Build Solutions across VPCs with Peering

01:38:54 Exit Lab ✓ Complete Lab

Guided Mode 2 hours duration Apprentice Rate this lab

Videos Guide

Build Solutions across VPCs with Peering

Introduction

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses. In this lab, you will create a new VPC for your WordPress blog to run from. You will then create a VPC peering connection between the new VPC and an existing database VPC. By the end of this lab, you will understand how to create a new VPC from scratch, attach internet gateways, edit routing tables, and peer multiple VPCs together.

Solution

Log in to the AWS Management Console using the credentials provided on the lab instructions page. Make sure you're in the N. Virginia (`us-east-1`) Region throughout the lab.

Create `Web_VPC` Subnets and Attach a New Internet Gateway

Create a VPC

1. Use the top search bar to look for and navigate to **VPC**.

2. Under **Resources by Region**, click **VPCs**.
3. Use the top search bar to look for and navigate to **RDS** in a new tab.
4. Click **DB Instances**, and observe the instance created for this lab.

> **Note:** Keep this tab open for use later on in the lab.

5. Go back to your VPC tab, and click **Create VPC**.
6. Ensure the `VPC only` option is selected.
7. Set the following values:
   - **Name tag:** Enter *Web_VPC*.
   - **IPv4 CIDR block:** Enter `192.168.0.0/16`.
8. Leave the rest of the settings as their defaults, and click **Create VPC**.

Create a Subnet

1. On the left menu under **VIRTUAL PRIVATE CLOUD**, select **Subnets**.
2. Click **Create subnet**.
3. For **VPC ID**, select the newly created `Web_VPC`.
4. Under **Subnet settings**, set the following values:
   - **Subnet name:** Enter *WebPublic*.
   - **Availability Zone:** Select **us-east-1a**.
   - **IPv4 CIDR block:** Enter `192.168.0.0/24`.
5. Click **Create subnet**.

Create an Internet Gateway

1. On the left menu, select **Internet Gateways**.
2. Click **Create internet gateway**.
3. For **Name tag**, enter *WebIG*.
4. Click **Create internet gateway**.
5. In the green notification at the top of the page, click **Attach to a VPC**.
6. In **Available VPCs**, select the `Web_VPC` and click **Attach internet gateway**.
7. On the left menu, select **Route Tables**.
8. Select the checkbox for the `Web_VPC`.
9. Underneath, select the **Routes** tab and click **Edit routes**.
10. Click **Add route**.

11. Set the following values:
    - **Destination:** Enter `0.0.0.0/0`.
    - **Target:** Select **Internet Gateway**, and select the internet gateway that appears in the list.
12. Click **Save changes**.

Create a Peering Connection

1. On the left menu, select **Peering Connections**.
2. Click **Create peering connection**.
3. Set the following values:
    - **Name:** Enter *DBtoWeb*.
    - **VPC (Requester):** Select the `DB_VPC`.
    - **VPC (Accepter):** Select the `Web_VPC`.
4. Click **Create peering connection**.
5. At the top of the page, click **Actions** > **Accept request**.
6. Click **Accept request**.
7. On the left menu, select **Route Tables**.
8. Select the checkbox for the `Web_VPC`.
9. Underneath, select the **Routes** tab, and click **Edit routes**.
10. Click **Add route**.
11. Set the following values:
    - **Destination:** Enter `10.0.0.0/16`.
    - **Target:** Select **Peering Connection**, and select the peering connection that appears in the list.
12. Click **Save changes**.
13. Go back to **Route Tables**, and select the checkbox for the `DB_VPC` instance with a **Main** column value of **Yes**.
14. Underneath, select the **Routes** tab, and click **Edit routes**.
15. Click **Add route**.
16. Set the following values:
    - **Destination:** Enter `192.168.0.0/16`.
    - **Target:** Select **Peering Connection**, and select the peering connection that appears in the list.
17. Click **Save changes**.

