181-[JAWS]-Activity - Troubleshoot a VPC

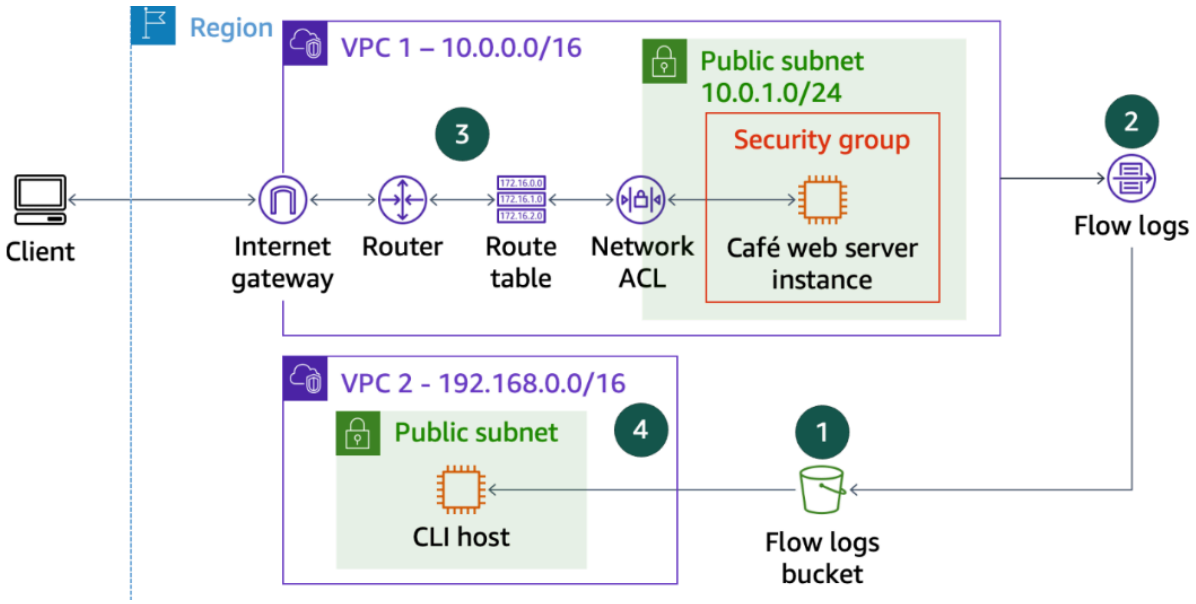
Troubleshooting a VPC

Lab overview

In this lab, you troubleshoot virtual private cloud (VPC) configurations and analyze VPC Flow Logs.

You begin with an environment that includes two VPCs, Amazon Elastic Compute Cloud (Amazon EC2) instances, and other networking components shown in the following diagram.

This diagram also shows four numbered circles (#1–4) that indicate the order in which you work through this lab.



Your tasks include the following:

1. Creating an Amazon Simple Storage Service (Amazon S3) bucket to hold VPC Flow Log data
2. Creating a flow log to capture all IP traffic passing through network interfaces in the VPC
3. Troubleshooting the VPC configuration issues to allow access to the resources
4. Downloading and analyzing the flow log data

Objectives

By the end of this lab, you will be able to do the following:

- Create VPC Flow Logs.
- Troubleshoot VPC configuration issues.
- Analyze flow logs.

Duration

This lab requires approximately **75 minutes** to complete.

Accessing the AWS Management Console

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

   A **Start Lab** panel opens displaying the lab status.
2. Wait until the message "Lab status: ready" appears, and then choose **X** to close the **Start Lab** panel.

   Note: It takes approximately **10** minutes for the lab to be ready for use.
3. After the Lab is ready, at the top of these instructions, choose Details, and then choose Show.
4. From the **Credentials** panel, copy the values from the table, and paste them into a text editor. You use these values throughout the lab.
5. At the top of these instructions, choose AWS to open the AWS Management Console on 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**.
6. Arrange the AWS Management Console so that it appears alongside these instructions. Ideally, you should be able to see both browser tabs at the same time to follow the lab steps.

