



Immersion Day

Getting Started with Amazon RDS

August 2018

Table of Contents

Overview.....	3
Launch an RDS Instance	4
Add EC2 Instance Security Group to RDS Security Group	4
Configure Instance to Leverage RDS.....	12
Appendix – Additional RDS Features.....	16
Create an RDS Snapshot	16
Modify RDS Instance Size	18

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.



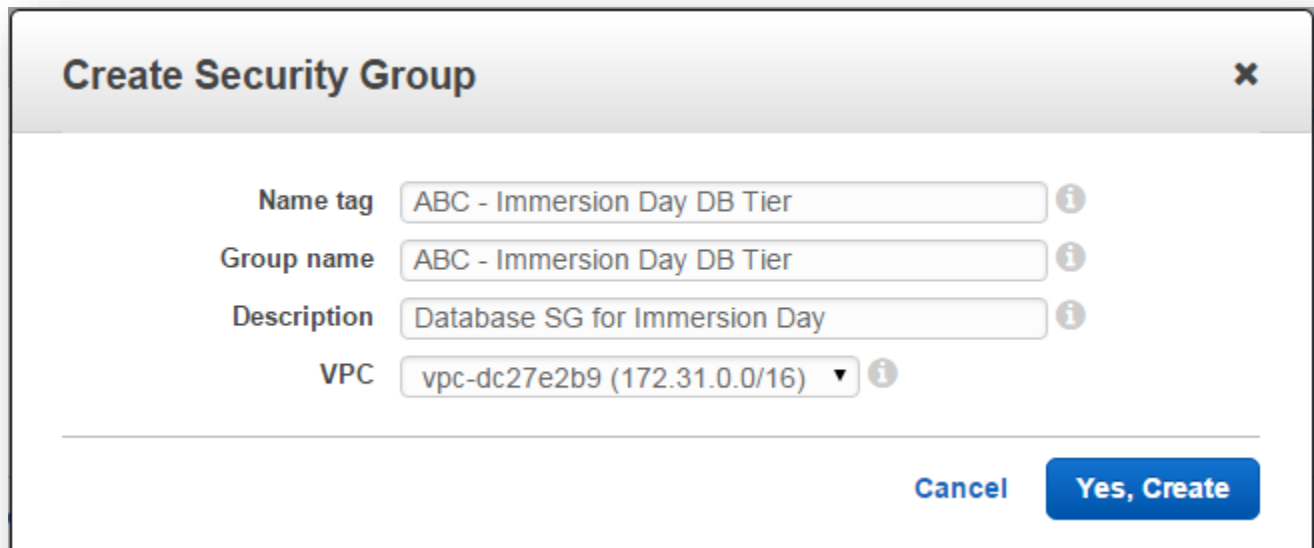
This lab has a prerequisite of *Immersion Day – Getting Started with EC2* in order to complete. This part of the lab will demonstrate configuring a previously created web server in the *Immersion Day – Getting Started with EC2* lab to use RDS for its Relational Database Management System (RDBMS) needs.

Add a VPC Instance Security Group

Prerequisite: Immersion Day – Getting Started with EC2

The RDS servers have the same security model as Amazon EC2 overall: trust nothing. A common use of an RDS instance in a VPC is to share data with an application server running in an EC2 instance in the same VPC and that is accessed by a client application outside the VPC. To this end, we'll need to utilize a VPC security group to allow this access.

If you've already completed the instructions in the "Immersion Day – Getting Started with EC2" lab manual, you'll have an existing EC2 instance with an existing security group. The name will be "[Initials] – Immersion Day Web Tier." Let's create a new VPC security group for our database tier that only allows traffic from our web tier. In the VPC dashboard, click **Security Groups**, then the **Create Security Group** button. Set *Name tag* and *group name* to "[Initials] - Immersion Day DB Tier." Write a short description, and keep the VPC setting to the same VPC you've launched your EC2 instance in (**My VPC**). Then click **Yes, Create**.



After your VPC security group is created, you'll see the details of it in the lower pane on the screen. Click **Inbound Rules**, then the **Edit rules** button.

