AWS Network Firewall Fundamentals

**SPL-TF-200-NWANFF-1 - Version 1.0.1**

© 2024 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).

**Lab overview**

You are a Network Security Engineer at AnyCompany. You are responsible for setting up and deploying network firewall security across Amazon Virtual Private Cloud (Amazon VPC) in your Organization. Your company is concerned about network security and would like to standardize network configurations.

Your task is to set up a baseline configuration of AWS Network Firewall to ensure traffic is inspected and filtered as necessary.

This lab is designed to provide a hands-on walkthrough of setting up an AWS environment that properly utilizes AWS Network Firewall. In this lab, you walk through each step to learn the fine grained details of what it takes to have the proper networking infrastructure to route and block traffic with AWS Network Firewall. You start with a simple, single VPC and single firewall and build from there. After completing this lab, you end up with one VPC and up to three subnets. Multiple EC2 instances are used during the course of this lab to help test network connectivity.

OBJECTIVES

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

* Review the basic network architecture required to support the Network Firewall scenarios.
* Configure AWS Network Firewall.
* Configure and test routing configuration using a basic subnet architecture.
* Configure and test routing configuration using a more sophisticated architecture involving a NAT Gateway.

DURATION

This lab requires approximately *75* 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:

* **Command:** A command that you must run.
* **Expected output:** A sample output that you can use to verify the output of a command or edited file.
* **Note:** A hint, tip, or important guidance.
* **Consider:** 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.
* **Knowledge check:** An opportunity to check your knowledge and test what you have learned.
* **Refresh:** A time when you might need to refresh a web browser page or list to show new information.
* **Task complete:** A conclusion or summary point in the 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.

LAB ENVIRONMENT

SERVICES USED IN THIS LAB

**AWS Network Firewall**

AWS Network Firewall can be configured as either stateful or stateless. It is a managed, network firewall and intrusion detection and prevention service for your virtual private cloud (VPC). Network Firewall enables you to filter traffic entering and exiting your VPC. This includes filtering traffic directed towards an internet gateway, NAT gateway, or even a VPN or AWS Direct Connect. Network Firewall uses the open source intrusion prevention system (IPS), Suricata, for stateful inspection.

**Learn more**

Refer to *What is AWS Network Firewall?* in the **Additional resources** section for more information.

**Amazon EC2**

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

**Learn more**

Refer to *What is Amazon EC2?* in the **Additional resources** section for more information.

**AWS Systems Manager Session Manager**

Session Manager is a fully managed AWS Systems Manager capability. With Session Manager, you can manage your Amazon Elastic Compute Cloud (Amazon EC2) instances, edge devices, on-premises servers, and virtual machines (VMs). You can use either an interactive one-click browser-based shell or the AWS Command Line Interface (AWS CLI). Session Manager provides secure and auditable node management without the need to open inbound ports, maintain bastion hosts, or manage SSH keys. Session Manager also allows you to comply with corporate policies that require controlled access to managed nodes, strict security practices, and fully auditable logs with node access details, while providing end users with simple one-click cross-platform access to your managed nodes.

**Learn more**

Refer to *AWS Systems Manager Session Manager* in the **Additional resources** section for more information.

**Amazon VPC**

With Amazon Virtual Private Cloud (Amazon VPC), you can launch AWS resources in a logically isolated virtual network that you’ve defined. This virtual network closely resembles a traditional network that you’d operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

**Learn more**

Refer to *What is Amazon VPC?* in the **Additional resources** section for more information.

AWS SERVICES NOT USED IN THIS LAB

AWS service capabilities used in this lab are limited to what the lab requires. Expect errors when accessing other services or performing actions beyond those provided in this lab guide.

**Task 1: Review the basic network architecture**

In this task, you review the basic network architecture of the lab and test the connectivity for two Amazon EC2 instances. This basic infrastructure is used to support the Network Firewall scenarios in the later tasks.

TASK 1.1: REVIEW THE LAB VPC

In this task, you review the VPC and its related resources.

1. At the top of the AWS Management Console, in the search bar, search for and choose

VPC

.

1. In the navigation pane at the left of the page, choose **Your VPCs**.

**Note:** On **Your VPCs** page, you see two VPCs that are already created for this lab.

1. In the **Lab-VPC** row, choose the **VPC ID** link to view its details.

On the **Lab-VPC** details page, you see additional details about the VPC, including but not limited to:

* VPC ID
* State
* IPv4 CIDR
* Tenancy

1. On the **Lab-VPC** details page, choose the **Resource map** tab.

The resource map shows relationships between resources inside a VPC and how traffic flows from subnets to NAT gateways, internet gateway and gateway endpoints.

**Learn more**

Refer to *Visualize the resources in your VPC* in the **Additional resources** section for more information.

1. Review the VPC resources, including the subnets, route tables and network connections.

TASK 1.2: REVIEW EC2 INSTANCES AND TEST CONNECTIVITY

In this task, you review the EC2 instances and test their connectivity.

1. At the top of the AWS Management Console, in the search bar, search for and choose

EC2

.

1. In the navigation pane, under **Instances**, choose **Instances**.

**Note:** On the **Instances** page, you see two EC2 instances that are already configured for this lab.

1. Select **Lab-PublicInstance**.
2. Choose **Connect**.
3. On the **Connect to instance** page, choose the **Session Manager** tab.
4. Choose **Connect**.

**Expected output:** After successful connection, you are redirected to a terminal session with a secured connection to the *Lab-PublicInstance* instance.