   Leave this browser tab open. You return to it later in this lab.

Task 1: Connecting to the CLI Host instance

In this task, you use EC2 Instance Connect to connect to the CLI Host instance. You use this instance to run AWS Command Line Interface (AWS CLI) commands.

7. On the **AWS Management Console**, in the **Search** bar, enter and choose `EC2` to open the **EC2 Management Console**.
8. In the navigation pane, choose **Instances**.
9. From the list of instances, select the **CLI Host** instance.
10. Choose **Connect**.
11. On the **EC2 Instance Connect** tab, choose **Connect**.

    This option opens a new browser tab that shows the EC2 Instance Connect terminal window.

    Note: If you prefer to use an SSH client to connect to the EC2 instance, see the guidance to Connect to Your Linux Instance.

    You use this terminal window to complete the tasks throughout the lab. If the terminal becomes unresponsive, refresh the browser or use the steps in this task to connect again.

    Now that you are connected to the CLI Host instance, you can configure and use the AWS CLI to call AWS services.

Task 1.1: Configuring the AWS CLI on the CLI Host instance

12. To configure the AWS CLI profile with credentials, in the EC2 Instance Connect terminal, run the following command:

```
aws configure
```

13. At the prompts, copy the following values that you pasted into your text editor, and paste them into the terminal window as directed.
    - **AWS Access Key ID:** Enter the value for **AccessKey**.
    - **AWS Secret Access Key:** Enter the value for **SecretKey**.
    - **Default region name:** Enter `us-west-2`.
    - **Default output format:** Enter `json`.

You run CLI commands on this CLI Host terminal window as instructed throughout the lab.

Task 2: Creating VPC Flow Logs

In this task, you create an S3 bucket to publish data from VPC Flow Logs. Then you create VPC Flow Logs on VPC1 to capture information about IP traffic between network interfaces in VPC1. The flow logs are then published to the S3 bucket.

14. To create the S3 bucket where the flow logs will be published, run the following command. In the command, replace ###### with six random numbers:

    ```
    aws s3api create-bucket --bucket flowlog###### --region 'us-west-2' --create-bucket-configuration
    LocationConstraint='us-west-2'
    ```

    The JSON-formatted output similar to the following shows a bucket location: http://flowlog######.s3.amazonaws.com

    In this command, **flowlog######** is your bucket name. You use this bucket name in a later step.

    Note: If you receive an error message indicating that *Bucket name already exists*, use different set of random numbers to replace **######** and run the command again.

15. To get the VPC ID for VPC1 to create VPC Flow Logs, run the following command:

    ```
    aws ec2 describe-vpcs --query 'Vpcs[*].[VpcId,Tags[?Key==`Name`].Value,CidrBlock]' --filters
    "Name=tag:Name,Values='VPC1'"
    ```

    The JSON-formatted output similar to the following shows the VPC ID: vpc-01edacbe1c31959d2

16. To create VPC Flow Logs on VPC1, run the following command. In the command, replace *<flowlog######>* with the bucket name from the previous steps, and replace *<vpc-id>* with the VPC ID for VPC1 from the previous step.

    The VPC ID was returned by the describe-vpcs command that you ran. You can also find it in the list of values (**VPC1ID**) that you copied to a text editor at the beginning of the lab.

    ```
    aws ec2 create-flow-logs --resource-type VPC --resource-ids <vpc-id> --traffic-type ALL --log-
    destination-type s3 --log-destination arn:aws:s3:::<flowlog######>
    ```

    The command output returns **FlowLogIds** and a **ClientToken**.

    Note: If you see an "Unsuccessful" message, ignore it.

17. To confirm that the flow log was created, run the following command:

    ```
    aws ec2 describe-flow-logs
    ```

    The command output should show that a single flow log was created with a **FlowLogStatus** of *ACTIVE* and a log destination that points to your S3 bucket.

    Now that the flow log has been created, you can continue to the next task, which involves some troubleshooting.

Task 3: Troubleshooting VPC configuration issues to allow access to resources

In this task, you analyze access to the web server instance and troubleshoot some networking issues. Recall that the cafe web server instance runs in the public subnet in VPC1. Refer to the diagram at the start of this lab to see details about how the network should be configured.