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

Immersion Day

Getting Started with Amazon RDS

group(s) that match that name should be presented to you. Select your security group, then click the **Save** button.

sg-cb0e76ae | ABC - Immersion Day DB Tier

Summary

Inbound Rules

Outbound Rules

Tags

Cancel

Save

Type	Protocol	Port Range	Source	Remove
MySQL/Aurora (3306) ▼	TCP (6) ▼	3306	sg-cb i x	
<div>Add another rule</div>				

sg-cb0e76ae | ABC - Immersion Day DB Tier

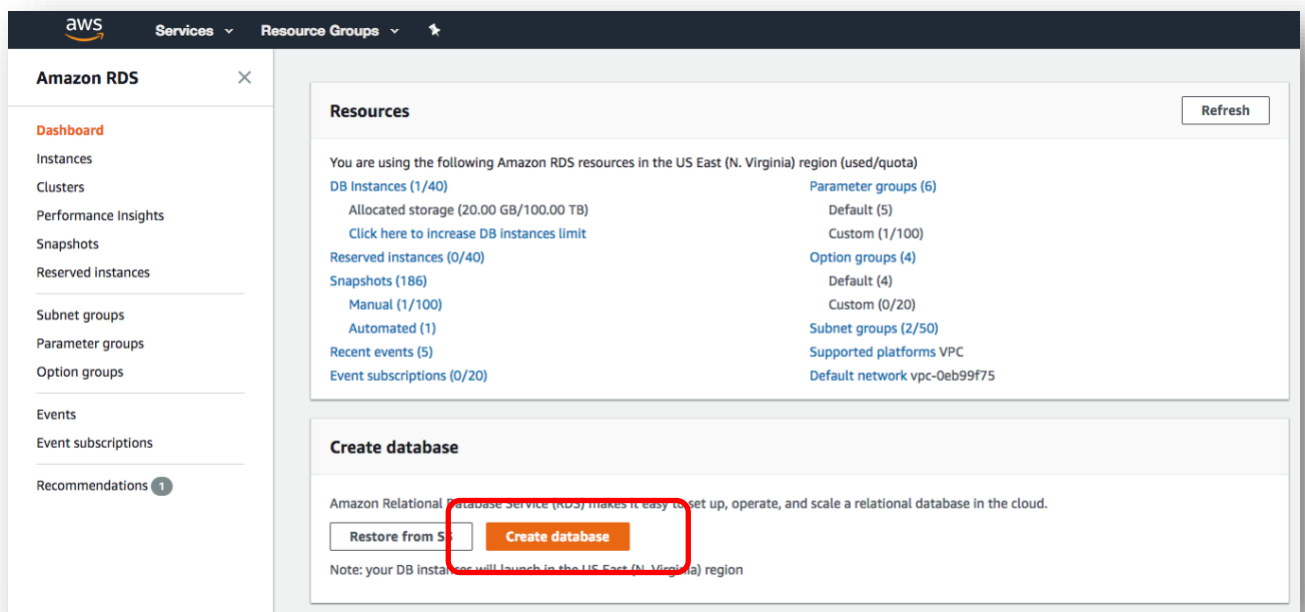
Create a subnet group

1. Sign into the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds>.
2. Click on **Subnet groups** and click on **Create DB Subnet Group**.
3. Name your subnet group as **PrivateDBGroup** and a short description.
4. For the VPC select **My VPC**.
5. In Add subnets section select the Private Subnets from us-east-1a (10.1.2.0/24) and us-east-1b (10.1.3.0/24).
6. Click on **Create**.

