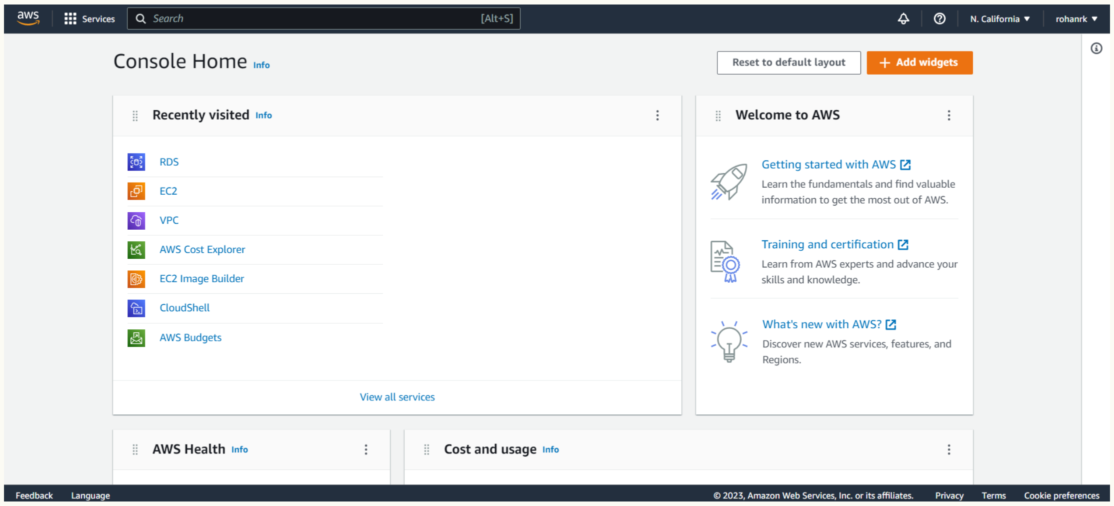
Learning Sprint #1

OBJECTIVE: This learning sprint closely models what you will be doing in project sprint #1 which is approving a person's loan request based off of some predetermined criteria (i.e. credit card score). In this learning sprint, you will choose how you will structure your tables to represent a person and write code that processes a loan request and assigns a risk category to the individual making the loan request.

Prerequisites:

1. Create an AWS account using your Berkeley email account. When you sign in, you should see a console like the one below:



Learn the fundamentals and find valuable information to get the most out of AWS.

Learn from AWS experts and advance your skills and knowledge.

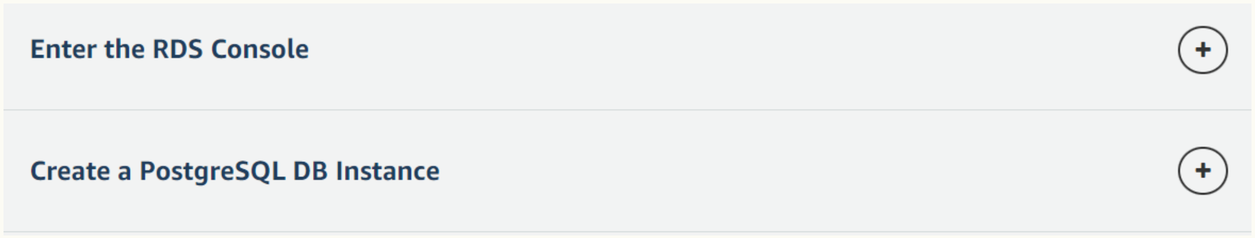
Discover new AWS services, features, and Regions.

© 2023, Amazon Web Services, Inc. or its affiliates.

2. Familiarize yourself with Postgres. It is a relational database system; so, your experience with SQL in 61A should prep you well to use it. Here is a quick reference guide, take a look through the syntax section to get a feel for how to to do some basic operations like creating a database, defining a function, creating a new entry, etc: https://www.tutorialspoint.com/postgresql/postgresql_syntax.htm
3. Make sure to have python downloaded: https://www.python.org/downloads

Time to Dive in!