18. From your text editor, copy the **WebServerIP** IP address, and paste it into a new browser tab.
19. After a few moments, the page fails to load, and you receive a message indicating that the site can't be reached or the connection has timed out. This message is expected.

    Leave this browser tab open so that you can return to it later.
20. In the CLI Host terminal, to find details about the web server instance, run the following command. In the command, replace <*WebServerIP*> with the **WebServerIP** address that you used in the previous steps:

    ```
    aws ec2 describe-instances --filter "Name=ip-address,Values='<WebServerIP>'"
    ```

    A large JSON document is returned that provides more details than you need for your troubleshooting.

    To return only relevant details, you filter the results on the client side by using the query parameter. The command in the next step returns only the state of the instance, the private IP address, the instance ID, the security groups that are applied to it, the subnet in which it runs, and the key pair name that is associated with it.
21. To filter the results, run the following command. In the command, replace <*WebServerIP*> with the same **WebServerIP** address that you've used in the previous steps:

    ```
    aws ec2 describe-instances --filter "Name=ip-address,Values='<WebServerIP>'" --query
    'Reservations[*].Instances[*].[State,PrivateIpAddress,InstanceId,SecurityGroups,SubnetId,KeyName]'
    ```

    The command results indicate that the instance is running and return additional information that you can use later.

    Next, you try to establish an SSH connection to the web server instance by using EC2 Instance Connect.
22. In the browser tab with the **AWS Management Console**, in the **Search** bar, enter and choose `EC2` to open the **EC2 Management Console**.
23. In the navigation pane, choose **Instances**.
24. From the list of instances, select the **Cafe Web Server** instance.
25. Choose **Connect**.
26. On the **EC2 Instance Connect** tab, choose **Connect**.

    After a few seconds, the attempt to connect fails. You get an error on the browser window that says, "Failed to connect to your instance."

    This behavior is expected.

Troubleshooting challenge #1

You have established that the web server instance is running but the webpage is not loading. What could the issue be?

Challenge yourself to conduct your investigation by using only AWS CLI programmatic access. Avoid using the AWS Management Console.

Hints:

- Use the nmap utility to check which ports are open on the web server EC2 instance.
  - To do this, you must first install the utility on the CLI Host instance by running the `sudo yum install -y nmap` command.
  - Then run the `nmap <WebServerIP>` command. In this command, replace <*WebServerIP*> with the actual public IP address.

  If nmap cannot find any open ports, could there be something else blocking access to the instance?
- Check the security group details by using the `aws ec2 describe-security-groups` command.
  - You might find it helpful to analyze the results of the command if you use the group-ids parameter. This value is also available in the text editor (**WebServerSgId**) with the other values that you've used in this lab.

    You can use the following command to look up the connectivity to port 22:

    ▶ command
  - You can also use the `describe-instances` command to return the security group ID.
  - After you run the `describe-security-groups` command, analyze the resulting output.

    Do the security group settings that are applied to the web server EC2 instance look like they are allowing connectivity to port 22?

- Check the route table settings for the route table that is associated with the subnet where the web server is running.
  - Use the `aws ec2 describe-route-tables` command.
    - When you run this command, you might find it helpful to apply a filter like the following example: `--filter "Name=association.subnet-id,Values='<VPC1PubSubnetID>'"`
      - In this command, replace <*VPC1PubSubnetID*> with the actual subnet ID value from the text editor.
      - The subnet ID value was also returned when you ran the `describe-instances` command.

    You can use the following command:

    ▶ command
- When you analyze the output of the `describe-route-tables` command, recall that the subnet is labeled as public.

  Do you notice any issues with the routes?
  - If you must define a new route, use the `aws ec2 create-route` command.
    - You must know the **route-table-id** and **gateway-id** to successfully create a route. Both of these values are available in the text editor. You should also have the route-table-id from when you ran the `describe-route-tables` command earlier.

