Lab: Build Your DB Server and Interact With Your DB Using an App

Version 4.6.6 (TESS2)

This lab is designed to reinforce the concept of leveraging an AWS-managed database instance for solving relational database needs.

Amazon Relational Database Service (Amazon RDS) 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, which allows you to focus on your applications and business. Amazon RDS provides you with six familiar database engines to choose from: Amazon Aurora, Oracle, Microsoft SQL Server, PostgreSQL, MySQL and MariaDB.

Objectives

After completing this lab, you can:

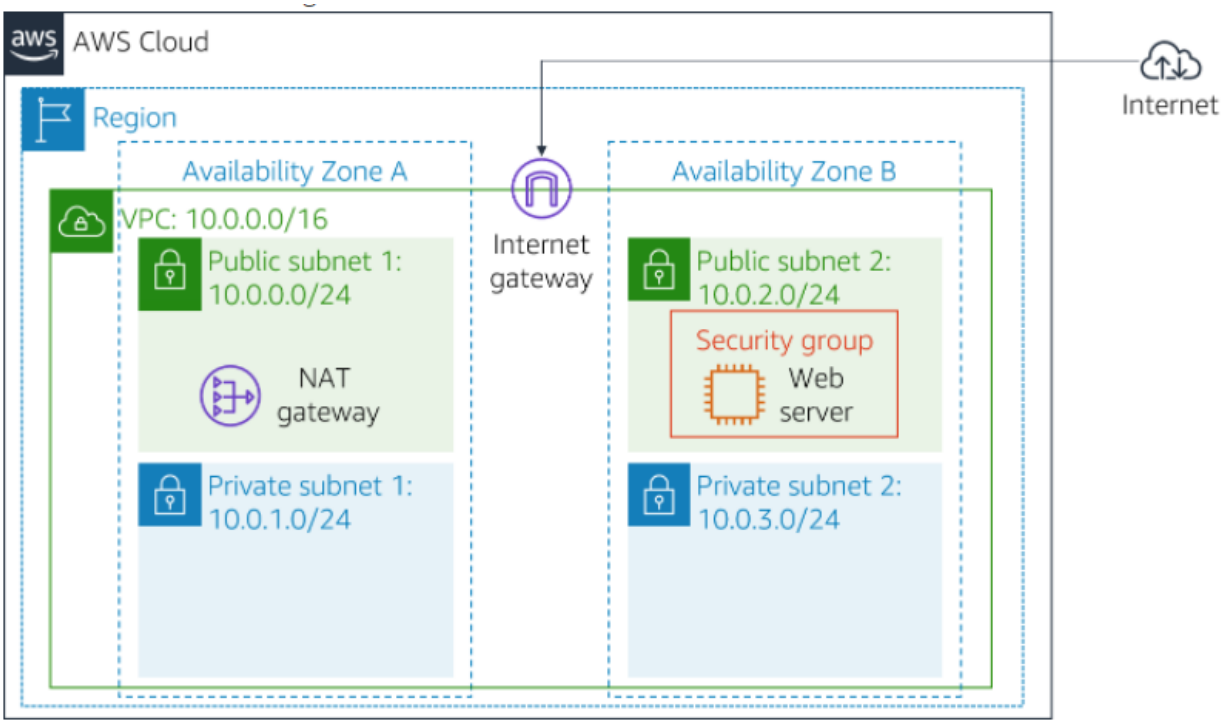
- Launch an Amazon RDS DB instance with high availability.
- Configure the DB instance to permit connections from your web server.
- Open a web application and interact with your database.

