Controlling the Network

**SPL-TF-200-SISBPN-10-EN - Version 1.0.4**

© 2023 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. All trademarks are the property of their owners.

Note: Do not include any personal, identifying, or confidential information into the lab environment. Information entered may be visible to others.

Corrections, feedback, or other questions? Contact us at [*AWS Training and Certification*](https://support.aws.amazon.com/#/contacts/aws-training).

**Overview**

You are a network security engineer at AnyCompany. You are responsible for creating a secure network infrastructure in AWS to prepare for AnyCompany’s upcoming migration to the cloud. AnyCompany currently has a three-tier network security infrastructure on-premises:

* The Public Access Zone hosts load balancers that serve as the primary connection point to your web servers.
* The Web Server Zone hosts the frontend servers for your website.
* The Database Zone hosts the backend database servers that provide data to your website.

You must ensure each zone is securely segmented from each other and only certain types of traffic are allowed to flow between them to support the company’s websites and applications.

In this lab, you use public and private subnets, security groups, and network ACLs to create a three-security zone network infrastructure. You then use VPC flow logs to monitor the traffic that reaches the resources in each zone to verify only the required traffic is allowed.

OBJECTIVES

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

* Create a three-security zone network infrastructure
* Implement network segmentation using security groups, network ACLs, and public and private subnets
* Monitor network traffic to EC2 instances using VPC flow logs

TECHNICAL KNOWLEDGE PREREQUISITES

To successfully complete this lab, you should be familiar with navigation of the AWS Management Console and have an understanding of basic networking concepts.

DURATION

This lab requires approximately *45* minutes to complete.