    You can use the following command to create routes as needed:

    ▶ command
  - You can also use the `aws ec2 describe-internet-gateways` command to get the gateway-id. You might also need to specify other parameters to run the command successfully.

After you think you have solved the issue, return to the browser tab where you tried to load the web server page, and refresh the webpage. The browser page should display a message that says, "Hello From Your Web Server!"

Congratulations! You have resolved the issue that prevented you from accessing the website. However, another issue remains, and you investigate this issue in the next section.

Troubleshooting challenge #2

Now that you resolved the web access issue, try connecting to the web server instance using EC2 instance Connect.

This attempt also fails. An error similar to the message that you received earlier displays on the browser. Again, this behavior is expected.

What could be the remaining issue?

You already verified that the web server is running. You successfully created a route table entry to connect the subnet where the web server instance is running to the internet. You also verified that the security group allows connections on port 22, which is the default SSH port.

Hints:

- On the CLI Host instance terminal, check the network access control list (network ACL) settings for the network ACL that is associated with the subnet where the instance is running.
- To do this, run the following command. In the command, replace *<VPC1PublicSubnetID>* with the subnet ID from the text editor:

```
aws ec2 describe-network-acls --filter "Name=association.subnet-id,Values='VPC1PublicSubnetID'" --query
'NetworkAcls[*].[NetworkAclId,Entries]'
```

- Analyze the output that results from running the command. Do any of the entries look like they might be causing the issue?
- To delete any network ACL entries that might be causing an issue, use the `delete-network-acl-entry` command. Note the network acl-id retrieved by the previous command.

  You can use following command to delete the rule:

  ▶ command

  After you think you have solved the issue, try connecting to the web server instance using EC2 Instance Connect again and confirm that you can connect. If you can connect, you have successfully resolved the issue. To confirm that you are connected to the correct EC2 instance, run the `hostname` command after you are connected. It should indicate **web-server** as the hostname.

Congratulations! You have resolved the SSH access issue that prevented you from connecting to the web server.

Task 4: Analyzing flow logs

You have resolved the network issues. While doing so, you created some useful entries in the flow logs that you created when you created VPC Flow Logs at the beginning of this lab.

In this final task, you query the flow logs to observe the activities that they capture.

Task 4.1: Downloading and extracting flow logs

27. In the CLI Host terminal window, to create a local directory where you can download the flow log files, run the following command:

    `mkdir flowlogs`

28. To change the directory to the new directory, run the following command:

    `cd flowlogs`

29. To list the S3 buckets to recall the bucket name, run the following command:

    `aws s3 ls`

30. To download the flow logs, run the following command. In the command, replace *<flowlog######>* with the bucket name that you used earlier in the lab:

    `aws s3 cp s3://<flowlog######>/ . --recursive`

    If the command is successful, you should see that many files are downloaded to a subdirectory similar to the following: AWSLogs/AccountID/vpcflowlogs/us-west-2/yyyy/mm/dd/

    Next, you move down the folder structure to the subdirectory where you downloaded the files.

31. To reach the required subdirectory, run following cd command. In the command, replace *<AWSLogs/AccountID/vpcflowlogs/us-west-2/yyyy/mm/dd/>* with the subdirectory from the output of the previous command:

    `cd <AWSLogs/AccountID/vpcflowlogs/us-west-2/yyyy/mm/dd/>`

    Tip: You can also use the cd command and repeatedly press the Tab key to reach the required subdirectory.

32. To see all the downloaded log files, run the `ls` command. The logs are located in an AWSLogs/<AccountID>/vpcflowlogs/<region>/yyyy/mm/dd subdirectory.

    The file names all end in log.gz, which indicates that they are compressed as GNU .zip files.

33. To extract the logs, run the following command:

    `gunzip *.gz`

34. Run the `ls` command again.

    All the files are now extracted.

Task 4.2: Analyzing the logs

In this section, you analyze the flow logs to check if your failed SSH connection attempts were captured in the logs.

First, you analyze the structure of the logs.

35. Copy one of the file names that were returned by the ls command that you ran in the previous steps.
36. In the terminal window, run the following command. In the command, replace *<file name>* with the file name that you copied in the previous step.

