Guided Lab: Creating a VPC Peering Connection

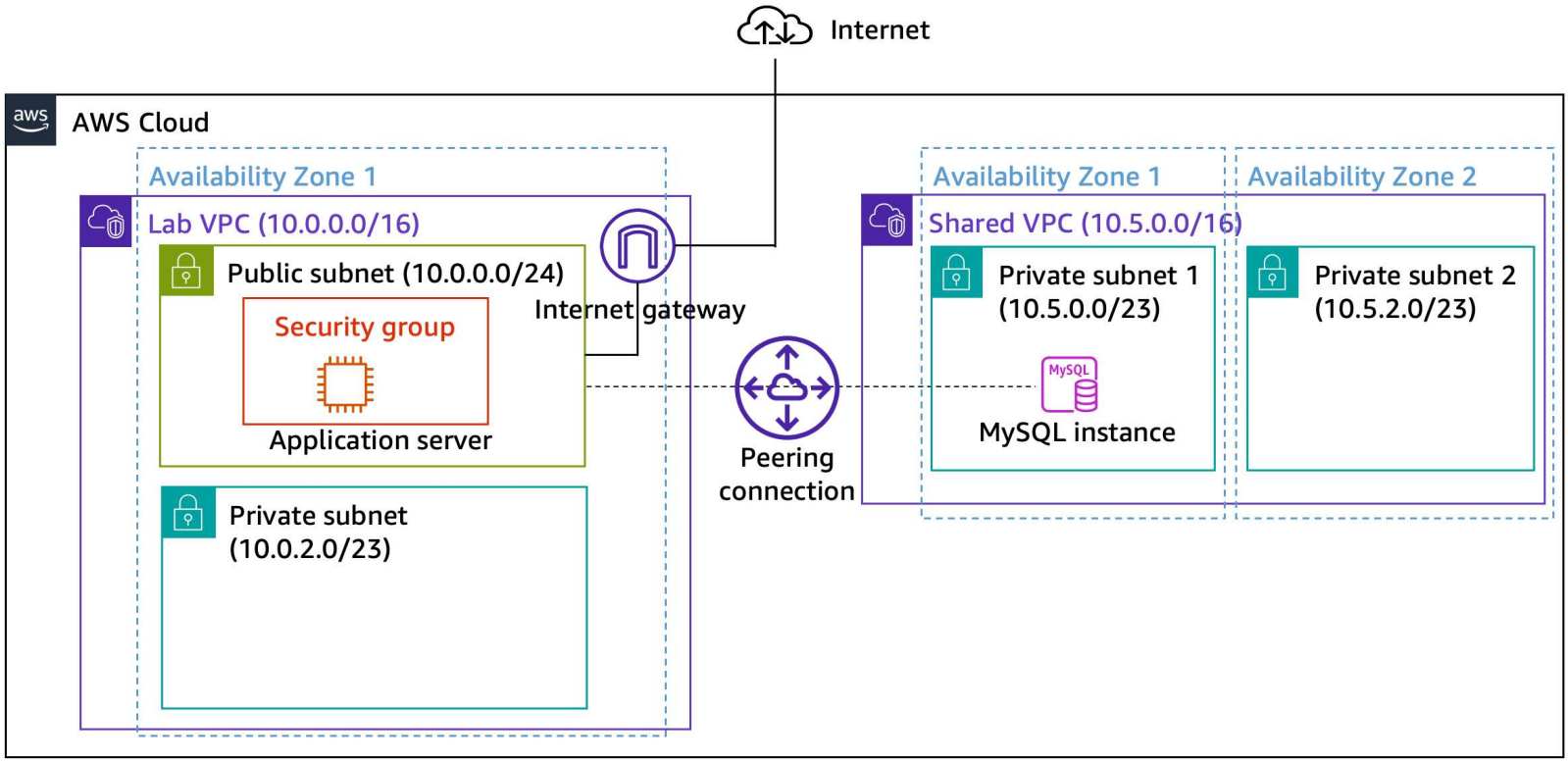
Lab overview and objectives

You need to connect your virtual private clouds (VPCs) when you must transfer data between them. This lab shows you how to create a private VPC peering connection between two VPCs.

