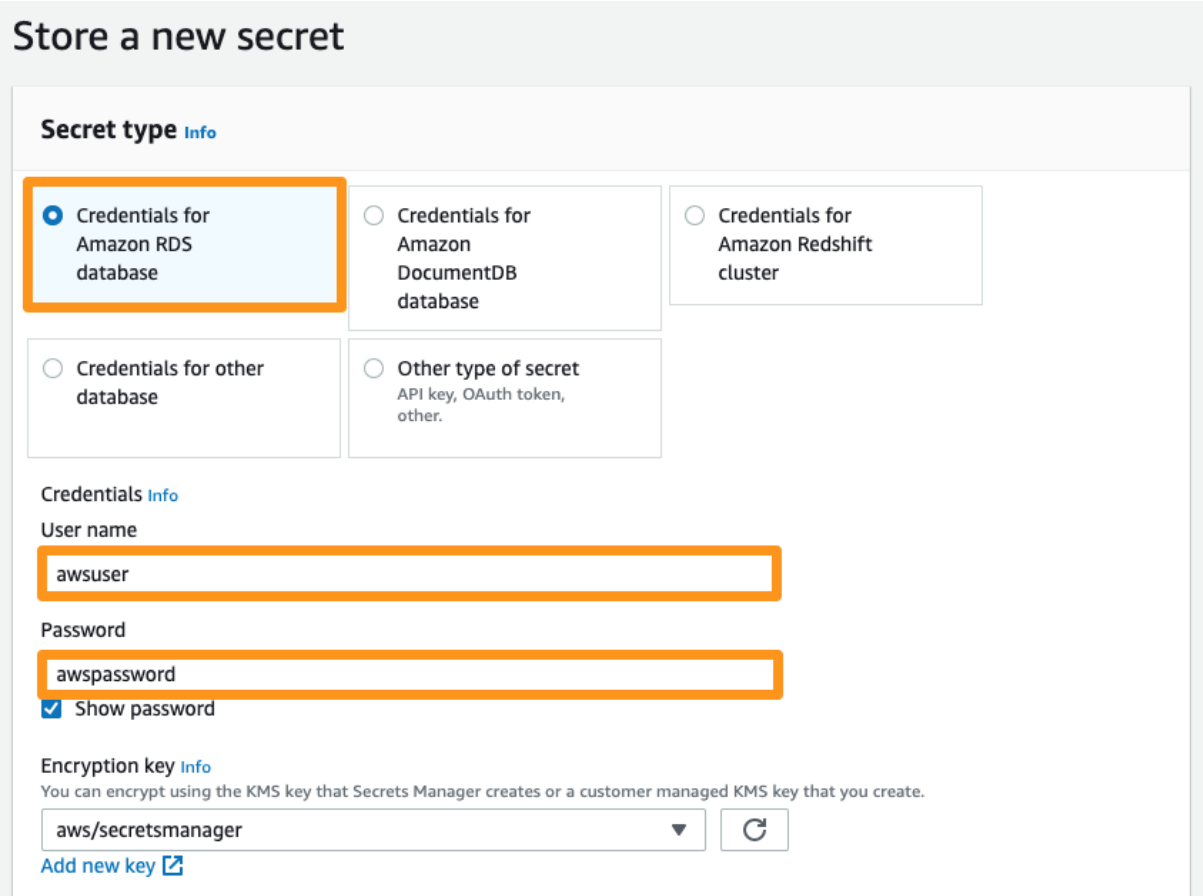
The web server you created contains sample code for a simple address book. We must tell the sample code how to find the database and connect to it. We will store this information in AWS Secrets Manager.

In this section, we will create a secret containing the database connection information. Later, we will grant permission to the web server to retrieve this secret.

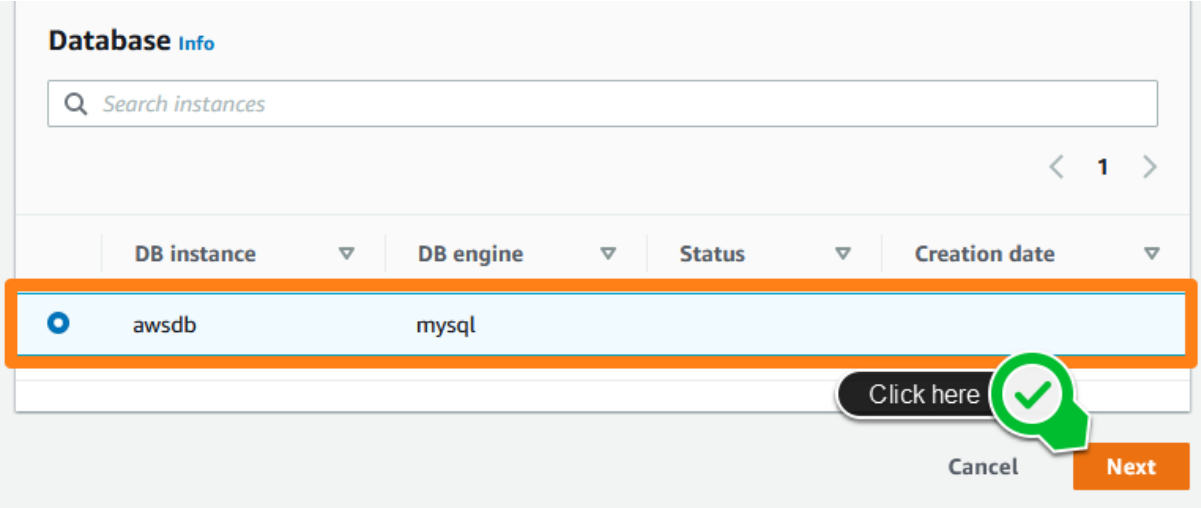
1. In the console, open the AWS Secrets Manager . Click **Store a new secret**.