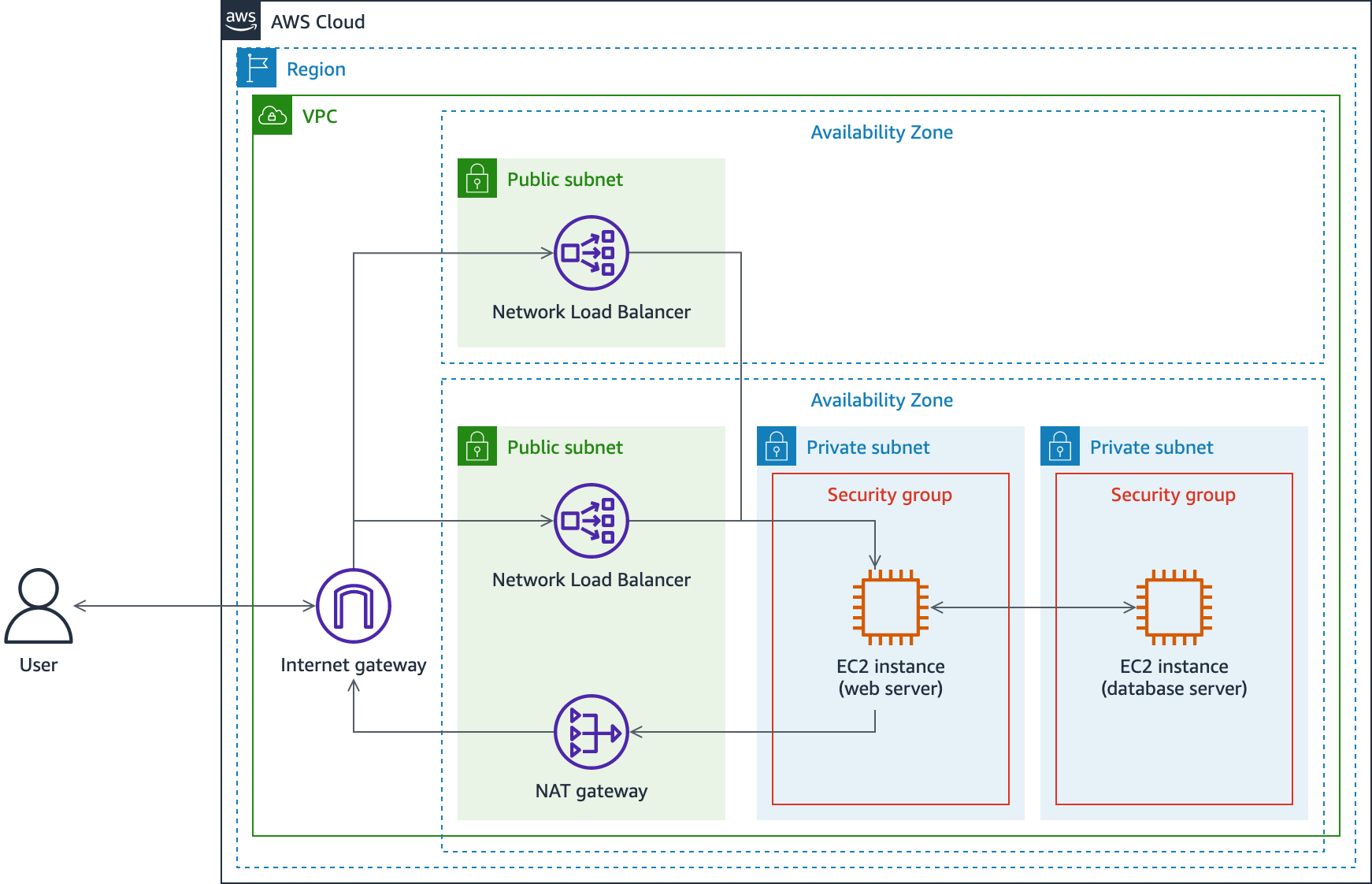
ICON KEY

Various icons are used throughout this lab to call attention to different types of instructions and notes. The following list explains the purpose for each icon:

* A sample output that you can use to verify the output of a command or edited file
* A hint, tip, or important guidance
* Where to find more information
* Information of special interest or importance (not so important to cause problems with the equipment or data if you miss it, but it could result in the need to repeat certain steps)
* A moment to pause to consider how you might apply a concept in your own environment or to initiate a conversation about the topic at hand
* An opportunity to check your knowledge and test what you have learned
* A hint to a question or challenge

ENVIRONMENT OVERVIEW

The following diagram shows the basic architecture of the lab environment:



The following list details the major resources in the diagram:

* A **VPC** with one **public subnet** and two **private subnets** in one Availability Zone, and one **public subnet** in a second Availability Zone.
* A **Network Load Balancer** with two nodes, one in each public subnet.
* An **EC2 instance** acting as a web server is the first private subnet.
* An **EC2 instance** acting as a database server in the second subnet.
* Two **security groups**, one for each instance based its purpose.

The network traffic flows from an external user, though an internet gateway to one of the two Network Load Balancer nodes, to the web server. If the URL of the WordPress blog site running on the web server is requested, traffic flows to the database server as well.

 Due to current restrictions in the lab platform with creating certificates for an Application Load Balancer, a Network Load Balancer is used instead. The load balancer acts as a pass-through for incoming traffic to the web server - it does not analyze or transform the traffic in any way. In a production environment you might consider using an Application Load Balancer to take advantage of its additional features and security measures, such as accepting and processing an HTTPS request from a client before it reaches the web server. For more information about the features of each type of Elastic Load Balancer, refer to *Elastic Load Balancing product comparisons* in the **Additional resources** section at the end of this lab.

**Start lab**

1. To launch the lab, at the top of the page, choose **Start lab**.

 You must wait for the provisioned AWS services to be ready before you can continue.

1. To open the lab, choose **Open Console**.

You are automatically signed in to the AWS Management Console in a new web browser tab.

**Do not change the Region unless instructed.**

COMMON SIGN-IN ERRORS

**Error: You must first sign out**



If you see the message, **You must first log out before logging into a different AWS account:**

* Choose the **click here** link.
* Close your **Amazon Web Services Sign In** web browser tab and return to your initial lab page.
* Choose **Open Console** again.

**Error: Choosing Start Lab has no effect**

In some cases, certain pop-up or script blocker web browser extensions might prevent the **Start Lab** button from working as intended. If you experience an issue starting the lab:

* Add the lab domain name to your pop-up or script blocker’s allow list or turn it off.
* Refresh the page and try again.

**Task 1: Restrict network traffic using security groups**

As part of your task of replicating your on-premises network zones in AWS, you must ensure that your hosts are properly secured. You decide to use security groups for each resource type to control the traffic that is allowed to flow between them.

In this task, you create new rules for existing security groups to address your traffic needs.

TASK 1.1: VERIFY CURRENT CONNECTIVITY STATE

First, verify that you are unable to access the web server.

1. Copy the **TestSiteUrl** value from the list to the left of these instructions. Paste the URL into a new web browser tab and press **Enter** to navigate to an Apache test page on the web server.

 The connection should time out after approximately 1 minute, with an error stating the page could not be reached.

Currently, there are no security group rules that allow traffic to pass from the load balancer to the web server, so the connection times out.