After completing this lab, you should be able to do the following:

- Create a VPC peering connection.
- Configure route tables to use the VPC peering connection.
- Enable VPC Flow Logs to provide insight on the data moving across the network.
- Test a peering connection.
- Analyze the VPC flow logs.

At the **end** of this lab, your architecture will look like the following example:



Duration

This lab requires approximately **30 minutes** to complete.

Accessing the AWS Management Console

1. At the top of these instructions, choose **Start Lab** to launch your lab.

   Tip: If you need more time to complete the lab, choose **Start Lab** again to restart the timer for the environment.

   The following information indicates the lab status:

   - A red circle next to **AWS** at the upper-left corner of this page indicates that the lab has not been started.
   - A yellow circle next to **AWS** at the upper-left corner of this page indicates that the lab is starting.

- A green circle next to **AWS** at the upper-left corner of this page indicates that the lab is ready.

Wait for the lab to be ready before proceeding.

2. At the top of these instructions, choose the green circle next to **AWS**.

   This option opens the AWS Management Console in a new browser tab. The system automatically signs you in.

   Tip: If a new browser tab does not open, a banner or icon at the top of your browser will indicate that your browser is preventing the site from opening pop-up windows. Choose the banner or icon, and choose **Allow pop-ups**.

3. Arrange the AWS Management Console tab so that it displays along side these instructions. Ideally, you should be able to see both browser tabs at the same time so that you can follow the lab steps.

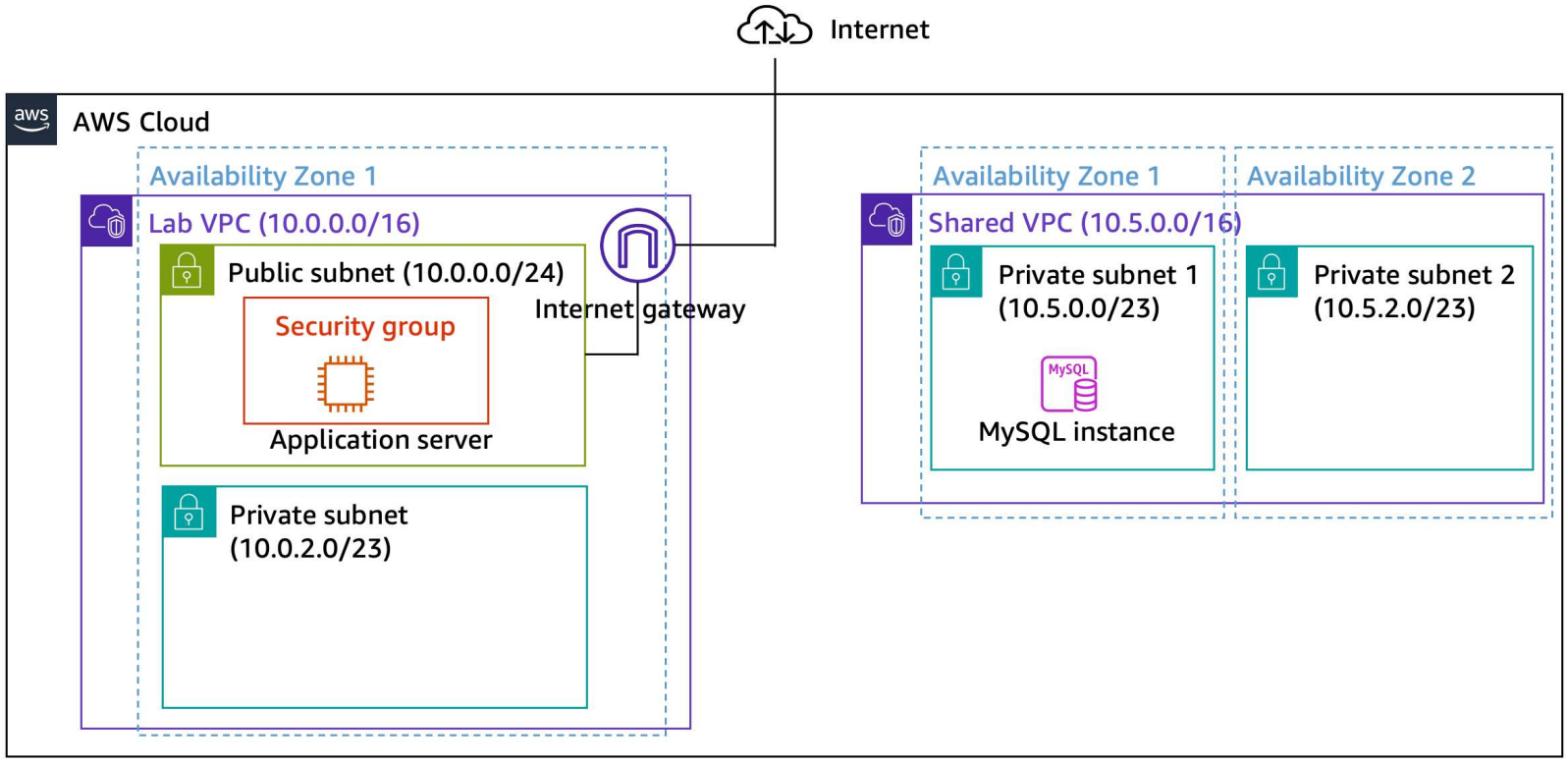
   Do not change the lab Region unless specifically instructed to do so.

