# 06-12-2020

# AWS Lab Session - Callahan

# Relational Database Service (RDS)

# Session starts at 5.30pm EST

# Part 3

# create MySQL DB Instance on RDS

# log into AWS RDS Console

# create database with standard creation method

# engine option: MySQL

# version: 8.0.19 (latest)

# template:free tier

# db instance identifier: call-mysql-db-server

# master username: admin

# master password: Clarusway\_1

# db instance size: db.t2.micro

# storage type:ssd

# storage size:default 20GiB

# storage autoscaling: disabled

# vpc:default

# subnet group: default

# publicly accessible: yes

# security group: existing (allowing ssh and sql)

# az: default no preference

# db port: 3306

# db authentication: password

# ---- additional configuration -----

# initial db name: clarusway

# db parameter group & option group:default

# automatic backups:enabled

# backup retention period: 7 days

# select window for backup to show snapshots

10.15pm UTC -> 6.15pm EST, DUR: 30mins

# Enable auto minor version upgrade: enabled

# deletion protection: enabled

# show configuration details of creating db for production

# show and explain newly created db instance description details

# Part 4

# launch EC2 Instance (Ubuntu 18.04) and name it as "MariaDB-Client on Ubuntu"

# update instance

sudo apt update -y

# install the mariadb-client

sudo apt-get install mariadb-client -y

# connect the RDS MySQL DB instance with admin user, and paste the password when prompted

mysql -h call-mysql-db-server.cbanmzptkrzf.us-east-1.rds.amazonawsst-1.rds.amazonaws.com -u admin -p

# show default databases in the MySQL server

SHOW DATABASES;

# choose a database ('mysql' db) to work with.

USE mysql;

# show tables within the mysql db

SHOW TABLES;

# show users defined in the db server currently.

SELECT Host, User, authentication\_string FROM user;

# create a user named "hr\_guy";

CREATE USER hr\_guy IDENTIFIED BY 'Hr\_guy1234';

# grant permissions to the user "hr\_guy" for database "clarusway"

# you can not use grant all command on RDS DB Instance due to the restriction, following command will give an error.

GRANT ALL ON clarusway.\* TO hr\_guy IDENTIFIED BY 'Hr\_guy1234' WITH GRANT OPTION;

# ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server verersion for the right syntax to use near 'IDENTIFIED BY 'Hr\_guy1234' WITH GRANT OPTION' at line 1

# to grant permissions, you need to specify each permission explicitly

GRANT SELECT, INSERT, UPDATE, DELETE, EXECUTE, REFERENCES, LOCK TABLES, ALTER, CREATE, CREATE TEMPORARY TABLES, DROP, INDEX, EVENT, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, TRIGGER ON clarusway.\* TO hr\_guy WITH GRANT OPTION;

# update privileges

FLUSH PRIVILEGES;

# close the mysql terminal

EXIT;

# download rds-ca-2019-root.pem from https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.SSL.html

wget https://s3.amazonaws.com/rds-downloads/rds-ca-2019-root.pem

# login back to the RDS MySQL DB instance as "hr\_guy" using the password defined with SSL

mysql -h call-mysql-db-server.cbanmzptkrzf.us-east-1.rds.amazonaws.com --ssl-ca=rds-ca-2019-root.pem -u hr\_guy -p clarusway

# create a table named "offices"

CREATE TABLE `offices` (

`office\_id` int(11) NOT NULL,

`address` varchar(50) NOT NULL,

`city` varchar(50) NOT NULL,

`state` varchar(50) NOT NULL,

PRIMARY KEY (`office\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

# insert some data into the table named "offices"

INSERT INTO `offices` VALUES (1,'03 Reinke Trail','Cincinnati','OH');

INSERT INTO `offices` VALUES (2,'5507 Becker Terrace','New York City','NY');

INSERT INTO `offices` VALUES (3,'54 Northland Court','Richmond','VA');

INSERT INTO `offices` VALUES (4,'08 South Crossing','Cincinnati','OH');

INSERT INTO `offices` VALUES (5,'553 Maple Drive','Minneapolis','MN');

INSERT INTO `offices` VALUES (6,'23 North Plaza','Aurora','CO');

INSERT INTO `offices` VALUES (7,'9658 Wayridge Court','Boise','ID');

INSERT INTO `offices` VALUES (8,'9 Grayhawk Trail','New York City','NY');

INSERT INTO `offices` VALUES (9,'16862 Westend Hill','Knoxville','TN');