TASK 1.2: ALLOW HTTPS TRAFFIC TO THE WEB SERVER

1. Return to your web browser tab with the **AWS Management Console**.

If you no longer have the AWS Management Console tab open, follow the steps in the [Start Lab](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-200-SISBPN-1%3A1.0.4-63e7457a/en-US#StartLab) section to log into the AWS Management Console.

1. At the top of the page, in the unified search bar, search for and choose

EC2

.

1. In the navigation list at the left of the page, under **Network & Security**, choose **Security Groups**.
2. On the **Security Groups** page, select the **Web Server SG** security group.

 Verify *Web Server SG* is the only item selected.

1. In the **Details** pane at the bottom of the page, choose the **Inbound rules** tab.
2. At the right side of the **Inbound rules** section, choose **Edit inbound rules**.
3. On the **Edit inbound rules** page, choose **Add rule**.
4. In the row for the new rule:

* For **Type**, select **HTTPS**.
* For **Source**, select **Anywhere-IPv4**.
  + Notice **0.0.0.0/0** is automatically added as the source IP CIDR.
* For **Description**, enter

Allow HTTPS traffic from any source.

 The rule you just added allows inbound traffic on port 443 to any resource in the *WebServerSg* security group from any IPv4 source IP. Because this lab utilizes a Network Load Balancer to pass the traffic directly to the web server, you must allow HTTPS connections from any source to the web server.

1. At the lower-right of the page, choose **Save rules**.

TASK 1.3: VERIFY TRAFFIC FLOW TO THE WEB SERVER

Next, verify the rules you added to the security group allows you to access the web server.

1. Copy the **TestSiteUrl** value from the list to the left of these instructions. Paste the URL into a new web browser tab and press **Enter** to navigate to an Apache test page on the web server, or refresh the page if you still have the web browser tab open.

 It can take approximately 15 seconds for the page to load the first time. The web server uses a self-signed SSL certificate. If your web browser warns you of a potential security risk due to a self-signed certificate, choose to continue to the site.

 On MacOS Catalina and later, some Chromium-based web browsers, such as Google Chrome or Microsoft Edge, might not display a link to continue to the site, with a *NET::ERR\_CERT\_INVALID* error message. If you experience this situation, try using an alternative web browser, such as Mozilla Firefox.

 An Apache HTTP server test page should load.

1. Close the **Test page for the Apache HTTP server** web browser tab.

Now that you’ve verified you can access the web server over HTTPS, attempt to access the WordPress site.

1. Copy the **WordPressUrl** value from the list to the left of these instructions. Paste the URL into a new web browser tab and press **Enter** to navigate to the WordPress page.

 The connection should time out after approximately 1 minute, with a *Gateway Timeout* error on the page.

WordPress is configured so that the web page is hosted from one instance and its database is running on a different instance in a separate private subnet. Currently, there are no security group rules that allow traffic to pass from the web server EC2 instance to the database server EC2 instance, so the connection times out.

TASK 1.4: ALLOW TRAFFIC FROM THE WEB SERVER TO THE DATABASE SERVER

1. Return to your web browser tab with the **EC2 management console**.
2. On the **Security Groups** page, select the **Web Server SG** security group.
3. In the **Details** pane at the bottom of the page, choose the **Outbound rules** tab.
4. At the right side of the **Outbound rules** section, choose **Edit outbound rules**.
5. On the **Edit outbound rules** page, choose **Add rule**.
6. In the row for the new rule:

* For **Type**, select **MYSQL/Aurora**.
  + Notice **Protocol** and **Port range** are automatically set to TCP and 3306, respectively.
* For **Destination**, select **Custom**.
* In the search box to the right of the **Destination** parameter, search for and select

DatabaseServerSG

.

* For **Description**, enter

Allow MYSQL database traffic to resources in the DatabaseServerSG security group.

 The rule you just added allows outbound traffic on port 3306 from any resource in the *WebServerSg* security group to any resource in the *DatabaseServerSg* security group. For the purposes of this lab, it allows MYSQL traffic to flow from the web server EC2 instance to the database server EC2 instance.

1. At the lower-right of the page, choose **Save rules**.

Next, add an ingress (inbound) rule to the database server security group.

1. On the **Security Groups** page, select the **Database Server SG** security group.

 Verify *Database Server SG* is the only item selected.

