Forcepoint Next Generation Firewall

6.3 and higher

How to deploy Forcepoint Next Generation Firewall in the Amazon Web Services cloud

Revision I

Contents

Introduction

You can deploy Forcepoint Next Generation Firewall in the Amazon Web Services (AWS) cloud to provide VPN connectivity, access control, and inspection for services in the AWS cloud.

Forcepoint Next Generation Firewall (Forcepoint NGFW) is available in the Amazon marketplace as an Amazon machine image (AMI) that allows you to run a Forcepoint NGFW Engine instance in Amazon EC2. You deploy Forcepoint NGFW Engines in the same way as other virtual machines in Amazon EC2.

Note

AWS objects are only unique within a region. For more information about regions, see https://aws.amazon.com/about-aws/global-infrastructure/.

This document provides an overview of the configuration tasks in Amazon EC2. For more information, see the *Amazon Elastic Compute Cloud Documentation* at https://docs.aws.amazon.com/ec2/.

Note

All configuration values shown in this document are examples. Your configuration might be different depending on your environment.

Licensing models for Forcepoint NGFW in the AWS cloud

Two licensing models are supported for Forcepoint NGFW in the AWS cloud.

There are two AMIs, depending on the licensing model:

- **Bring Your Own License** — You pay only Amazon's standard runtime fee for the engine instance. You must install a license for the engine in the Forcepoint NGFW Security Management Center (SMC). Forcepoint Customer Hub is provided according to your support contract. For more information, see Support Programs.
- **Hourly** (pay as you go license) — You pay Amazon's standard runtime fee for the engine instance plus an hourly license fee based on the runtime of the engine. No license installation is needed for the engine in the SMC. Your subscription includes Forcepoint essential support. For more information, see Support Information.

For more information about Amazon's infrastructure prices, see https://aws.amazon.com/ec2/pricing/on-demand/. For more information about hourly license fees, see Forcepoint in the AWS marketplace.

For information about supported Forcepoint NGFW versions, see Knowledge Base article 10156.

Considerations for deploying Forcepoint NGFW in the AWS cloud

There are some additional considerations when you deploy Forcepoint NGFW in the AWS cloud.

- Only the Firewall/VPN role is supported.
- Only single-node NGFW Engines are supported. NGFW Engine Clusters are not supported.
- Master NGFW Engines and Virtual Security Engines are not supported.
- VLAN interfaces and link aggregation are not supported.
- FIPS mode is not supported.
- Memory dump diagnostics are not supported.
- The engine does not limit the number of network interfaces but some types of instances might have limitations.

Note

AWS does not allow the root user to log on to the command line. Instead, you must log on as the aws user and use sudo to gain root permissions.

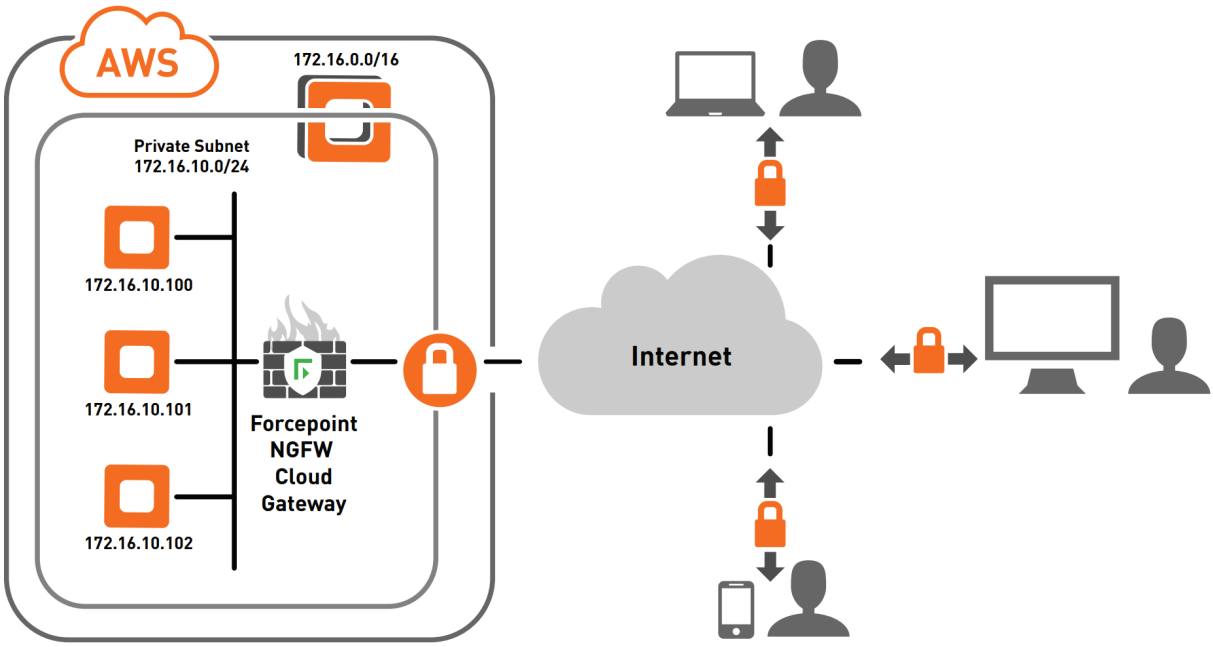
Use cases for Forcepoint NGFW in the AWS cloud

These deployment examples show how you can use Forcepoint NGFW in the AWS cloud environment.

Remote access connectivity

You can use Forcepoint NGFW as a cloud edge gateway to connect your remote users to Amazon Virtual Private Cloud (VPC).

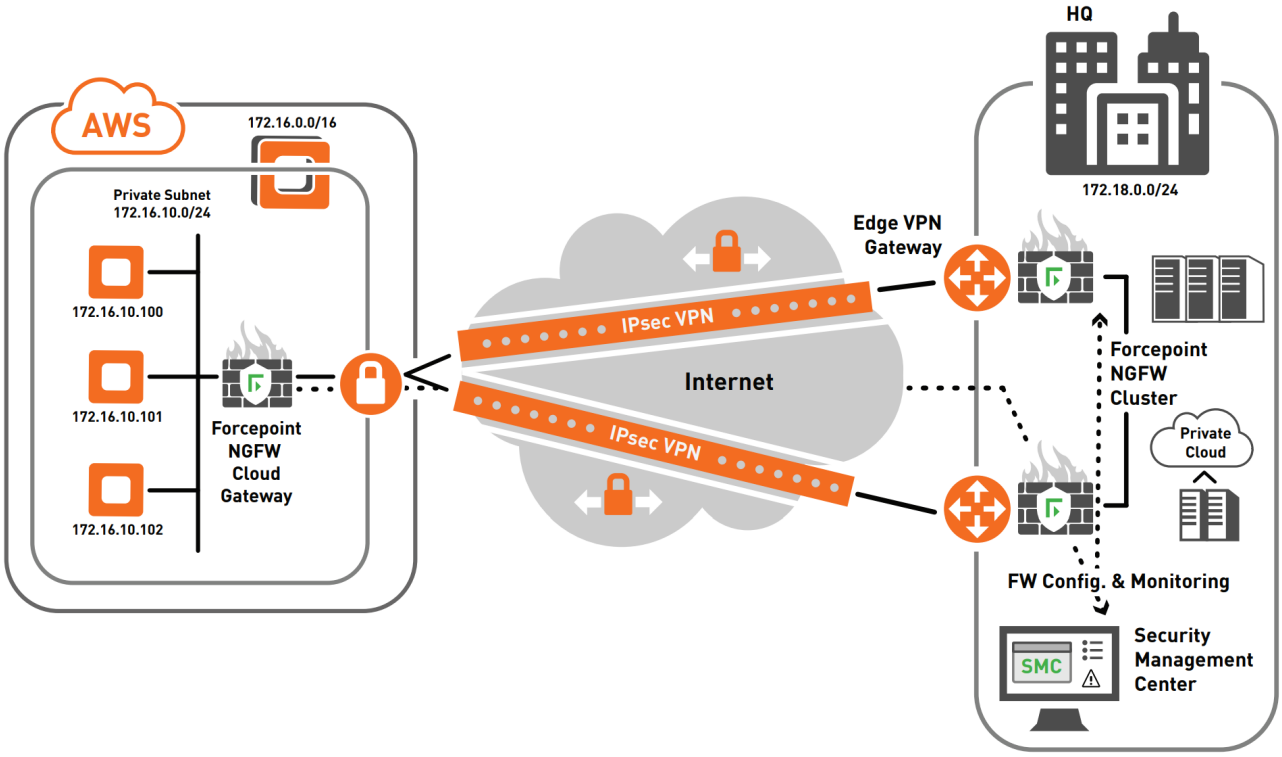
You can deploy Forcepoint NGFW as a cloud gateway in an Amazon Elastic Compute Cloud (EC2) instance. Forcepoint NGFW provides advanced firewall features, such as application awareness and user identity capabilities, to protect your EC2 instances for all inbound and outbound access.



Corporate data center connectivity

Physical and virtual Forcepoint NGFW gateways securely connect your corporate on-premises data centers to your virtual ones in AWS VPCs.

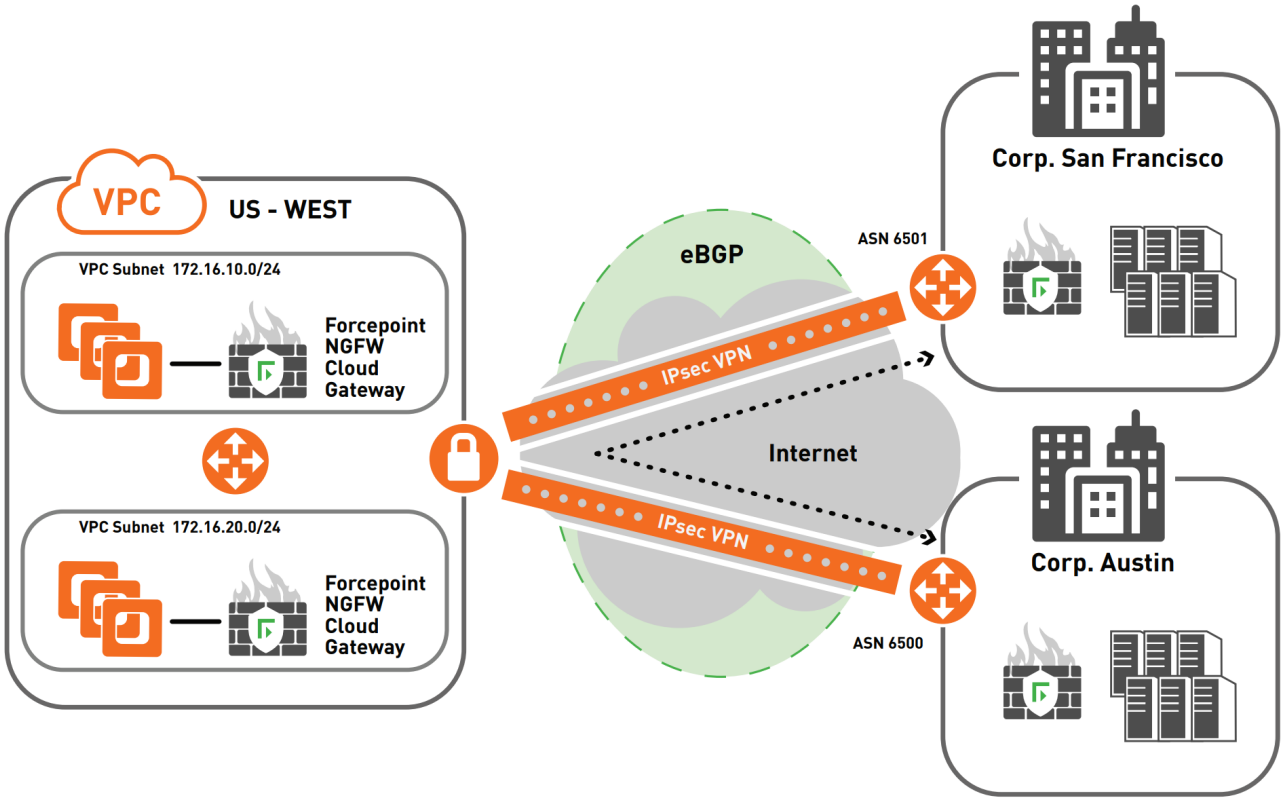
Simply create one or more VPN connections between your data center network and your Forcepoint NGFW running in your Amazon VPC network. Manage and control all your software and physical Forcepoint NGFW Engines at both ends of the VPN connections using the Security Management Center (SMC). You can also use a cluster of physical Forcepoint NGFW Engines to provide high availability for business continuity on the on-premises side of the VPN connection.



VPN CloudHub

Securely connect remote branch offices using the AWS VPN CloudHub, operating on a simple hub-and-spoke model, for primary and backup connectivity between remote offices.

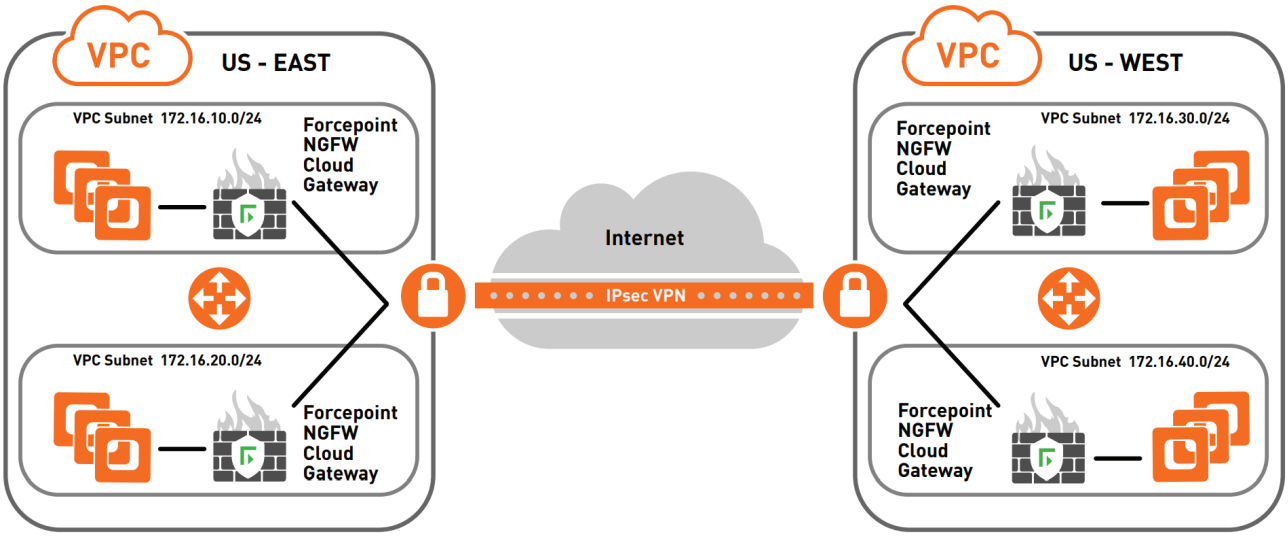
Each remote site must have a unique ASN to send data to and receive data from other sites. The choice between static routing and dynamic routing for your VPN connections depends on how you want to handle failover. Both static and dynamic connectivity types use IPsec VPN tunnels. Dynamic routing uses BGP peering to exchange routes and routing priorities between AWS and the remote endpoints. Dynamic routing using Forcepoint NGFW is more flexible than dynamic routing in AWS, because AWS automatically changes BGP gateway routes when the gateway changes.



VPC-to-VPC routing between regions

Create secure VPN tunnels between two or more Forcepoint NGFW Engines to connect VPCs across multiple AWS regions.

You can manage and enforce security policies at both ends of the VPN connection using the Security Management Center (SMC).



