QATIP Intermediate
AWS Lab08
Deploying SQL on AWS

Contents

Part 1

Overview

This part of the lab deploys an AWS EC2 instance running SQL Server 2022 Standard using Terraform onto which a database backup will be restored. This simulates a simple option for migrating an on-premises databases into the cloud. You will be provided with terraform files that will generate errors. You are to remedy the issues and then successfully restore a provided database bak file.

By the end of this lab, you will:

- Define and configure an AWS Virtual Private Cloud (VPC).
- Create a subnet for the EC2 instance.
- Assign an Elastic IP and configure networking.
- Deploy a Windows-based EC2 instance with SQL Server 2022 Standard Edition.

- Implement security rules to allow RDP and SQL access.
- Upload and restore a sample database backup (demo.bak).

Before you begin

Ensure you have completed Lab0 before attempting this lab.

In the IDE terminal pane, enter the following command…

```
cd ~/environment/aws-tf-int/labs/08/EC2
```

This shifts your current working directory to labs/08/EC2. Ensure all commands are executed in this directory

Close any open files and use the Explorer pane to navigate the labs/08/EC2 folder.

Solution

There are no detailed step-by-step instructions for this lab. A solution can be at /aws-tf-int/labs/solutions/08/EC2 but try to use this only as a last resort if you are struggling.

Step 1: Create a key-pair

1. Go to the AWS Console

- Sign in to the AWS Console
- Select the **region** (top-right corner) where you plan to launch your EC2 instance (e.g. eu-west-2)

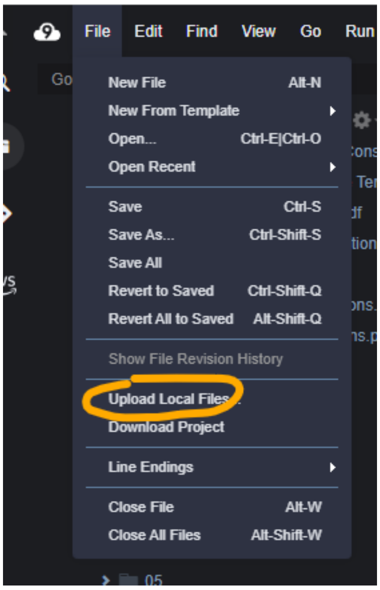
2. Create a key pair

- Go to: **EC2 > Key Pairs**
- Click **Create Key Pair**
- Give it a name like **my-sql-lab-key**
- Leave it as .pem format
- Click **Create Key Pair**

- **my-sql-lab-key.pem** will download to your local machine

3. Upload the pem file to Cloud9

Select the /aws-tf-in/labs/08/EC2 folder and then upload the pem file



4. Set secure permissions

Run **chmod 400 my-sql-lab-key.pem**

This makes sure only *you* can read the file (required for SSH/RDP security)

Step 2: Troubleshoot the terraform files

Step 3: Restore Database from Backup

Retrieve the **Public IP** from Terraform output.

In the AWS console, retrieve the administrator password, providing your pem.

On your local machine, Use **RDP** to connect to the EC2 instance:

Open **Remote Desktop (mstsc.exe)**.

Enable **Local Drives** access.

Enter the **Public IP** and log in using adminuser credentials.

Copy demo.bak from your local machine to C:\Backup\demo.bak on the EC2 instance.

Restore the database using **SQL Server Management Studio (SSMS)**:

```
RESTORE DATABASE Demo
FROM DISK = 'C:\backup\demo.bak'
WITH
 MOVE 'ReactSpringDB' TO 'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\DATA\Demo.mdf',
 MOVE 'ReactSpringDB_log' TO 'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\DATA\Demo.ldf',
 REPLACE,
 RECOVERY,
 STATS = 10;
```

Validate the restore by performing a query against the database

Step 4: Lab Cleanup

Remove all deployed resources: `terraform destroy -autoapprove`