1. In the **Details** pane at the bottom of the page, choose the **Inbound rules** tab.
2. At the right side of the **Inbound rules** section, choose **Edit inbound rules**.
3. On the **Edit inbound rules** page, choose **Add rule**.
4. In the row for the new rule:

* For **Type**, select **MYSQL/Aurora**.
* For **Source**, select **Custom**.
* In the search box to the right of the **Source** parameter, search for and select

WebServerSg

.

* For **Description**, enter

Allow MYSQL database traffic from resources in the WebServerSg security group.

 The rule you just added allows inbound traffic to any resource in the *DatabaseServerSg* security group from any resource in the *WebServerSg* security group. For the purposes of this lab, it allows MYSQL traffic to flow to the database server EC2 instance from the web server EC2 instance.

1. At the lower-right of the page, choose **Save rules**.

TASK 1.5: VERIFY TRAFFIC FLOW TO THE DATABASE SERVER AND CONFIGURE WORDPRESS

Now that you have added security group rules to allow traffic to flow to the database server, you should be able to load the WordPress page.

1. Copy the **WordPressUrl** value from the list to the left of these instructions. Paste the URL into a new web browser tab and press **Enter** to navigate to the WordPress page, or refresh the page if you still have the web browser tab open.

 The WordPress welcome page should load.

Next, finalize the WordPress configuration. You have decided to use a WordPress blog site as an easy way to replicate and test traffic flow to the application servers you plan to migrate from your on-premises environment. The WordPress server configuration uses a frontend web server in one subnet, with a database server is a separate subnet, similar to how your application servers are configured.

1. In the **Information needed** section:

* For **Site Title**, enter

AWS Security Best Practices

.

* For **Username**, enter

wpadmin

.

* For **Password**, copy and paste the **AdministratorPassword** value listed to the left of these instructions.
* For **Your Email**, enter

wpadmin@example.corp

.

1. At the bottom of the page, choose **Install WordPress**.
2. On the **Success!** page, choose **Log In**.
3. On the WordPress login page:

* For **Username or Email Address**, enter

wpadmin

.

* For **Password**, copy and paste the **AdministratorPassword** value listed to the left of these instructions.

1. Choose **Log In**.

 It can take approximately 1-2 minutes for the WordPress dashboard to load the first time.

You should now see the WordPress dashboard.

1. Close your web browser tab with the **WordPress dashboard**.

 Congratulations! You have successfully added security group rules to allow only the appropriate traffic to connect to the WordPress site hosted in your private subnet.

**Task 2: Restrict traffic to the public subnet**

Now that you have secured traffic to and from your instances, you would like to further harden the network security posture.

In this task, you modify the network access control lists (ACLs) for each subnet to allow only the traffic required for access to the website to pass through at the subnet level.

TASK 2.1: CREATE NETWORK ACL RULES FOR THE LOAD BALANCER SUBNETS

1. Return to your web browser tab with the **EC2 management console**.
2. At the top of the page, in the unified search bar, search for and choose

VPC

.

1. In the navigation list at the left of the page, under **SECURITY**, choose **Network ACLs**.
2. At the upper-right corner of the **Network ACLs** page, choose **Create network ACL**.
3. On the **Create network ACL** page:

* In the **Network ACL settings** section, for **Name**, enter

load-balancer-nacl

.

* For **VPC**, select **Lab VPC**.

1. Choose **Create network ACL**.
2. On the **Network ACLs** page, select **load-balancer-nacl**.
3. In the **Details** pane at the bottom of the page, choose the **Inbound rules** tab.

 Notice there is only one inbound rule that denies all incoming traffic.

1. Choose the **Outbound rules** tab.

 Notice there is only one outbound rule that denies all incoming traffic.

1. Choose the **Subnet associations** tab.
2. Choose **Edit subnet associations**.
3. On the **Edit subnet associations** page:

* Select **Load Balancer Subnet 1 (Public)**.
* Select **Load Balancer Subnet 2 (Public)**.

1. Choose **Save changes**.
2. Copy the **TestSiteUrl** value from the list to the left of these instructions. Paste the URL into a new web browser tab and press **Enter** to navigate to an Apache test page on the web server.

 The connection should time out after approximately 30 seconds, with an error stating the page could not be reached.

1. Return to your web browser tab with the **VPC management console**.
2. On the **Network ACLs** page, select **load-balancer-nacl**.
3. In the **Details** pane at the bottom of the page, choose the **Inbound rules** tab.
4. Choose **Edit inbound rules**.
5. On the **Edit inbound rules** page, choose **Add new rule**.
6. In the row for the new rule:

* For **Rule number**, enter

100

.