INSERT INTO `offices` VALUES (10,'4 Bluestem Parkway','Savannah','GA');

# create a table named "employees"

CREATE TABLE `employees` (

`employee\_id` int(11) NOT NULL,

`first\_name` varchar(50) NOT NULL,

`last\_name` varchar(50) NOT NULL,

`job\_title` varchar(50) NOT NULL,

`salary` int(11) NOT NULL,

`reports\_to` int(11) DEFAULT NULL,

`office\_id` int(11) NOT NULL,

PRIMARY KEY (`employee\_id`),

KEY `fk\_employees\_offices\_idx` (`office\_id`),

CONSTRAINT `fk\_employees\_offices` FOREIGN KEY (`office\_id`) REFERENCES `offices` (`office\_id`) ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

# insert some data into the table named "employees"

INSERT INTO `employees` VALUES (37270,'Yovonnda','Magrannell','Executive Secretary',63996,NULL,10);

INSERT INTO `employees` VALUES (33391,'Darcy','Nortunen','Account Executive',62871,37270,1);

INSERT INTO `employees` VALUES (37851,'Sayer','Matterson','Statistician III',98926,37270,1);

INSERT INTO `employees` VALUES (40448,'Mindy','Crissil','Staff Scientist',94860,37270,1);

INSERT INTO `employees` VALUES (56274,'Keriann','Alloisi','VP Marketing',110150,37270,1);

INSERT INTO `employees` VALUES (63196,'Alaster','Scutchin','Assistant Professor',32179,37270,2);

INSERT INTO `employees` VALUES (67009,'North','de Clerc','VP Product Management',114257,37270,2);

INSERT INTO `employees` VALUES (67370,'Elladine','Rising','Social Worker',96767,37270,2);

INSERT INTO `employees` VALUES (68249,'Nisse','Voysey','Financial Advisor',52832,37270,2);

INSERT INTO `employees` VALUES (72540,'Guthrey','Iacopetti','Office Assistant I',117690,37270,3);

INSERT INTO `employees` VALUES (72913,'Kass','Hefferan','Computer Systems Analyst IV',96401,37270,3);

INSERT INTO `employees` VALUES (75900,'Virge','Goodrum','Information Systems Manager',54578,37270,3);

INSERT INTO `employees` VALUES (76196,'Mirilla','Janowski','Cost Accountant',119241,37270,3);

INSERT INTO `employees` VALUES (80529,'Lynde','Aronson','Junior Executive',77182,37270,4);

INSERT INTO `employees` VALUES (80679,'Mildrid','Sokale','Geologist II',67987,37270,4);

INSERT INTO `employees` VALUES (84791,'Hazel','Tarbert','General Manager',93760,37270,4);

INSERT INTO `employees` VALUES (95213,'Cole','Kesterton','Pharmacist',86119,37270,4);

INSERT INTO `employees` VALUES (96513,'Theresa','Binney','Food Chemist',47354,37270,5);

INSERT INTO `employees` VALUES (98374,'Estrellita','Daleman','Staff Accountant IV',70187,37270,5);

INSERT INTO `employees` VALUES (115357,'Ivy','Fearey','Structural Engineer',92710,37270,5);

# show newly created tables;

SHOW TABLES;

# list all records within employees table

SELECT \* FROM offices;

# list all records within offices table

SELECT \* FROM employees;

# close the mysql terminal

EXIT;

# Part 5

# take a manual snapshot of call-mysql-db-server and name it as manual-snapshot-call-db

# login back to the RDS MySQL DB instance (call-mysql-db-server) as "hr\_guy" using the password defined with SSL pass: Hr\_guy1234

mysql -h call-mysql-db-server.cbanmzptkrzf.us-east-1.rds.amazonaws.com -u hr\_guy -pHr\_guy1234 clarusway

# delete employees who earns salary above $100000 from the clarusway db on call-mysql-db-server (DELETE FROM employees WHERE salary > 100000;)

SELECT \* FROM employees;

DELETE FROM employees WHERE salary > 100000;

# show that only 0 records left in employees tables

SELECT \* FROM employees;

# restore to the point in time as new db instance and name it as "restored-from-time-call-mysqld-db-server"

# login back to the RDS MySQL DB instance (restored-from-time-call-mysqld-db-server) as "hr\_guy" using the password defined Hr\_guy1234

mysql -h restored-from-time-call-mysqld-db-s

erver.cbanmzptkrzf.us-east-1.rds.amazonaws.com -u hr\_guy -p clarusway

# show that deleted records of employees are back in restored-from-time-call-mysqld-db-server

SELECT \* FROM employees;