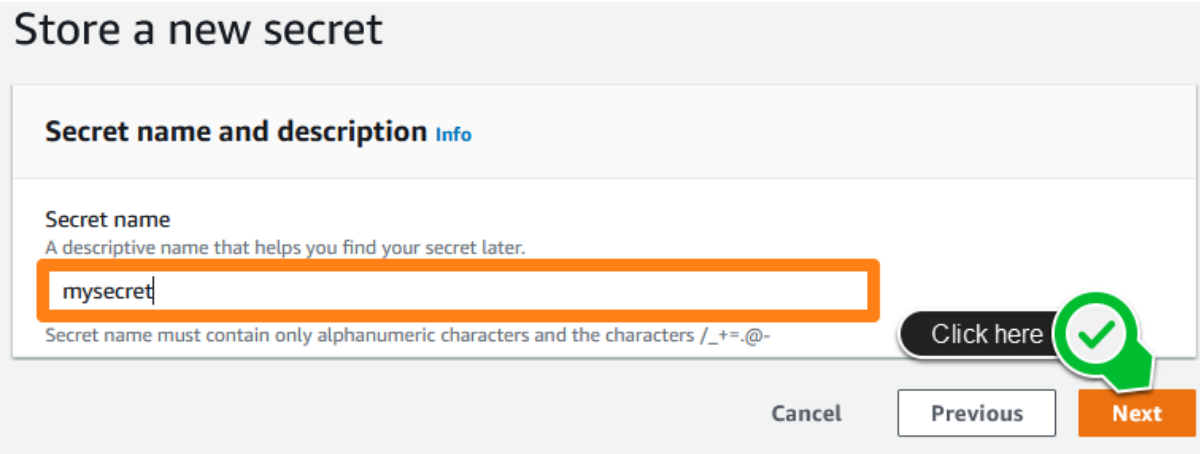
2. Under **Secret Type**, choose **Credentials for Amazon RDS database**. Provide the user name and password you entered when you created the database.



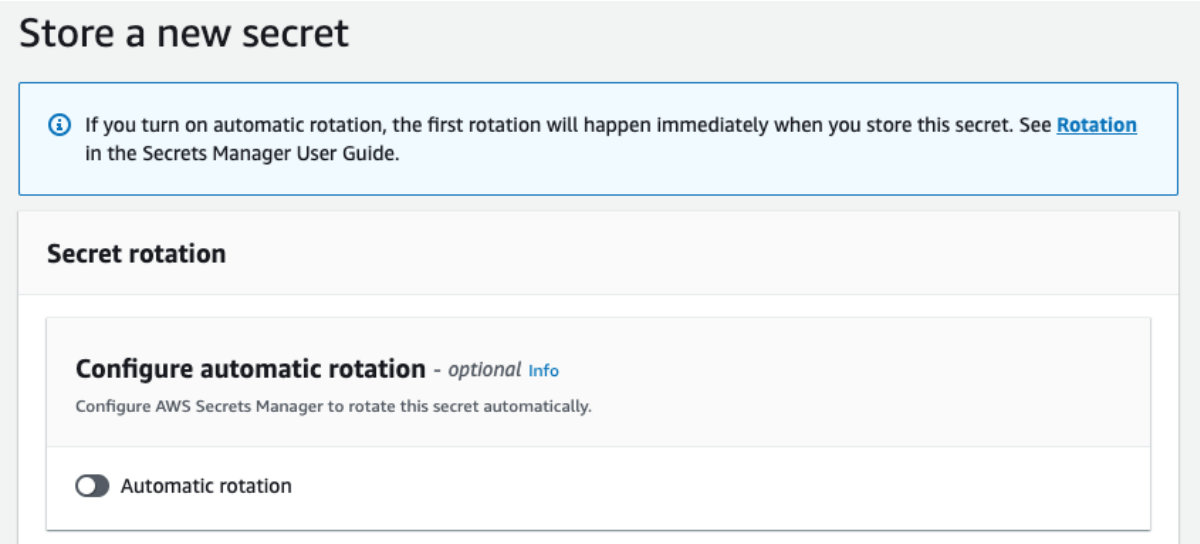
3. Under **Database**, choose the database you just created. Click **Next**.