* For **Type**, select **HTTPS**.
  + Notice the **Protocol** (TCP) and **Port range** (443) fields are filled automatically.
* For **Source**, enter

0.0.0.0/0

.

* For **Allow/Deny**, select **Allow**.

1. Create a second rule with the following configuration:

* For **Rule number**, enter

101

.

* For **Type**, select **Custom TCP**.
* For **Port range**, enter

1024-65535

.

* For **Source**, enter

10.10.0.0/16

.

* For **Allow/Deny**, select **Allow**.

 The rule you just created allows inbound connections on ephemeral ports from any resource within the lab VPC. Devices temporarily use ephemeral ports to initiate a connection with one another. After the initial connection is made, traffic is then allowed to propagate on the required port, such as HTTPS (port 443). For more information about, refer to *Ephemeral ports* in the **Additional resources** section.

1. Choose **Save changes**.
2. On the **Network ACLs** page, select **load-balancer-nacl** if it is not already selected.
3. In the **Details** pane at the bottom of the page, choose the **Outbound rules** tab.
4. Choose **Edit outbound rules**.
5. On the **Edit outbound rules** page, choose **Add new rule**.
6. In the row for the new rule:

* For **Rule number**, enter

100

.

* For **Type**, select **HTTPS**.
* For **Destination**, enter the value of **WebServerSubnet** listed to the left of these instructions.
* For **Allow/Deny**, select **Allow**.

1. Create a second rule with the following configuration:

* For **Rule number**, enter

101

.

* For **Type**, select **Custom TCP**.
* For **Port range**, enter

1024-65535

.

* For **Destination**, enter

0.0.0.0/0

.

* For **Allow/Deny**, select **Allow**.

 For more information about the recommended traffic rules for a load balancer, refer to *Network ACLs for load balancers in a VPC* in the **Additional resources** section at the end of this lab.

1. Choose **Save changes**.

TASK 2.2: CREATE NETWORK ACL RULES FOR THE WEB SERVER SUBNET

 Based on the steps you just followed, create a new network ACL for the web server subnet with the following configuration:

* **Name:**

web-server-nacl

.

* **VPC:** Lab VPC
* Inbound rules:
  + Allow **HTTPS** (port 443) traffic from any source,

0.0.0.0/0

.

* + Allow TCP traffic on ports

1024-65535

 from the entire VPC subnet,

10.10.0.0/16

.

* Outbound rules:
  + Allow TCP traffic on ports

1024-65535

 to any destination,

0.0.0.0/0

.

* + Allow **MYSQL/Aurora** (port 3306) traffic to the database server subnet, which is listed to the left of these instructions as **DatabaseServerSubnet**.
* **Subnet associations**: Web Server Subnet (Private)

 If you get stuck, refer to [Task 2.2 solution](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-200-SISBPN-1%3A1.0.4-63e7457a/en-US#Task2.2Solution) in the **Answer key** section at the end of this lab.

TASK 2.3: VERIFY TRAFFIC FLOW TO THE WEB SERVER

Next, verify the network ACL rules you added allow you to access the web server.

1. Copy the **TestSiteUrl** value from the list to the left of these instructions. Paste the URL into a new web browser tab and press **Enter** to navigate to an Apache test page on the web server, or refresh the page if you still have the web browser tab open.

 An Apache HTTP server test page should load.

1. Close the **Test page for the Apache HTTP server** web browser tab.

TASK 2.4: CREATE NETWORK ACL RULES FOR THE DATABASE SERVER SUBNET

Finally, create network ACL rules to allow only the required traffic to the database server subnet.

 Based on the steps you just followed, create a new network ACL for the web server subnet with the following configuration:

* **Name:**

database-server-nacl

.

* **VPC:** Lab VPC
* Inbound rules:
  + Allow **MYSQL/Aurora** (port 3306) traffic from the web server subnet, which is listed to the left of these instructions as **WebServerSubnet**.
  + Allow TCP traffic on ports

1024-65535

 from the entire VPC subnet,

10.10.0.0/16

.

* Outbound rules:
  + Allow TCP traffic on ports

1024-65535

 to the entire VPC subnet,

10.10.0.0/16

.

* **Subnet associations**: Database Server Subnet (Private)

 If you get stuck, refer to [Task 2.4 solution](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-200-SISBPN-1%3A1.0.4-63e7457a/en-US#Task2.4Solution) in the **Answer key** section at the end of this lab.