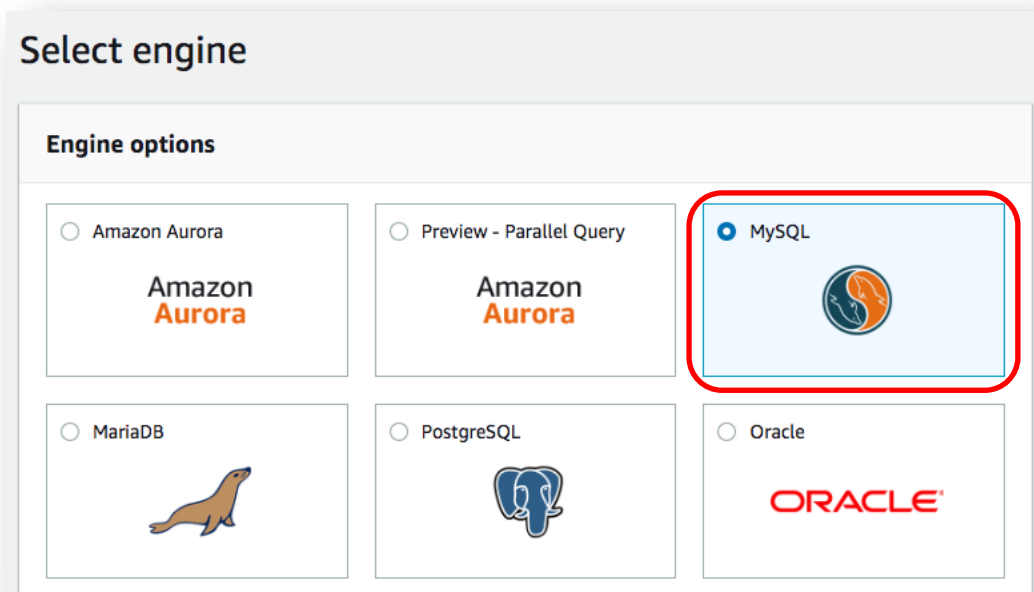
Launch an RDS Instance

Now that our VPC security group and subnet group are ready, let's configure and launch a MySQL RDS Instance.

7. Sign into the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds>.
8. Click on **Create database** or **Get Started Now**



9. We will be using a MySQL database, so choose MySQL from the available engines.



10. Click **Next**.

11. Check **Dev/Test - MySQL** for the Use Case, at the bottom of the page, and then click **Next**.
(Note: this is not recommended for production databases, as this option will disable options such as Multi-AZ deployments or read replicas, but it is OK for the purposes of this lab.)

12.


13. Fill out the DB Instance details with the following information and click **Next**:

- DB Engine Version: Use the default engine version ("5.7.22" as of August, 2018)
- DB Instance Class: db.t2.micro
- Storage Type: General Purpose (SSD)
- Allocated Storage: 20 GB
- DB Instance Identifier: awssdb
- Master Username: awsuser
- Master Password: awspassword


Getting Started with Amazon RDS

DB engine version [Info](#)

MySQL 5.7.22 ▼

 **Known Issues/Limitations**

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

 **Free tier**

The Amazon RDS Free Tier provides a single db.t2.micro instance as well as up to 20 GiB of storage, allowing new AWS customers to gain hands-on experience with Amazon RDS. Learn more about the RDS Free Tier and the instance restrictions [here](#).

☒ Only enable options eligible for RDS Free Usage Tier [Info](#)

DB instance class [Info](#)

db.t2.micro — 1 vCPU, 1 GiB RAM ▼

Multi-AZ deployment [Info](#)

☐ Create replica in different zone
Creates a replica in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.

☒ No

Storage type [Info](#)

General Purpose (SSD) ▼

Allocated storage

20

GiB

(Minimum: 20 GiB, Maximum: 20 GiB) Higher allocated storage [may improve](#) IOPS performance.

Getting Started with Amazon RDS

Settings

DB instance identifier [Info](#)

Specify a name that is unique for all DB instances owned by your AWS account in the current region.

DB instance identifier is case insensitive, but stored as all lower-case, as in "mydbinstance". Must contain from 1 to 63 alphanumeric characters or hyphens (1 to 15 for SQL Server). First character must be a letter. Cannot end with a hyphen or contain two consecutive hyphens.

Master username [Info](#)

Specify an alphanumeric string that defines the login ID for the master user.

Master Username must start with a letter. Must contain 1 to 16 alphanumeric characters.