Deploying Forcepoint NGFW in the AWS cloud

You can deploy Forcepoint NGFW in the AWS cloud using 1-Click Launch or using Manual Launch when you have an existing SMC installation.

Related tasks

Deploy Forcepoint NGFW using 1-Click Launch on page 6
Deploy Forcepoint NGFW in AWS when you have an existing SMC installation on page 15

Deploy Forcepoint NGFW using 1-Click Launch

Create a Forcepoint NGFW instance, then deploy the SMC on your own hardware or in a separate instance on AWS.

Create a Forcepoint NGFW instance using 1-Click Launch

Configure and launch an instance of the Forcepoint NGFW AMI using 1-Click Launch.

CAUTION

If required for regulatory compliance, or in environments with stricter security requirements, we recommend using dedicated instances when you deploy Forcepoint NGFW in AWS.

We recommend using the following instance types depending on the Forcepoint NGFW product:

Forcepoint NGFW product	EC2 instance type
NGFW 2 CPU	M4.large
NGFW 4 CPU	M4.xlarge or C4.xlarge
NGFW 8 CPU	M4.2xlarge or C4.2xlarge
NGFW 16 CPU	C4.4xlarge

For information about VM size and network performance, see the Amazon documentation at https://aws.amazon.com/ec2/instance-types/. Enabling some Forcepoint NGFW features, such as inspection, might decrease the network throughput.

Forcepoint NGFW is designed to receive and manage all traffic on all ports. Use a security group that allows connections on all ports for inbound and outbound for the instance in which Forcepoint NGFW is running.

Steps

1) In the AWS Marketplace, start the launch for the Forcepoint NGFW AMI.

2) On the **1-Click Launch** tab, configure the following settings:

Setting	Configuration
Version	Select the most recent version.
Region	Select the region that is the best match for your existing infrastructure and geographic location.
EC2 Instance Type	Select an instance type that meets your performance needs. The AMI automatically restricts the instance types so that only compatible instance types are available. **Note** If you want to change the instance type later, you must create a new instance.
VPC Settings	Select a VPC and a subnet that correspond to the management interface of the NGFW Engine.
Security Group	Select a security group based on the seller settings. If the default security group is too limited for your environment, you can use a different security group or change the rules. You can also configure the NGFW Engine to restrict access.
Key Pair	Select a key pair for SSH connections to the NGFW engine. **Note** The key is the only allowed authentication method for SSH connections to the engine command line.

3) Click **Launch with 1-click**.

4) When the instance is running, connect to the command line of the NGFW Engine and verify the SSH server identity.

   a) In the AWS web management console, select the NGFW Engine instance, then select **Actions** > **Instance Settings** > **Get system log** to show the SSH server fingerprints.

   The SSH server fingerprint are shown at the end of the NGFW Engine boot messages.

b) On your computer, open a terminal program, then enter the following command to open an SSH connection to the command line of the NGFW Engine using the aws user account:

```
ssh -i <your ssh private key>.pem aws@<aws instance public ip address>
```

The SSH key fingerprints are shown when you connect.

c) Compare the SSH key fingerprints to the SSH server fingerprints from the system log.

d) To confirm that you want to continue connecting, type `yes`.
The IP address of the NGFW Engine is added to the SSH known hosts list.

5) If the AMI does not support the use of sudo without a password, enter the following command to set a sudo password for the aws user:

```
sudo passwd
```

Next steps

If you do not have existing SMC installation, deploy the SMC.

Deploy the SMC

When the NGFW Engine launch is complete, deploy the SMC.

Before you begin

Create a Forcepoint NGFW instance in AWS.

Note

If you already have existing SMC installation, it is not necessary to install an additional SMC for controlling NGFW Engines deployed in AWS.

All configuration information for the NGFW Engines is stored on the Management Server component of the SMC. The NGFW Engines continue to operate normally even when the Management Server is unreachable, so there is no interruption to any network services.

To deploy the SMC on your own hardware, you must have a computer with a 64-bit Linux operating system, such as Ubuntu 16.04 LTS. For compatible operating systems, see the *Forcepoint NGFW Security Management Center Release Notes*.

If you deploy the SMC in an instance on AWS, we recommend using the M4.xlarge instance type. If the SMC manages a large number of NGFW Engines, the M4.2xlarge or M4.4xlarge instance types might provide improved performance. Use a 64-bit Linux operating system, such as Ubuntu 16.04 LTS, and a 64-bit JRE. For compatible operating systems, see the *Forcepoint NGFW Security Management Center Release Notes*.

CAUTION

Do not deploy the SMC in the same instance as the Forcepoint NGFW Engine. Forcepoint NGFW Engine image includes a custom operating system that is dedicated to running the Forcepoint NGFW Engine. The custom operating system is not suitable for general purpose computing.

Steps

1) If you deploy the SMC in an instance on AWS, implement security groups for the instance to allow traffic only on the ports that the SMC uses.

Note

If the SMC is already behind a firewall that restricts access, it is not necessary to implement security groups for the instance in which the SMC runs.

a) To allow traffic on the necessary ports for system communication, add the following rules to the security group:

TCP ports	UDP ports	Direction	Purpose
53	53	Outbound	DNS queries
443		Outbound	HTTPS connections to the Forcepoint NGFW update service for downloading dynamic update packages, engine upgrades, and licenses
3020		Inbound	Alert sending from the Log Server and optional Web Portal Server. Log and alert messages from NGFW Engines. Monitoring of blacklists, connections, status, and statistics for NGFW Engines.
3021		Inbound	Certificate requests or certificate renewal for system communications
3023		Inbound	Status monitoring for the Log Server and the optional Web Portal Server
8914-8918		Inbound	Log browsing connections from the Management Client to the Log Server. Database replication (push) to the Log Server, log browsing on the optional Web Portal Server.

b) To allow traffic on ports for optional features, add the following rules for the optional features that you use:

TCP ports	UDP ports	Direction	Purpose
389		Outbound	External LDAP queries for display/editing users from external LDAP domains in the Management Client. This port is only needed if you store user information in external LDAP domains.
1812		Outbound	RADIUS. Only needed if you use RADIUS to authenticate administrator logons to the Management Client.
514, 5514	514, 5514	Outbound	Log data forwarding to syslog servers. Only needed if you forward data from the Log Server or Management Server to external syslog servers.
514, 5514	514, 5514	Inbound	Syslog reception from third-party components. Only needed if you have configured monitoring of third-party devices.
8082		Inbound	SMC API. Only needed if you have enabled the SMC API.
8083		Inbound	Communication from SMC Web Access clients to the optional Web Portal Server. Only needed if you use the optional Web Portal Server and have enabled SMC Web Access.
8085		Inbound	Communication from SMC Web Access clients to the Management Server. Only needed if you have enabled SMC Web Access.
8902-8913		Inbound and Outbound	Database replication from the active Management Server to additional Management Servers for high availability. Only needed if you have configured multiple Management Servers for high availability.
8931		Outbound	Connections from the Log Server to the Web Portal Server. Only needed if you have installed the optional Web Portal Server component of the SMC.
	161	Outbound	SNMP status probing to external IP addresses. Only needed if you have configured monitoring of third-party devices.
	2055	Inbound	NetFlow or IPFIX forwarding to third-party components. Only needed if you have configured monitoring of third-party devices.
	162, 5162	Inbound	SNMPv1 trap reception from third-party components. Only needed if you have configured monitoring of third-party devices.

2) On the computer or instance where you want to deploy the SMC, open a terminal program, then enter the following command to copy the SMC installation files from the NGFW Engine EC2 instance to the local computer:

```
scp -p -i <your ssh private key>.pem aws@<aws instance public ip address>:/spool/<smc
 installation files>.zip .
```

The SMC installation files are included in the NGFW Engine instance.

3) Decompress the SMC installation files using compression utilities in your operating system.
For example:

```
unzip <smc installation files>.zip
```

4) Navigate to the <smc installation files>/Forcepoint_SMC_Installer/Linux-x64 directory.

5) To start the SMC installation, enter the following command:

```
sudo ./setup.sh
```

6) Install the SMC components.
For detailed instructions, see the *Forcepoint Next Generation Firewall Installation Guide*.

Next steps

Configure the network connections and contact addresses for the SMC.

Configure the SMC

Configure the network connections and contact addresses for the SMC.

Before you begin

You must have an existing SMC installation.

These steps provide an overview of the SMC configuration process. For detailed instructions, see the following documentation:

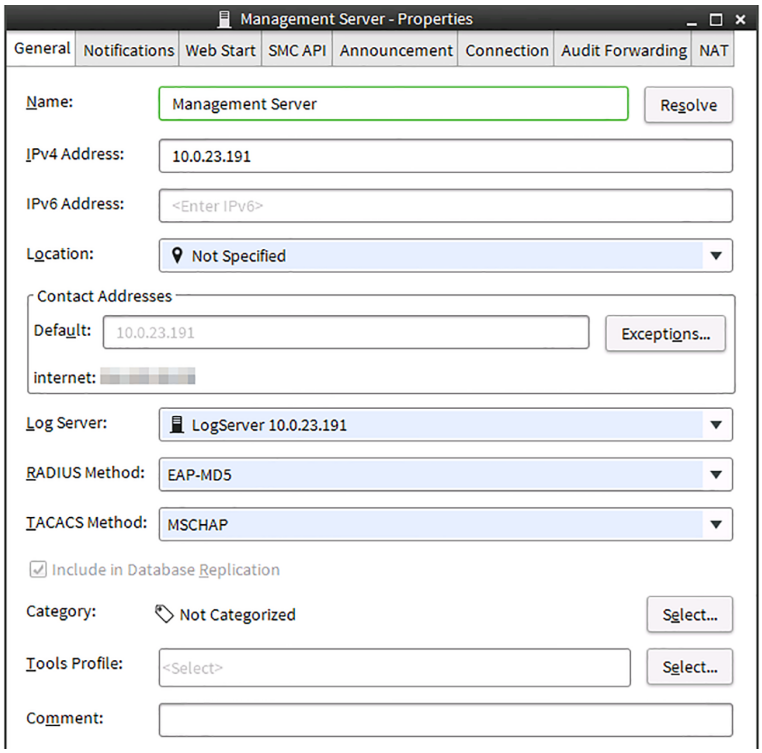
- *Forcepoint Next Generation Firewall Installation Guide*
- *Forcepoint Next Generation Firewall Product Guide*

Steps

1) In the Management Client component of the SMC, create a Location element for elements that are located in networks outside of the local network for the SMC servers.
In the example configuration, a Location element called "internet" has been created.

2) Configure contact addresses for the Management Server.

In the example configuration, the external IP address that is used to reach the SMC from AWS has been configured as the contact address for the "internet" Location.



a) In the **Management Server Properties** dialog box, click **Exceptions**.

b) Click **Add**, select the Location element that you created, then click **Select**.

c) In the **Contact Address** cell, enter the external IP address that is used to reach the SMC from AWS, then click **OK**.

d) Click **OK** to close the **Management Server Properties** dialog box.

3) Configure contact addresses for the Log Server.

a) In the **Log Server Properties** dialog box, click **Exceptions**.

b) Click **Add**, then select the Location element that you created and click **Select**.

c) In the **Contact Address** cell, enter the external IP address of the Log Server, then click **OK**.

d) Click **OK** to close the **Log Server Properties** dialog box.

Next steps

Create a Single Firewall element for each Forcepoint NGFW engine that you deploy in the AWS cloud.

Create Single Firewall elements

Create a Single Firewall element for each Forcepoint NGFW engine that you deploy in the AWS cloud.

Before you begin

Configure the network connections and contact addresses for the SMC.

These steps provide an overview of the NGFW configuration process. For detailed instructions, see the following documentation:

- *Forcepoint Next Generation Firewall Installation Guide*
- *Forcepoint Next Generation Firewall Product Guide*

Steps

1) In the Management Client component of the SMC, add a Single Firewall element.

2) From the **Location** drop-down list on the **General** pane, select the Location element for elements outside of the local network of the SMC servers.
In the example configuration, the "internet" Location element is used.

3) Add a layer 3 physical interface and configure it as the primary control interface.
   a) To add a layer 3 physical interface, select **Add** > **Layer 3 Physical Interface**.
   b) To add a dynamic IP address to the interface, select **Add** > **IPv4 Address**.
   c) From the IP address type drop-down list, select **Dynamic**.
   d) From the **Dynamic Index** drop-down list, select **First DCHP Interface**.
   e) In the **Interface Options**, select Interface ID 0 as the primary control interface.
      The **Node-Initiated Contact to Management Server** option is automatically selected when the control IP address is dynamic. When the option is selected, the engine opens a connection to the Management Server and maintains connectivity.

4) (Optional) Add more physical interfaces and IPv4 addresses according to your environment.

5) If the SMC is located outside of the VPC where the NGFW Engine is deployed, add a route to the Management Server on the **Routing** pane in one of the following ways:
   - Add a static route through Interface 0 to the IP address of the Management Server.

     Note

     The routing configuration in the SMC must be the same as the routing configuration in AWS.

   - Add a default route through Interface 0 to the Internet through Interface 0.

6) Add more routes and configure other settings according to your environment, then click **Save** to save and validate changes.

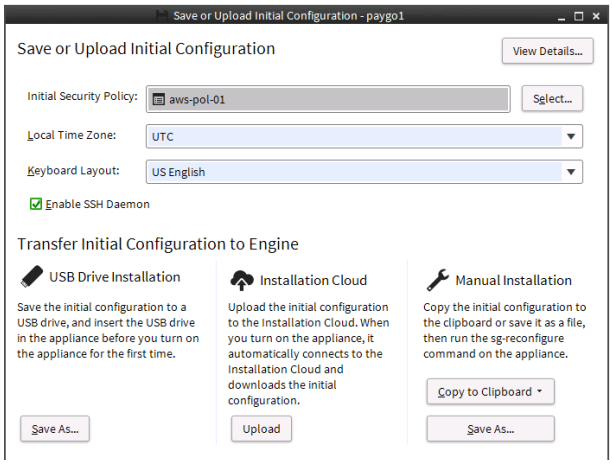
7) Install a license for the Forcepoint NGFW engine and bind the license to the Single Firewall element.

Note

When you use the Bring Your own License image, you must install a license for the engine in the SMC.

8) Save the initial configuration.

a) Right-click the engine, then select **Configuration** > **Save initial Configuration**.



b) Next to the **Initial Security Policy** field, click **Select** and select a policy for the engine.

c) Select **Enable SSH Daemon**.

d) Keep the **Save or Upload Initial Configuration** dialog box open.

This dialog box shows the one-time password that you enter when you establish contact between the NGFW Engine and the Management Server.

Next steps

Connect the NGFW Engine to the SMC.

Connect the NGFW Engine to the SMC

Establish contact between the NGFW Engine and the Management Server.

Before you begin

Create a Single Firewall element for each Forcepoint NGFW engine that you deploy in the AWS cloud.

Steps

1) On your computer, open a terminal program, then enter the following command to open an SSH connection to the command line of the NGFW Engine using the aws user account:

```
ssh -i <your ssh private key>.pem aws@<aws instance public ip address>
```

2) On the command line of the NGFW Engine, enter the following command to start the NGFW Configuration Wizard:

```
sudo sg-reconfigure
```

3) Configure the general settings and network interfaces for the NGFW Engine.
For detailed instructions, see the *Forcepoint Next Generation Firewall Installation Guide*.

4) On the **Prepare for Management Contact** page, select **DHCPv4** or **DHCPv6**.

5) Select **Contact**, then press the spacebar.

6) Enter the Management Server contact IP address and the one-time password.
You can copy and paste the one-time password from the **Save or Upload Initial Configuration** dialog box.

