Create VPC and other cool stuff:

**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.

1. Go back to your VPC tab, and click **Create VPC**.
2. Ensure the VPC only option is selected.
3. Set the following values:
   * **Name tag:** Enter *Web\_VPC*.
   * **IPv4 CIDR block:** Enter 192.168.0.0/16.
4. 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 20.04 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:
14. #!/bin/bash
15. sudo apt update -y
16. sudo apt install php-curl php-gd php-mbstring php-xml php-xmlrpc php-soap php-intl php-zip perl mysql-server apache2 libapache2-mod-php php-mysql -y
17. wget https://github.com/ACloudGuru-Resources/course-aws-certified-solutions-architect-associate/raw/main/lab/5/wordpress.tar.gz
18. tar zxvf wordpress.tar.gz
19. cd wordpress
20. wget https://raw.githubusercontent.com/ACloudGuru-Resources/course-aws-certified-solutions-architect-associate/main/lab/5/000-default.conf
21. cp wp-config-sample.php wp-config.php
22. perl -pi -e "s/database\_name\_here/wordpress/g" wp-config.php
23. perl -pi -e "s/username\_here/wordpress/g" wp-config.php
24. perl -pi -e "s/password\_here/wordpress/g" wp-config.php
25. perl -i -pe'
26. BEGIN {
27. @chars = ("a" .. "z", "A" .. "Z", 0 .. 9);
28. push @chars, split //, "!@#$%^&\*()-\_ []{}<>~\`+=,.;:/?|";
29. sub salt { join "", map $chars[ rand @chars ], 1 .. 64 }
30. }
31. s/put your unique phrase here/salt()/ge
32. ' wp-config.php
33. mkdir wp-content/uploads
34. chmod 775 wp-content/uploads
35. mv 000-default.conf /etc/apache2/sites-enabled/
36. mv /wordpress /var/www/
37. apache2ctl restart
38. At the bottom, click **Launch Instance**.

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

1. From the green box that appears after the instance launches, open the link for the instance in a new browser tab.
2. Observe the **Instance state** column, and check to ensure it is **Running** before you proceed.
3. Select the checkbox for the new instance and click **Connect**.
4. 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.

1. To confirm WordPress installed correctly, view the configuration files:
2. cd /var/www/wordpress
3. ls
4. To configure WordPress, open wp-config.php:
5. sudo vim wp-config.php
6. Go back to your browser tab with RDS.
7. Click the link to open the provisioned RDS instance.
8. Under **Connectivity & security**, copy the RDS **Endpoint**.
9. Go back to the tab with the terminal, and scroll down to /\*\* MySQL hostname \*/.
10. Press i to enter Insert mode.
11. Replace localhost with the RDS endpoint you just copied. Ensure it remains wrapped in single quotes.
12. 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. At the bottom, select the **Inbound rules** tab.
4. Click **Edit inbound rules**.
5. Click **Add rule**.
6. Under **Type**, search for and select **MYSQL/Aurora**.
7. Under **Source**, search for and select 192.168.0.0/16.
8. Click **Save rules**.
9. Return to the terminal page.
10. Below the terminal window, copy the public IP address of your server.
11. Open a new browser tab and paste the public IP address in the address bar. You should now see the WordPress installation page.
12. Set the the following values:
    * **Site Title:** Enter *A Blog Guru*.
    * **Username:** Enter *guru*.
    * **Your Email:** Enter [*test@test.com*](mailto:test@test.com).
13. Click **Install WordPress**.
14. 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.