Master password [Info](#)

Master Password must be at least eight characters long, as in "mypassword". Can be any printable ASCII character except "/", "", or "@".

Confirm password [Info](#)

14. In **Configure Advanced Settings**, fill out *Network & Security* with the following information:

- VPC: My VPC
- Subnet Group: privatedbfroup
- Public Accessibility: No
- Availability Zone: No Preference
- VPC Security Group(s): Select *Choose existing VPC security groups*, then pick [Initials] – Immersion Day DB Tier, remove the default security group.

15. Under *Database Options*, enter a DB name called "immersionday" and accept the defaults for *database port*, *parameter group*, *option group* and *IAM DB authentication*. Leave the default options for the rest of the configuration groups (*Encryption*, *Backup*, *Monitoring*, *Log exports* and *Maintenance*).

Immersion Day

Getting Started with Amazon RDS

Database options

Database name

immersionday

Note: if no database name is specified then no initial MySQL database will be created on the DB instance.

Database port

TCP/IP port the DB instance will use for application connections.

3306

DB parameter group [Info](#)

default.mysql5.7

Option group [Info](#)

default:mysql-5-7

IAM DB authentication [Info](#)

☐ Enable IAM DB authentication

Manage your database user credentials through AWS IAM users and roles.

☒ Disable

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



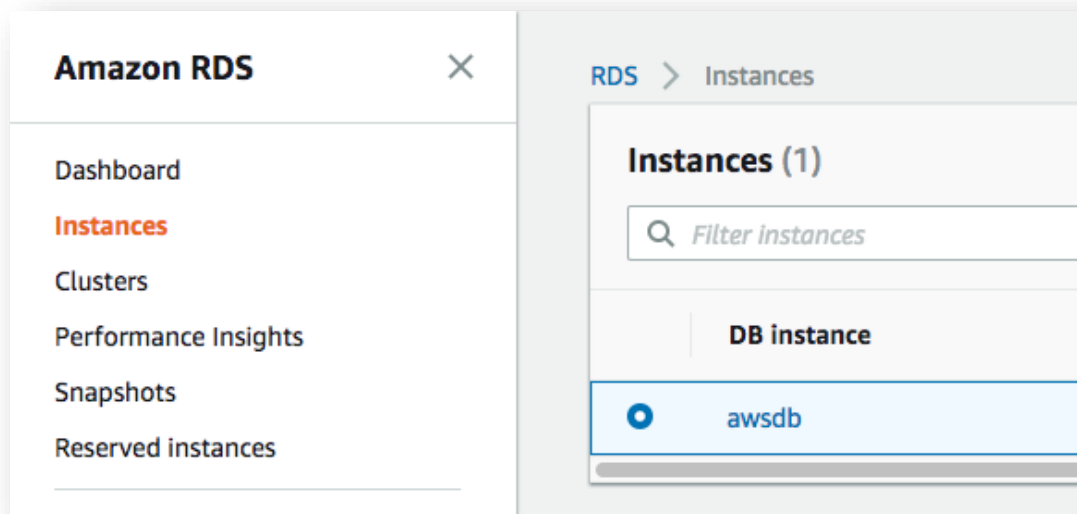
Note: This may take up to 5 minutes as the database is being created and backed up.

Configure Instance to Leverage RDS

Prerequisite: Immersion Day – Getting Started with EC2

We provided an example database table and sample code for creating a simple address book. Before configuring your instance, you will need to get the URL for your database endpoint.

1. In the RDS console, click on **Instances** and then select your database instance, *awsdb*.



2. Scroll down to the *Connect* section and check the value under *Endpoint*. Remember this because you will need it in a minute.