Duration

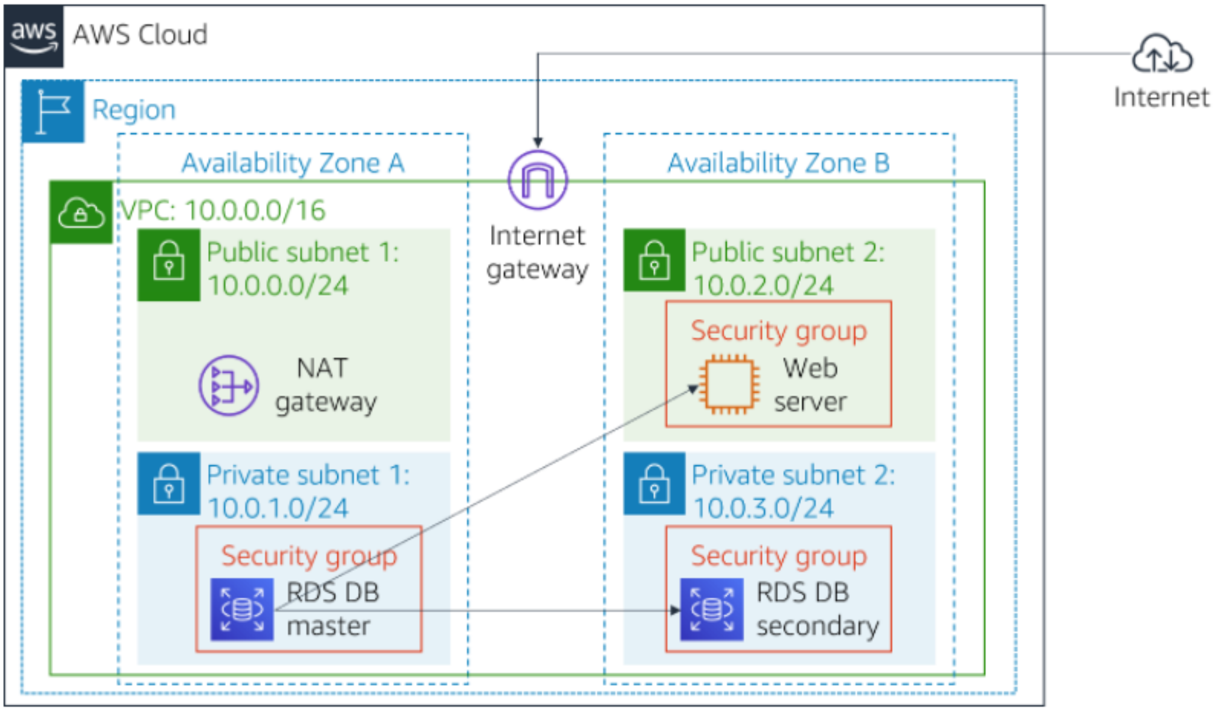
This lab takes approximately **30 minutes**.

Scenario

You start with the following infrastructure:

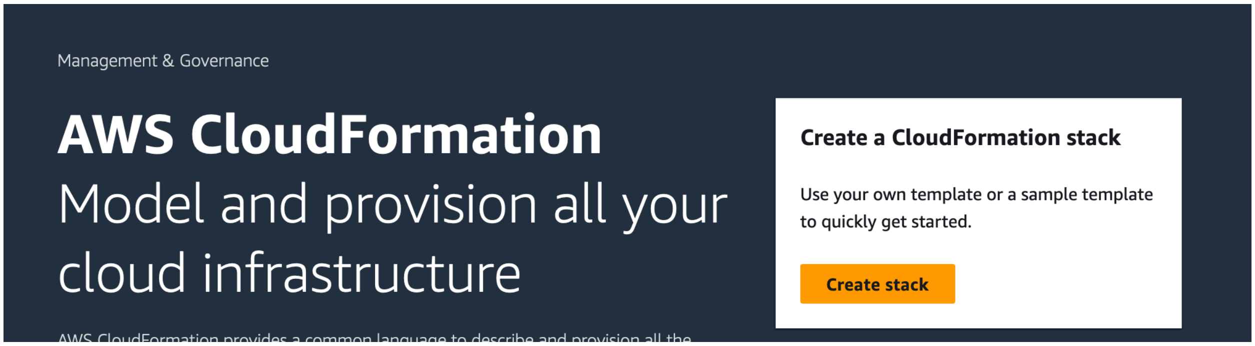


At the end of the lab, this is the infrastructure:



Must do this before doing the Lab

1. On the AWS console, search for **CloudFormation** service. From the search results, click on the **CloudFormation** and once it opens, you will see the option **Create stack**. Choose it.