7) Highlight **Finish**, then press **Enter**.
The engine now tries to make initial contact with the Management Server. The progress is shown on the command line. If you see a connection refused message, make sure that the one-time password is correct and that a route to the Management Server IP address has been configured for the NGFW Engine. Save a new initial configuration if you are unsure about the password.

Note

If the initial management contact fails for any reason, you can start the configuration again with the `sg-reconfigure` command.

Result

After you see notification that Management Server contact has succeeded, the engine installation is complete and the engine is ready to receive a policy.
When the initial configuration is complete, the status of the NGFW Engine element changes in the Management Client from **Unknown** to **No Policy Installed**. The connection state is **Connected**, indicating that the Management Server can connect to the node.

Next steps

Install a policy on the engine using the Management Client.

Deploy Forcepoint NGFW in AWS when you have an existing SMC installation

If you already have an existing SMC installation, you can deploy additional NGFW Engines in AWS.

Configure the SMC

Configure the network connections and contact addresses for the SMC.

Before you begin

You must have an existing SMC installation.

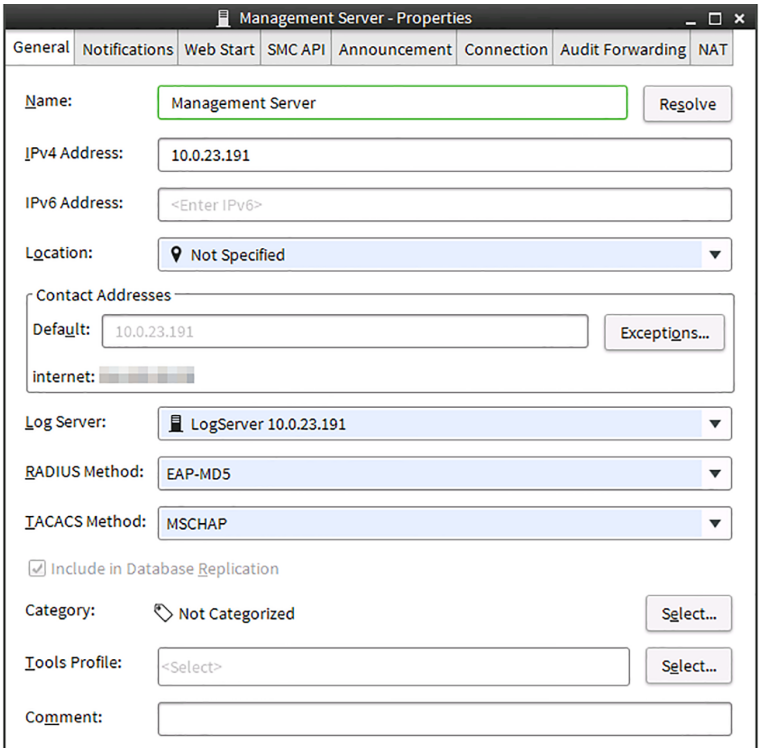
These steps provide an overview of the SMC configuration process. For detailed instructions, see the following documentation:

- *Forcepoint Next Generation Firewall Installation Guide*
- *Forcepoint Next Generation Firewall Product Guide*

Steps

1) In the Management Client component of the SMC, create a Location element for elements that are located in networks outside of the local network for the SMC servers.
In the example configuration, a Location element called "internet" has been created.

2) Configure contact addresses for the Management Server.
In the example configuration, the external IP address that is used to reach the SMC from AWS has been configured as the contact address for the "internet" Location.



a) In the **Management Server Properties** dialog box, click **Exceptions**.

b) Click **Add**, select the Location element that you created, then click **Select**.

c) In the **Contact Address** cell, enter the external IP address that is used to reach the SMC from AWS, then click **OK**.

d) Click **OK** to close the **Management Server Properties** dialog box.

3) Configure contact addresses for the Log Server.
   a) In the **Log Server Properties** dialog box, click **Exceptions**.
   b) Click **Add**, then select the Location element that you created and click **Select**.
   c) In the **Contact Address** cell, enter the external IP address of the Log Server, then click **OK**.
   d) Click **OK** to close the **Log Server Properties** dialog box.

Next steps

Create a Single Firewall element for each Forcepoint NGFW engine that you deploy in the AWS cloud.

Create Single Firewall elements

Create a Single Firewall element for each Forcepoint NGFW engine that you deploy in the AWS cloud.

Before you begin

Configure the network connections and contact addresses for the SMC.

These steps provide an overview of the NGFW configuration process. For detailed instructions, see the following documentation:

- *Forcepoint Next Generation Firewall Installation Guide*
- *Forcepoint Next Generation Firewall Product Guide*

Steps

1) In the Management Client component of the SMC, add a Single Firewall element.
2) From the **Location** drop-down list on the **General** pane, select the Location element for elements outside of the local network of the SMC servers.
   In the example configuration, the "internet" Location element is used.
3) Add a layer 3 physical interface and configure it as the primary control interface.
   a) To add a layer 3 physical interface, select **Add** > **Layer 3 Physical Interface**.
   b) To add a dynamic IP address to the interface, select **Add** > **IPv4 Address**.
   c) From the IP address type drop-down list, select **Dynamic**.
   d) From the **Dynamic Index** drop-down list, select **First DCHP Interface**.

e) In the **Interface Options**, select Interface ID 0 as the primary control interface.

The **Node-Initiated Contact to Management Server** option is automatically selected when the control IP address is dynamic. When the option is selected, the engine opens a connection to the Management Server and maintains connectivity.

4) (Optional) Add more physical interfaces and IPv4 addresses according to your environment.

5) If the SMC is located outside of the VPC where the NGFW Engine is deployed, add a route to the Management Server on the **Routing** pane in one of the following ways:

- Add a static route through Interface 0 to the IP address of the Management Server.

Note

The routing configuration in the SMC must be the same as the routing configuration in AWS.

- Add a default route through Interface 0 to the Internet through Interface 0.

6) Add more routes and configure other settings according to your environment, then click **Save** to save and validate changes.

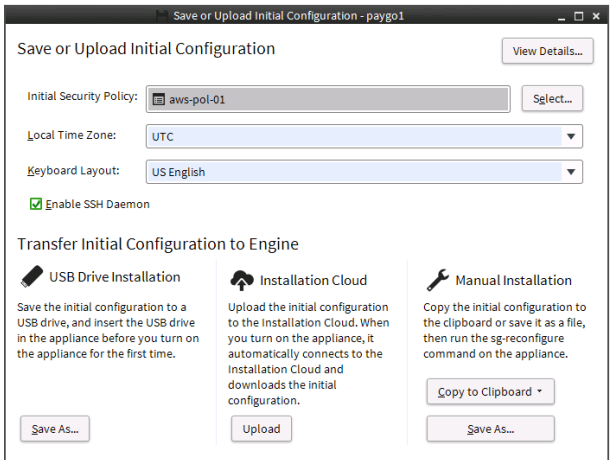
7) Install a license for the Forcepoint NGFW engine and bind the license to the Single Firewall element.

Note

When you use the Bring Your own License image, you must install a license for the engine in the SMC.

8) Save the initial configuration.

a) Right-click the engine, then select **Configuration** > **Save initial Configuration**.



b) Next to the **Initial Security Policy** field, click **Select** and select a policy for the engine.

c) Select **Enable SSH Daemon**.

d) Keep the **Save or Upload Initial Configuration** dialog box open.

This dialog box shows the one-time password that you enter when you establish contact between the NGFW Engine and the Management Server.

Next steps

Prepare the AWS environment for the NGFW deployment.

Configure the AWS environment

Prepare the AWS environment for the NGFW deployment.

These instructions use the AWS web management console. For automated and large scale deployment, we recommend using the AWS command line interface (CLI) tools or lower level programming libraries to communicate with the AWS REST API directly.

These steps provide an overview of the configuration process. For detailed instructions, see the Amazon Elastic Compute Cloud Documentation and the Amazon Virtual Private Cloud Documentation.

Steps

1) Create the virtual private clouds (VPCs) and the subnet that the NGFW Engine will be deployed in.

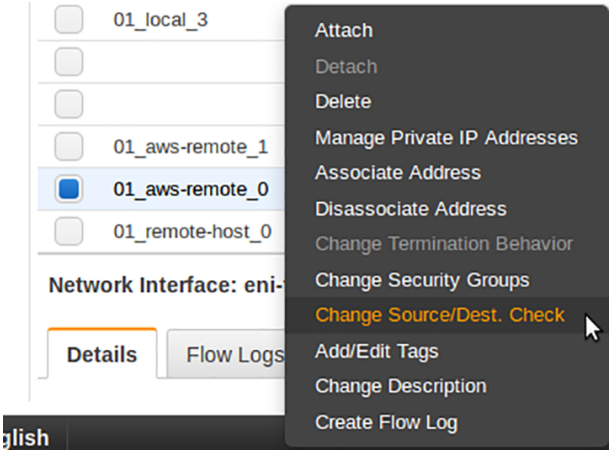
   You must deploy the NGFW Engine in a dedicated subnet.

2) In the subnet that the NGFW Engine will be deployed in, create one or more elastic network interfaces (ENIs).

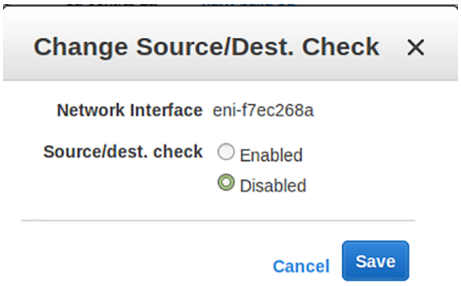
   Only one ENI is required. You can optionally create more ENIs depending on your environment. Create one ENI for each physical interface that you added to the Single Firewall element.

3) Disable the **Source/Dest. check** option for each engine interface.

The **Source/Dest. check** option prevents packet forwarding to destinations on other interfaces. When the option is enabled, the firewall cannot act as a router.



a) Right-click the ENI interface, then select **Change Source/Dest. Check**.



b) From the **Source/Dest. check** options, select **Disabled**.

c) Click **Save**.

4) Create the required gateways and routing tables and assign them to subnets.

Create a Forcepoint NGFW instance using Manual Launch

Configure and launch an instance of the Forcepoint NGFW AMI using Manual Launch.

CAUTION

If required for regulatory compliance, or in environments with stricter security requirements, we recommend using dedicated instances when you deploy Forcepoint NGFW in AWS.

We recommend using the following instance types depending on the Forcepoint NGFW product:

Forcepoint NGFW product	EC2 instance type
NGFW 2 CPU	M4.large

Forcepoint NGFW product	EC2 instance type
NGFW 4 CPU	M4.xlarge or C4.xlarge
NGFW 8 CPU	M4.2xlarge or C4.2xlarge
NGFW 16 CPU	C4.4xlarge

For information about VM size and network performance, see the Amazon documentation at https://aws.amazon.com/ec2/instance-types/. Enabling some Forcepoint NGFW features, such as inspection, might decrease the network throughput.

Forcepoint NGFW is designed to receive and manage all traffic on all ports. Use a security group that allows connections on all ports for inbound and outbound for the instance in which Forcepoint NGFW is running.

Steps

1) In the AWS Marketplace, start the launch for the Forcepoint NGFW AMI.

2) Click the **Manual Launch** tab.

3) Select an instance type that meets your performance needs.
   The AMI automatically restricts the instance types so that only compatible instance types are available.

4) Add one or more interfaces and map ENIs to the interfaces.
   a) To add an interface, click **Add Device**.

   Note

   The wizard only allows you to add two interfaces. If you need to add more interfaces, use the command line tools.

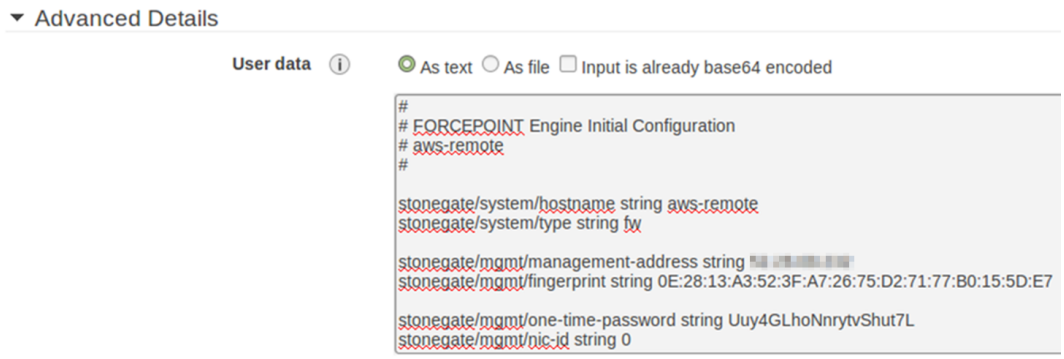
   Add all required interfaces while creating the instance. If you add interfaces later, a reboot is required before the interfaces become available.

   b) From the **Network Interface** drop-down list for eth0, select the ENI for the control interface.

   c) From the **Network Interface** drop-down list for the other interfaces, select the ENI to connect to each interface.

5) If you want to transfer the initial configuration file to the instance, add the initial configuration as user data.

We recommend transferring the engine's initial configuration as user data when you launch the Forcepoint NGFW instance. When you provide user data, the engine automatically makes initial contact with the Management Server when it starts. After it is launched, the Forcepoint NGFW instance automatically appears in the Management Client.



a) In the **User Data** options, select **As Text**.

b) In the **Save or Upload Initial Configuration** dialog box in the Management Client, click **Copy to Clipboard**.

c) In the EC2 Management Console, paste the text that you copied from the **Save or Upload Initial Configuration** dialog box into the **User Data** field.

6) Click **Review and Launch**.

7) On the **Review Instance Launch** page, select an existing key pair or create a new key pair for SSH connections to the NGFW engine.

Note

The key is the only allowed authentication method for SSH connections to the engine command line.

If the default security group is too limited for your environment, you can select a different security group or change the rules. You can also configure the NGFW Engine to restrict access.

Result

When the NGFW Engine installation is complete and the engine is ready to process traffic, the status of the NGFW Engine element changes in the Management Client to **Online**. The connection state is **Connected**, indicating that the Management Server can connect to the node.

You can also check the status of the NGFW Engine in the AWS console. To check the status, select **Actions** > **Instance Settings** > **Get system log**. The system log shows the following information:

```
Management server contact successful
Sg-auto-contact done
```

Configure HA

After you have deployed two NGFW Engines, configure high availability (HA).

Before you begin

- To use HA, the NGFW Engine must be able to resolve host names. Configure a DNS server in the Management Client component of the SMC.
- In the Management Client, add a rule to the Firewall Policy to allow HTTP connections from the NGFW Engine to the AWS API, and from the AWS API to the NGFW Engine.

For detailed instructions, see the *Forcepoint Next Generation Firewall Product Guide*.

HA requires NGFW version 6.4.4 or higher.

In an HA configuration, one NGFW instance acts as the default gateway for outbound traffic in one VPC. If the active NGFW instance becomes unavailable, the other NGFW instance becomes the default gateway.

The HA configuration consists of the following files:

- run-at-boot script — The HA script that runs on each NGFW instance. The script uses AWS API calls to enumerate the Route Tables of one or more subnets of a VPC and to change the NGFW instance that acts as the default gateway in case of a failover.
- policy.json — Example rules that you can copy and paste into the identity and access management (IAM) policy that allows the NGFW instance to access the AWS API.

Steps

1) Obtain the run-at-boot script and the policy.json file from https://github.com/Forcepoint/fp-NGFW-AWS-ha.

2) Create an IAM policy to allow the NGFW instance to access the AWS API.
   a) Open the AWS console, then select **IAM** from the **Services** drop-down list at the top of the page.
   b) From the menu on the left, select **Policies**.
   c) Click **Create Policy**.
   d) Copy the contents of the policy.json file and paste them into the web editor on the **JSON** tab.
   e) Click **Review Policy**.
   f) Enter a name and description for the policy.
   g) Click **Create Policy**.