Task 1: Creating a VPC peering connection

Your task is to create a VPC peering connection between two VPCs.

A *VPC peering connection* is a one-to-one networking connection between two VPCs that enables you to route traffic between them privately. Instances in either VPC can communicate with each other like they are in the same network. You can create a VPC peering connection between your own VPCs, in a VPC in another AWS account, or with a VPC in a different AWS Region.

Two VPCs are provided as part of this lab: *Lab VPC* and *Shared VPC*. *Lab VPC* has an inventory application that runs on an Amazon Elastic Compute Cloud (Amazon EC2) instance in a public subnet. *Shared VPC* has a database instance that runs in a private subnet.



4. On the **AWS Management Console**, in the **Search** bar at the top, enter and choose `VPC` to open the **VPC Management Console**.
5. In the left navigation pane, choose **Peering connections**.
6. Choose **Create peering connection** and configure the following settings:
   - **Name - optional:** `Lab-Peer`
   - **VPC ID (Requester):** *Lab VPC*

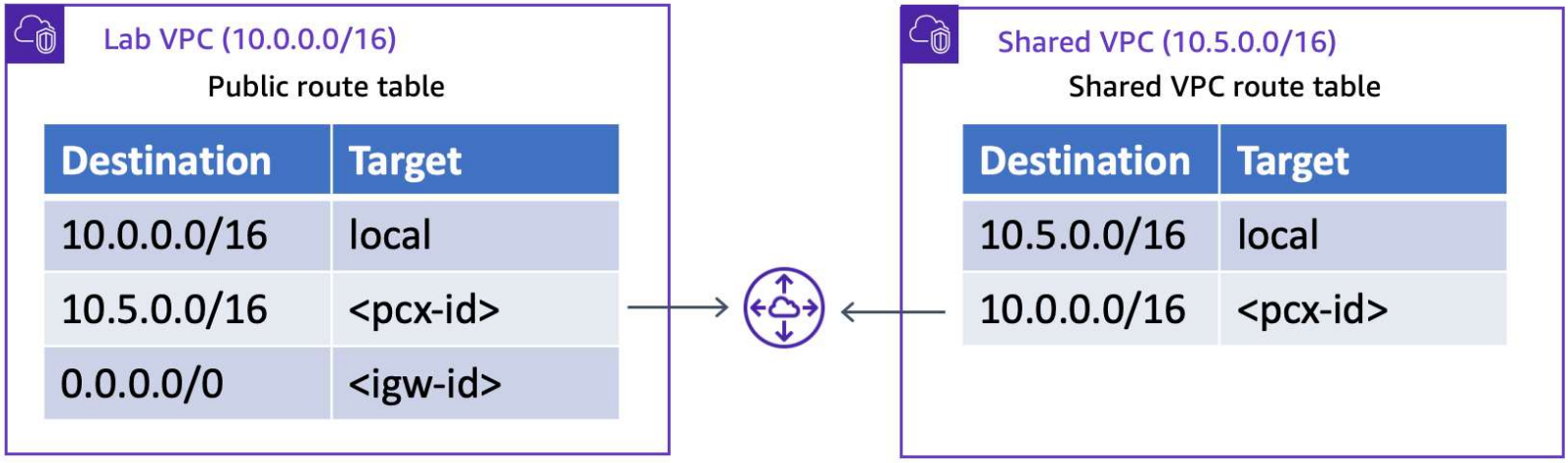
- **VPC ID (Accepter):** *Shared VPC*
- Choose **Create peering connection**.