Connect

Endpoint

awsdb.cxo6a707kcvz.us-east-1.rds.amazonaws.com

Security group rules (2)

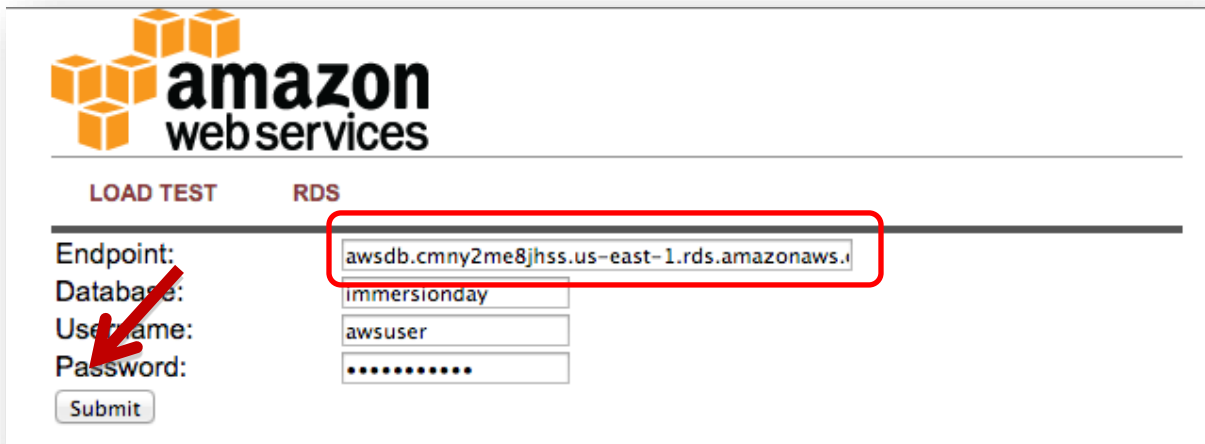
Filter security group rules

Security group

ABC - immersion Day DB Tier (sg-0e71b89ccec38a847)

ABC - immersion Day DB Tier (sg-0e71b89ccec38a847)

3. Navigate to the browser tab connected to web application you launched previously in the *Immersion Day – Getting Started with EC2* lab (or open a new tab and reconnect to your web server's URL) and click on **RDS**. You should see a prompt to enter the DB endpoint (do NOT include :3306 at the end of the DB endpoint), username (*awsuser*), password (*awspassword*) and database (*immersionday*) information you just created. Click the **Submit** button.



amazon web services

LOAD TEST RDS

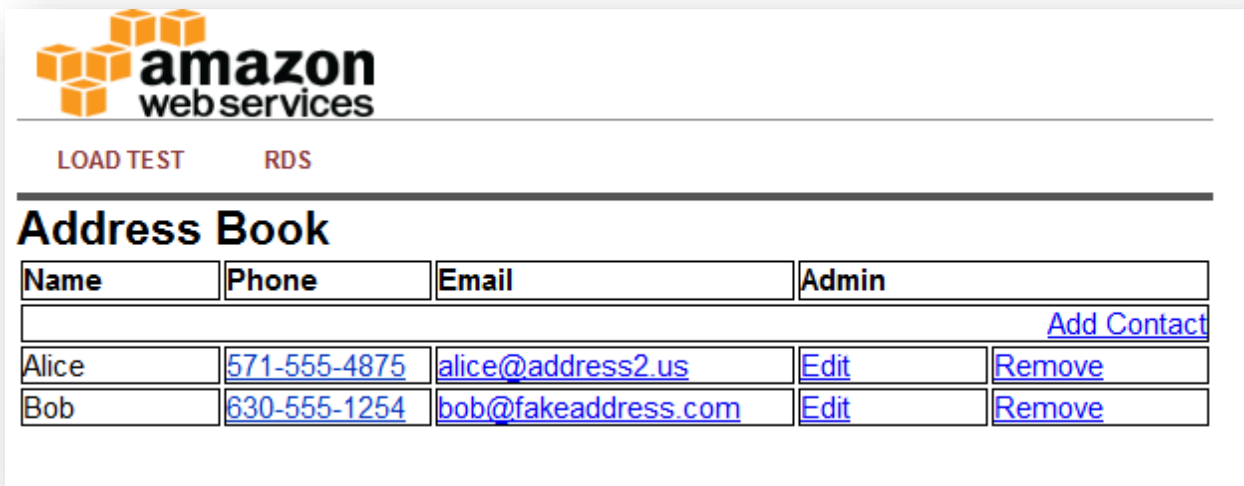
Endpoint:

Database:

User name:

Password:

4. When complete, you will be redirected to a simple page displaying all of the information from the database you just created.



amazon web services

LOAD TEST RDS

Address Book

Name	Phone	Email	Admin	
			Add Contact	
Alice	571-555-4875	alice@address2.us	Edit	Remove
Bob	630-555-1254	bob@fakeaddress.com	Edit	Remove

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.