Management & Governance

AWS CloudFormation

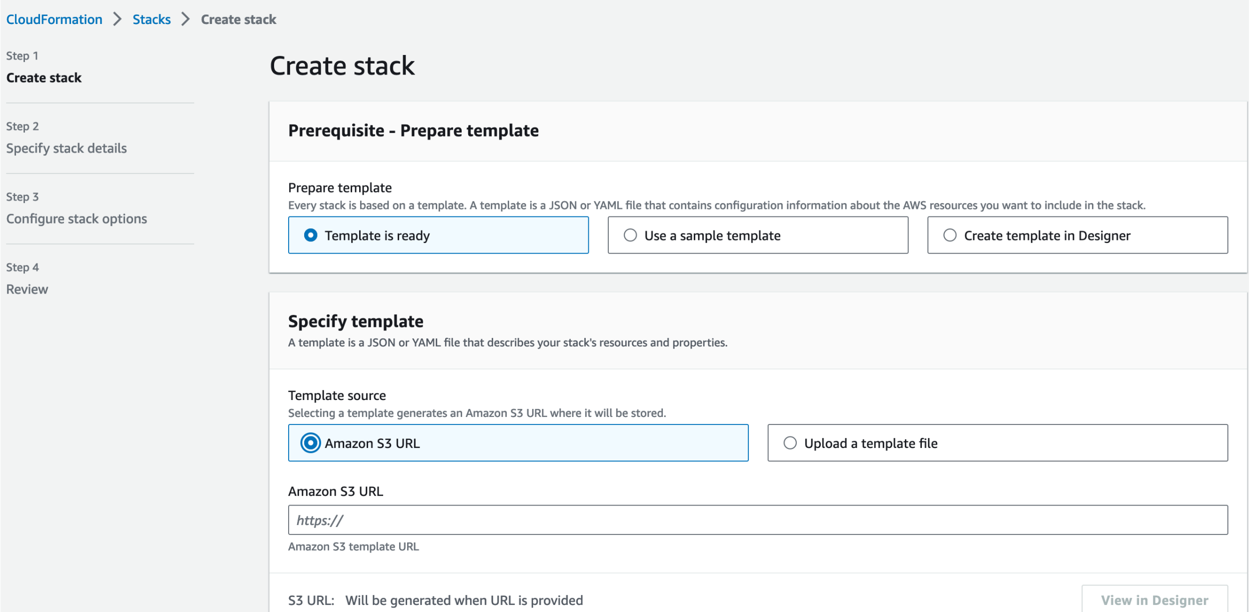
Model and provision all your cloud infrastructure

AWS CloudFormation provides a common language to describe and provision all the

Create a CloudFormation stack

Use your own template or a sample template to quickly get started.

2. We can see that we have multiple options to create infra. Currently we are uploading a template, but you can explore the other options as well.



Create stack

Prerequisite - Prepare template

Prepare template

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

● Template is ready ○ Use a sample template ○ Create template in Designer

Specify template

A template is a JSON or YAML file that describes your stack's resources and properties.

Template source

Selecting a template generates an Amazon S3 URL where it will be stored.