1. **Command:** To confirm the connection from the public EC2 instance to a remote host, run the following command:

curl www.example.com

**Expected output:** The output should display the HTML code for the website address you entered, similar to this:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

<!doctype html>

<html>

<head>

<title>Example Domain</title>

<meta charset="utf-8" />

<meta http-equiv="Content-type" content="text/html; charset=utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<style type="text/css">

body {

background-color: #f0f0f2;

margin: 0;

padding: 0;

font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;

}

div {

width: 600px;

margin: 5em auto;

padding: 2em;

background-color: #fdfdff;

border-radius: 0.5em;

box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);

}

a:link, a:visited {

color: #38488f;

text-decoration: none;

}

@media (max-width: 700px) {

div {

margin: 0 auto;

width: auto;

}

}

</style>

</head>

<body>

<div>

<h1>Example Domain</h1>

<p>This domain is for use in illustrative examples in documents. You may use this

domain in literature without prior coordination or asking for permission.</p>

<p><a href="https://www.iana.org/domains/example">More information...</a></p>

</div>

</body>

</html>

1. **Command:** To test connectivity from the public instance to the private instance, run the following command:

* Replace the **IP\_ADDRESS** placeholder value with the **PrivateIP1** value that is listed to the left of these instructions.

ping -c 5 IP\_ADDRESS

**Note:** *PrivateIP1* is the private IP of the *Lab-PrivateInstance* instance.

**Expected output:** The output should display the successful transmission of five data packets to the *Lab-PrivateInstance* instance, similar to this:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PING XX.X.X.XXX (XX.X.X.XXX) 56(84) bytes of data.

64 bytes from XX.X.X.XXX: icmp\_seq=1 ttl=255 time=0.403 ms

64 bytes from XX.X.X.XXX: icmp\_seq=2 ttl=255 time=0.442 ms

64 bytes from XX.X.X.XXX: icmp\_seq=3 ttl=255 time=0.462 ms

64 bytes from XX.X.X.XXX: icmp\_seq=4 ttl=255 time=5.02 ms

64 bytes from XX.X.X.XXX: icmp\_seq=5 ttl=255 time=0.410 ms

--- XX.X.X.XXX ping statistics ---

5 packets transmitted, 5 received, 0% packet loss, time 4074ms

rtt min/avg/max/mdev = 0.403/1.348/5.024/1.838 ms

1. Close your **AWS Systems Manager - Session Manager** web browser tab.
2. Return to your web browser tab with the **AWS Management Console**.
3. At the top of the AWS Management Console, in the search bar, search for and choose

EC2

.

1. In the navigation pane, choose **Instances**.
2. Select **Lab-PrivateInstance**.
3. Choose **Connect**.
4. On the **Connect to instance** page, choose the **Session Manager** tab.
5. Choose **Connect**.

**Expected output:** After successful connection, you are redirected to a terminal session with a secured connection to the *Lab-PrivateInstance* instance.

1. **Command:** To confirm the connection from the private EC2 instance to the public EC2 instance, run the following command:

* Replace the **IP\_ADDRESS** placeholder value with the **PrivateIP2** value that is listed to the left of these instructions.

curl IP\_ADDRESS

**Expected output:** The output should display a message, similar to this:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Hello from Public Instance!

1. **Command:** To test connectivity from the private instance to the public instance, run the following command:

* Replace the **IP\_ADDRESS** placeholder value with the **PrivateIP2** value that is listed to the left of these instructions.

ping -c 5 IP\_ADDRESS

**Note:** **PrivateIP2** is the private IP of **Lab-PublicInstance**.

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PING XX.X.X.X (XX.X.X.XX) 56(84) bytes of data.

64 bytes from XX.X.X.X: icmp\_seq=1 ttl=255 time=0.392 ms

64 bytes from XX.X.X.X: icmp\_seq=2 ttl=255 time=0.432 ms

64 bytes from XX.X.X.X: icmp\_seq=3 ttl=255 time=0.398 ms

64 bytes from XX.X.X.X: icmp\_seq=4 ttl=255 time=0.479 ms

64 bytes from XX.X.X.X: icmp\_seq=5 ttl=255 time=0.447 ms

--- XX.X.X.X ping statistics ---

5 packets transmitted, 5 received, 0% packet loss, time 4085ms

rtt min/avg/max/mdev = 0.392/0.429/0.479/0.039 ms

1. Close the **AWS Systems Manager - Session Manager** browser tab.

**Task complete:** You have successfully reviewed the basic network architecture for you lab, and tested the connectivity for two Amazon EC2 instances.

**Task 2: Create a basic Network Firewall, configure routing and test the firewall**

In this task, you deploy an AWS Network Firewall, change the routing configuration of your *VPC* to send traffic from the public subnet to the *Network Firewall*, and test your configuration using the *EC2* instances already created for this lab.

**Note:** A firewall needs a dedicated subnet with a /28 or larger IP space. Hence, you create this dedicated subnet before creating the firewall.

1. At the top of the AWS Management Console, in the search bar, search for and choose

VPC

.

1. In the navigation pane, under **Virtual private cloud**, choose **Subnets**.
2. Choose **Create subnet**.

On the **Create subnet** page, configure the following:

* For **VPC ID**, choose **Lab-VPC**.
* For **Subnet name**, enter

FirewallSubnet

.

* For **Availability Zone**, choose the value that matches with the value of **LabAZ** that is listed to the left of these instructions.
* For **IPv4 subnet CIDR block**, enter

10.0.2.0/28

.

1. Choose **Create subnet**.

**Expected output:** A *You have successfully created 1 subnet: subnet-xxxxxx* message is displayed at the top of the page.

TASK 2.1: FIREWALL SETUP

In this task, you create a firewall, some basic rule groups, and firewall policies. These resources are later used for testing the configuration of your *AWS Network Firewall* deployment.

An *AWS Network Firewall* rule group is a reusable set of criteria for inspecting and handling network traffic. You add one or more rule groups to a firewall policy as part of policy configuration.

**Learn more**

Refer to *Rule groups in AWS Network Firewall* in the **Additional resources** section for more information.