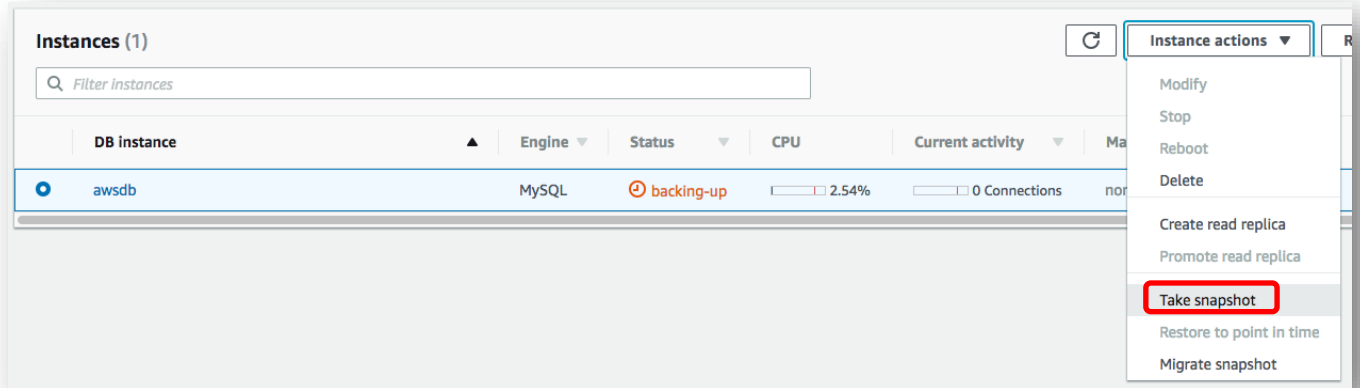
Great Job: You have successfully deployed and utilized an AWS managed MySQL database!!!

Appendix – Additional RDS Features

Create an RDS Snapshot

Now is a good time to take a snapshot of your RDS database. Taking a snapshot enables you to back up your DB Instance in a known state as frequently as you wish, and then restore to that specific state at any time.

In the RDS section of the of the AWS management console, select your RDS instance, click on **Instance actions** and select **Take snapshot**:

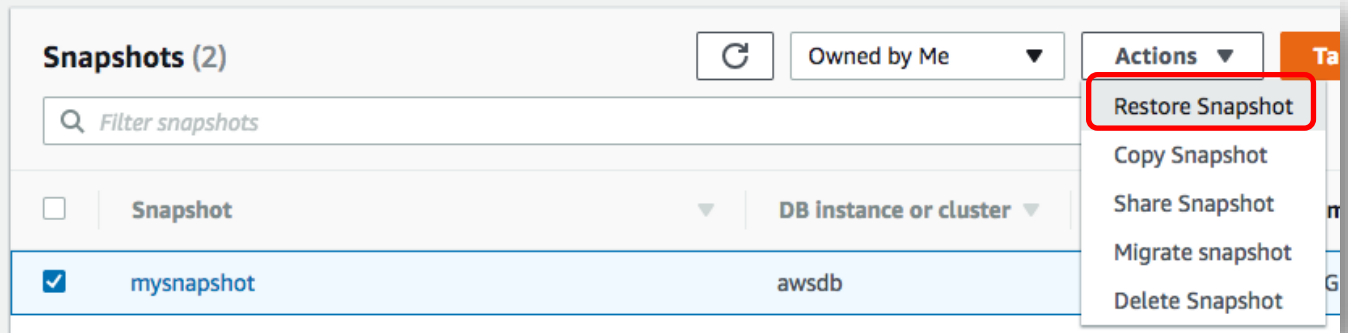


The next screen will ask you for a name. Enter **mysnapshot** and click **Take snapshot**.