3) Create an IAM role that uses the IAM policy that you created.
   a) In the AWS console, select **IAM** from the **Services** drop-down list at the top of the page.
   b) From the menu on the left, select **Roles**.

c) Click **Create role**.

d) In the **service that will use this role** options, select **EC2**, then click **Next**.

e) Attach the IAM policy that you created, then click **Next**.

f) Click **Review**.

g) Enter a name and description for the role, then click **Create role**.

4) Attach the IAM role to the NGFW instances in AWS.

a) In the AWS console, select **EC2** from the **Services** drop-down list at the top of the page.

b) From the menu on the left, open the **Instances** page.

c) Right-click the NGFW instances on which you want to enable HA, then select **Instance Settings** > **Attach/Replace IAM Role**.

d) From the drop-down list, select the role that you created, then click **Apply**.

5) Perform these steps on each NGFW instance:

a) On your computer, open a terminal program, then enter the following command to open an SSH connection to the command line of the NGFW Engine using the aws user account:

```
ssh -i <your ssh private key>.pem aws@<aws instance public ip address>
```

b) Create a /data/route-tables file and populate it with the “rtb-*” entries from the Route Tables to be configured with HA.

Select the route tables of the subnets that use the HA NGFW Engines as a default route. Enter each route table entry on a separate line.

Example /data/route-tables file:

```
rtb-0123456789
rtb-0123456788
rtb-0123456787
rtb-0123456786
```

c) Copy the run-at-boot script to the instance.

d) If the NGFW instance only has one interface, edit the run-at-boot script and change 1 to 0 in the following line:

```
if interface['Attachment']['DeviceIndex'] == 1
```

e) To move the run-at-boot script to the /data directory, enter the following command:

```
mv run-at-boot /data
```

f) To make the run-at-boot file executable, enter the following command:

```
chmod +x /data/run-at-boot
```

g) Edit the /data/run-at-boot file and change the region on the following line to the region that your instance is operating in:

```
ec2 = boto3.resource('ec2', region_name='<region>',api_version='2016-09-15')
```

h) To make sure that there are no errors, enter the following command to run the run-at-boot script manually:

```
python /data/run-at-boot
```

i) Reboot the NGFW Engine.

AWS Transit Gateway

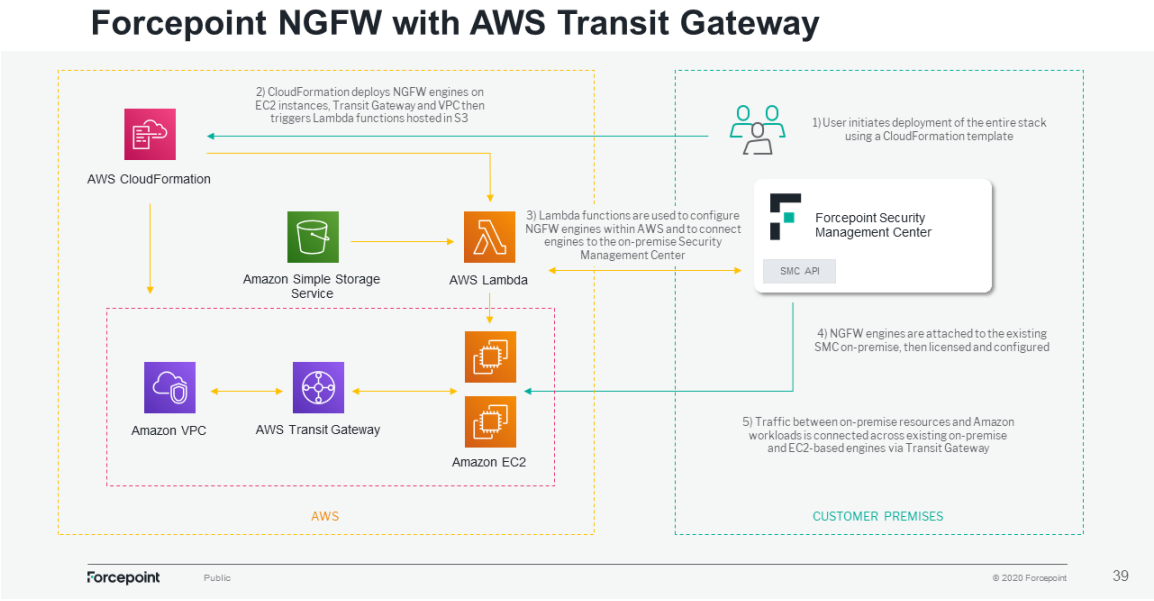
The AWS transit gateway service provides inter-connectivity across thousands of VPCs, AWS accounts, and on-premises networks. It lets you to control communications between VPCs and to connect to the on-premises networks using a single gateway.

This section provides detailed instructions on how to integrate Forcepoint Next Generation Firewall (NGFW) and AWS Transit Gateway using a CloudFormation template, which includes an auto-scaling template that connects the AWS Transit Gateway using an AWS Lambda function and configures NGFW engines in the existing Forcepoint Security Management Center (SMC). This deployment provides connectivity for on-premise traffic to networks within AWS VPCs and vice versa.

This CloudFormation template lets system administrators to automatically:

- Deploy all AWS resources necessary to setup NGFW Engines and AWS Transit Gateway.
- Connect Forcepoint NGFW engines deployed as EC2 instances from the auto-scaling template with an existing Forcepoint SMC.
- Configure and connect on-premise and EC2-based NGFW engines to bridge traffic between on-premise and AWS workloads.

The following diagram provides a description of the workflow between the components involved in this solution:



Product Compatibility

The integration described in this document is developed and tested with the following product versions:

- Forcepoint NGFW 6.9.2
- Forcepoint SMC 6.9.2

This interoperability uses:

- **AWS CloudFormation:** for modeling and provisioning AWS and third-party application resources in your cloud environment.
- **AWS Transit Gateway**: connects VPCs and on-premises networks through a central hub.
- **AWS Lambda:** to run code without provisioning or managing servers.
- **AWS S3:** an object storage service that offers industry-leading scalability, data availability, security, and performance.
- **AWS EC2:** a web service that provides secure, re-sizable compute capacity in the cloud.
- **Amazon EventBridge:** a serverless event bus that makes it easier to build event-driven applications.
- **Auto Scaling groups** an auto-scaling group contains a collection of Amazon EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and management.

Implementation

This implementation requires the following resources:

- `fp-ngfw-aws-TransitGateway-autoscaling.zip` available at this link: https://frcpnt.com/fp-ngfw-aws-transitgateway-latest.
- A single VPC, which is created during the CloudFormation deployment workflow.
- Multiple Elastic IPs in AWS (based on number of engines running).

Note: The standard limit for each AWS region is 5 VPCs and 5 Elastic IPs (EIP), therefore the region chosen for the deployment must allow to create a new VPC and new EIPs (you can contact AWS support to check if the limit of creating new VPC and EIPs can be increased.).

This implementation has been tested working with the following requirements:

- Ubuntu 20.04.1 LTS (with at least 2 GB RAM and 20 GB free disk space).
- Python 3.8
- The following Python modules:
  - `fp-NGFW-SMC-python`
  - `crhelper`
  - `xmltodict`
  - `boto3`

Networking requirements

The CloudFormation template performs deployment and configuration tasks that involve network traffic between the existing SMC and NGFW engines, which are installed on AWS as EC2 instances. Therefore, network traffic to/from AWS and on-premise locations must be allowed accordingly.

For more information on how to identify the necessary ports and protocols needed to allow SMC API, NGFW engines and IPSEC network traffic, see *Default communication ports* in the *Next Generation Firewall Product Guide*.

Configure Forcepoint SMC

Forcepoint SMC must be reachable from AWS components that provision the necessary configuration between the NGFW engines hosted in AWS and the existing SMC using the SMC API. If SMC is not already reachable from outside the private company network, do as follows:

1) Sign into the **SMC**.

2) Navigate to **Configuration** > **Administration**.

3) Expand **Access Rights** and then select **API Clients**.

4) Right-click **API Clients** and select **New API Client**. The **API Client Properties** screen is displayed.

5) Add a name in the **Name** field, and then click **Generate Authentication Key**. You can save the authentication key in your local drive for future reference.

6) Select the **Permissions** tab.

7) Select **Unrestricted Permissions (Superuser)** option.

8) Click **OK**.

9) From the left navigate panel navigate to **Certificates**, and then select **TLS Credentials**.

10) Right-click **TLS Credentials** and select **New TLS Credentials**. Perform the following:

   a) Type a name for the certificate.

   b) Type the publicly accessible IP address into the **Common Name [CN]** field. Rest of the fields must have existing default values.

   c) Click **Next**.

11) Select the **Self-Sign** option, and then click **Finish**.

12) Right-click the newly created Credential and select **Properties**.

13) From the **Certificate properties** window, select the **Certificate** tab, then copy the entire content **including** the lines:**—BEGIN CERTIFICATE—** and **—END CERTIFICATE—**.

14) Save the certificate in your local drive for future reference.

15) Click **OK**.

16) Close the **Certificate** window.

17) From the left navigation pane, select **Other Elements** and right-click **Locations**.

18) Select **New Location**. The **Location Properties** window is displayed.

19) Type "cloud" in the **Name** field using only lower-case characters

20) Click **OK**.

21) In the SMC header select **Home**.

22) From the left navigation menu, select **Others**. Right-click **Management Server** and select **Properties**.

23) Click **General** tab and then select **Exceptions**.

24) Click **Add** and browse the location "cloud" created in step 19. Select "cloud" and enter the public IP of the SMC into the **Contact Addresses** section.

25) Click **OK**.

26) Navigate to the **SMC API** tab and select **Enable**.

27) From the **Server Credentials** section click the option **Select**.

28) From the **Select Element** windows select the **TLS Credentials** that has been created already.

29) From the **Server TLS Cryptography Suite Set** section, click the option **Select**.

30) From the **Select Element** window, select the option `NIST(SP 800-52 Rev.2) Compatible TLS Cryptographic Algorithms`.

31) Click **Select** and then **OK** in the **Management Server-Properties** window when finished.

32) Click **Yes**.

33) Navigate to the **Home** tab of the SMC.

34) Right-click **Log Server** and select **Properties**.

35) Add an exception same as done in step 23.

Provision AWS S3 bucket for Lambda code

`CloudFormation` template deploys AWS Lambda functions code, which is stored in a folder inside an S3 bucket. You can either use an existing bucket or a new one can be provisioned.

1) Search for **S3** in the AWS console.

2) Once you get the search result, select **S3** from the drop-down list.

3) On the **S3** page, select **Create new Bucket** (or use an existing bucket if you have already created one).

4) Create a folder named **Lambda-Functions** (case specific) either in the newly created bucket, or in the existing one.

5) Within the **Lambda-Functions** folder, create another folder named `config-smc` (case specific).

You need to upload the code for AWS Lambda function in this location. Save the name of the bucket in your local drive for future reference.

Generate key pairs and identify AMI

The Amazon Machine Image (AMI) ID is required to deploy the NGFW engines within an AWS region. Both the AMI ID and the AWS Region name are used in the configuration file for this integration.

1) Using the **AWS console** search for **EC2**.

2) Once you get the search result, select **EC2** from the drop-down list.

3) From the left navigation pane in the **Network & Security** section, select **Key Pairs**.

4) Select **Create key pair** in the top right.

5) Do the following on the **Create key pair** screen:

a) In the Name field, type "ngfw-tgw-keypair" (all lower case).

b) Select file format as **pem** and then click **Create Key Pair**.

This re-directs to the page where you created the key and automatically downloads the keypair file.

6) Save this file in your local drive as it will be needed to access the EC2 instances deployed as part of this integration.

7) Once the key pair is created, select **Instances** from the left navigation pane.

8) Click **Launch Instance** and select an **Amazon Machine Image (AMI)**.

9) In the AMI wizard search for **Forcepoint NGFW** and select the **AWS Marketplace** tab on the left navigation area.

10) Click **Previous versions** link in the **Forcepoint NGFW (BYOL) - Next Generation Firewall** option.

11) On the next page select **Continue to Configuration**.

12) On the next page select the region you want to use, everything else can be left as default. The **AMI ID** will appear below the region drop-down menu.

13) Save this value in a safe location for future reference.

Unpack and configure SMC Connector

1) Download the latest version of **fp-ngfw-aws-TransitGateway-autoscaling.zip** available at this link: https://github.com/Forcepoint/fp-bd-aws-transitgateway-ngfw/releases/latest to a directory on your Linux machine and unzip it.

2) Open **config.json** and **smc.pem** using a text editor and add the necessary values to each field. For the **smc.pem** file, refer to **Configure Forcepoint SMC**.

Note: Description of each field with examples is provided in the Configuration File on page 31 and Pem files on page 30.

Pem files

The following two **.pem** files are mentioned in this section:

PEM files	Description
YOUR_AWS_KEY_PAIR.pem	Specifies the key generated by AWS when the key pair is created. This key is only required to SSH into the EC2 instances.

PEM files	Description
Smc.pem	Specifies the file included in the **fp-ngfw-aws-TransitGateway-v1.** This file will be populated with the certificate created in the SMC. For more information, see **Configure Forcepoint SMC.**

Configuration File

This table provides a description for the values required in the configuration file.

Field	Example	Description	Requires to be changed
url	https://13.25.14.2:8082	The public endpoint of the SMC, used for accessing the SMC API from Internet.	Yes
api_key	abcdefgh1234567	API key required to use the SMC API.	Yes
api_version	6.8	Version of the SMC API to be used. Default is 6.8.	No
region	ap-south-1	This is the region of AWS required to deploy the CloudFormation template.	Yes
availability_zone_1	ap-south-1a	This is the first availability zone of the AWS region required to deploy the CloudFormation template.	Yes
availability_zone_2	ap-south-1b	This is the second availability zone of the AWS region required to deploy the CloudFormation template.	Yes
ngfw_ami	ami-021207f5865d6b9a9	AMI ID of the NGFW EC2 instance required.	Yes
lambda_bucket_name	smc-lambda-bucket	Name of the bucket used to host the AWS Lambda code that will be deployed by the CloudFormation template.	Yes

Check SMC API connectivity

The following steps provide information on how to check SMC API connectivity:

1) In the folder where **fp-ngfw-aws-TransitGateway-autoscaling.zip** was unpacked, run the following command:

```
chmod +x ApiTest
```

2) Check whether **config.json** and **smc.pem** are configured correctly.

3) Run the **ApiTest** with the following command:

```
./ApiTest
```

4) The following message is displayed if the SMC API is reached successfully:

```
Your API Client: 'smc-api-client' can be reached
```

Deploy Lambda code and CloudFormation Template

The code for AWS Lambda functions is packed in advanced before it is made available to the AWS Lambda.

1) On the Linux machine, open a terminal window where the file **fp-ngfw-aws-TransitGateway-autoscaling.zip** was unzipped.

2) Navigate to the unzipped directory

3) Run the script named **package.sh**. This creates the following two files:
   - **autoscale-tg-ngfw.json** - this file is uploaded to the **CloudFormation**.
   - **myDeploymentPackage.zip** - this archive file is uploaded to the S3 bucket.

4) Navigate back to the AWS console, and then navigate to the S3 bucket that will be used to store the archive.

5) In the **config-smc** folder upload **myDeploymentPackage.zip**.

6) Now search for **CloudFormation** in the AWS console. Navigate to **CloudFormation**.

7) The displayed console might look different if you already have a stack created in the region. Using the drop-down menu in the top right of the page, select the region you want to deploy to (same as the one used inside the configuration file) using.

8) Select **Create Stack** > **With new resources(standard)**.