● Amazon S3 URL ○ Upload a template file

Amazon S3 URL

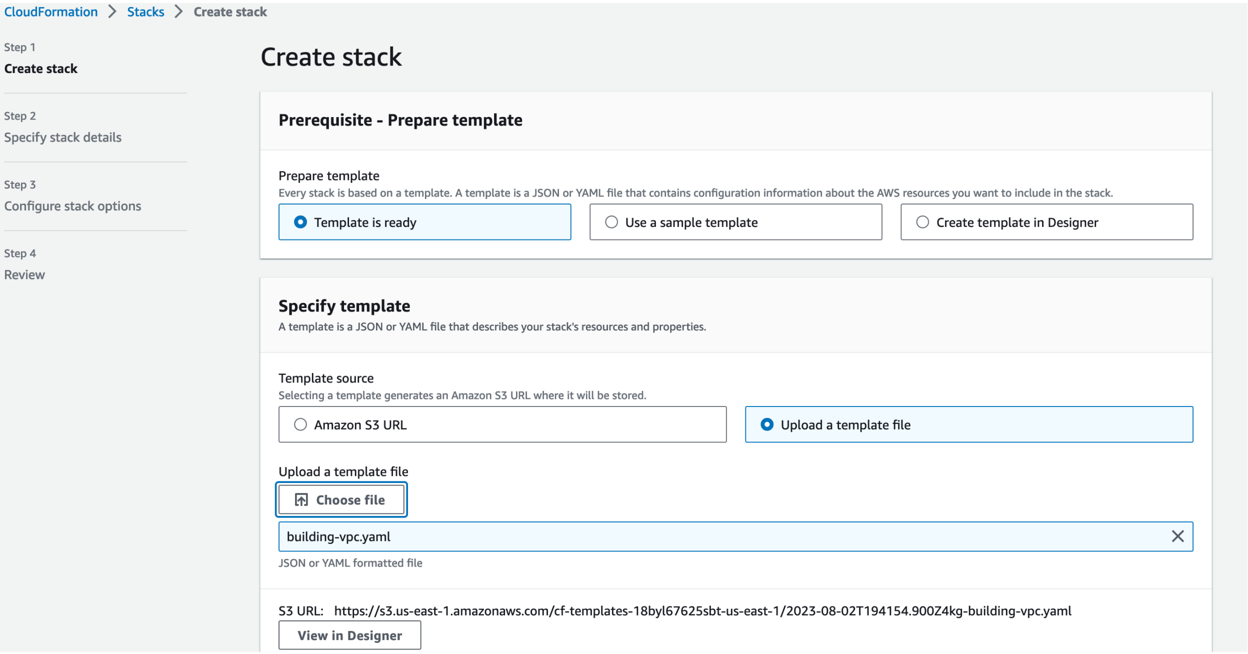
https://

Amazon S3 template URL

S3 URL: Will be generated when URL is provided

Download the code for this step from the URL: https://cloudformation-assignment.s3.amazonaws.com/RDSLabCF.yml

3. Upload the downloaded template. After uploading the template, CloudFormation takes care of syntax validation and prompts an error if there are any. Otherwise, if everything is good, you can successfully upload the template. Click on **next button.**



CloudFormation > Stacks > Create stack

Step 1
Create stack

Step 2
Specify stack details

Step 3
Configure stack options

Step 4
Review

Create stack

Prerequisite - Prepare template

Prepare template
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

● Template is ready ○ Use a sample template ○ Create template in Designer

Specify template
A template is a JSON or YAML file that describes your stack's resources and properties.

Template source
Selecting a template generates an Amazon S3 URL where it will be stored.

○ Amazon S3 URL ● Upload a template file

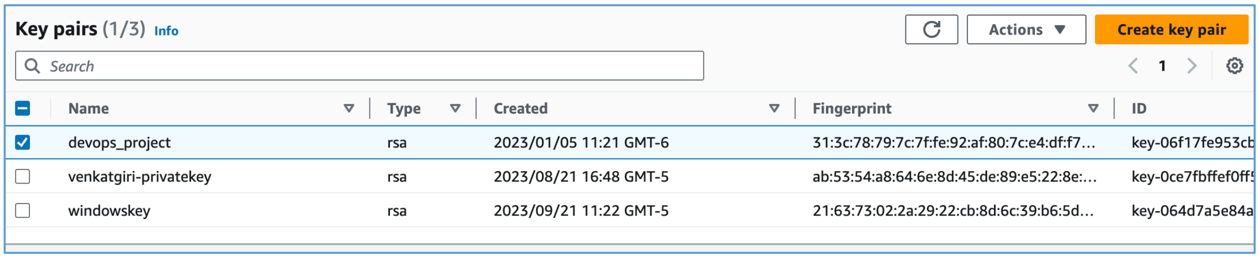
Upload a template file
Choose file
building-vpc.yaml
JSON or YAML formatted file

S3 URL: https://s3.us-east-1.amazonaws.com/cf-templates-18byl67625sbt-us-east-1/2023-08-02T194154.900Z4kg-building-vpc.yaml