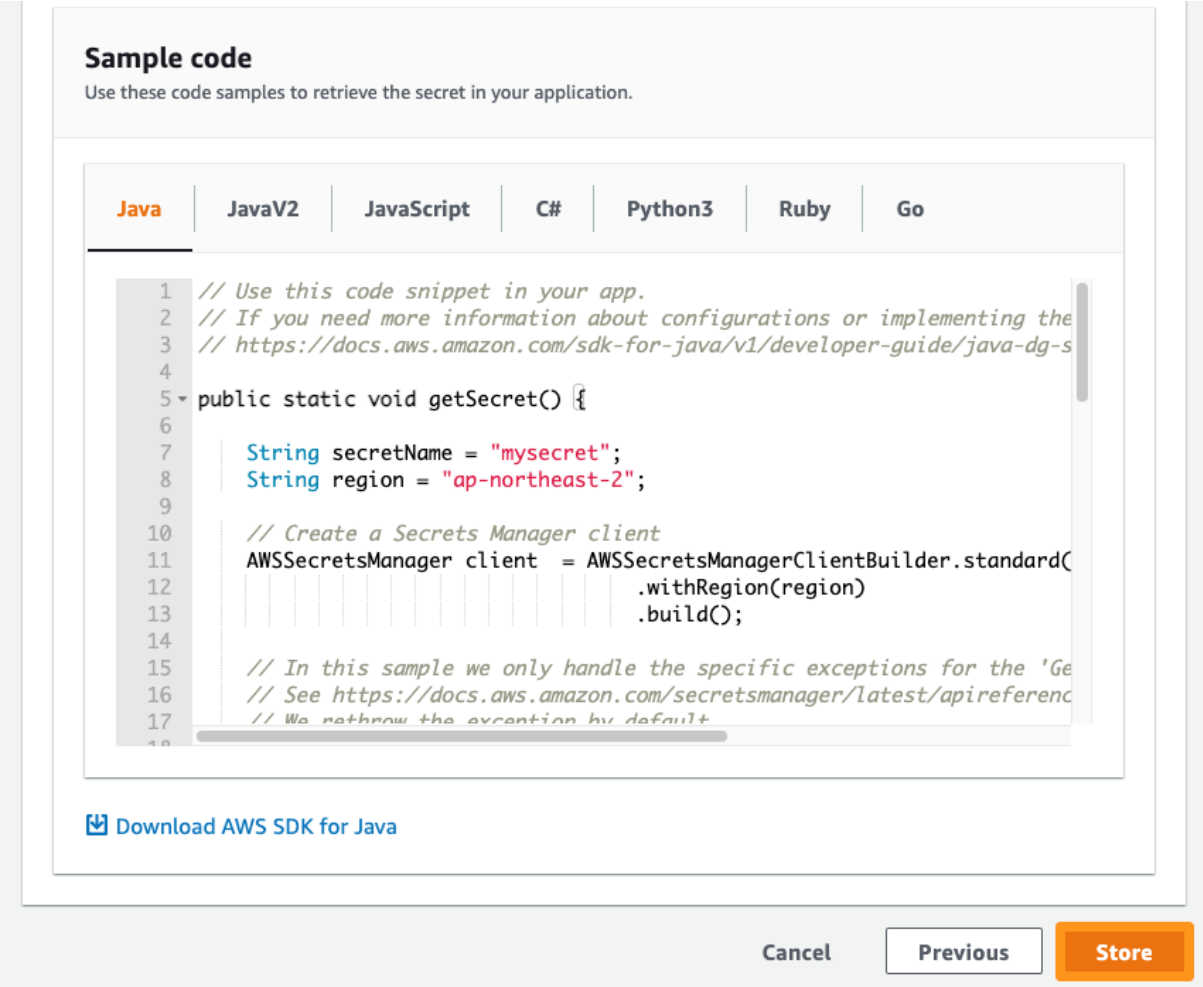
4. Name your secret, "**mysecret**". The sample code is written to ask for the secret by this specific name. Click **Next**.



5. Leave **Secret rotation** at default values. Click **Next**.



6. Review your choices. Click **Store**.

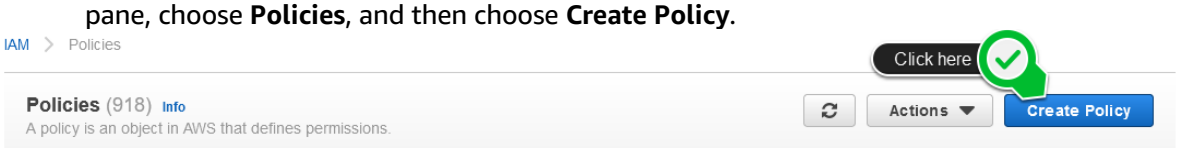


5-4 Access RDS from EC2

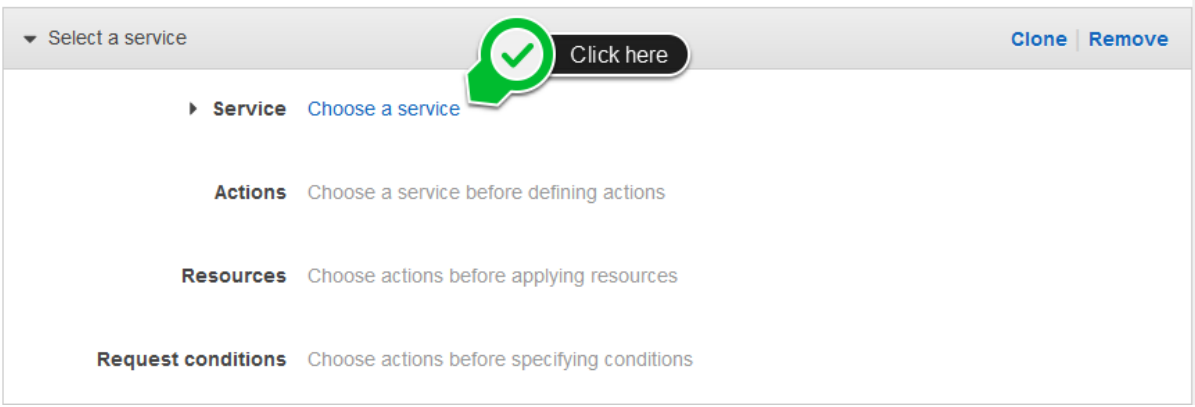
Allow the web server to access the secret

Now that you have created a secret, you must give your web server permission to use it. To do this, we will create a **Policy** that allows the web server to read a secret. We will add this policy to the **Role** you previously assigned to the web server.