**Create the Network Firewall rule groups**

First, you create the following two **stateful** rule groups:

* A stateful rule group to address traffic to certain domains.
* A stateful rule group for ICMP traffic.

**Note:** A *stateful* rule group defines criteria for examining a packet in the context of traffic flow and of other traffic that’s related to the packet.

1. In the navigation pane, under **Network Firewall**, choose **Network Firewall rule groups**.
2. Choose **Create rule group**.
3. On the **Choose rule group type** page, configure the following:

* For **Rule group type**, choose **Stateful rule group**.
* For **Rule group format**, choose **Domain list**.
* For **Rule evaluation order**, choose **Action order**.

1. Choose **Next**.
2. On the **Describe rule group** page, configure the following:

* For **Name**, enter

StatefulRuleGroupDomainList

.

* For **Description - *optional***, enter

A stateful rule group specific to certain domains

.

* For **Capacity**, enter

1000

.

1. Choose **Next**.
2. On the **Configure rules** page, configure the following:

* For **Domain names**, enter the following domains, each on its own line:
  + www.example.com
  + www.facebook.com
* For **CIDR ranges**, choose **Default**.
* For **Protocols**, select **HTTP** and **HTTPS**.
* For **Action**, choose **Allow**.

1. Choose **Next**.
2. On the **Configure advanced settings - *optional*** page, keep the default settings, and choose **Next**.
3. On the **Add tags - *optional*** page, choose **Next**.
4. At the bottom of the **Review and create** page, choose **Create rule group**.

**Expected output:** A *You’ve successfully created rule group StatefulRuleGroupDomainList* message is displayed on top of the page.

1. On the **Rule groups** page, choose **Create rule group**.
2. On the **Choose rule group type** page, configure the following:

* For **Rule group type**, choose **Stateful rule group**.
* For **Rule group format**, choose **Standard stateful rule**.
* For **Rule evaluation order**, choose **Action order**.

1. Choose **Next**.
2. On the **Describe rule group** page, configure the following:

* For **Name**, enter

StatefulRuleGroupICMP

.

* For **Description - *optional***, enter

A stateful rule group for ICMP traffic

.

* For **Capacity**, enter

1000

.

1. Choose **Next**.
2. On the **Configure rules** page, in the **Standard stateful rules** section, configure the following:

* For **Protocol**, choose **ICMP**.
* For **Source**, keep the default setting of **Any**.
* For **Source port**, keep the default setting of **Any port**.
* For **Destination**, keep the default setting of **Any**.
* For **Destination port**, keep the default setting of **Any port**.
* For **Traffic direction**, keep the default setting of **Forward**.
* For **Action**, keep the default setting of **Pass**.

1. Choose **Add rule**.
2. Choose **Next**.
3. On the **Configure advanced settings - *optional*** page, keep the default settings, and choose **Next**.
4. On the **Add tags - *optional*** page, choose **Next**.
5. At the bottom of the **Review and create** page, choose **Create rule group**.

**Expected output:** A *You’ve successfully created rule group StatefulRuleGroupICMP* message is displayed at the top of the page.

Next, you create one *stateless* rule group that allows all traffic.

**Note:** A *stateless* rule group defines standard network connection attributes for examining a packet on its own, with no additional context.

1. On the **Rule Groups** page, choose **Create rule group**.
2. On the **Choose rule group type** page, configure the following:

* For **Rule group type**, choose **Stateless rule group**.

1. Choose **Next**.
2. On the **Describe rule group** page, configure the following:

* For **Name**, enter

StatelessRuleGroup

.

* For **Description**, enter

A stateless rule group that allows all traffic

.

* For **Capacity**, enter

1000

.

1. Choose **Next**.
2. On the **Configure rules** page, configure the following:

* For **Priority**, enter

10

.

* For **Protocol**, keep the default setting of **All**.
* For **Source**, choose **Custom** and then enter

10.0.0.0/16

 in the textbox below the drop-down menu.

* For **Destination**, choose **Custom** and then enter

10.0.0.0/16

 in the textbox below the drop-down menu.

* For **Rule action**, keep the default setting of **Pass**.

**Note:** In this lab environment, *10.0.0.0/16* is the IPv4 CIDR for *Lab-VPC*.

1. Choose **Add rule**.
2. Verify that the rule that you just added is visible in the **Rules** section.
3. Choose **Next**.
4. On the **Configure advanced settings - *optional*** page, keep the default settings, and choose **Next**.
5. On the **Add tags - *optional*** page, choose **Next**.
6. At the bottom of the **Review and create** page, choose **Create rule group**.

**Expected output:** A *You’ve successfully created rule group StatelessRuleGroup* message is displayed at the top of the page.

**Create the Firewall Policy**

An *AWS Network Firewall* firewall policy defines the monitoring and protection behavior for a firewall. The details of the behavior are defined in the rule groups that you add to your policy, and in some policy default settings. To use a firewall policy, you associate it with one or more firewalls.

**Learn more**

Refer to *Firewall policies in AWS Network Firewall* in the **Additional resources** section for more information.

1. In the navigation pane, under **Network Firewall**, choose **Firewall policies**.
2. On the **Firewall policies** page, choose **Create firewall policy**.
3. On the **Describe firewall policy** page, configure the following:

* For **Name**, enter

FirewallLabPolicy

.

* For **Description - *optional***, enter

Sample firewall policy for this lab

.

* For **Stream exception policy**, choose **Drop**.