9) In the **Specify template** section, select **Upload a template file**

10) Select **Choose file**.

11) Upload the **autoscale-tg-ngfw.json** template file created in step 1.

12) Select **Next**.

13) Enter a name for the stack, and then click **Next.**

14) Scroll to the bottom of the **Configure stack options** page, and then select **Next**.

15) On the **Review NGFW-TransitGateway,** scroll to the bottom of the screen.

16) Select the box to allow the necessary requirements.

17) Click **Create stack** to proceed.

Note: To prevent unexpected failures in the deployment workflow, the AWS Security Groups are configured in a permissive way allowing both inbound and outbound traffic. This must be changed once deployment is completed, allowing only traffic from intended sources. Outbound traffic must be controlled as well based on existing security policies within the organization.

Configuring Auto-scaling group

1) From the AWS webpage search for **EC2**.

2) From the left navigation pane, select **Auto Scaling Groups**.

3) Select the group name that looks similar to "xxxxx-NGFWAutoscalingGroup-xxxxx".

4) On the group details screen, select the **Edit** button.

5) Increase the desired capacity, minimum and maximum capacity as required, and then click **Update** when finished.

Optional, but recommended

The auto-scaling group is deployed without a dynamic scaling policy to let the user the flexibility to customize this. They can be configured in a dew options like Average CPU, network in, and network out.

Once the deployment is completed, NGFW engines listed inside the SMC web interface are displayed in green color within 5 minutes.

Refer to Engine states on page 34 for more information on the different status colors visible in the SMC UI during the configuration of the NGFW engines.

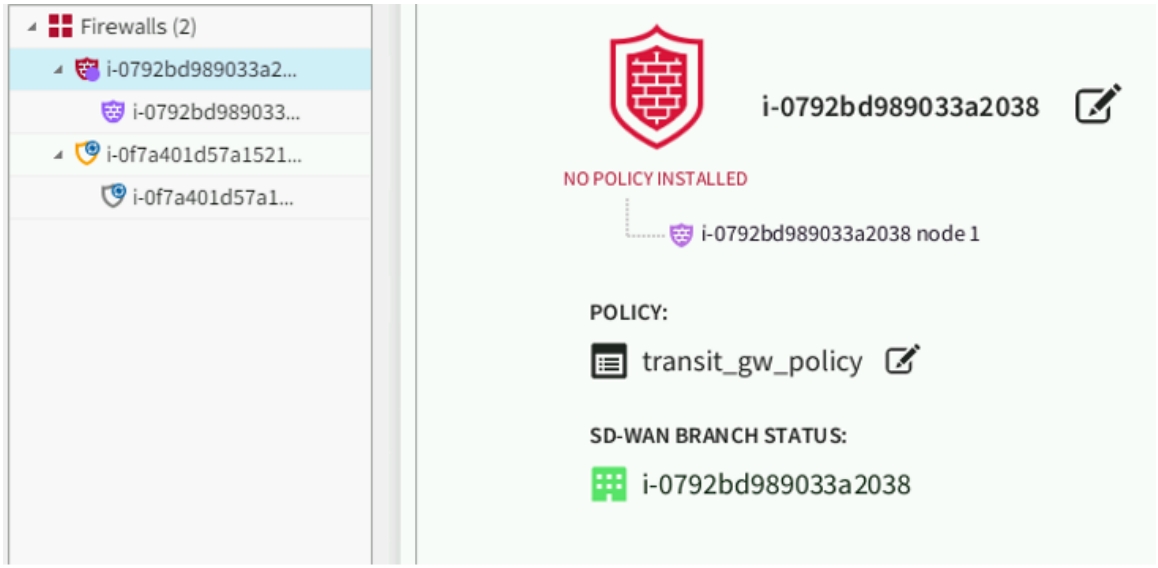
Engine states

NGFW engines deployed in AWS are displayed on Forcepoint SMC, once the CloudFormation stack has been created in full. During the operations of the CloudFormation workflow, engines status will change as the configuration and setup process progresses.

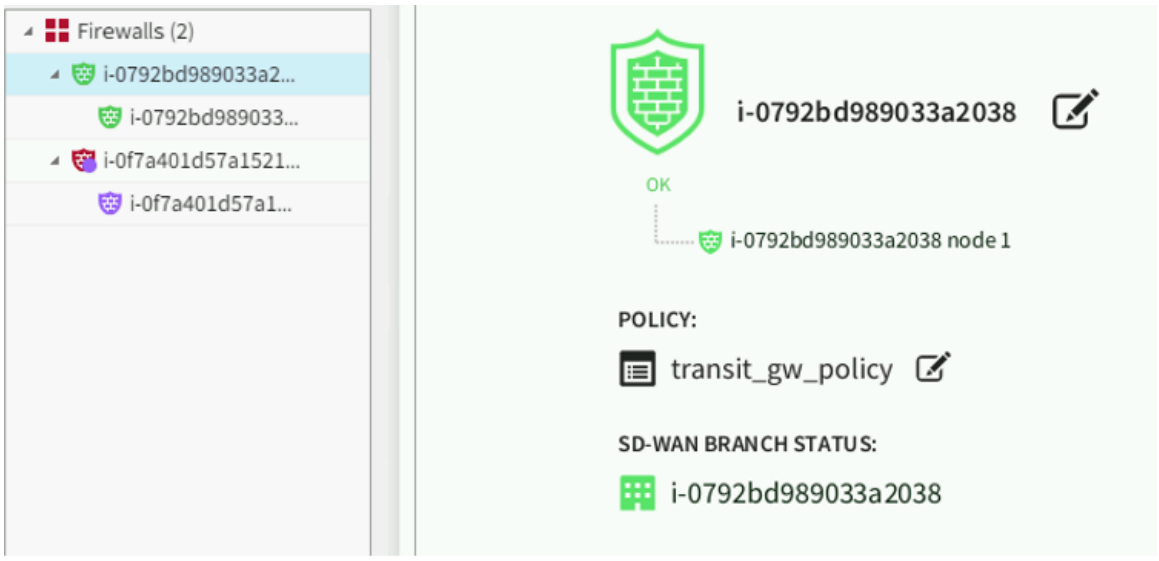
Here is a list of the possible engine statuses, with explanation.

- **Engines initial creation:** NGFW engines deployed as EC2 instances have been created and have contacted the SMC.
- **Engines waiting for configuration and policy upload:** NGFW engines are waiting for policy upload after initial contact is completed. At this point, the AWS Lambda is not triggered yet.
- **Engines have been configured and policy is being uploaded:** At this point, the AWS Lambda has been triggered, and policy upload has started.
- **Policy has been uploaded:** After the policy is uploaded, it initially looks like it has not succeeded. This state is expected temporarily and will change to normal operating status.

The preceding image shows the first NGFW engine in an error state (even though the policy has been uploaded) while the second engine is still uploading.



The following image shows the first NGFW engine has completed uploading the policy and is now in normal working status, while the second NGFW engine has now finished receiving the policy and appears in an error state. This state will change to green shortly after, and both engines will display no error.



- **Engines are connected and traffic flows without problems:** Both NGFW engines are connected and there are no errors.

Troubleshooting

Follow these steps to identify issues impacting the normal operation of the integration:

- Validate the prerequisites on page 35
- Check network connectivity on page 36
- Check dependencies are installed on page 36
- Check all components are configured and running properly on page 36

Validate the prerequisites

Make sure the following prerequisites are met:

- The compatible versions of Forcepoint NGFW and Forcepoint SMC are:
  - Forcepoint NGFW 6.9.2
  - Forcepoint SMC 6.9.2
- Verify the integration is operating on an Ubuntu version 20.04.1 machine with at least 2 GB RAM and 20 GB free disk space.
- Verify necessary ports are open on the SMC machine to allow SMC API, NGFW engines, and IPSec network traffic.
- Make sure the user selects the same AWS region for the following steps:
  - Generating the keypair **ngfw-tgw-keypair**.
  - AMI ID for **Forcepoint NGFW (BYOL) – Next Generation Firewall**.
  - Creating stack on AWS CloudFormation.
- Check the user has downloaded the necessary files from the following location: https://github.com/Forcepoint/fp-bd-aws-transitgateway-ngfw/releases/latest

Check network connectivity

Make sure firewalls or other security appliances are not impacting the network connectivity necessary for the operation of all components involved in this integration. Run the following command on the host machine to check whether the host Ubuntu machine has internet connectivity:

```
ping -c 2 www.aws.com
```

The result must be similar to the following sample:

```
PING www.aws.com (10.10.120.12) 56(84) bytes of data.
64 bytes from 10.10.120.12 (10.10.120.12): icmp_seq=1 ttl=128 time=179 ms
64 bytes from 10.10.120.12 (10.10.120.12): icmp_seq=1 ttl=128 time=181 ms
```

Check dependencies are installed

Make sure the software dependencies needed by the components involved in this integration are installed. Run the following command on the host machine to check python3.8 is installed:

```
python3 –version
```

The output must look like the following:

```
Python 3.8.2
```

Run the following command on the host machine to check pip3 is installed:

```
pip3 –version
```

The output must look similar to the following:

```
pip 20.0.2 from /usr/lib/python3/dist-packages/pip (python 3.8)
```

Check all components are configured and running properly

Make sure the products and services are configured as expected and are running. To check the AWS CloudFormation stack gets created properly, make sure the user sees a **CREATE_COMPLETE** status for the newly created stack.

To check the NGFW engines are installed properly, make sure the NGFW engines are displayed in the SMC UI with green color within 10 minutes once the stack is created successfully.

Managing Forcepoint NGFW Engines using the SSM Agent

You can use the AWS Systems Manager Agent (SSM Agent) to manage Forcepoint NGFW Engines that are deployed in the AWS cloud using the same AWS tools that are used for other AWS resources.

The SSM Agent allows you to:

- Run commands remotely on Forcepoint NGFW Engines.
- Open interactive command line sessions on Forcepoint NGFW Engines.

To use the SSM Agent, the Forcepoint NGFW Engine instance must have an IAM role that allows administration using the SSM Agent, and your AWS account must have permissions to use the SSM Agent.

For more information about the Amazon Systems Manager, see https://docs.aws.amazon.com/systems-manager/.

Create an IAM role for administration using the SSM Agent

Create an IAM role that allows administration using the SSM Agent.

Steps

1) In the AWS console, select **IAM** from the **Services** drop-down list at the top of the page.
2) From the menu on the left, select **Roles**.
3) Click **Create role**.
4) From the **Select type of trusted entity** options, select **AWS service**.
5) From the **Choose a use case** options, select **EC2**, then click **Next: Permissions**.
6) On the Permissions tab, attach one or more policies that allow the use of SSM, then click **Next: Tags**.
   Recommended policies include the following:
   - AmazonSSMFullAccess — Allows interactive sessions and running commands remotely.
   - AmazonSSMAutomationRole — Allows running commands remotely.
7) On the Tags tab, click **Next: Review**.
8) In the **Role name** field, enter a unique name for the IAM role, then click **Create role**.
9) Attach the IAM role to the NGFW instances in AWS.

Run commands on Forcepoint NGFW Engines remotely using the SSM Agent

The SSM Agent allows you to remotely run commands on Forcepoint NGFW Engines that are deployed in the AWS cloud.

You can use the SSM Agent to run the same command on multiple Forcepoint NGFW Engines at the same time, rather than separately connecting to each Forcepoint NGFW Engine and running the command.

Steps

1) Open the AWS Systems Manager console.
2) In the navigation pane, select **Run Command**.
3) Select **Run Command**.
4) In the **Command document** list, select a Systems Manager document, such as AWS-RunShellScript.
5) In the **Command parameters** section, specify values for required parameters.
6) In the **Targets** section, specify the instances on which you want to run the command.
7) Click **Run**.

Open interactive command line sessions on Forcepoint NGFW Engines using the SSM Agent

You can use the AWS SSM Agent to connect to the command line of individual Forcepoint NGFW Engines that are deployed in the AWS cloud using the predefined ssm-user account.

Steps

1) Open the AWS Systems Manager console.
2) In the navigation pane, **Instances & Nodes** > **Managed Instances**.
3) Select the instance to which you want to connect, then select **Actions** > **Start Session**.

Maintenance

All configuration information for the NGFW Engines is stored on the Management Server component of the SMC. After deployment, you can manage NGFW Engines in the AWS cloud using the Management Client component of the SMC in the same way as other NGFW Engines.

Upgrading Forcepoint NGFW Engines

You can remotely upgrade Forcepoint NGFW Engines deployed in the AWS cloud using the Management Client component of the SMC.

For information about supported Forcepoint NGFW versions, see Knowledge Base article 10156.

The upgrade package is imported to the Management Server manually or automatically. Upgrade package digests are calculated using an SHA-512 hash and signed with an ECDSA key.

Before the import, the Management Server verifies the digital signature of the upgrade package using a valid Trusted Update Certificate. The signature must be valid for the import to succeed. Verification might fail for the following reasons:

- The SMC version is out of date. Upgrade the SMC before upgrading the engines.
- A signature is invalid or missing in the upgrade files. Obtain an official upgrade package.

After the upgrade package has been imported, you can apply it to selected engines through the Management Client. Before the upgrade is installed on the engines, the Management Server again verifies the digital signature of the upgrade package. The engines also verify the digital signature of the upgrade package before the upgrade is installed.

The engines have two alternative partitions for the software. When you install a new software version, it is installed on the inactive partition and the current version is preserved. This configuration allows rollback to the previous version in case there are problems with the upgrade. If the engine is not able to return to operation after the upgrade, it automatically changes back to the previous software version at the next restart. You can also change the active partition manually.

Upgrade NGFW Engines remotely

The Management Server can remotely upgrade NGFW Engine components that it manages. You can upgrade several NGFW Engines of the same type in the same operation.

Before you begin

Read the Release Notes for the new version, especially the required SMC version and any other version-specific upgrade issues that might be listed. To access the release notes, select **Configuration**, then browse to **Administration** > **Other Elements** > **Engine Upgrades**. Select the type of NGFW Engine you are upgrading. A link to the release notes is included in the upgrade file's information. If the Management Server has no Internet connectivity, you can find the release notes at https://support.forcepoint.com/Documentation.

Steps

For more details about the product and how to configure features, click **Help** or press **F1**.

1) In the Management Client component of the SMC, select **Home**.

2) Right-click the NGFW Engine that you want to upgrade, then select **Commands** > **Go Offline**.

3) When prompted to confirm that you want to set the node offline, click **Yes**.
   The node goes offline shortly.

4) When the node is offline, right-click the node, then select **Configuration** > **Upgrade Software**.

5) From the **Operation** drop-down list, select the type of operation that you want to perform:
   - Select **Remote Upgrade (transfer + activate)** to install the new software and reboot the node with the new version of the software.
   - Select **Remote Upgrade (transfer)** to install the new software on the node without an immediate reboot and activation. The node continues to operate with the currently installed version until you choose to activate the new version.
   - Select **Remote Upgrade (activate)** to reboot the node and activate the new version of the software that was installed earlier.

6) If necessary, add or remove NGFW Engines in the **Target** list.
   All NGFW Engines in the same Upgrade Task must be of the same type.

7) Click **Select** next to the **Engine Upgrade** field, select the upgrade file, then click **OK**.
   If you choose to activate the new configuration, you are prompted to acknowledge a warning that the node will be rebooted. A new tab opens showing the progress of the upgrade. The time the upgrade takes varies depending on the performance of your system and the network environment. The NGFW Engine is automatically rebooted and brought back online.

   The upgrade overwrites the inactive partition and then changes the active partition. To undo the upgrade, use the `sg-toggle-active` command or the NGFW Engine's boot menu to change back to the previous software version on the other partition. This change can also happen automatically at the next reboot if the NGFW Engine is not able to successfully return to operation when it boots up after the upgrade.