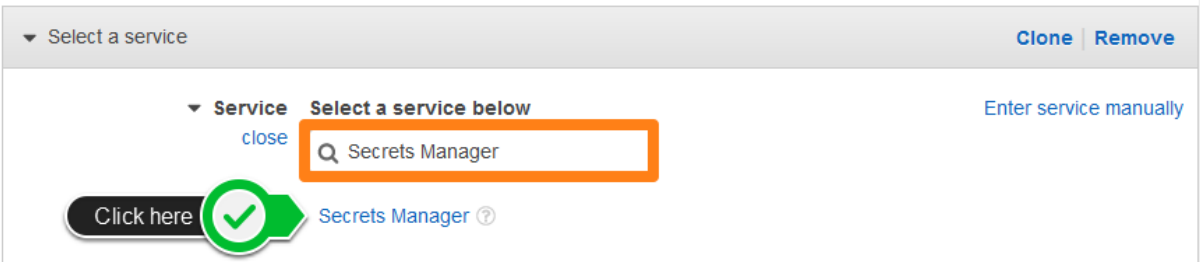
1. If you have not already done so, create an **IAM Instance Profile** as described in Connect to your Linux instance using Session Manager.
2. Sign in to the AWS Management Console and open the IAM console . In the navigation pane, choose **Policies**, and then choose **Create Policy**.



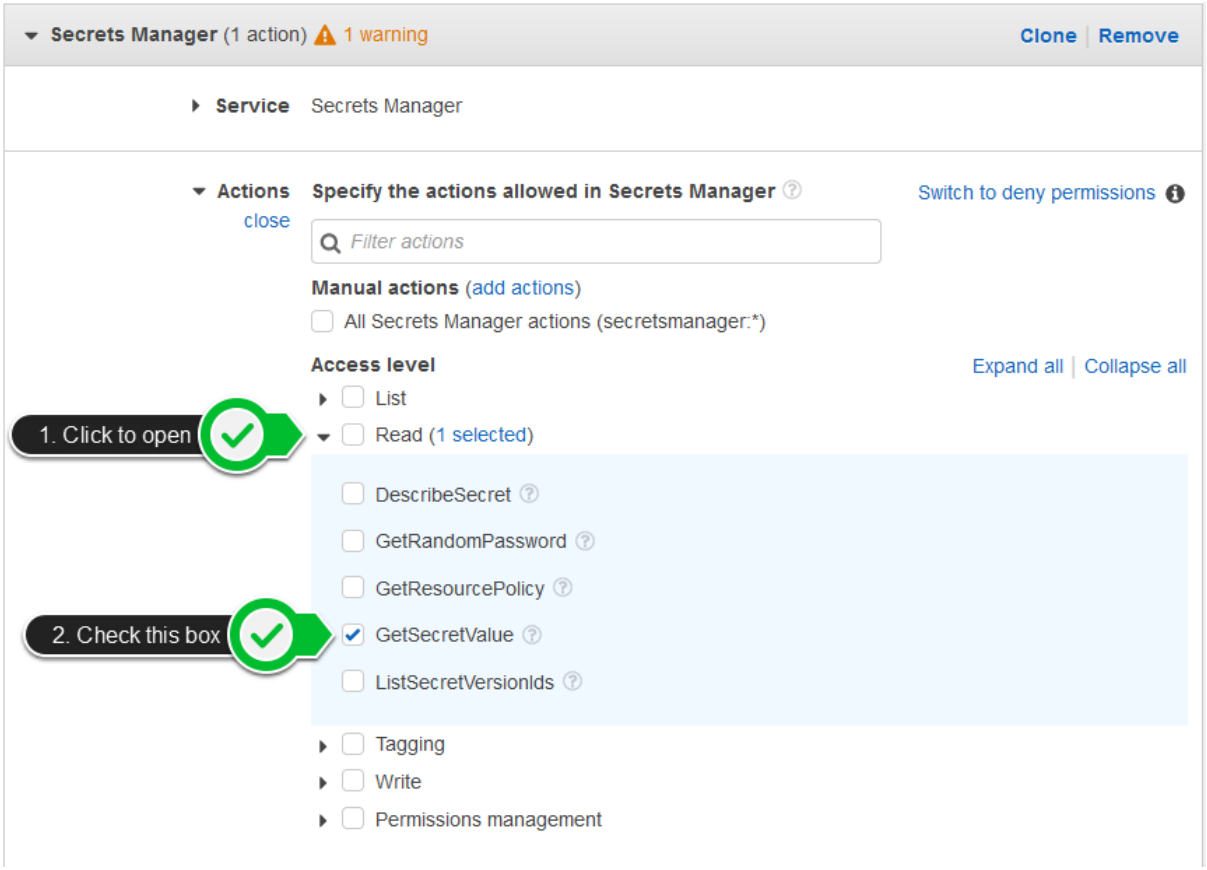
3. Click **Choose a service**.