1. Choose **Next**.
2. On the **Add rule groups - *optional*** page, in the **Stateless default actions** section, keep the default settings.
3. In the **Stateless rule group** section, choose **Add stateless rule groups**.
4. In the **Add from existing rule groups** pop-up window, select **StatelessRuleGroup**.
5. Choose **Add rule groups**.
6. In the **Stateful rule evaluation order and default actions** section, choose **Action order**.
7. In the **Stateful rule group** section, choose **Add stateful rule groups**.
8. In the **Add from existing rule groups** pop-up window, select **StatefulRuleGroupDomainList** and **StatefulRuleGroupICMP**.
9. Choose **Add rule groups**.
10. Choose **Next**.
11. On the **Configure advanced settings - *optional*** page, choose **Next**.
12. On the **Add TLS inspection configuration - *optional*** page, choose **Next**.
13. On the **Add tags** page, choose **Next**.
14. At the bottom of the **Review and create** page, choose **Create firewall policy**.

**Expected output:** A *You’ve successfully created firewall policy FirewallLabPolicy* message is displayed at the top of the page.

**Create Network Firewall**

An *AWS Network Firewall* firewall connects a firewall policy, which defines network traffic monitoring and filtering behavior, to the VPC that you want to protect. The firewall configuration includes specifications for the Availability Zones and subnets where the firewall endpoints are placed. It also defines high-level settings like the firewall logging configuration and tagging on the AWS firewall resource.

**Learn more**

Refer to *Firewalls in AWS Network Firewall* in the **Additional resources** section for more information.

1. In the navigation pane, under **Network Firewall**, choose **Firewalls**.
2. On the **AWS Network Firewall** getting started page, choose **Create firewall**.
3. On the **Describe firewall** page, configure the following:

* For **Firewall name**, enter

LabFirewall

.

* For **Description - *optional***, enter

Lab Network Firewall

.

1. Choose **Next**.
2. On the **Configure VPC and subnets** page, configure the following:

* For **VPC**, choose **Lab-VPC**.
* Under **Firewall subnets**, configure the following:
  + For **Availability Zone**, choose the value that matches the value of **LabAZ** that is listed to the left of these instructions.
  + For **Subnet**, choose **FirewallSubnet**.
  + For **IP address type**, choose **IPv4**.

1. Choose **Next**.
2. On the **Configure advanced settings - *optional*** page, in the **Protection against changes** section, configure the following:

* For **Delete protection**, clear **Enable**.

1. Choose **Next**.
2. On the **Associated firewall policy** page, configure the following:

* For **Firewall policy**, choose **Associate an existing firewall policy**.
* For **Choose firewall policy**, choose **FirewallLabPolicy**.

1. Choose **Next**.
2. On the **Add tags - *optional*** page, choose **Next**.
3. At the bottom of the **Review and create** page, choose **Create firewall**.

**Expected output:** A *You’ve successfully created firewall LabFirewall* message is displayed at the top of the page.

**Note:** The firewall provisioning takes approximately 5-10 minutes to complete. While it is being provisioned, you can continue to configure logging for your firewall.

1. On the **LabFirewall** page, choose the **Firewall details** tab.
2. In the **Logging** section, choose **Edit**.
3. On the **Edit firewall logging configuration** page, configure the following:

* In the **Log type** section:
  + For **Log type**, select **Alert** and **Flow**.
* In the **Log destination for alerts** section:
  + For **Log destination**, choose **CloudWatch log group**.
  + For **CloudWatch log group**, search for and choose

/nfw/alert

.

* In the **Log destination for flows** section:
  + For **Log destination**, choose **CloudWatch log group**.
  + For **CloudWatch log group**, search for and choose

/nfw/flow

.

**Learn more**

You can record alert logs and flow logs from your *Network Firewall* stateful engine.

* *Alert* logs report traffic that matches your stateful rules that have an action that sends an alert. A stateful rule sends alerts for the the following rule actions:
  + DROP
  + ALERT
  + REJECT
* *Flow* logs are standard network traffic flow logs. Each flow log record captures the network flow for a specific standard stateless rule group.

Refer to *Logging network traffic from AWS Network Firewall* in the **Additional resources** section for more information.

1. Choose **Save**.

**Expected output:** A *You’ve successfully updated the firewall LabFirewall* message is displayed at the top of the page.

**Refresh:** Refresh the page until the **Firewall status** changes from **Provisioning** to **Ready**.

TASK 2.2: ROUTING CONFIGURATION

Before you begin testing your firewall rules, you need to adjust routing to send traffic through the firewall.

In this task, you create the following two route tables:

* Internet gateway (IGW) ingress route table.
* Firewall subnet route table.

**Create IGW ingress route table**

1. In the navigation pane, under **Virtual private cloud**, choose **Route tables**.
2. Choose **Create route table**.
3. On the **Create route table** page, in the **Route table settings** section, configure the following:

* For **Name - *optional***, enter

igw-ingress-route-table

.

* For **VPC**, choose **Lab-VPC**.

1. Choose **Create route table**.

**Expected output:** A *Route table rtb-xxxxxxx | igw-ingress-route-table was created successfully.* message is displayed at the top of the page.

**Create firewall subnet route table**

1. In the navigation pane, under **Virtual private cloud**, choose **Route tables**.
2. Choose **Create route table**.
3. On the **Create route table** page, in the **Route table settings** configure the following:

* For **Name - *optional***, enter

firewall-subnet-route-table

.

* For **VPC**, choose **Lab-VPC**.

1. Choose **Create route table**.

**Expected output:** A *Route table rtb-xxxxxxx | igw-ingress-route-table was created successfully.* message is displayed at the top of the page.

**Configure the igw-ingress-route-table to direct incoming traffic to the Firewall**

1. In the navigation pane, under **Virtual private cloud**, choose **Route tables**.
2. On the **Route tables** page, in the row for **igw-ingress-route-table**, choose the **Route table ID** link to view its details.
3. On the **igw-ingress-route-table** details page, choose the **Edge associations** tab.
4. Choose **Edit edge associations**.
5. On the **Edit edge associations** page, select **Internet gateway**.
6. Choose **Save changes**.

**Expected output:** A *You have successfully updated edge associations for rtb-xxxxxxx / igw-ingress-route-table.* message is displayed on top of the screen.