View in Designer

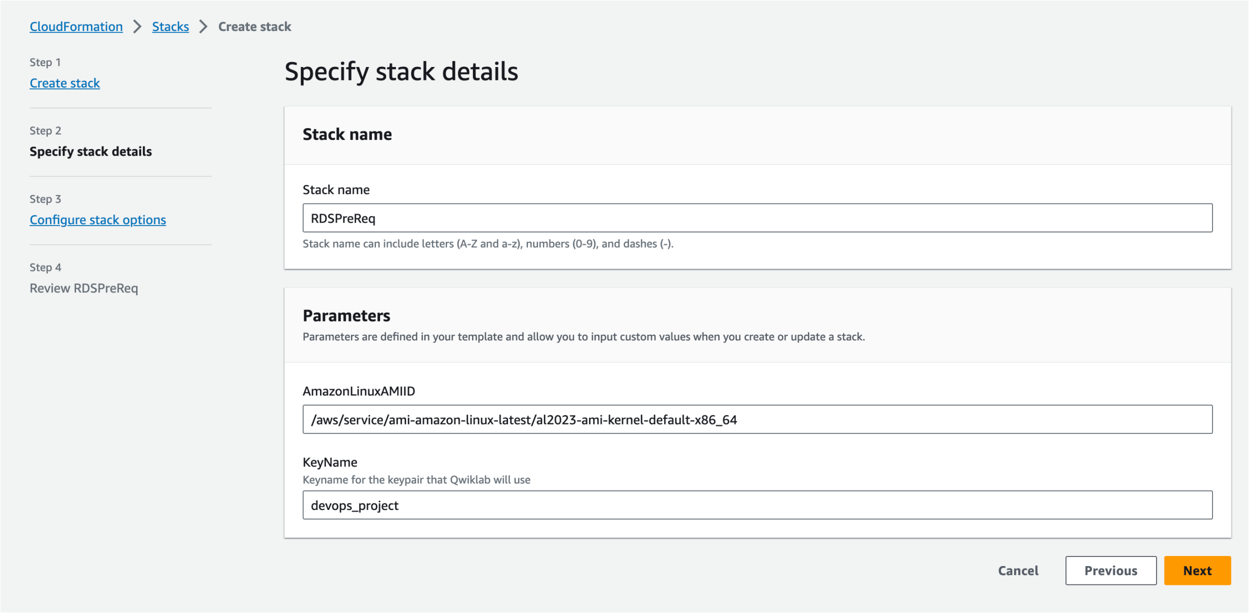
4. Enter **Stack name**. Our infra is maintained through this **stack**. We can modify the stack, delete and replace. The parameters are prompted when we define them in our code. Refer the parameters that were in the code.

You can find the **KeyName** under Keypairs section of the EC2 Service. You should have a keypair by now as you have done many labs. Otherwise, create a keypair and give it under **KeyName**.



Key pairs (1/3) Info

	Name	Type	Created	Fingerprint	ID
☑	devops_project	rsa	2023/01/05 11:21 GMT-6	31:3c:78:79:7c:7f:fe:92:af:80:7c:e4:df:f7…	key-06f17fe953cb
□	venkatgiri-privatekey	rsa	2023/08/21 16:48 GMT-5	ab:53:54:a8:64:6e:8d:45:de:89:e5:22:8e:…	key-0ce7fbffef0ff5
□	windowskey	rsa	2023/09/21 11:22 GMT-5	21:63:73:02:2a:29:22:cb:8d:6c:39:b6:5d…	key-064d7a5e84a