4. Type **Secrets Manager** into the search box. Click **Secrets Manager**.

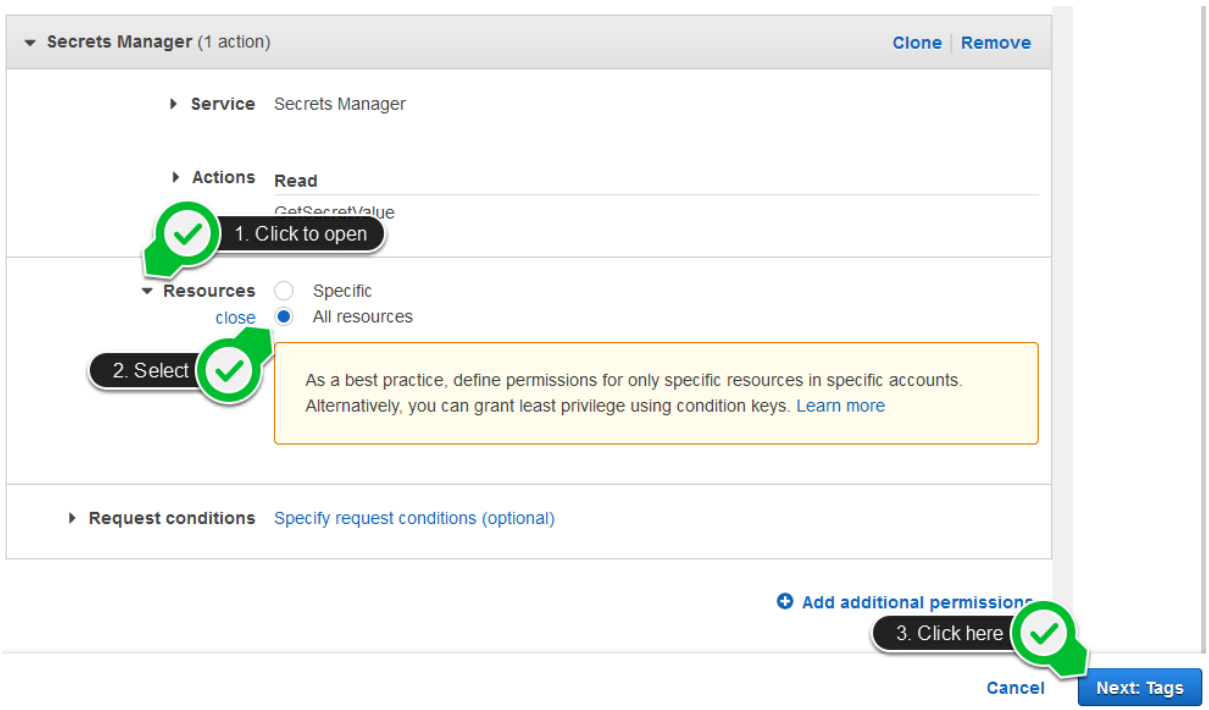


5. Under **Access level**, click on the carat next to **Read** and then check the box by **GetSecretValue**.

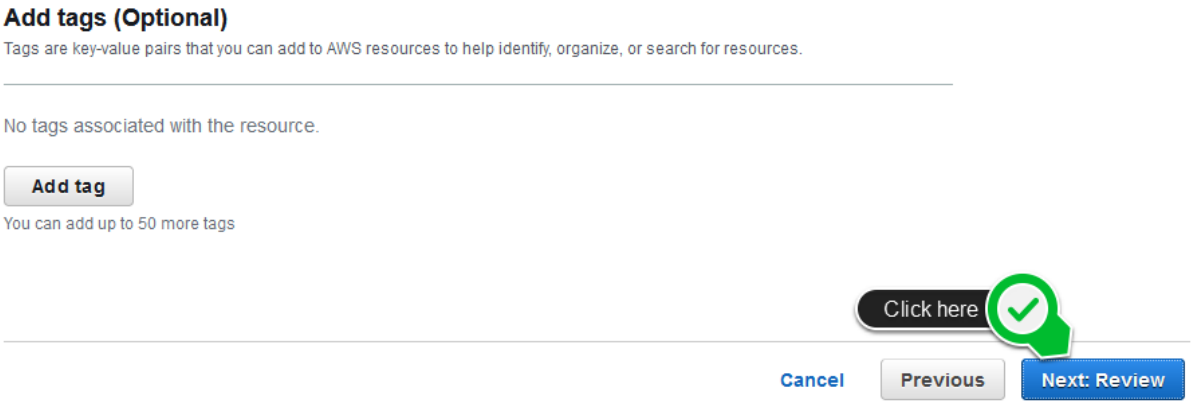