A screenshot of the 'Settings' dialog box in the AWS Management Console. The dialog has a title 'Settings' and a subtitle 'To take a snapshot of this DB instance you must provide a name for the snapshot.' There are two input fields: 'DB instance' with the value 'awsdb' and 'Snapshot name' with the value 'mysnapshot'. At the bottom right, there are two buttons: 'Cancel' and 'Take Snapshot' (highlighted with an orange background).

Note: Using single-instance RDS, you will incur downtime for as long as it takes to make a backup. Of course our example database is so small that total time to back up is very small too!

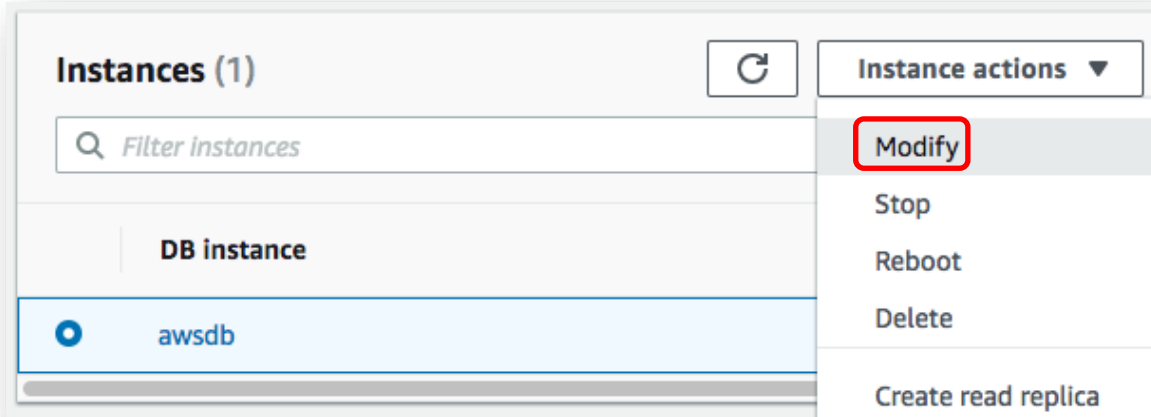
DB snapshots show up under the **Snapshots** link on the left side of the screen. Notice that you can easily launch new RDS instances from any previous snapshot!



Modify RDS Instance Size

Scaling up and down with RDS is simple via the AWS Console. You can grow the database or change the underlying server size, etc. – all from the AWS Console.

Select your RDS DB instance, click **Instance actions** and then **Modify**.



Try changing to a Large instance, and if you want, also grow the database at the same time. Click **Next**.

In the next screen, don't forget to click "**Apply Immediately**" – otherwise changes will be queued for the next maintenance window.



Tip: You can change instance sizes up or down at any time. However you cannot shrink a database once you grow it.

Just like backups, there will be an outage while you perform these operations. In general, major RDS reconfigurations such as scaling database sizes or machine size take between 4 and 12 minutes.

DB instance class

Contains the compute and memory capacity of the DB instance.

db.t2.micro — 1 vCPU, 1 GiB RAM
db.t2.micro — 1 vCPU, 1 GiB RAM
db.t2.small — 1 vCPU, 2 GiB RAM
db.t2.medium — 2 vCPU, 4 GiB RAM
db.t2.large — 2 vCPU, 8 GiB RAM
db.t2.xlarge — 4 vCPU, 16 GiB RAM
db.t2.2xlarge — 8 vCPU, 32 GiB RAM
db.m4.large — 2 vCPU, 8 GiB RAM
db.m4.xlarge — 4 vCPU, 16 GiB RAM

Scheduling of Modifications**When to Apply Modifications**

☐ Apply during the next scheduled maintenance window

Current maintenance window: sat:09:14-sat:09:44

☒ Apply immediately

The modifications in this request and any pending modifications will be asynchronously applied as soon as possible, regardless of the maintenance window setting for this database instance.