When a VPC peering connection is created, the target VPC must accept the connection request because it might be owned by a different account. Alternatively, the user that creates the peering connection might not have permission to accept the connection request for the target VPC. However, in this lab, you will accept the connection yourself.

7. On the next screen, from the **Actions** dropdown list, choose **Accept request**.

Task 2: Configuring route tables

You will now update the route tables in both VPCs to send traffic from *Lab VPC* to the peering connection for *Shared VPC*.



8. In the left navigation pane, choose **Route Tables**.
9. Select **Lab Public Route Table** (for *Lab VPC*).

   You will configure the *Public Route Table* associated with *Lab VPC*. If the destination IP address falls in the range of *Shared VPC*, the *Public Route Table* will send traffic to the peering connection.
10. In the **Routes** tab, choose **Edit routes**, and then configure the following settings:
    - Choose **Add route**.
    - **Destination:** `10.5.0.0/16` (The setting is the Classless Inter-Domain Route, or CIDR, block range of *Shared VPC*.)
    - **Target:** Choose **Peering Connection** from the dropdown list, and then from the search bar, choose *Lab-Peer*.
    - Choose **Save changes**.

      You will now configure the reverse flow for traffic that comes from *Shared VPC* and goes to *Lab VPC*.
11. Go back to Route tables and select **Shared-VPC Route Table**. If the check boxes for any other route tables are selected, clear them.

    This route table is for *Shared VPC*. You will now configure it to send traffic to the peering connection if the destination IP address falls in the range of *Lab VPC*.
12. In the **Routes** tab, choose **Edit routes**, and then configure the following settings:
    - Choose **Add route**.
    - **Destination:** `10.0.0.0/16` (This setting is the CIDR block range of *Lab VPC*.)
    - **Target:** Choose *Peering Connection* from the dropdown list, and then from the search bar, choose *Lab-Peer*.
    - Choose **Save changes**.

The route tables are now configured to send traffic through the peering connection when the traffic is destined for the other VPC.

Task 3: Enabling VPC Flow Logs to provide insight on the data moving across the network

Now that the peering connection is established between the two VPCs, you setup VPC Flow Logs to monitor the network traffic moving between two networks. In this lab, you setup VPC Flow Logs to monitor the traffic on the VPC hosting database.

13. In the left navigation pane, choose **Your VPCs**, and then select **Shared VPC**.
14. From the bottom panel, choose the **Flow logs** tab.
15. Choose **Create flow log**.
16. On the **Create flow log** page, configure the following settings:
    - Name (optional): `SharedVPCLogs`
    - Maximum aggregation interval: **1 minute**.
    - Destination: **Send to CloudWatch Logs**.
    - Destination log group: Enter `ShareVPCFlowLogs` to create a new log group with the same name.
    - IAM Role: Choose **vpc-flow-logs-Role**.
    - Choose **Create flow log**.

      An alert is displayed at the top indicating the flow log was created for **Shared VPC**.
17. From the bottom pane, choose the **Flow logs** tab and notice that **SharedVPCLogs** was created.
18. Below Destination name, choose the hyperlink **ShareVPCFlowLogs** to display the CloudWatch log group that was created.

    Note: Refresh after few minutes if you get a message that says *Log group does not exist*.

    Keep this window open.

Task 4: Testing the VPC peering connection

Now that you configured VPC peering, you will test the VPC peering connection. You will perform the test by configuring the inventory application to access the database across the peering connection.