6. Click on the carat next to **Resources**. For this lab, select **All resources**. Click **Next: Tags**.

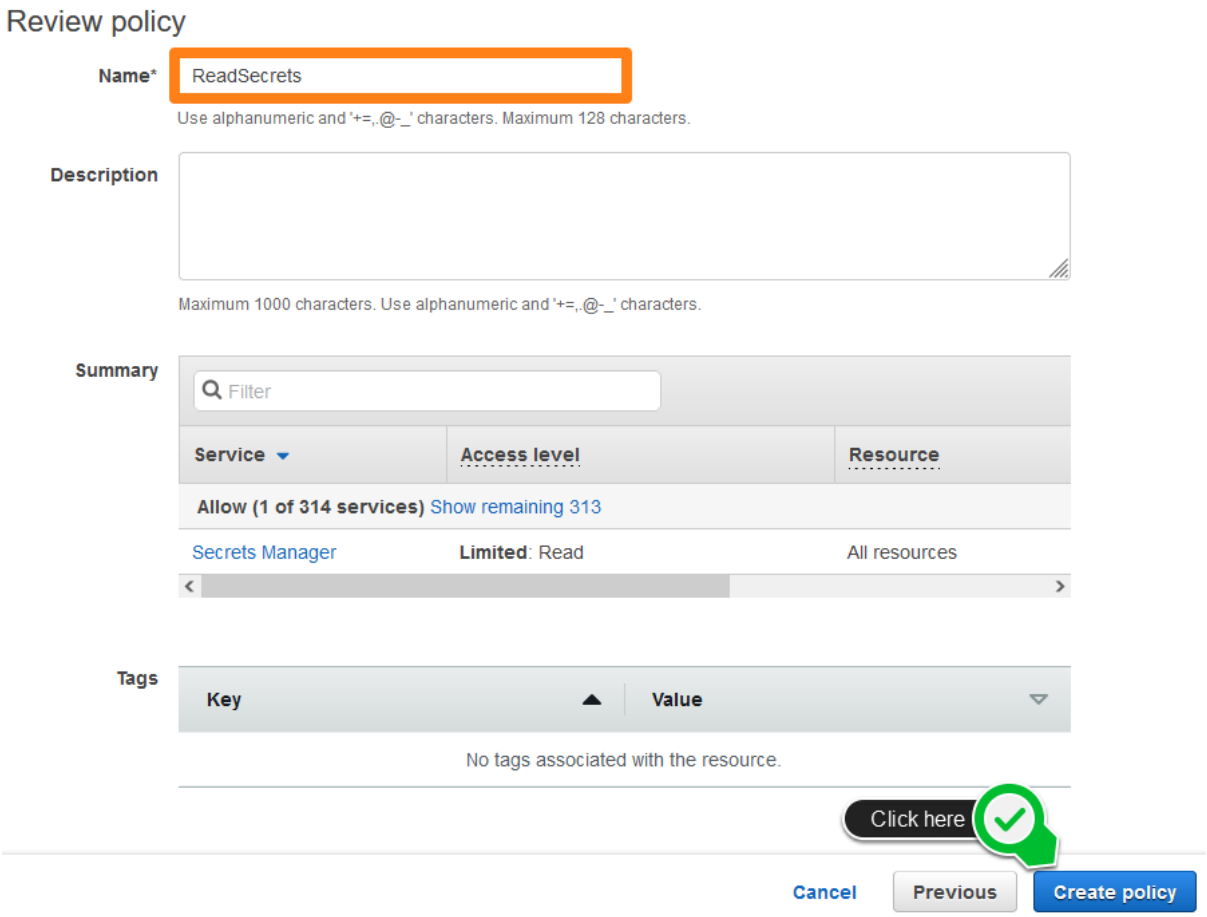
Note: For the lab, we're allowing EC2 to access all secrets. With a real workload, you should consider allowing access to specific secrets.



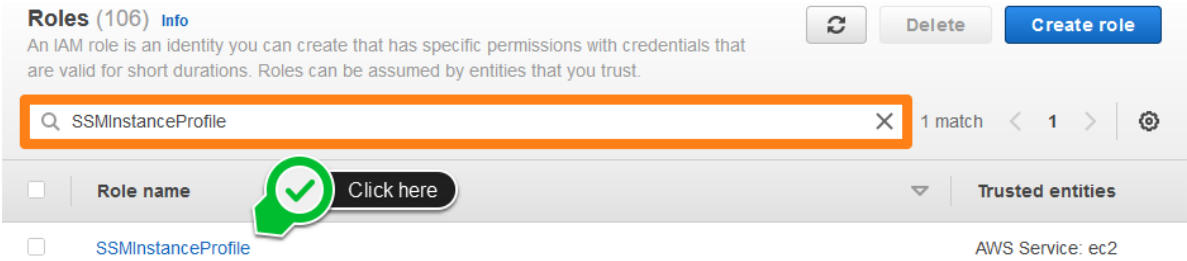
7. Click **Next: Review**.