Specify stack details

Stack name

Stack name: RDSPreReq

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

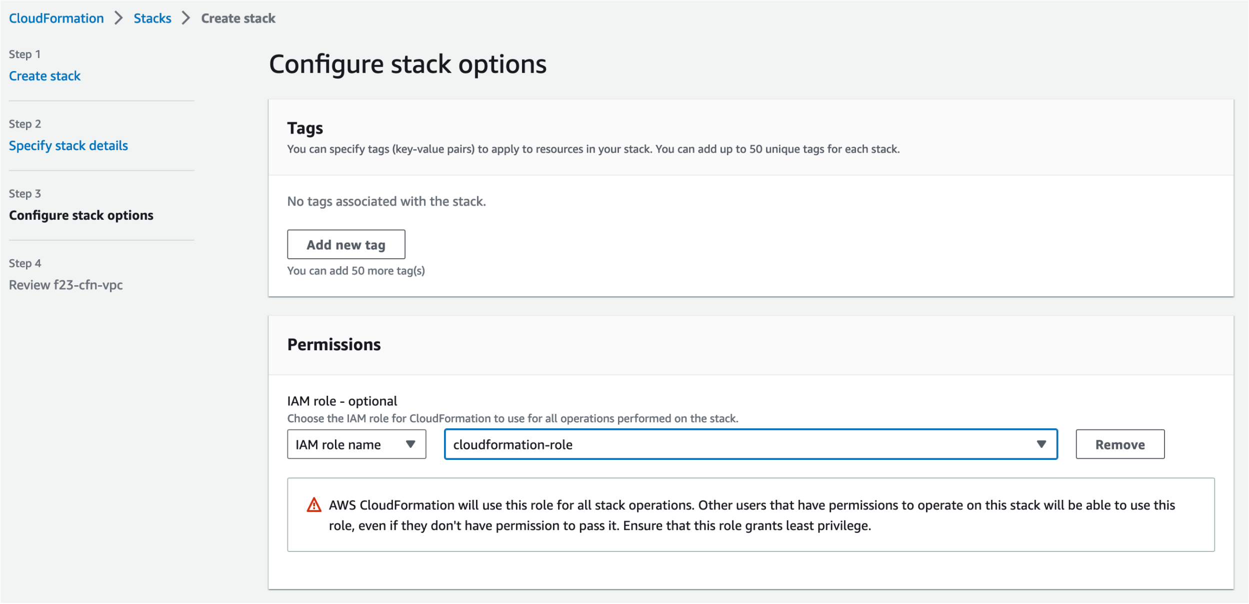
Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

AmazonLinuxAMIID: /aws/service/ami-amazon-linux-latest/al2023-ami-kernel-default-x86_64

KeyName
Keyname for the keypair that Qwiklab will use: devops_project

5. I have created a role named **cloudformation-role** with **poweruser** policy attached. You can create your own or leave it with no option selected. **CloudFormation** takes care of it. Leave the rest of the options to the default and click next.



Configure stack options

Tags

You can specify tags (key-value pairs) to apply to resources in your stack. You can add up to 50 unique tags for each stack.

No tags associated with the stack.

You can add 50 more tag(s)

Permissions

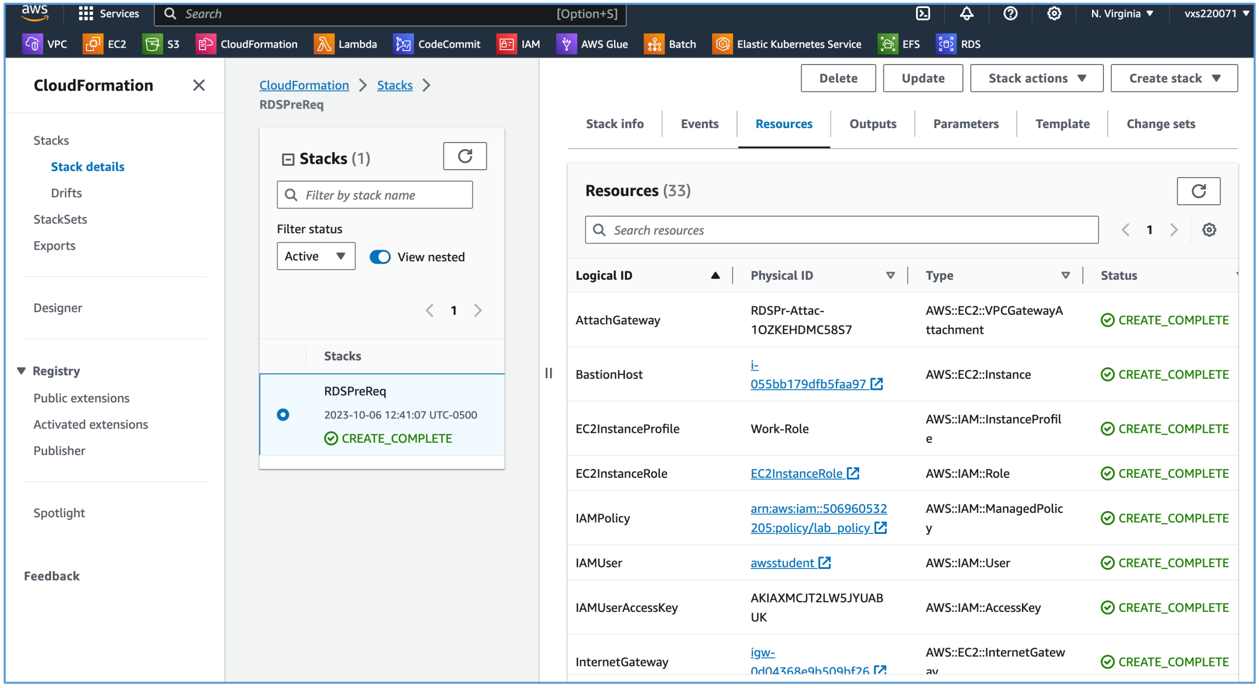
IAM role - optional
Choose the IAM role for CloudFormation to use for all operations performed on the stack.