19. From the top of this guide, choose **AWS Details**.
20. Copy the value for *EC2PublicIP* and paste it into a new web browser tab.

    You should see the inventory application and the following message: *Please configure Settings to connect to database*.
21. Choose **Settings**, and configure the following settings:

- **Endpoint:** Paste the database endpoint. To find this endpoint, choose *AWS Details* on the lab instructions page. Then, copy the *Endpoint*.
- **Database:** `inventory`
- **Username:** `admin`
- **Password:** `lab-password`
- Choose Save.

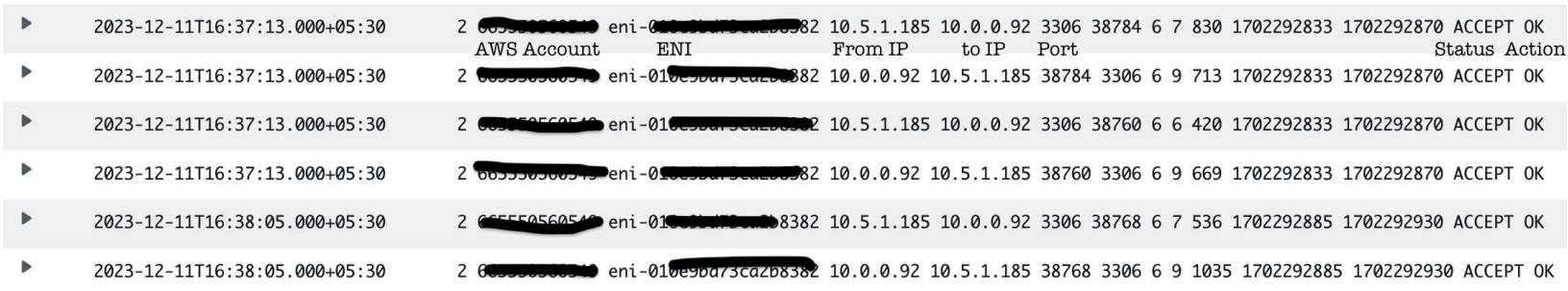
The application should now show data from the database.

This step confirms that the VPC peering connection was established because *Shared VPC* does not have an internet gateway. The only way to access the database is through the VPC peering connection.

Task 5: Analyzing the VPC flow logs

In this task, you analyze the VPC flow logs to understand the traffic between application and database in peered VPCs.

22. Go to the browser tab or window that's displaying **ShareVPCFlowLogs**.
23. Select the **Log stream eni-****.
24. After few minutes, the network traffic starts showing up.
25. Notice the traffic pattern from the logs, which looks similar to the following:



You might notice lot of lines in the log, but you analyze few lines with port 3306; this represents the traffic between the application server and database.

Explanation:

AWS Accountid	**Shows the account in which the VPC is hosted (Hidden here)**	
Network interface	eni-* - The elastic network interface ID on which network traffic is recorded.	
From IP	10.5.1.185 - The private IP of the database instance.	
To IP	10.0.0.02 - The private IP of the destination, which is the EC2 instance running the application.	
Port	3306 - RDS database port	
action	Accept/Reject	
status	OK	

Note: There are many fields in the record, and the table is describing the important ones.

Conclusion

Congratulations! You successfully completed the following:

- Created a VPC peering connection
- Configured route tables to use the VPC peering connection
- Enabled VPC Flow Logs to provide insight on the data moving across the network
- Tested a peering connection
- Analyzed the VPC flow logs

Submitting your work

26. At the top of these instructions, choose Submit to record your progress and when prompted, choose **Yes**.
27. If the results don't display after a couple of minutes, return to the top of these instructions and choose Grades.

Tip: You can submit your work multiple times. After you change your work, choose **Submit** again. Your last submission is what's recorded for this lab.

28. To find detailed feedback on your work, choose Details followed by **View Submission Report**.

Lab complete

Congratulations! You have completed the lab.

29. At the top of this page, choose **End Lab** and then choose Yes to confirm that you want to end the lab.

    The message "Ended AWS Lab Successfully" is briefly displayed, indicating that the lab has ended.