TASK 2.5: VERIFY TRAFFIC FLOW TO THE DATABASE SERVER

Next, verify the network ACL rules you added allow you to access the WordPress blog site.

1. Copy the **WordPressUrl** value from the list to the left of these instructions. Paste the URL into a new web browser tab and press **Enter** to navigate to the WordPress blog site, or refresh the page if you still have the web browser tab open.

 The WordPress blog page should load.

1. Close the **WordPress blog** web browser tab.

 Congratulations! You have successfully created network ACL rules to limit inbound and outbound traffic to only what is required to successfully access your blog site.

**Task 3: Inspect network traffic with VPC flow logs**

You have now configured security groups and network ACLs to limit allowed network traffic to only what is absolutely necessary for your blog site to function properly. Next, you would like to verify that non-essential traffic is being denied successfully.

In this task, you create VPC flow logs to send network traffic data to Amazon CloudWatch. You then analyze the logs to confirm which types of traffic are allowed, and which are rejected.

TASK 3.1: CREATE THE CLOUDWATCH LOG GROUP

First, you must create the CloudWatch log group to send the VPC flow logs to.

1. At the top of the page, in the unified search bar, search for and choose

CloudWatch

.

1. In the navigation pane at the left of the page, in the **Logs** section, choose **Log groups**.
2. At the upper-right of the **Log groups** page, choose **Create log group**.
3. On the **Create log group** page:

* In the **Log group details** section, for **Log group name**, enter

blog-access-logs

.

* For **Retention setting**, choose **1 day**.

 For the purposes of this lab, you don’t need to retain the logs beyond the duration of the lab. However, in a production environment, select the retention time based on your organization’s policy or regulation requirements.

1. Choose **Create**.

TASK 3.2: CREATE A VPC FLOW LOG

Now that you have the CloudWatch log group, you can create and configure a VPC flow log to record traffic flow within the VPC.

1. At the top of the page, in the unified search bar, search for and choose

VPC

.

1. In the navigation pane at the left of the page, in the **VIRTUAL PRIVATE CLOUD** section, choose **Your VPCs**.
2. On the **Your VPCs** page, select **Lab VPC**.
3. In the **Details** pane at the bottom of the page, choose the **Flow logs** tab.
4. At the right of the **Flow logs** section, choose **Create flow log**.
5. On the **Create flow log** page:

* In the **Flow log settings** section, for **Name**, enter

vpc-flow-logs-for-blog

.

* For **Filter**, select **All**.
* For **Maximum aggregation interval**, select **1 minute**.
  + You select 1 minute in this lab to shorten the time before you can see the resulting logs. For more information, refer to *Aggregation interval* in the **Additional resources** section at the end of this lab.
* For **Destination**, select **Send to CloudWatch Logs**.
* For **Destination log group**, select **blog-access-logs**.
* For **IAM role**, select **VpcFlowLogsRole**.
* For **Log record format**, select **Custom format**.
* On the **Log format** drop-down menu, select the following options in this order:
  + account-id
  + interface-id
  + srcaddr
  + srcport
  + dstaddr
  + dstport
  + subnet-id
  + flow-direction
  + action

 The **Format preview** box should look like this:

${account-id} ${interface-id} ${srcaddr} ${srcport} ${dstaddr} ${dstport} ${subnet-id} ${flow-direction} ${action}

1. Choose **Create flow log**.

 The *VpcFlowLogsRole* IAM role that you selected while configuring the flow logs contains the following policy:

{

"Version": "2012-10-17",

"Statement": [

{

"Action": [

"logs:CreateLogGroup",

"logs:CreateLogStream",

"logs:PutLogEvents",

"logs:DescribeLogGroups",

"logs:DescribeLogStreams"

],

"Resource": "\*",

"Effect": "Allow"

}

]

}

The policy allows the VPC Flow Logs service to create CloudWatch log groups and log streams, and write events to the log streams.

 For more information about VPC Flow Logs, refer to *Logging IP traffic with VPC Flow Logs* and *Publish flow logs to CloudWatch Logs* in the **Additional resources** section at the end of this lab.

TASK 3.3: GENERATE TRAFFIC TO THE BLOG

Next, visit the WordPress blog site to generate traffic to the web and database servers.

1. Copy the **WordPressUrl** value from the list to the left of these instructions. Paste the URL into a new web browser tab and press **Enter** to navigate to the WordPress blog site, or refresh the page if you still have the web browser tab open.
2. Refresh the page three times to generate additional traffic.
3. Update the URL to **HTTP://** and then attempt to access the blog site again.