1. Choose the **Routes** tab.
2. Choose **Edit routes**.
3. On the **Edit routes** page, choose **Add route** and configure the following:

* For **Destination**, enter

10.0.0.0/24

.

* For **Target**, choose **Gateway Load Balancer Endpoint** to search for the available endpoints, and then choose the VPC firewall endpoint that starts with **vpce-**.

**Learn more**

A firewall endpoint is similar to a PrivateLink VPC interface endpoint and it shows up as *vpce-id* in your VPC route table target selection.

Refer to *What is AWS PrivateLink?* in the **Additional resources** section for more information.

1. Choose **Save changes**.

**Expected output:** A *Updated routes for rtb-xxxxxxx / igw-ingress-route-table successfully* message is displayed at the top of the page.

**Configure the firewall-subnet-route-table to direct outgoing traffic to the Internet Gateway**

1. In the navigation pane, under **Virtual private cloud** choose **Route tables**.
2. On the **Route tables** page, in the row for **firewall-subnet-route-table**, choose the **Route table ID** link to view its details.
3. Choose the **Routes** tab.
4. Choose **Edit routes**.
5. On the **Edit routes** page, choose **Add route** and configure the following:

* For **Destination**, enter

0.0.0.0/0

.

* For **Target**, choose **Internet Gateway** , and then choose **Lab-InternetGateway**.

1. Choose **Save changes**.

**Expected output:**

An *Updated routes for rtb-xxxxxxx / firewall-subnet-route-table successfully* message is displayed at the top of the page.

1. Choose the **Subnet associations** tab.
2. In the **Explicit subnet associations** section, choose **Edit subnet associations**.
3. On the **Edit subnet associations** page, select **FirewallSubnet**.
4. Choose **Save associations**.

**Expected output:** A *You have successfully updated subnet associations for rtb-xxxxxxx / firewall-subnet-route-table.* message is displayed at the top of the page.

**Configure the Lab-PublicRouteTable to direct outgoing traffic to the firewall**

1. In the navigation pane, under **Virtual private cloud**, choose **Route tables**.
2. On the **Route tables** page, in the row for **Lab-PublicRouteTable**, choose the **Route table ID** link to view its details.
3. Choose the **Routes** tab.
4. Choose **Edit routes**.
5. On the **Edit routes** page, for the existing **0.0.0.0/0** destination, do the following:

* For **Target**, change the value to **Gateway Load Balancer Endpoint**, and then choose the VPC firewall endpoint that starts with **vpce-**.

1. Choose **Save changes**.

**Expected output:** An *Updated routes for rtb-xxxxxxx / Lab-PublicRouteTable successfully* message is displayed at the top of the page.

You should now have three route tables as follows:

**Lab-PublicRouteTable**

| **Destination** | **Target** |
| --- | --- |
| 10.0.0.0/16 | local |
| 0.0.0.0/0 | vpce-id |

**igw-ingress-route-table**

| **Destination** | **Target** |
| --- | --- |
| 10.0.0.0/16 | local |
| 10.0.0.0/24 | vpce-id |

**firewall-subnet-route-table**

| **Destination** | **Target** |
| --- | --- |
| 10.0.0.0/16 | local |
| 0.0.0.0/0 | igw-id |

TASK 2.3: TESTING FIREWALL

In this task, you test the configuration of your AWS Network Firewall configuration.

**Test internet access from your instance in the public subnet**

1. At the top of the AWS Management Console, in the search bar, search for and choose

EC2

.

1. In the navigation pane, under **Instances**, choose **Instances**.
2. Select **Lab-PublicInstance**.
3. Choose **Connect**.
4. On the **Connect to instance** page, choose the **Session Manager** tab.
5. Choose **Connect**.

**Expected output:** You are redirected to a terminal session with a secured connection to the **Lab-PublicInstance** instance.

First, you use the *curl* command to connect to a public website.

**Learn more**

curl\_ is a command-line tool and library used for transferring data to or from a server, using any of the supported protocols including but not limited to HTTP, HTTPS, FTP, SCP, SMTP, TELNET, and FILE.

Refer to *curl* in the **Additional resources** section for more information.

1. **Command:** To confirm the connection from the public EC2 instance to a public website, run the following command:

curl -Iv www.example.com --max-time 5

**Expected output:** The following output confirms that you have access over port 80.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Trying 93.184.216.34:80...

\* Connected to www.example.com (93.184.216.34) port 80 (#0)

> HEAD / HTTP/1.1

> Host: www.example.com

> User-Agent: curl/8.0.1

> Accept: \*/\*

>

< HTTP/1.1 200 OK

HTTP/1.1 200 OK

< Accept-Ranges: bytes

Accept-Ranges: bytes

< Age: 603858

Age: 603858

< Cache-Control: max-age=604800

Cache-Control: max-age=604800

< Content-Type: text/html; charset=UTF-8

Content-Type: text/html; charset=UTF-8

< Date: Fri, 07 Jul 2023 04:09:22 GMT

Date: Fri, 07 Jul 2023 04:09:22 GMT

< Etag: "3147526947"

Etag: "3147526947"

< Expires: Fri, 14 Jul 2023 04:09:22 GMT

Expires: Fri, 14 Jul 2023 04:09:22 GMT

< Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT

Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT

< Server: ECS (oxr/830F)

Server: ECS (oxr/830F)

< X-Cache: HIT

X-Cache: HIT

< Content-Length: 1256

Content-Length: 1256

<

\* Connection #0 to host www.example.com left intact

Next, try the *ping* command to confirm if you are able to communicate over ICMP.

**Learn more**

*ping* is a networking utility tool used for testing reachability of a host on an Internet Protocol (IP) network.

Refer to *ping (networking utility)* in the **Additional resources** section for more information.