IAM role name: cloudformation-role

⚠ AWS CloudFormation will use this role for all stack operations. Other users that have permissions to operate on this stack will be able to use this role, even if they don't have permission to pass it. Ensure that this role grants least privilege.

6. You can review the options in this step and make changes if needed. Scroll down to the bottom, acknowledge and click on submit.

7. After submission, you can see the below page. We can see **create_complete** under **Status**. You can see various options. Go through them and try to understand.



Logical ID ▲	Physical ID ▽	Type ▽	Status
AttachGateway	RDSPr-Attac-1OZKEHDMC58S7	AWS::EC2::VPCGatewayAttachment	✓ CREATE_COMPLETE
BastionHost	i-055bb179dfb5faa97	AWS::EC2::Instance	✓ CREATE_COMPLETE
EC2InstanceProfile	Work-Role	AWS::IAM::InstanceProfile	✓ CREATE_COMPLETE
EC2InstanceRole	EC2InstanceRole	AWS::IAM::Role	✓ CREATE_COMPLETE
IAMPolicy	arn:aws:iam::506960532205:policy/lab_policy	AWS::IAM::ManagedPolicy	✓ CREATE_COMPLETE
IAMUser	awsstudent	AWS::IAM::User	✓ CREATE_COMPLETE
IAMUserAccessKey	AKIAXMCJT2LW5JYUABUK	AWS::IAM::AccessKey	✓ CREATE_COMPLETE
InternetGateway	igw-0d04368e9b509bf26	AWS::EC2::InternetGateway	✓ CREATE_COMPLETE

Note: This is a deliverable. Capture the above screenshot as shown above.

Task 1: Create a Security Group for the RDS DB Instance

In this task, you will create a security group to allow your web server to access your RDS DB instance. The security group will be used when you launch the database instance.

5. In the **AWS Management Console**, on the **Services** menu, click **VPC**.
6. In the left navigation pane, click **Security Groups**.
7. Click **Create security group** and then configure:

- **Security group name:** DB Security Group
- **Description:** Permit access from Web Security Group
- **VPC:** *Lab VPC*

You will now add a rule to the security group to permit inbound database requests.

8. In the **Inbound rules** pane, choose **Add rule**

   The security group currently has no rules. You will add a rule to permit access from the *Web Security Group*.

9. Configure the following settings:

- **Type:** *MySQL/Aurora (3306)*
- **CIDR, IP, Security Group or Prefix List:** Type `sg` and then select *Web Security Group*.

   This configures the Database security group to permit inbound traffic on port 3306 from any EC2 instance that is associated with the *Web Security Group*.

10. Choose **Create security group**

   You will use this security group when launching the Amazon RDS database.

   Note: This is a Deliverable. Once the security group is created, capture the screenshot with inbound rules tab.

Task 2: Create a DB Subnet Group

In this task, you will create a *DB subnet group* that is used to tell RDS which subnets can be used for the database. Each DB subnet group requires subnets in at least two Availability Zones.

11. On the **Services** menu, click **RDS**.
12. In the left navigation pane, click **Subnet groups**.

   If the navigation pane is not visible, click the menu icon in the top-left corner.