Because you have only allowed HTTPS traffic from the load balancer to the web server, the request should time out. When you view the flow logs, you should find rejection messages for traffic on port 80.

 If your web browser has *HTTPS-Only Mode* turned on, you might need to turn it off to successfully attempt to visit the site using HTTP.

1. Return to your web browser tab with the **VPC management console**.
2. At the top of the page, in the unified search bar, search for and choose

CloudWatch

.

1. In the navigation pane at the left of the page, in the **Logs** section, choose **Log groups**.
2. On the **Log groups** page, select the link for **blog-access-logs**.
3. On the **blog-access-logs** details page, in the **Log streams** section, choose **Search all**.

 The *Search all* option allows you to search through all of the log streams that are in the log group, rather than searching through them individually.

1. Review several of the log entries to determine what types of traffic were allowed and which were rejected. Use the right arrow  to expand each item to view the full message

If you followed the log format listed in task 3.1, the log messages should appear similar to this:

| **Account number** | **Interface ID** | **Source IP address** | **Source port** | **Destination IP address** | **Destination port** | **Subnet ID** | **Traffic direction** | **Action** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 111122223333 | eni-0cf0d60e2c2a9f674 | 10.10.1.52 | 2197 | 10.10.10.10 | 443 | subnet-0f2a1fafd992c336e | ingress | ACCEPT |
| 111122223333 | eni-0cf0d60e2c2a9f674 | 205.251.233.176 | 29803 | 10.10.1.52 | 80 | subnet-0f2a1fafd992c336e | ingress | REJECT |

 What types of traffic do you notice are accepted? Which are rejected? What do you think could be the cause of the traffic on ports other than 80, 443, or 3306?

 Congratulations! You have successfully created VPC Flow Logs to monitor the network traffic within your VPC.

**Conclusion**

 Congratulations! You now have successfully:

* Created a three-security zone network infrastructure
* Implemented network segmentation using security groups, network ACLs, and public and private subnets
* Monitored network traffic to EC2 instances using VPC flow logs

**End lab**

Follow these steps to close the console and end your lab.

1. Return to the **AWS Management Console**.
2. At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.
3. Choose **End lab** and then confirm that you want to end your lab.

**Additional resources**

* [Elastic Load Balancing product comparisons](https://aws.amazon.com/elasticloadbalancing/features/?nc=sn&loc=2&dn=1)
* [Ephemeral ports](https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html#nacl-ephemeral-ports)
* [Network ACLs for load balancers in a VPC](https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-security-groups.html#elb-vpc-nacl)
* [Aggregation interval](https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs.html#flow-logs-aggregration-interval)
* [Logging IP traffic with VPC Flow Logs](https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs.html)
* [Publish flow logs to CloudWatch Logs](https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs-cwl.html)

For more information about AWS Training and Certification, see [*https://aws.amazon.com/training/*](https://aws.amazon.com/training/).

*Your feedback is welcome and appreciated.*  
If you would like to share any feedback, suggestions, or corrections, please provide the details in our [*AWS Training and Certification Contact Form*](https://support.aws.amazon.com/#/contacts/aws-training).

**Answer key**

TASK 2.2 SOLUTION

1. Return to your web browser tab with the **EC2 management console**.
2. At the top of the page, in the unified search bar, search for and choose

VPC

.

1. In the navigation list at the left of the page, under **SECURITY**, choose **Network ACLs**.
2. At the upper-right corner of the **Network ACLs** page, choose **Create network ACL**.
3. On the **Create network ACL** page:

* In the **Network ACL settings** section, for **Name**, enter

web-server-nacl

.

* For **VPC**, select **Lab VPC**.

1. Choose **Create network ACL**.
2. On the **Network ACLs** page, select **web-server-nacl**.
3. In the **Details** pane at the bottom of the page, choose the **Subnet associations** tab.
4. Choose **Edit subnet associations**.
5. On the **Edit subnet associations** page:

* Select **Web Server Subnet (Private)**.

1. Choose **Save changes**.
2. Copy the **TestSiteUrl** value from the list to the left of these instructions. Paste the URL into a new web browser tab and press **Enter** to navigate to an Apache test page on the web server.

 The connection should time out after approximately 10-20 seconds, with an error stating the gateway timed out.