Back up system configurations

All configuration information for the NGFW Engines is stored on the Management Server component of the SMC. Backups are needed to recover from the loss of the system configurations, for example, due to hardware failure.

The Management Server is the only component that contains usable, complete configuration information for any individual engine component. The engines contain a working copy of the configuration details that allows them to carry out traffic inspection independently. It is not possible to extract this information from the engines if the Management Server is lost. For this reason, regular Management Server backups are essential and must be stored in a safe storage location outside of the computer where the SMC servers are installed.

Always take the backups using the proprietary backup tools in the Management Client, on the Management Server command line, or on the SMC Appliance command line. Third-party backup applications that back up the host system might not produce usable backups of your SMC servers, especially if the SMC servers are running when you take the backup.

Different types of backups contain different information:

- The Management Server backup contains the policies, elements, and other configuration details for all NGFW Engines that they manage. The Management Server backup also contains the configuration information of the Web Portal Server and of the Management Server itself.
- The Log Server backup contains the Log Server's local configuration and optionally the logs.

Note

To back up a Management Server, there must be enough free disk space on the server. Twice the size of the management database is required. If there is not enough available disk space, the backup process does not start.

Steps

For more details about the product and how to configure features, click **Help** or press **F1**.

1) In the Management Client component of the SMC, select **Home**.
2) Right-click the Management Server or Log Server you want to back up, then select **Backup**.
3) (Optional) To back up other servers, select the servers from the list on the left, then click **Add**.
4) (Optional) To encrypt the backup, select **Encrypted**, then enter and confirm a password.
   We recommend this option if the configuration contains TLS Credentials and Client Protection Certificate Authority elements.
5) (Optional) If you are creating a backup of Log Servers and you want to back up the log files, select **Back up Log Files**.
6) Click **OK**.
   The backup starts and the progress is shown on a new tab.

Next steps

Copy the backup files to a storage location.

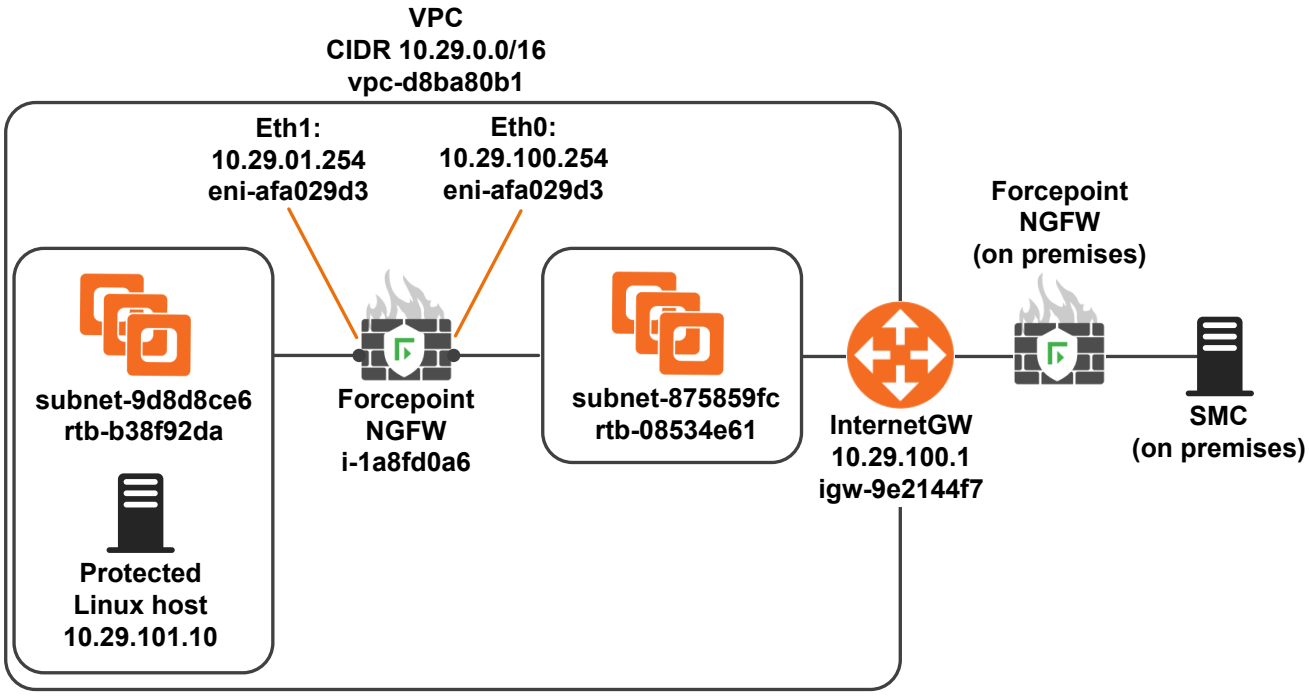
Troubleshooting in the AWS console

You can use diagnostics information provided by the AWS console for troubleshooting.

If the SSH service for the engine does not start automatically, use the **Actions** > **Instance Settings** > **Get system log** option to get diagnostics information.

Example deployment

This example shows a deployment in an example network environment.



Note

All configuration values shown in this document are examples. Your configuration might be different depending on your environment.

Begin the example deployment by preparing the VPC in which you deploy the NGFW instances.

Preparing your VPC for the example deployment

A *virtual private cloud* (VPC) is the virtual network in which you deploy Amazon EC2 instances.

The first four IP addresses and the last IP address in each subnet CIDR block are reserved. You cannot assign these IP addresses to an instance.

For example, in a subnet with CIDR block 10.29.100.0/24, the following five IP addresses are reserved:

- 10.29.100.0: Network address.
- 10.29.100.1: Reserved by AWS for the VPC router.
- 10.29.100.2: Reserved by AWS for mapping to the Amazon-provided DNS.
- 10.29.100.3: Reserved by AWS for future use.
- 10.29.100.255: Network broadcast address.

Note

AWS does not support broadcast in a VPC.

Begin by creating the VPC in which you deploy the NGFW instances.

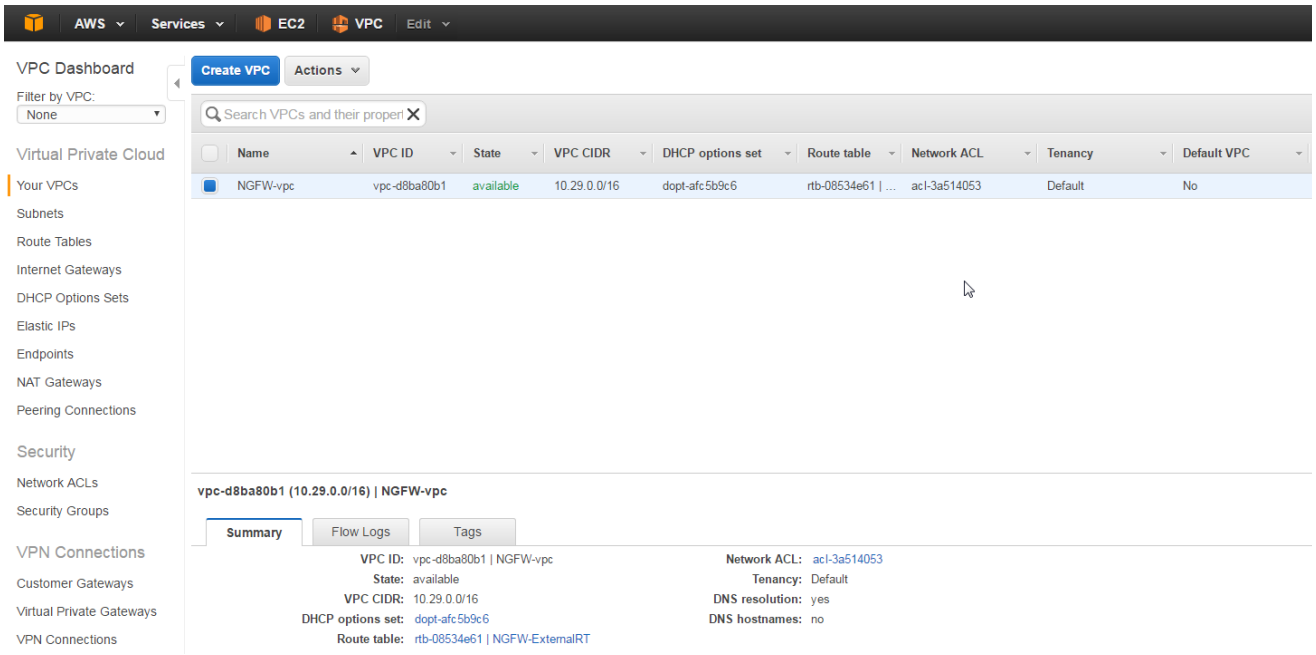
Create a VPC for the example deployment

Create the VPC in which you deploy the NGFW instances.

Steps

1) Select **VPC** > **Your VPCs** > **Create VPC**.

2) Create a test network VPC 10.29.0.0/16 (vpc-d8ba80b1).



Next steps

Create subnets.

Create subnets for the example deployment

After creating a VPC, create subnets.

Before you begin

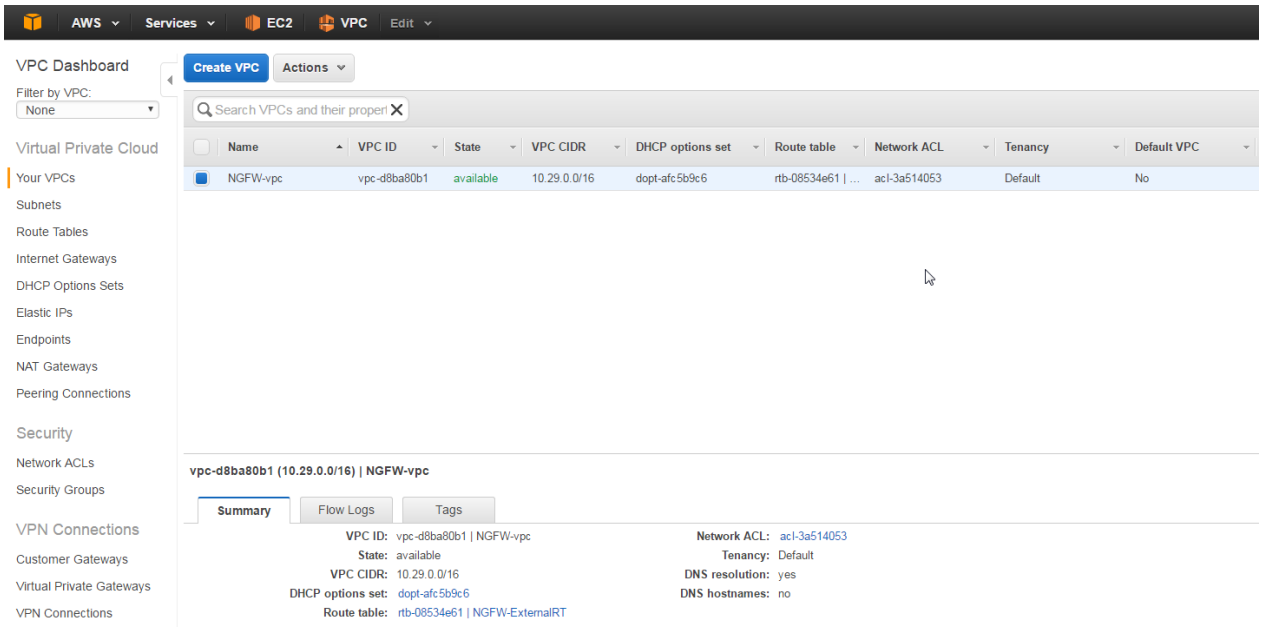
Create the VPC in which you deploy the NGFW instances.

When you create a subnet, you specify the CIDR block for the subnet. The CIDR block for the subnet is a subset of the VPC CIDR block.

Steps

1) Select **VPC** > **Subnets** > **Create Subnet**.

2) Create the following subnets:
   - External LAN 10.29.100.0/24 (subnet-875859fc)
   - Internal LAN 10.29.101.0/24 (subnet-9d8d8ce6)



Next steps

Associate route tables with subnets.

Associate route tables with subnets for the example deployment

The test environment has route tables for internal and external connections.

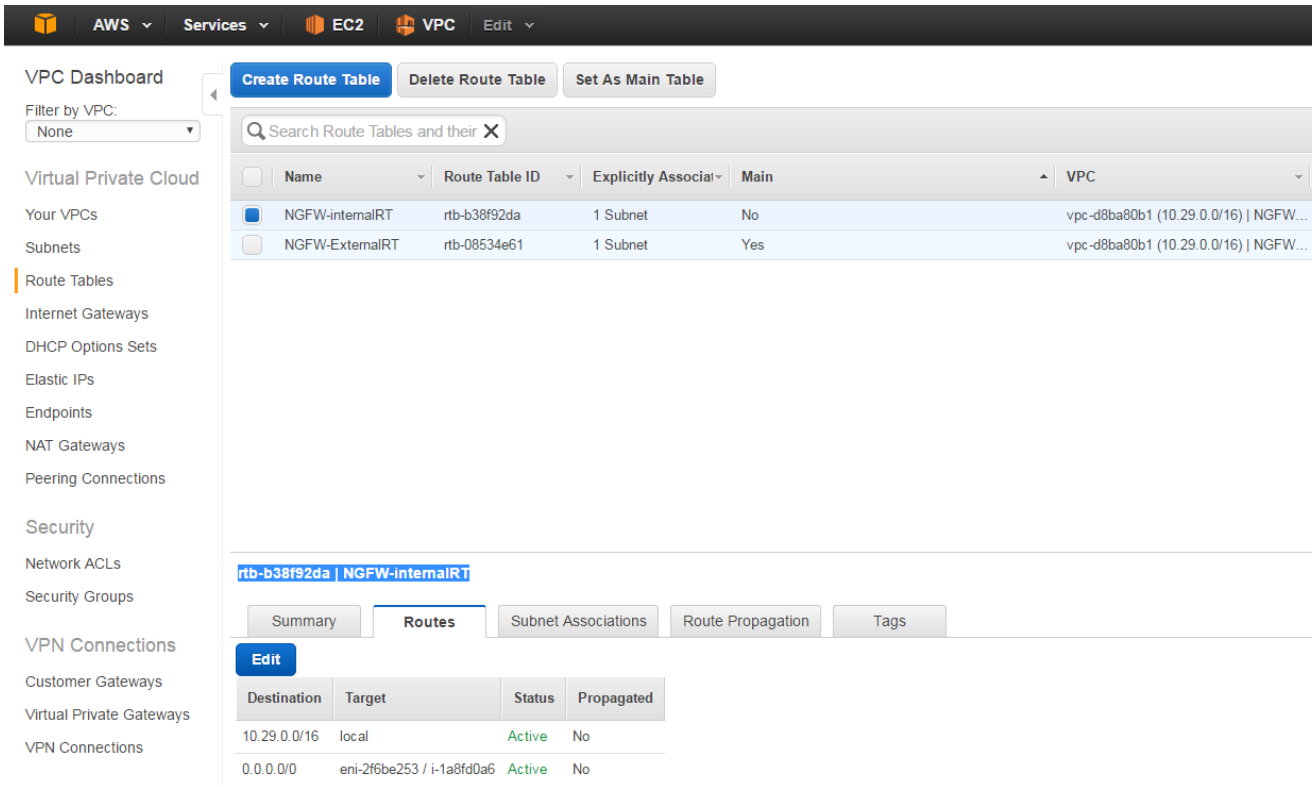
Before you begin

Create subnets.

Steps

1) Associate the **NGFW-ExternalRT** route table with the External LAN (10.29.100.0/24) subnet.
   This route table has a default route to the InternetGW (igw-9e2144f7).