    ```
    head <file name>
    ```

    The header row indicates the kind of data that each log entry contains. Each entry contains information, such as the IP address of the source of the event (in the fourth column), the destination port (seventh column), start and end timestamps (in Unix timestamp format), and the action that resulted (ACCEPT or REJECT).
37. To search each log file in the current directory and return lines that contain the word **REJECT**, run the following command:

    ```
    grep -rn REJECT .
    ```

    This command should return a large dataset because it includes every event where the VPC settings rejected the request.
38. To find out how many records were returned, run the following command:

    ```
    grep -rn REJECT . | wc -l
    ```

    The results show the number of lines in your result set.
39. To refine your search by looking for only lines that contain 22 (which is the port number where you attempted to connect to the web server when access was blocked), run the following command:

    ```
    grep -rn 22 . | grep REJECT
    ```

    This command should return a smaller number of results.

    To isolate the result set so that it displays only the log entries that correspond to the failed SSH connection attempts that you made, you must filter the results further.

    Recall that your failed attempts to use SSH to connect the web server were initiated from your local machine. In the next step, you determine the IP address by which your local machine is addressable from the internet.
40. On the AWS Management Console, go to the Amazon EC2 service in the same Region where your EC2 instances are running.
41. Choose **Security Groups**.
42. Choose the link for **WebSecurityGroup**, and then choose the **Inbound rules** tab.
43. Choose **Edit inbound rules**, and then choose **Add Rule**.
44. In the third row that you just created, for **Source**, choose **My IP**.
45. Copy the IP address from the Classless Inter-Domain Routing (CIDR) block that is automatically populated (it will end in /32), and paste it into a text editor. Copy only the IP address, not the /32 suffix.
46. Choose **Cancel**.

    You do not need to modify any security groups in this account. The purpose of this step is to capture this IP address.
47. In the CLI Host terminal session, run the following refined query on the flow logs. In the following command, replace *<ip-address>* with the IP address from the CIDR block that you copied in the previous steps:

    ```
    grep -rn 22 . | grep REJECT | grep <ip-address>
    ```

    The number of lines in the result set should now match the number of times you tried and failed to use SSH to connect the web server instance.

Notice that the elastic network interface ID is in each of the log entries that were returned by your query.

48. To confirm that the network interface ID that is recorded in the flow log matches the network interface that is assigned to the web server instance (as part of the network interface), run the following command. In the command, replace *<WebServerIP>* with the IP address from text editor:

```
aws ec2 describe-network-interfaces --filters "Name=association.public-ip,Values='<WebServerIP>'" --query
'NetworkInterfaces[*].[NetworkInterfaceId,Association.PublicIp]'
```

Next, you translate the timestamps into a human-readable format.

Notice the two long numbers that appear toward the end of each log entry before the REJECT term.

These numbers are Unix-formatted timestamps. The first timestamp indicates the start time of each event that was captured. The second timestamp indicates the end time. You can convert them into a human-readable format by using the Linux date command line utility. For example, if the timestamp is **1554496931**, then you would run the following command:

```
date -d @1554496931
```

49. To translate one of the timestamps into a human-readable format, run the `date -d @` command for one of the captured timestamps from one of the filtered REJECT results. It should indicate a time from today that corresponds to when you were working through this lab.
50. To compare the result to the current time, run the following command:

```
date
```

Using grep is a powerful but basic way to pull meaningful data out of VPC Flow Log files. The market offers many tools for running reports or generating analytic dashboards from logs. One solution is to use the Amazon Athena service. You can use Athena to ingest logs so that they become data in a database table. You can then run SQL queries to extract meaningful information from the logs. For more information about Athena, see Querying Amazon VPC Flow Logs.

Conclusion

Congratulations! You now have successfully done the following:

- Created VPC Flow Logs
- Troubleshot VPC configuration issues
- Analyzed flow logs

Lab complete

Congratulations! You have completed the lab.

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

    A panel appears indicating that "You may close this message box now. Lab resources are terminating."
52. To close the **End Lab** panel, choose the **X** in the upper-right corner.