1. **Command:** To confirm if you are able to communicate over ICMP, run the following command:

ping -c 5 8.8.8.8

**Expected output:** The following output confirms that you are able to communicate over ICMP.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.

64 bytes from 8.8.8.8: icmp\_seq=1 ttl=97 time=10.2 ms

64 bytes from 8.8.8.8: icmp\_seq=2 ttl=97 time=8.94 ms

64 bytes from 8.8.8.8: icmp\_seq=3 ttl=97 time=9.05 ms

64 bytes from 8.8.8.8: icmp\_seq=4 ttl=97 time=9.05 ms

64 bytes from 8.8.8.8: icmp\_seq=5 ttl=97 time=9.02 ms

--- 8.8.8.8 ping statistics ---

5 packets transmitted, 5 received, 0% packet loss, time 4005ms

rtt min/avg/max/mdev = 8.947/9.258/10.205/0.486 ms

Now that you have tested internet access from your instance in the public subnet, you edit your rule groups so you can test if the firewall is working correctly.

**Test the StatefulRuleGroupDomainList rule group**

1. Return to your web browser tab with the **AWS Management Console**.
2. At the top of the AWS Management Console, in the search bar, search for and choose

VPC

.

1. In the navigation pane, under **Network Firewall**, choose **Network Firewall rule groups**.
2. Choose the **StatefulRuleGroupDomainList** link to view its details.
3. On the **StatefulRuleGroupDomainList** details page, in the **Traffic to inspect and action** section, choose **Edit**.
4. On the **Edit rules** page, for **Action** choose **Deny**.
5. Choose **Save rule group**.

Now verify if the firewall is working correctly. Once again, use curl commands on two different domains to see if you can access them.

1. Return to your web browser tab with the **AWS Systems Manager - Session Manager** session.
2. **Command:** To confirm the connection from the public EC2 instance to the **www.example.com** domain, run the following command:

curl -Iv www.example.com --max-time 5

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Trying 93.184.216.34:80...

\* Connected to www.example.com (93.184.216.34) port 80 (#0)

> HEAD / HTTP/1.1

> Host: www.example.com> User-Agent: curl/8.0.1

> Accept: \*/\*

>

\* Operation timed out after 5000 milliseconds with 0 bytes received

\* Closing connection 0

curl: (28) Operation timed out after 5000 milliseconds with 0 bytes received

As you can see, the curl command times out after 5 seconds and you can no longer connect to the *www.example.com* domain.

Next, use the curl command on the *www.facebook.com* domain.

1. **Command:** To confirm the connection from the public EC2 instance to a the **www.facebook.com** domain, run the following command:

curl -Iv www.facebook.com --max-time 5

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Trying 157.240.249.35:80...

\* Connected to www.facebook.com (157.240.249.35) port 80 (#0)

> HEAD / HTTP/1.1

> Host: www.facebook.com

> User-Agent: curl/8.0.1

> Accept: \*/\*

>

\* Operation timed out after 5001 milliseconds with 0 bytes received

\* Closing connection 0

curl: (28) Operation timed out after 5001 milliseconds with 0 bytes received

As you can see, the curl command again times out after 5 seconds and cannot connect to the *www.facebook.com* domain.

This confirms that the **StatefulRuleGroupDomainList** Network Firewall rule group is working as expected and traffic is denied to the *www.example.com* and *www.facebook.com* domains.

**Knowledge check:** Using the above example commands, try connecting to other domains to see if you are able to connect to them.

**Test the StatefulRuleGroupICMP rule group**

1. Return to your web browser tab the **AWS Management Console**.
2. In the navigation pane, under **Network Firewall**, choose **Network Firewall rule groups**.
3. Choose the **StatefulRuleGroupICMP** link to view its details.
4. On the **StatefulRuleGroupICMP** details page, in the **Rules** section, choose **Edit**.
5. At the bottom of the **Edit rules** page, in the **Rules** section, choose the **ICMP** rule.
6. Choose **Edit**.
7. In the **Edit rule** pop-up window, for **Action**, choose **Drop**.
8. Choose **Save**.
9. Choose **Save rule group**.

Now verify that the firewall is working correctly. You use *ping* commands to see if you can communicate over ICMP.

1. Return to your web browser tab with the **AWS Systems Manager - Session Manager** session.
2. **Command:** To test connectivity to a public IP over ICMP, run the following command:

ping -c 5 8.8.8.8

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.

--- 8.8.8.8 ping statistics ---

5 packets transmitted, 0 received, 100% packet loss, time 4088ms

As you can see, the *ping* command times out after 5 attempts and you cannot communicate to a public IP over ICMP.

**Note:** If the *ping* command still works, wait for a minute and try again. This temporary inconsistency can occur when you make changes to an existing firewall. Generally, any inconsistencies of this type last only a few seconds.

Next, use the *ping* command on the *www.amazon.com* domain.

1. **Command:** To test connectivity to *www.amazon.com* domain over ICMP, run the following command:

ping -c 5 www.amazon.com

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PING d3ag4hukkh62yn.cloudfront.net (3.163.28.135) 56(84) bytes of data.

--- d3ag4hukkh62yn.cloudfront.net ping statistics ---

5 packets transmitted, 0 received, 100% packet loss, time 4101ms

As you can see, the *ping* command times out again after 5 attempts and you cannot communicate with the *www.amazon.com* domain over ICMP.

This confirms that the **StatefulRuleGroupICMP** Network Firewall rule group is working as expected and ICMP traffic is being dropped.

1. Close the **AWS Systems Manager - Session Manager** browser tab.

**Task complete:** You successfully deployed an AWS Network Firewall, changed the routing configuration of your VPC to send traffic from the public subnet to the Network Firewall, and tested your configuration using the EC2 instances already created for this lab.