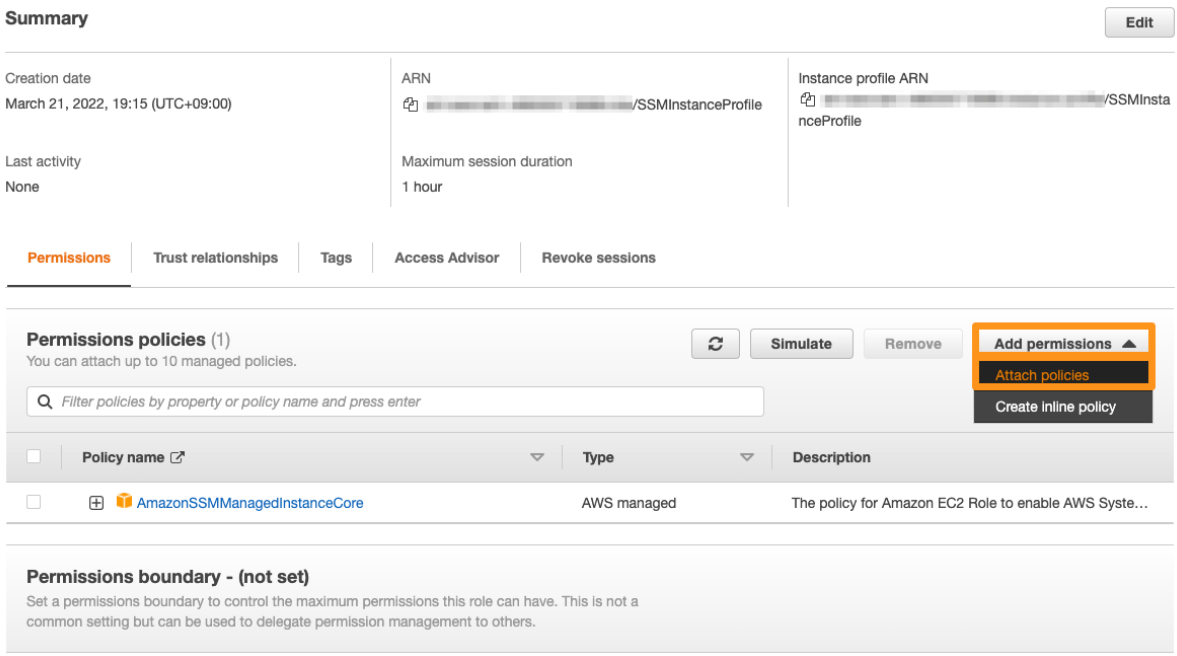
8. On the **Review Policy** screen, give your new policy the name **ReadSecrets**. Click **Create policy**.



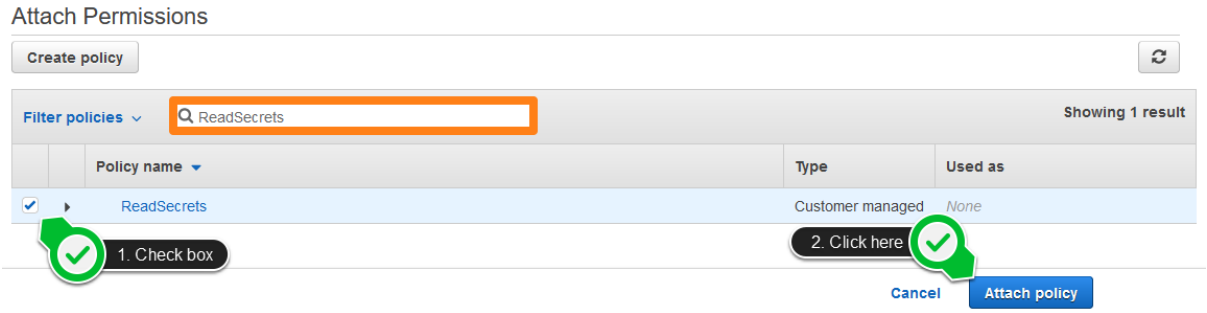
9. In the navigation pane, choose **Roles** and type **SSMInstanceProfile** into the search box. This is the role you created previously in "Connect to your Linux instance using Session Manager". Click **SSMInstanceProfile**.



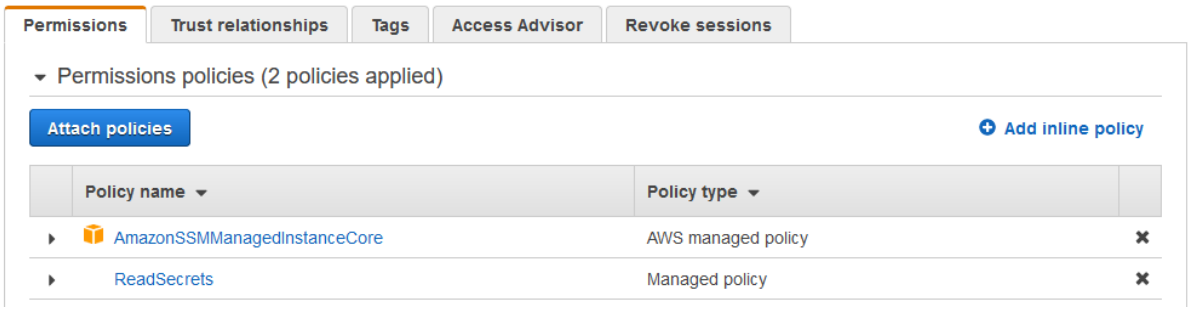
10. Under **Permissions policies**, click **Attach policies**.