# delete employees who earn salary above $60000 from the clarusway db on call-mysql-db-server (DELETE FROM employees WHERE salary > 60000;)

DELETE FROM employees WHERE salary > 60000;

# show that only 4 records left in employees tables

SELECT \* FROM employees ORDER BY salary ASC;

# restore from the snapshot (manual-snapshot-call-db) and name it as "restored-from-manual-snapshot-call-db"

# login back to the RDS MySQL DB instance (restored-from-manual-snapshot-call-db) as "hr\_guy" using the password defined Hr\_guy1234

mysql -h restored-from-manual-snapshot-call-db.cbanmzptkrzf.us-east-1.rds.amazonaws.com -u hr\_guy -p clarusway

# show that deleted records of employees are back in restored-from-manual-snapshot-call-db

SELECT \* FROM employees ORDER BY salary ASC;

# delete the restored-from-time-call-mysqld-db-server and restored-from-manual-snapshot-call-db

DELETE FROM employees WHERE salary > 40000;

EXIT;

# Part 6

# show that the clarusway db on RDS DB instance (the restored-from-time-call-mysqld-db-server) has only 4 employee records

# back up the clarusway db from RDS DB instance (the restored-from-time-call-mysqld-db-server) to the file named backup-clarusway.sql on EC2 instance

mysqldump -h restored-from-time-call-mysqld-db-server.cbanmzptkrzf.us-east-1.rds.amazonaws.com -u admin -pClarusway\_1 clarusway > backup-clarusway.sql

# create a database named "migrated\_clarusway" on the "MySQL DB Server" EC2 instance

mysql -u root -pClarusway\_1

CREATE DATABASE migrated\_clarusway;

EXIT;

# restore the backup of clarusway db to the "MySQL DB Server" on EC2 instance with db name migrated\_clarusway

mysql -u root -pClarusway\_1 migrated\_clarusway < backup-clarusway.sql

mysql -u root -pClarusway\_1;

SHOW DATABASES;

USE migrated\_clarusway;

SELECT \* FROM employees;

# delete the call-mysql-db-server, can not delete since it has deletion protection

# modify the call-mysql-db-server and disable deletion protection, then delete the call-mysql-db-server

# Part 7

# launch MySQL Server on EC2 Instance (Amazon Linux 2) and name it 'SQL Course MySQL DB Server'

# update os

sudo yum update -y

# download rpm file using the MySQL community Yum Repository (https://dev.mysql.com/downloads/repo/yum/)

# mysql57-community-release-el7-11.noarch.rpm

sudo wget https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm

# update the yum repo

sudo rpm -ivh mysql57-community-release-el7-11.noarch.rpm

# install the mysql-server

sudo yum install mysql-server -y

# start MySQL service

sudo systemctl start mysqld

# show status of MySQL service. note that MySQL is automatically enabled to start at boot when it is installed.

sudo systemctl status mysqld

# get the temporary password from log file '/var/log/mysqld.log' temp pass: ofBH=MYd2z%r

sudo grep 'temporary password' /var/log/mysqld.log

# setup security installation of MySQL, using following command root: Clarusway\_1

sudo mysql\_secure\_installation

# connect to the MySQL-server and open mysql cli with root user and paste the password when prompted

mysql -u root -p

# create new database named "sql\_course";

CREATE DATABASE sql\_course;

# create user and grant permission to use sql\_course

CREATE USER sql\_guy IDENTIFIED BY 'Sql\_guy1234';

GRANT ALL ON sql\_course.\* TO sql\_guy;

FLUSH PRIVILEGES;

# close the mysql terminal

EXIT;

# back up the sql\_course db from RDS DB instance (the call-mysql-db-server) on "SQL Course MySQL DB Server" EC2 instance using the following command

mysqldump -h call-mysql-db-server.cbanmzptkrzf.us-east-1.rds.amazonaws.com -u admin -pClarusway\_1 sql\_course > backup-sql\_course.sql

# restore the backup of sql\_course db to the "SQL Course MySQL DB Server" on EC2 instance with db name sql\_course

mysql -u root -pClarusway\_1 sql\_course < backup-sql\_course.sql

# connect to the SQL Course MySQL DB Server and open mysql cli with root user and paste the password when prompted Clarusway\_1

mysql -u root -p

# show that sql\_course db ready and available

USE sql\_course;

SHOW TABLES;

SELECT \* FROM countries;