**Task 3: Modify the Network Firewall to include NAT capabilities**

You can add a network address translation (NAT) gateway to your AWS Network Firewall architecture for the areas of your VPC where you need NAT capabilities.

In this task, you expand on what you have already deployed in the previous tasks by deploying a NAT Gateway. You then configure routes from your private subnet to route through your NAT Gateway and then through your firewall.

TASK 3.1: CREATING NAT GATEWAY

In this task, you create a new subnet to support the NAT Gateway. You then create a NAT Gateway.

**Create a subnet to support NAT Gateway**

1. Return to your web browser tab with the **AWS Management Console**.
2. At the top of the AWS Management Console, in the search bar, search for and choose

VPC

.

1. In the navigation pane, under **Virtual private cloud**, choose **Subnets**.
2. Choose **Create subnet**.

On the **Create subnet** page, configure the following:

* For **VPC ID**, select **Lab-VPC** from the dropdown menu.
* For **Subnet name**, enter

NATGWSubnet

.

* For **Availability Zone**, choose the value that matches the value of **LabAZ** listed to the left of these instructions.
* For **IPv4 subnet CIDR block**, enter

10.0.3.0/24

.

1. Choose **Create subnet**.

**Expected output:** A *You have successfully created 1 subnet: subnet-xxxxxx* message is displayed at the top of the page.

**Create a NAT Gateway**

1. In the navigation pane, under **Virtual private cloud**, choose **NAT gateways**.
2. Choose **Create NAT gateway**.
3. On the **Create NAT gateway** page, configure the following:

* For **Name - *optional***, enter

NATGWForFirewall

.

* For **Subnet**, choose **NATGWSubnet**.
* For **Connectivity type**, leave **Public** selected.
* For **Elastic IP allocation ID**, choose **Allocate Elastic IP** to create a new EIP.

**Important**

Make sure that you see an *Elastic IP address XX.XX.XXX.XXX (eipalloc-XXXXXXX) allocated.* message displayed at the top of the page before proceeding further.

1. Choose **Create NAT gateway**.

**Expected output:** A *NAT gateway nat-xxxxxxx | NATGWForFirewall was created successfully.* message is displayed at the top of the page.

TASK 3.2: CONFIGURE ROUTING

In this task, you create a route table for the NAT Gateway subnet and configure routing.

1. In the navigation pane, under **Virtual private cloud**, choose **Route tables**.
2. Choose **Create route table**.

1 On the **Create route table** page, in the **Route table settings** configure the following:

* For **Name- *optional***, enter

nat-gateway-route-table

.

* For **VPC**, select **Lab-VPC**.

1. Choose **Create route table**.

**Expected output:** A *Route table rtb-xxxxxxx | nat-gateway-route-table was created successfully.* message is displayed at the top of the page.

1. Choose the **Routes** tab.
2. Choose **Edit routes**.
3. On the **Edit routes** page, choose **Add route**, and then configure the following:

* For **Destination**, enter

0.0.0.0/0

.

* For **Target**, choose **Gateway Load Balancer Endpoint**, and then select the VPC firewall endpoint that starts with **vpce-**.

1. Choose **Save changes**.

**Expected output:** An *Updated routes for rtb-xxxxxxx / nat-gateway-route-table successfully* message is displayed at the top of the page.

1. Choose the **Subnet associations** tab.
2. In the **Explicit subnet associations** section, choose **Edit subnet associations**.
3. On the **Edit subnet associations** page, select **NATGWSubnet**.
4. Choose **Save associations**.

**Expected output:** A *You have successfully updated subnet associations for rtb-xxxxxxx / nat-gateway-route-table.* message is displayed at the top of the page.

Now you create a route table for the private subnet and add routes.

TASK 3.3: CREATE A ROUTE TABLE FOR THE PRIVATE SUBNET AND ADD ROUTES

1. In the navigation pane, under **Virtual private cloud**, choose **Route tables**.
2. Choose **Create route table**.
3. On the **Create route table** page, in the **Route table settings** configure the following:

* For **Name- *optional***, enter

private-subnet-route-table

.

* For **VPC**, choose **Lab-VPC**.

1. Choose **Create route table**.

**Expected output:** A *Route table rtb-xxxxxxx | private-subnet-route-table was created successfully.* message is displayed at the top of the page.

1. Choose the **Routes** tab.
2. Choose **Edit routes**.
3. On the **Edit routes** page, choose **Add route**, and then configure the following:

* For **Destination**, enter

0.0.0.0/0

.

* For **Target**, choose **NAT Gateway**, and then choose **NATGWForFirewall**.

1. Choose **Save changes**.

**Expected output:** An *Updated routes for rtb-xxxxxxx / private-subnet-route-table successfully* message is displayed at the top of the page.

1. Choose the **Subnet associations** tab.
2. In the **Explicit subnet associations** section, choose **Edit subnet associations**.
3. On the **Edit subnet associations** page, select **Lab-PrivateSubnet**.
4. Choose **Save associations**.

**Expected output:** A *You have successfully updated subnet associations for rtb-xxxxxxx / private-subnet-route-table.* message is displayed on top of the screen.

Since you have new subnets, you have to change your route table that was created earlier for the Internet Gateway.

1. In the navigation pane, under **Virtual private cloud**, choose **Route tables**.
2. On the **Route tables** page, in the row for **igw-ingress-route-table**, choose the **Route table ID** link to view its details.
3. Choose the **Routes** tab.
4. Choose **Edit routes**.
5. On the **Edit routes** page, choose **Add route**, and then configure the following:

* For **Destination**, enter

10.0.1.0/24

.

* For **Target**, select **Gateway Load Balancer Endpoint**, and then choose the VPC firewall endpoint that starts with **vpce-**.

1. Choose **Add route** again, and then configure the following:

* For **Destination**, enter

10.0.3.0/24

.