Create an EC2 Instance and Configure WordPress

1. In a new browser tab, navigate to EC2.
2. Click **Launch instance** > **Launch instance**.
3. Scroll down and under **Quick Start**, select the **Ubuntu** image box. (You can skip the **Name** field before this.)
4. Under **Amazon Machine Image (AMI)**, click the dropdown and select **Ubuntu Server 24.04 LTS**.
5. Under **Instance type**, click the dropdown and select **t3.micro**.
6. For **Key pair**, click the dropdown and select **Proceed without a key pair**.
7. In the **Network settings** section, click the **Edit** button.
8. Set the following values:
    - **VPC:** Select the `Web_VPC`.
    - **Subnet:** Ensure the **WebPublic** subnet is selected.
    - **Auto-assign public IP:** Select **Enable**.
9. Under **Firewall (security groups)**, ensure **Create security group** is selected (the default value).
10. Scroll down and click **Add security group rule**.
11. Set the following values for the new rule (i.e., **Security group rule 2**):
    - **Type:** Select **HTTP**.
    - **Source:** Select `0.0.0.0/0`.
12. Scroll to the bottom, and expand **Advanced details**.
13. At the bottom, under **User data**, copy and paste the following bootstrap script:

```
#!/bin/bash
sudo apt update
sudo apt install apache2 php libapache2-mod-php php-mysql php-curl php-gd
php-mbstring php-xml php-xmlrpc php-soap php-intl php-zip unzip -y
sudo ufw allow in "Apache"
sudo a2enmod rewrite
systemctl restart apache2
cd /tmp/ && wget https://wordpress.org/latest.zip
unzip latest.zip -d /var/www
```

```
chown -R www-data:www-data /var/www/wordpress/
mv /var/www/wordpress/wp-config-sample.php /var/www/wordpress/wp-config.php
cd /var/www/wordpress/
perl -pi -e "s/database_name_here/wordpress/g" wp-config.php
perl -pi -e "s/username_here/wordpress/g" wp-config.php
perl -pi -e "s/password_here/wordpress/g" wp-config.php
perl -i -pe'
BEGIN {
@chars = ("a" .. "z", "A" .. "Z", 0 .. 9);
push @chars, split //, "!@#$%^&*()-_ []{}<>~\`+=,.;:/?|";
sub salt { join "", map $chars[ rand @chars ], 1 .. 64 }
}
s/put your unique phrase here/salt()/ge
' wp-config.php
wget https://raw.githubusercontent.com/ACloudGuru-Resources/course-aws-certified-solutions-architect-associate/main/lab/5/000-default.conf
mkdir wp-content/uploads
chmod 775 wp-content/uploads
mv 000-default.conf /etc/apache2/sites-enabled/
systemctl restart apache2
```

14. At the bottom, click **Launch Instance**.

    > **Note:** It may take a few minutes for the new instance to launch.

15. From the green box that appears after the instance launches, open the link for the instance in a new browser tab.
16. Observe the **Instance state** column, and check to ensure it is **Running** before you proceed.
17. Select the checkbox for the new instance and click **Connect**.
18. Click **Connect**.

    > **Note:** The startup script for the instance may take a few minutes to complete and you may need to wait for it to complete before proceeding with the next step.

19. To confirm WordPress installed correctly, view the configuration files:

    ```
    cd /var/www/wordpress
    ls
    ```

20. To configure WordPress, open `wp-config.php`:

    ```
    sudo vim wp-config.php
    ```

21. Go back to your browser tab with RDS.
22. Click the link to open the provisioned RDS instance.
23. Under **Connectivity & security**, copy the RDS **Endpoint**.
24. Go back to the tab with the terminal, and scroll down to `/** MySQL hostname */`.
25. Press `i` to enter Insert mode.
26. Replace `localhost` with the RDS endpoint you just copied. Ensure it remains wrapped in single quotes.
27. Press **ESC** followed by `:wq`, and press **Enter**. Leave this tab open.

Modify the RDS Security Groups to Allow Connections from the `Web_VPC` VPC

1. Go back to your RDS browser tab.
2. In **Connectivity & security**, click the active link under **VPC security groups**.
3. Checkmark the **DatabaseSG** Security Group.
4. At the bottom, select the **Inbound rules** tab.
5. Click **Edit inbound rules**.
6. Click **Add rule**.
7. Under **Type**, search for and select **MYSQL/Aurora**.
8. Under **Source**, search for and select `192.168.0.0/16`.
9. Click **Save rules**.
10. Return to the terminal page.
11. Below the terminal window, copy the public IP address of your server.
12. Open a new browser tab and paste the public IP address in the address bar. You should now see the WordPress installation page.
13. Set the the following values:

   - **Site Title:** Enter *A Blog Guru.*
   - **Username:** Enter *guru.*
   - **Your Email:** Enter ***test@test.com***.

14. Click **Install WordPress**.
15. Reload the public IP address in the address bar to view your newly created WordPress blog.

Conclusion

Congratulations — you've completed this hands-on lab!

Troubleshooting

If the website isn't loading the way you expected at the end of this lab, here are some tips to help troubleshoot:

- Check the status of the lab objectives - are any not yet completed?
- Is everything you set up ready to use? Check things like the VPC peering connection, which requires you to specifically accept the connection request.
- Does the database error page load after a minute or so of waiting, or does no page load at all? This gives a hint on whether the issue may be with the peering or the security groups.

Tools

Lab Diagram Instant Terminal

Credentials

How do I connect?

AWS Account

Username

cloud_user

Password

$IFT0M4uhS4i%&5ao|jD

Open Link in Incognito Window

Additional Resources

Note: A few users reported issues using Firefox with this lab. If you have trouble, we recommend using a different browser. We expect this issue will be resolved with upcoming Firefox updates.

Logging In

Log in to the AWS Management Console using the credentials provided on the lab instructions page. Make sure you're in the N. Virginia (us-east-1) Region throughout the lab.

Learning Objectives

0 of 5 completed

Optional: Run progress checks to confirm you've completed the objectives

- ✓ **Create Web_VPC Subnets and Attach a New Internet Gateway**
- ✓ **Create a Peering Connection**
- ✓ **Create an EC2 Instance and configure Wordpress**
- ✓ **Modify the RDS Security Groups to Allow Connections from the Web_VPC VPC**
- ✓ **Test WordPress**