2) Associate the **NGFW-internalRT** with the Internal LAN (10.29.101.0/24) subnet.
   This route table has a default route to the NGFW internal interface (10.29.101.254 / eni-2f6be253).



Next steps

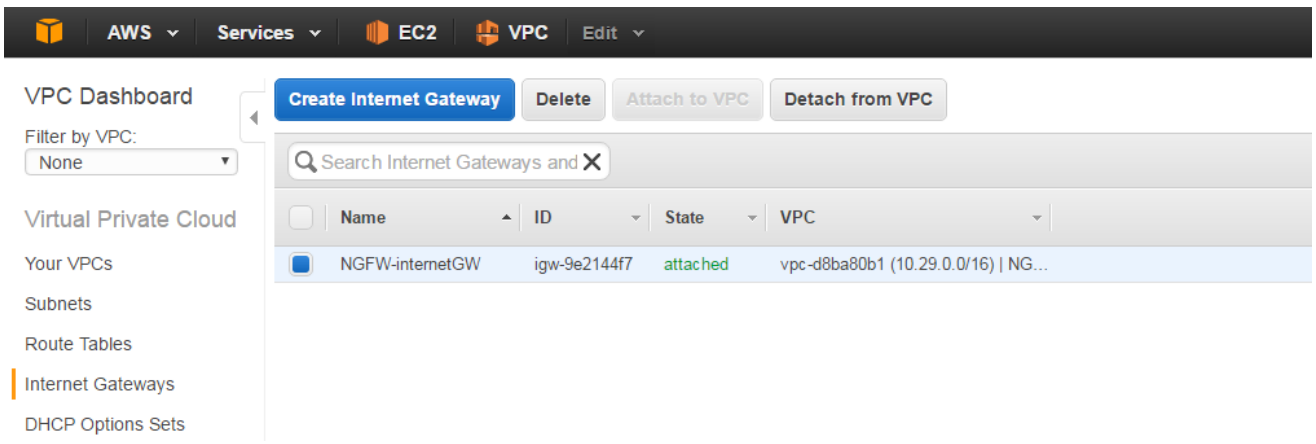
Attach an Internet gateway to your VPC.

Attach an Internet gateway to your VPC for the example deployment

To ensure that your instances can communicate with the Internet, you must also attach an Internet gateway to your VPC.

Before you begin

Associate route tables with subnets.



Next steps

Define a network ACL and a security group.

Define a network ACL and a security group for the example deployment

A *network access control list* (ACL) filters incoming and outgoing traffic for one or more subnets. A *security group* filters incoming and outgoing traffic for one or more instances.

Before you begin

Attach an Internet gateway to your VPC.

When you launch an instance, you associate one or more security groups with the instance. You add rules to each security group that allow traffic to or from the instance. In the example configuration, the NGFW Engine provides access control, and the ACL for the AWS network allows all traffic.

Note

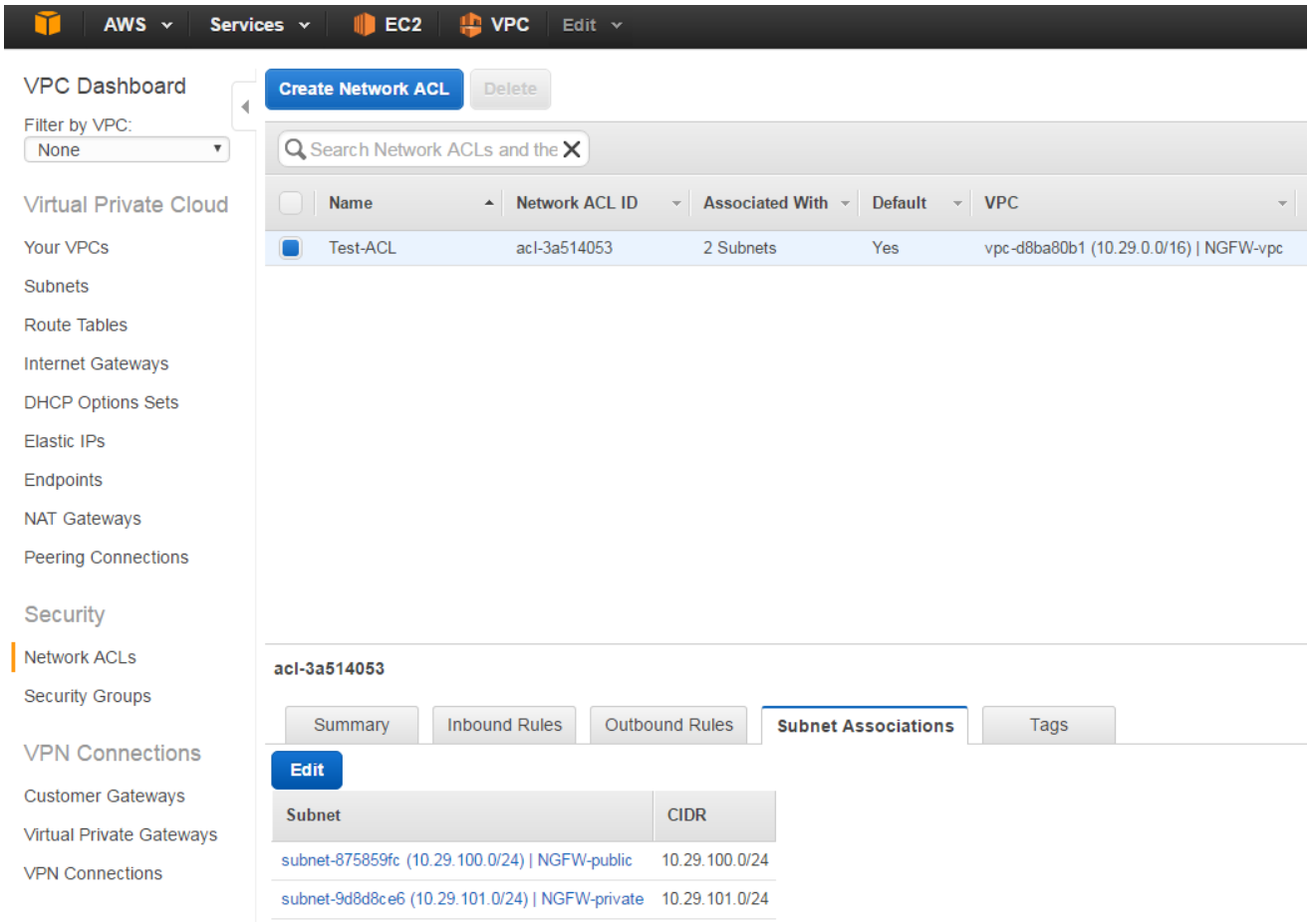
Network ACLs are stateless. They do not provide stateful connection tracking.

Steps

1) Create an ACL that allows all inbound and outbound traffic.
   In this example, Test-ACL (acl-3a514053) has ‘any-any-any-allow’ for inbound and outbound traffic.

2) Associate the ACL with internal and external networks.

3) Create a security group that allows all inbound and outbound traffic.



Next steps

Configure the SMC.

Configuring the SMC for the example deployment

In this example, the SMC is located on premises, and is reached through the public Internet. The SMC is protected by a firewall that allows the communication between NGFW Engines and the SMC, and translates the public IP address of the Management Server component of the SMC to a private IP address.

Configuring the SMC consists of the following tasks:

1) Create a new single NGFW Engine element.

2) Save the initial configuration for the NGFW Engine.

Begin by creating a new single NGFW Engine element.

Create a new single NGFW Engine element for the example deployment

In the Management Client component of the SMC, create a single NGFW Engine element.

Before you begin

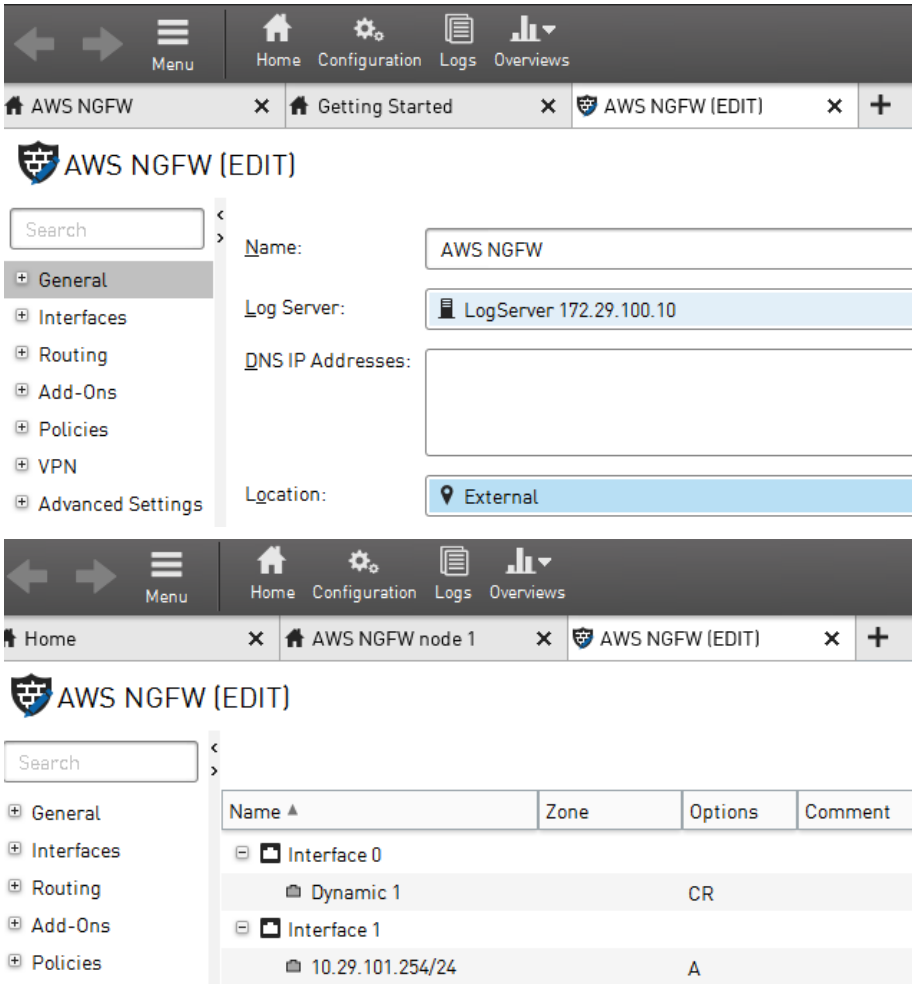
Prepare your VPC for the example deployment.

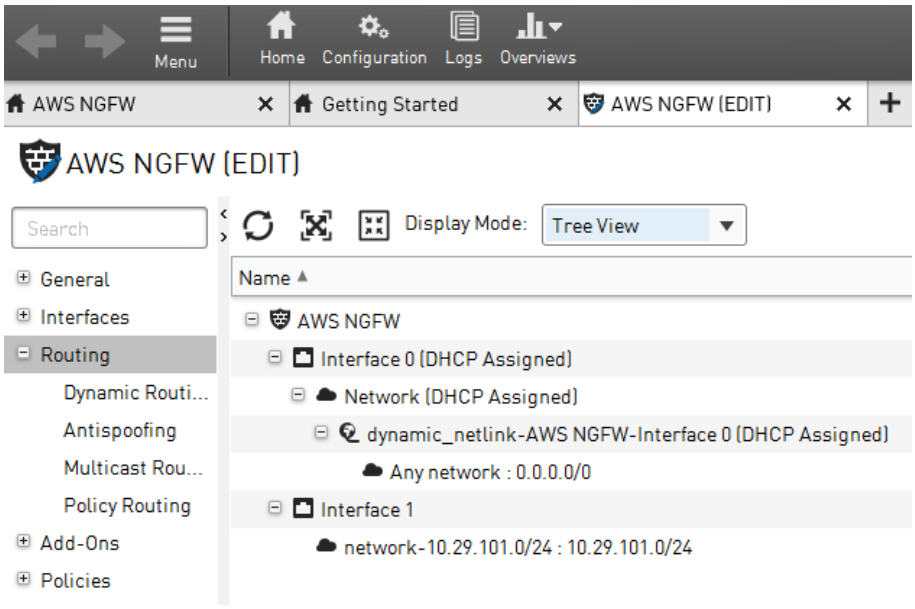
Steps

1) Create new Single Firewall element with a dynamic IP address.

2) Set the Location if the private IP address of the Management Server is not directly reachable.

3) Define a default route behind the management interface.





Next steps

Save the initial configuration for the NGFW Engine.

Save the initial configuration for the NGFW Engine for the example deployment

In the Management Client component of the SMC, save the initial configuration for the NGFW Engine.

Before you begin

Create a new single NGFW Engine element.

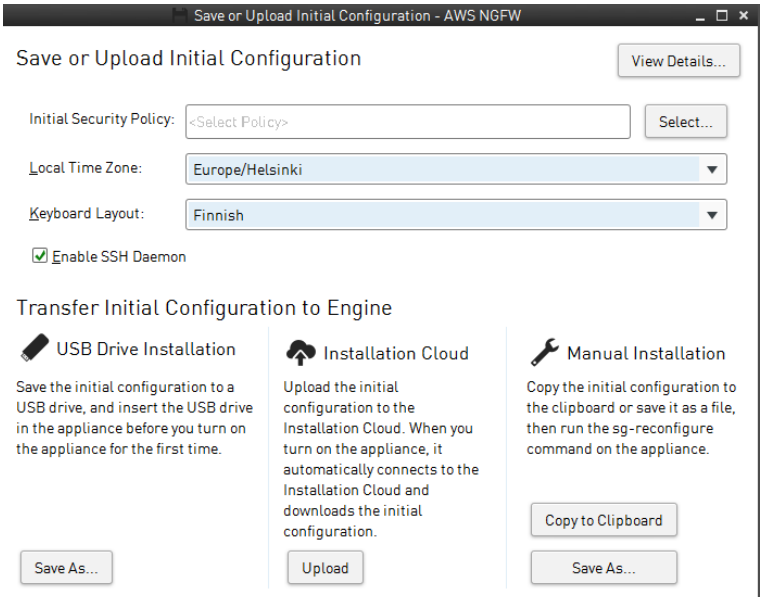
Steps

1) Right-click the engine, then select **Configuration** > **Save initial Configuration**.

2) To allow SSH connections to the NGFW Engine, select **Enable SSH daemon**.

3) Keep the **Save or Upload Initial Configuration** dialog box open.

This dialog box shows the one-time password that you enter when you establish contact between the NGFW Engine and the Management Server.



Next steps

Launch the Forcepoint NGFW instance in AWS.

Launch an instance for the example deployment

In AWS, launch the Forcepoint NGFW instance.