13. Click **Create DB Subnet Group** then configure:

- **Name:** `DB-Subnet-Group`
- **Description:** `DB Subnet Group`
- **VPC:** *Lab VPC*

14. Scroll down to the **Add Subnets** section.
15. Expand the list of values under **Availability Zones** and select the first two zones: **us-east-1a** and **us-east-1b**.
16. Expand the list of values under **Subnets** and select the subnets associated with the CIDR ranges **10.0.1.0/24** and **10.0.3.0/24**.

   These subnets should now be shown in the **Subnets selected** table.

17. Click **Create**

    You will use this DB subnet group when creating the database in the next task.

    Note: This is a deliverable. Once this is created, capture the screenshot.

Task 3: Create an Amazon RDS DB Instance

In this task, you will configure and launch a Multi-AZ Amazon RDS for MySQL database instance.

Amazon RDS ***Multi-AZ*** deployments provide enhanced availability and durability for Database (DB) instances, making them a natural fit for production database workloads. When you provision a Multi-AZ DB instance, Amazon RDS automatically creates a primary DB instance and synchronously replicates the data to a standby instance in a different Availability Zone (AZ).

18. In the left navigation pane, click **Databases**.
19. Click **Create database**

    If you see **Switch to the new database creation flow** at the top of the screen, please click it.

20. Select **MySQL**.
21. Under **Settings**, configure:
    - **DB instance identifier:** `lab-db`
    - **Master username:** `main`
    - **Master password:** `lab-password`
    - **Confirm password:** `lab-password`
22. Under **DB instance size**, configure:
    - Select **Burstable classes (includes t classes)**.
    - Select *db.t3.micro*
23. Under **Storage**, configure:
    - **Storage type:** *General Purpose (SSD)*
    - **Allocated storage:** *20*
24. Under **Connectivity**, configure:
    - **Virtual Private Cloud (VPC):** *Lab VPC*
25. Under **Existing VPC security groups**, from the dropdown list:
    - Choose *DB Security Group*.
    - Deselect *default*.
26. Expand **Additional configuration**, then configure:
    - **Initial database name:** `lab`

- Uncheck **Enable automatic backups**.
- Uncheck **Enable Enhanced monitoring**.

This will turn off backups, which is not normally recommended, but will make the database deploy faster for this lab.

27. Click **Create database**

Your database will now be launched.

If you receive an error that mentions "not authorized to perform: iam:CreateRole", make sure you unchecked *Enable Enhanced monitoring* in the previous step.

28. Click **lab-db** (click the link itself).

You will now need to wait **approximately 4 minutes** for the database to be available. The deployment process is deploying a database in two different Availability zones.

While you are waiting, you might want to review the Amazon RDS FAQs or grab a cup of coffee.

29. Wait until **Info** changes to **Modifying** or **Available**.

Note: This is a Deliverable. After the successful creation, click on the database created and capture the screenshot with all the details visible.

30. Scroll down to the **Connectivity & security** section and copy the **Endpoint** field.

It will look similar to: *lab-db.cggq8lhnxvnv.us-west-2.rds.amazonaws.com*

31. Paste the Endpoint value into a text editor. You will use it later in the lab.

Task 4: Interact with Your Database

In this task, you will open a web application running on your web server and configure it to use the database.

32. With the Cloudformation Code, you have created the webserver, i.e., EC2 instance. Open the EC2 console, and you will see a webserver is running. Copy the public IP of that.

33. Open a new web browser tab, paste the *WebServer* IP address and press Enter.

    The web application will be displayed, showing information about the EC2 instance.

34. Click the **RDS** link at the top of the page.

    You will now configure the application to connect to your database.

35. Configure the following settings:

- **Endpoint:** Paste the Endpoint you copied to a text editor earlier
- **Database:** `lab`
- **Username:** `main`
- **Password:** `lab-password`
- Click **Submit**

Note: This is a Deliverable. Capture the screenshot of the RDS tab.

Lab Complete

Congratulations! You have completed the lab.

Lab Clean Up:

Delete the RDS, and Cloudformation stack.

Attributions

Bootstrap v3.3.5 - http://getbootstrap.com/