1. Create a Postgres DB instance by following these steps. Follow the instructions only for the following two sections: https://aws.amazon.com/getting-started/hands-on/create-connect-postgresql-db/

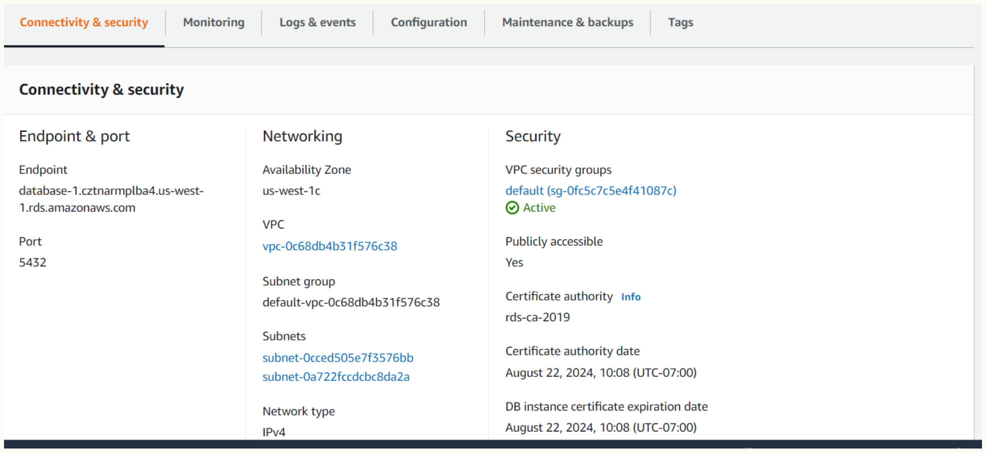


Enter the RDS Console +

Create a PostgreSQL DB Instance +

*Make sure to add your IP address to your security inbound rules:

a. First click on the database instance that you just created. You should see a page like this:



Connectivity & security

Endpoint & port

Endpoint
database-1.cztnarmplba4.us-west-1.rds.amazonaws.com

Port
5432

Networking

Availability Zone
us-west-1c

VPC
vpc-0c68db4b31f576c38

Subnet group
default-vpc-0c68db4b31f576c38

Subnets
subnet-0cced505e7f3576bb
subnet-0a722fccdcbc8da2a

Network type
IPv4

Security

VPC security groups
default (sg-0fc5c7c5e4f41087c)
Active

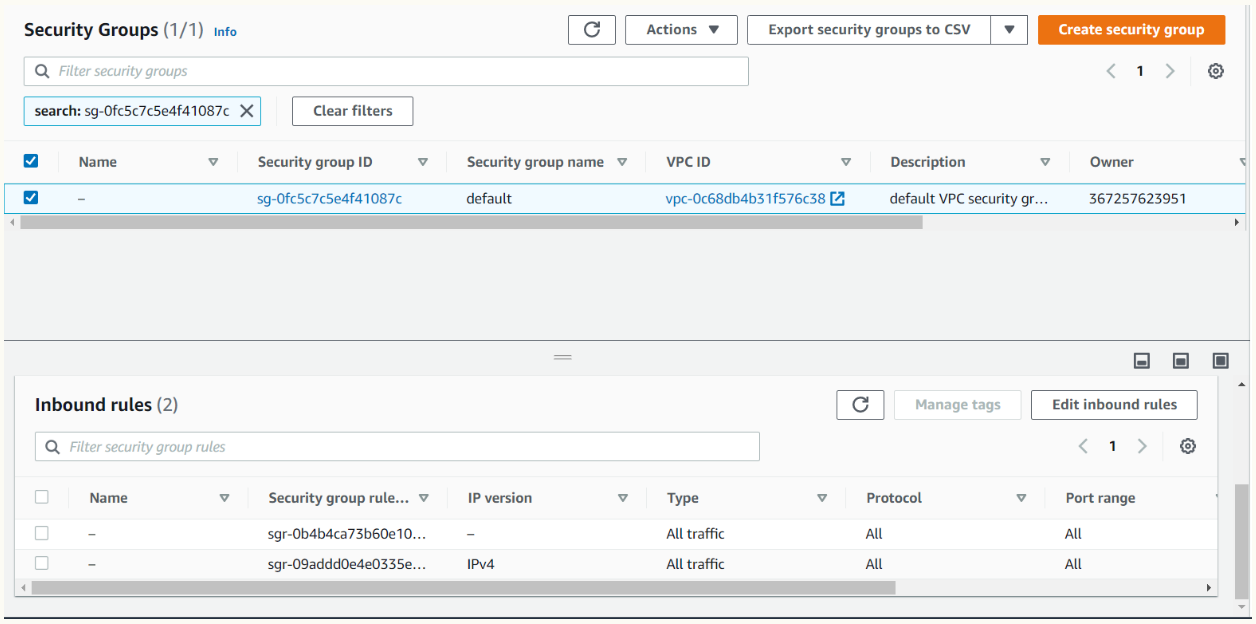
Publicly accessible
Yes

Certificate authority Info
rds-ca-2019

Certificate authority date
August 22, 2024, 10:08 (UTC-07:00)

DB instance certificate expiration date
August 22, 2024, 10:08 (UTC-07:00)

b. Click on the hyperlinked text under VPC security groups, which should take you to a page like this:



Security Groups (1/1) Info

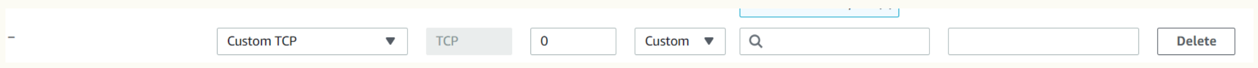
search: sg-0fc5c7c5e4f41087c

☑	Name	Security group ID	Security group name	VPC ID	Description	Owner
☑	–	sg-0fc5c7c5e4f41087c	default	vpc-0c68db4b31f576c38	default VPC security gr…	367257623951

Inbound rules (2)

☐	Name	Security group rule…	IP version	Type	Protocol	Port range
☐	–	sgr-0b4b4ca73b60e10…	–	All traffic	All	All
☐	–	sgr-09addd0e4e0335e…	IPv4	All traffic	All	All

c. Click edit inbound rules and click add rule. You should add your ip address in the search bar area (you can check your ip using the following link : http://checkip.amazonaws.com/)



2. Connect your instance to the PGAdmin
    a. Download PGAdmin with the following link: https://www.pgadmin.org/download/pgadmin-4-windows/
    b. Follow this video to connect your DB instance to PGAdmin (start from 3:00): Create an RDS Postgres instance and connect with pgAdmin

3. In a database table, each column represents an attribute (i..e first name, age) about the objects that the table contains. Each row is an instance of the object that the table contains (i.e. a person). A schema describes the different attributes (columns) that the table will keep track of. Watch the following tutorial on how to define a schema and create a table in a Postgres instance. Then, using the Query tool in PGAdmin, create your schema that describes a person. Think about what attributes you would want to include (aka what data do you think is relevant to store about a person from the POV of a loan approval company). This can be very simple (Credit Score, Age, etc).Then insert your own dumby data into the table.
Video: Creating PostgreSQL tables with pgAdmin

4. Platforms that experience a lot of user traffic require a lot of computational resources (like servers to run the code on). Rather than allocating these resources for each user to run 24/7, it would be more cost efficient to create them only when the user is actively interacting with the platform. AWS Lambda is a service that allows you to write functions that are invoked in response to a particular event taking place (i.e. a button click). Lambda takes care of creating and managing the underlying resources needed to execute the function. You will be creating a lambda function that in response to an event, will setup a connection with your Postgres DB and make a query that returns the information corresponding to the current user (Hint: query on a primary key). In the context of this sprint, we want you to write a lambda function that queries user information and determines if particular users should be allowed to get a loan. This can be a very simple algorithm (ex: assign a person to "low risk" if they have a credit score over 700)
Video on writing Lambda functions:
How to Query RDS MySQL From AWS Lambda in Python | Step by Step Tutorial