* For **Target**, choose **Gateway Load Balancer Endpoint**, and then choose the VPC firewall endpoint that starts with **vpce-**.

1. Choose **Save changes**.

**Expected output:** An *Updated routes for rtb-xxxxxxx / igw-ingress-route-table successfully* message is displayed on top of the screen.

**Task complete:** You successfully deployed a *NAT Gateway* and configured routes from your private subnet to route through your *NAT Gateway* and then through your firewall.

**Task 4: Test the firewall and routing configuration with NAT capabilities**

In this task, you test the firewall with *NAT Gateway* setup using the *EC2* instance in the private subnet to confirm you have configured everything correctly.

1. At the top of the AWS Management Console, in the search bar, search for and choose

EC2

.

1. In the navigation pane, choose **Instances**.
2. Select **Lab-PrivateInstance**.
3. Choose **Connect**.
4. On the **Connect to instance** page, choose the **Session Manager** tab.
5. Choose **Connect**.

**Expected output:** You are redirected to a terminal session with a secured connection to the **Lab-PrivateInstance** instance.

You now test that your firewall rules from earlier tasks are still working.

1. **Command:** To confirm the connection from the private EC2 instance to the **www.example.com** domain, run the following command:

curl -Iv www.example.com --max-time 5

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Trying 93.184.216.34:80...

\* Connected to www.example.com (93.184.216.34) port 80 (#0)

> HEAD / HTTP/1.1

> Host: www.example.com> User-Agent: curl/8.0.1

> Accept: \*/\*

>

\* Operation timed out after 5000 milliseconds with 0 bytes received

\* Closing connection 0

curl: (28) Operation timed out after 5000 milliseconds with 0 bytes received

As you can see, the curl command times out after 5 seconds and you cannot connect to the *www.example.com* domain.

Next, use the curl command on the *www.facebook.com* domain.

1. **Command:** To confirm the connection from the public EC2 instance to a the **www.facebook.com** domain, run the following command:

curl -Iv www.facebook.com --max-time 5

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Trying 31.13.70.36:80...

\* Connected to www.facebook.com (31.13.70.36) port 80 (#0)

> HEAD / HTTP/1.1

> Host: www.facebook.com

> User-Agent: curl/8.0.1

> Accept: \*/\*

>

\* Operation timed out after 5001 milliseconds with 0 bytes received

\* Closing connection 0

curl: (28) Operation timed out after 5001 milliseconds with 0 bytes received

As you can see, the curl command again times out after 5 seconds and you cannot connect to the *www.facebook.com* domain.

This confirms that the **StatefulRuleGroupDomainList** Network Firewall rule group that was created earlier is still working as expected and traffic is denied to the *www.example.com* and *www.facebook.com* domains.

Finally, test the **StatefulRuleGroupICMP** rule group.

1. **Command:** To test connectivity to a public IP over ICMP, run the following command:

ping -c 5 8.8.8.8

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.

--- 8.8.8.8 ping statistics ---

5 packets transmitted, 0 received, 100% packet loss, time 4088ms

As you can see, the *ping* command times out after 5 attempts and you are not able to communicate to a public IP over ICMP.

Now use the *ping* command on the *www.amazon.com* domain.

1. **Command:** To test connectivity to **www.amazon.com** domain over ICMP, run the following command:

ping -c 5 www.amazon.com

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PING d3ag4hukkh62yn.cloudfront.net (3.163.28.135) 56(84) bytes of data.

--- d3ag4hukkh62yn.cloudfront.net ping statistics ---

5 packets transmitted, 0 received, 100% packet loss, time 4101ms

As you can see, the *ping* command times out again after 5 attempts and you are not able to communicate with *www.amazon.com* domain over ICMP.

This confirms that the **StatefulRuleGroupICMP** Network Firewall rule group that was created earlier is still working as expected and ICMP traffic is being dropped.

**Knowledge check:** Can you modify these rules to allow traffic and then verify that you are able to connect?

**Consider:** At this point you might want to try some scenarios that apply directly to your environment from both public and private instances. What is relevant to you?

**Task complete:** You have successfully tested the Firewall with NAT Gateway setup using the EC2 instance in a private subnet to confirm you have configured everything correctly.

**Conclusion**

You have successfully done the following:

* Reviewed the basic infrastructure needed to support the Network Firewall scenarios.
* Configured AWS Network Firewall.
* Configured and tested routing configuration using a basic subnet architecture
* Configured and tested routing configuration using a more sophisticated architecture involving a NAT Gateway.

**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**

* [What is AWS Network Firewall?](https://docs.aws.amazon.com/network-firewall/latest/developerguide/what-is-aws-network-firewall.html)
* [What is Amazon EC2?](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html)
* [AWS Systems Manager Session Manager](https://docs.aws.amazon.com/systems-manager/latest/userguide/session-manager.html)
* [What is Amazon VPC?](https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html)
* [Visualize the resources in your VPC](https://docs.aws.amazon.com/vpc/latest/userguide/modify-vpcs.html#view-vpc-resource-map)
* [Rule groups in AWS Network Firewall](https://docs.aws.amazon.com/network-firewall/latest/developerguide/rule-groups.html)
* [Firewall policies in AWS Network Firewall](https://docs.aws.amazon.com/network-firewall/latest/developerguide/firewall-policies.html)
* [Firewalls in AWS Network Firewall](https://docs.aws.amazon.com/network-firewall/latest/developerguide/firewalls.html)
* [curl](https://curl.se/)
* [ping (networking utility)](https://en.wikipedia.org/wiki/Ping_(networking_utility))
* [Logging network traffic from AWS Network Firewall](https://docs.aws.amazon.com/network-firewall/latest/developerguide/firewall-logging.html)
* [What is AWS PrivateLink?](https://docs.aws.amazon.com/vpc/latest/privatelink/what-is-privatelink.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).