11. Search for the policy you created called **ReadSecrets**. Check the box and click **Attach policy**.

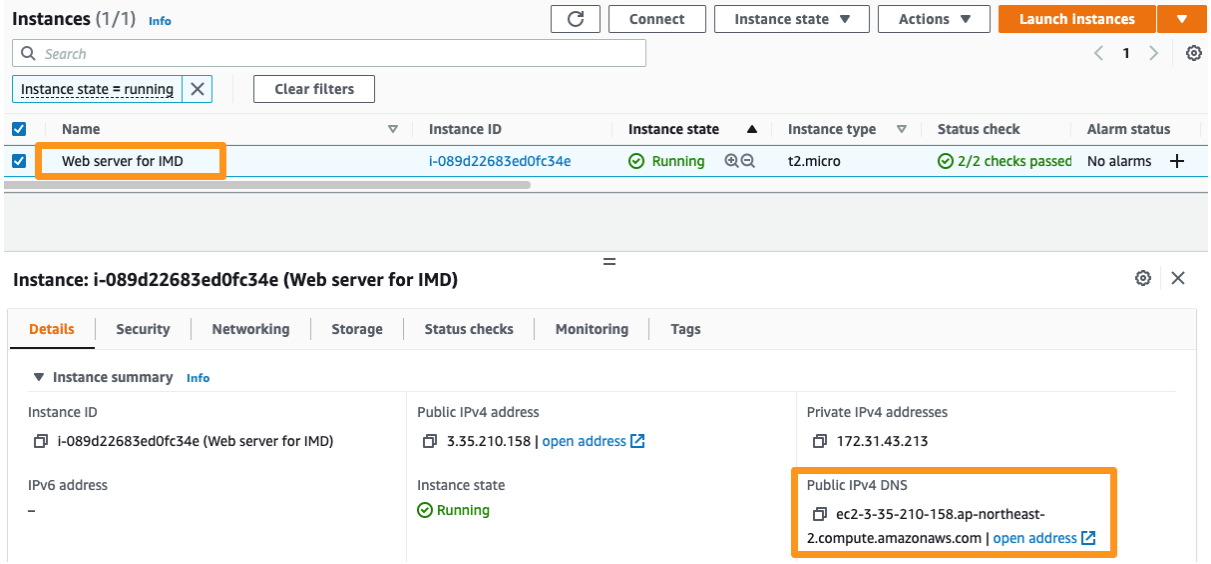


12. Under **Permissions policies**, verify that **AmazonSSMManagedInstanceCore** and **ReadSecrets** are both listed.

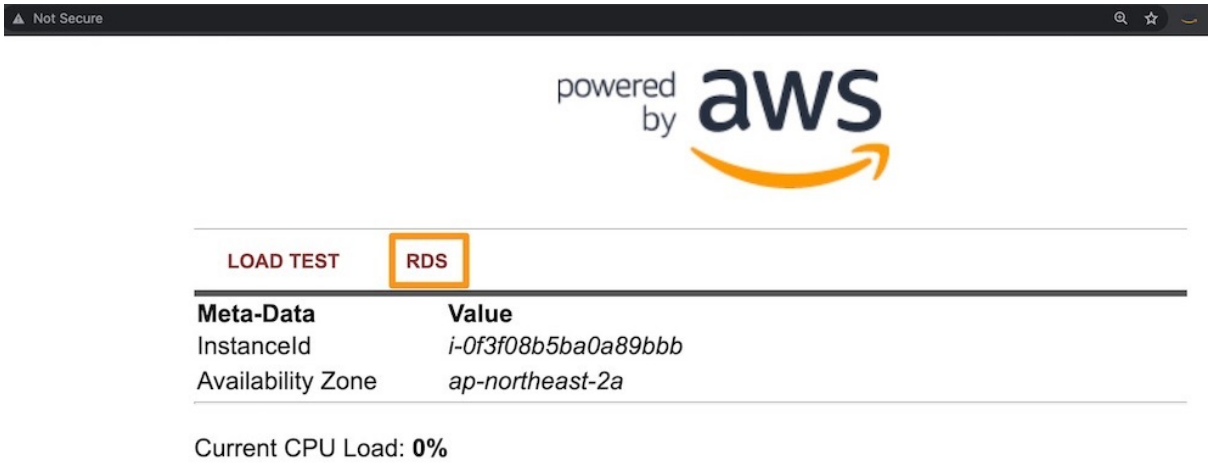


Try the Address Book

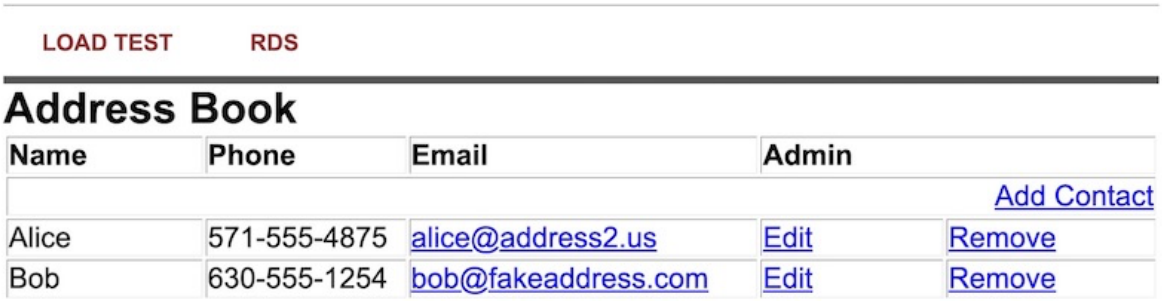
1. Navigate to the EC2 console and find the web server you launched in the EC2 Linux Hands-On Lab. Note your web server's public IP.



2. Open a new tab and reconnect to your web server's public IP. Click RDS.



3. You should now see a simple page displaying all of the information from the database you just created.



This is a very basic example of a simple address book interacting with a MySQL database managed by AWS. RDS can support much more complicated relational database scenarios, but we hope this simple example will suffice to demonstrate the point.

Feel free to play around with the address book and add/edit/remove content from your RDS database by using the **Add Contact, Edit**, and **Remove** links in the Address Book.

Great Job: You have successfully deployed and utilized an AWS managed MySQL database!!!