1. Return to your web browser tab with the **VPC management console**.
2. On the **Network ACLs** page, select **web-server-nacl**.
3. In the **Details** pane at the bottom of the page, choose the **Inbound rules** tab.
4. Choose **Edit inbound rules**.
5. On the **Edit inbound rules** page, choose **Add new rule**.
6. In the row for the new rule:

* For **Rule number**, enter

100

.

* For **Type**, select **HTTPS**.
* For **Source**, enter

0.0.0.0/0

.

* For **Allow/Deny**, select **Allow**.

1. Create a second rule with the following configuration:

* For **Rule number**, enter

101

.

* For **Type**, select **Custom TCP**.
* For **Port range**, enter

1024-65535

.

* For **Source**, enter

10.10.0.0/16

.

* For **Allow/Deny**, select **Allow**.

1. Choose **Save changes**.
2. On the **Network ACLs** page, select **web-server-nacl** if it is not already selected.
3. In the **Details** pane at the bottom of the page, choose the **Outbound rules** tab.
4. Choose **Edit outbound rules**.
5. On the **Edit outbound rules** page, choose **Add new rule**.
6. In the row for the new rule:

* For **Rule number**, enter

100

.

* For **Type**, select **MYSQL/Aurora**.
* For **Destination**, enter the value of **DatabaseServerSubnet** listed to the left of these instructions.
* For **Allow/Deny**, select **Allow**.

1. Create a second rule with the following configuration:

* For **Rule number**, enter

101

.

* For **Type**, select **Custom TCP**.
* For **Port range**, enter

1024-65535

.

* For **Destination**, enter

0.0.0.0/0

.

* For **Allow/Deny**, select **Allow**.

1. Choose **Save changes**.

[Return to Task 2.3](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-200-SISBPN-1%3A1.0.4-63e7457a/en-US#Task2.3)

TASK 2.4 SOLUTION

1. Return to your web browser tab with the **EC2 management console**.
2. At the top of the page, in the unified search bar, search for and choose

VPC

.

1. In the navigation list at the left of the page, under **SECURITY**, choose **Network ACLs**.
2. At the upper-right corner of the **Network ACLs** page, choose **Create network ACL**.
3. On the **Create network ACL** page:

* In the **Network ACL settings** section, for **Name**, enter

database-server-nacl

.

* For **VPC**, select **Lab VPC**.

1. Choose **Create network ACL**.
2. On the **Network ACLs** page, select **database-server-nacl**.
3. In the **Details** pane at the bottom of the page, choose the **Subnet associations** tab.
4. Choose **Edit subnet associations**.
5. On the **Edit subnet associations** page:

* Select **Database Server Subnet (Private)**.

1. Choose **Save changes**.
2. Copy the **WordPressUrl** value from the list to the left of these instructions. Paste the URL into a new web browser tab and press **Enter** to navigate to an Apache test page on the web server.

 The connection should time out after approximately 10-20 seconds, with an error stating the gateway timed out.

1. Return to your web browser tab with the **VPC management console**.
2. On the **Network ACLs** page, select **database-server-nacl**.
3. In the **Details** pane at the bottom of the page, choose the **Inbound rules** tab.
4. Choose **Edit inbound rules**.
5. On the **Edit inbound rules** page, choose **Add new rule**.
6. In the row for the new rule:

* For **Rule number**, enter

100

.

* For **Type**, select **MYSQL/Aurora**.
* For **Source**, enter the value of **WebServerSubnet** listed to the left of these instructions.
* For **Allow/Deny**, select **Allow**.

1. Create a second rule with the following configuration:

* For **Rule number**, enter

101

.

* For **Type**, select **Custom TCP**.
* For **Port range**, enter

1024-65535

.

* For **Source**, enter

10.10.0.0/16

.

* For **Allow/Deny**, select **Allow**.

1. Choose **Save changes**.
2. On the **Network ACLs** page, select **database-server-nacl** if it is not already selected.
3. In the **Details** pane at the bottom of the page, choose the **Outbound rules** tab.
4. Choose **Edit outbound rules**.
5. On the **Edit outbound rules** page, choose **Add new rule**.
6. In the row for the new rule:

* For **Rule number**, enter

100

.

* For **Type**, select **Custom TCP**.
* For **Port range**, enter

1024-65535

.

* For **Destination**, enter

10.10.0.0/16

.

* For **Allow/Deny**, select **Allow**.

1. Choose **Save changes**.

[Return to Task 2.5](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2FSPL-TF-200-SISBPN-1%3A1.0.4-63e7457a/en-US#Task2.5)