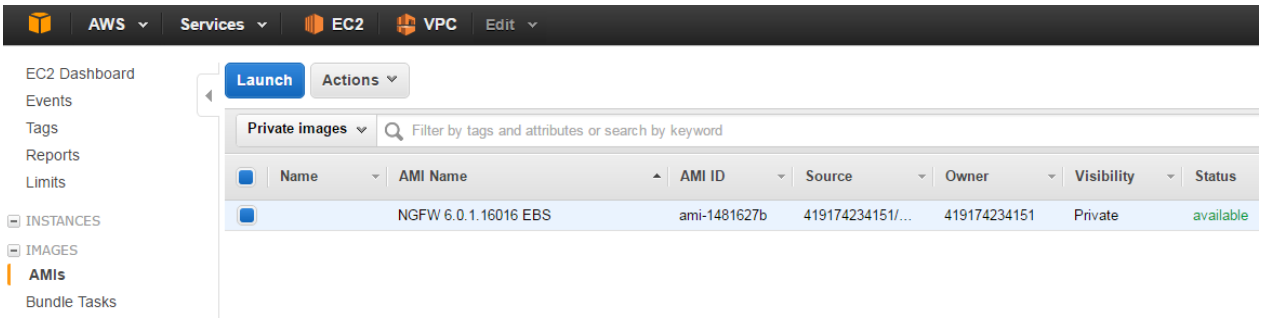
Before you begin

Complete these tasks before you launch the Forcepoint NGFW instance:

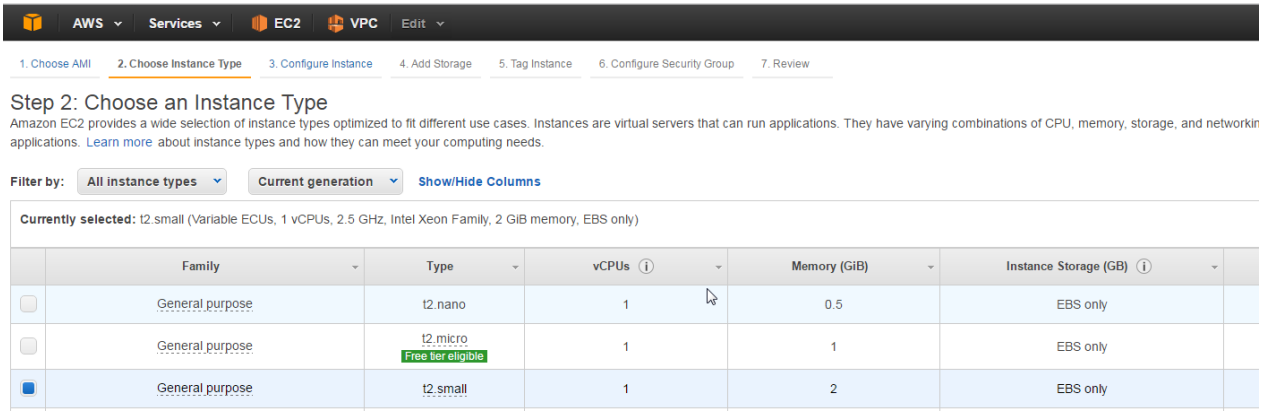
- Prepare the VPC for the example deployment.
- Configure the SMC.

Steps

1) Select **EC2** > **Instances** > **Launch**.



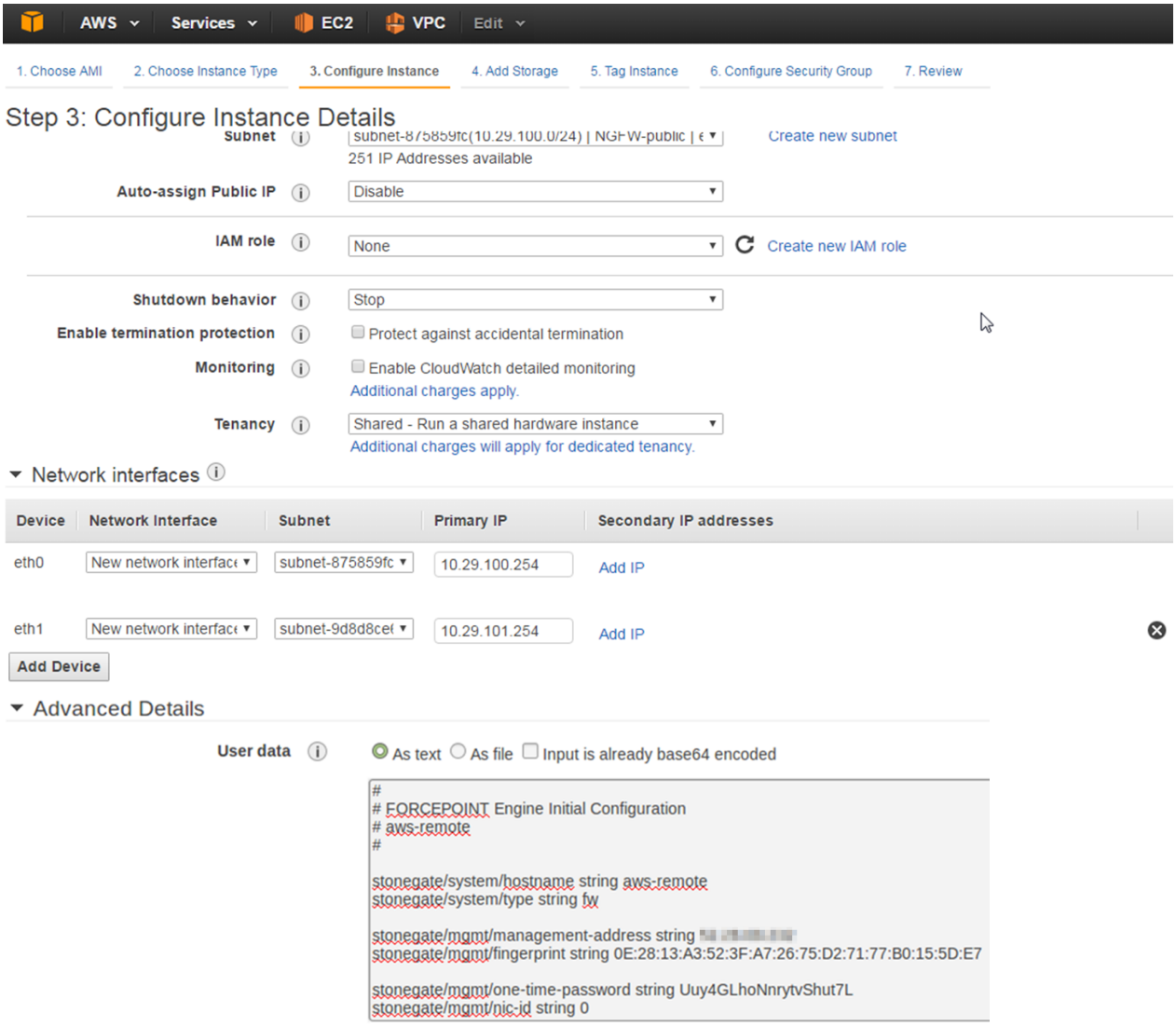
2) Select the latest available Forcepoint NGFW instance.
The minimum requirement for Forcepoint NGFW is 2GB of memory.



3) Use the first interface for management communication.

4) Define IP addresses for the interfaces.
In this example, the IP addresses are 10.29.100.254 and 10.29.101.254.

5) To automatically connect the NGFW Engine to the SMC when it starts up, transfer the initial configuration file that you created in the SMC to the instance.



a) In the **User Data** options, select **As Text**.

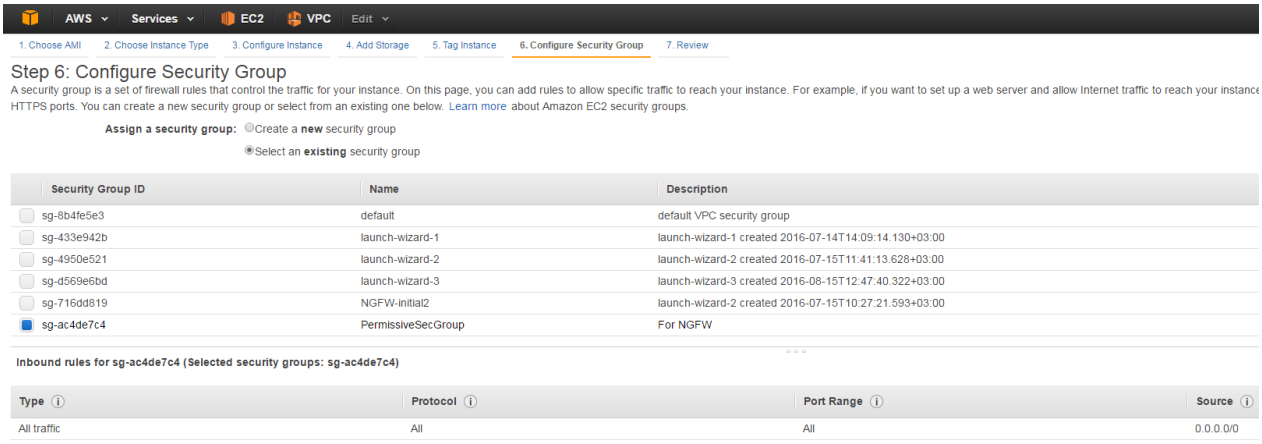
b) In the **Save or Upload Initial Configuration** dialog box in the Management Client, click **Copy to Clipboard**.

c) In the EC2 Management Console, paste the text that you copied from the **Save or Upload Initial Configuration** dialog box into the **User Data** field.

6) Click **Next**.

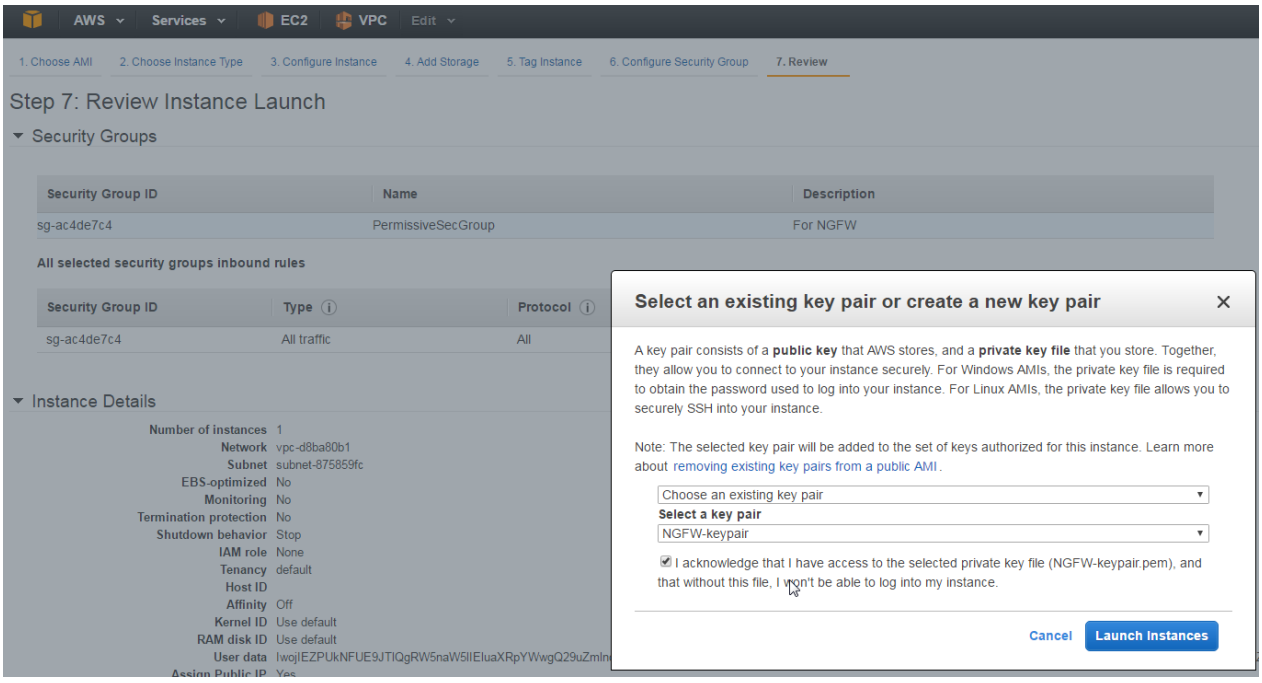
7) Select the security group for the instance.

Security groups filter traffic that reaches the NGFW Engine. In this example, the NGFW Engine is configured in a test environment. The NGFW Engine filters the traffic.



8) Create a key pair, then log on to your instance using SSH.

With Windows instances, use a key pair to obtain the administrator password, then log on using RDP.



a) Create a key pair.

b) Specify the name of the key pair when you launch the instance.

c) Enter the private key when you connect to the instance.

Next steps

Associate an Elastic IP address with your NGFW instance.

Related concepts

Preparing your VPC for the example deployment on page 42
Configuring the SMC for the example deployment on page 47

Associate an Elastic IP address with your NGFW instance for the example deployment

An elastic IP address is a static, public IP address that can be allocated by AWS. Elastic IP addresses can be associated with NGFW instances to allow initial contact with the Management Server to occur over the Internet.

Before you begin

Launch the Forcepoint NGFW instance.

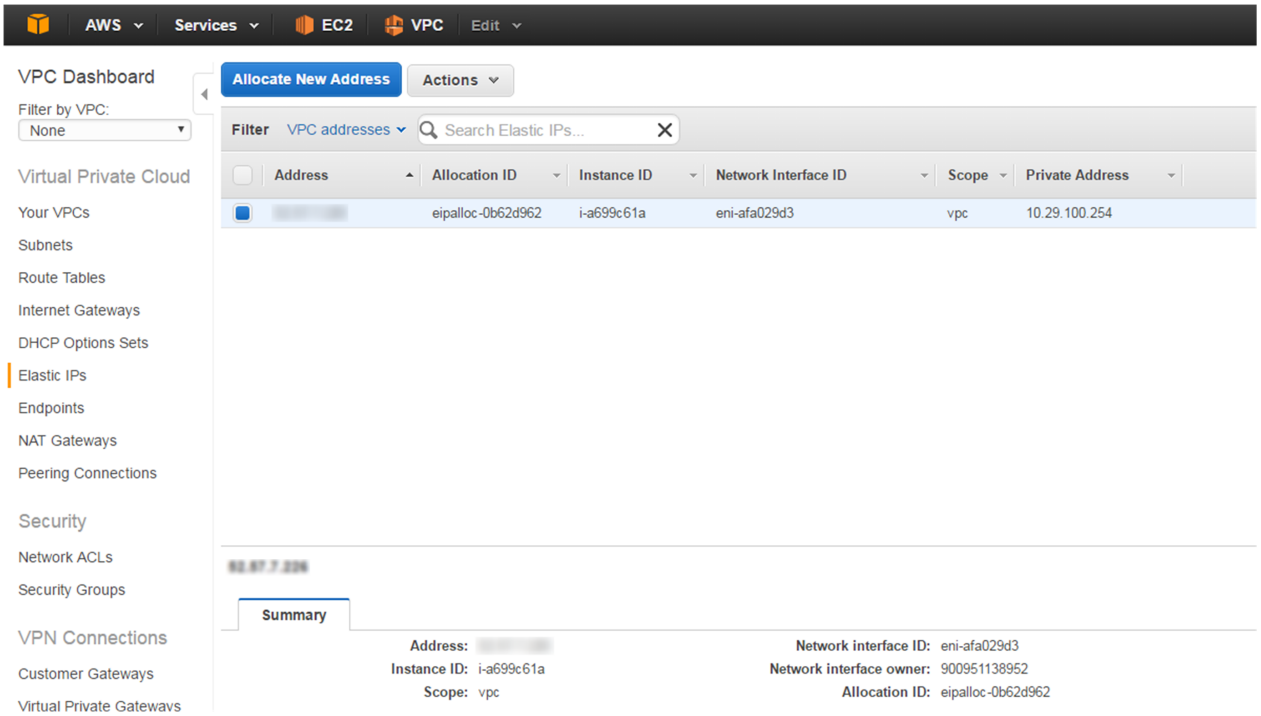
The NGFW Engine makes initial contact to the Management Server when the NGFW Engine starts up. If the Elastic IP address is not yet available when the NGFW Engine tries to connect to the Management Server, the initial contact fails and you must make initial contact manually. See *Log on to the engine using SSH*.

Steps

1) Select **VPC** > **Elastic IPs** > **Allocate New Address**.

2) Select the created address, then select **Actions** > **Associate address** and associate the address with the public IP address of the NGFW Engine (eni-afa029d3).

3) Make a note of the public IP address of the NGFW Engine.



Next steps

Disable source and destination checks.

Related tasks

Log on to the NGFW Engine using SSH in the example deployment on page 56

Disable source and destination checks for the example deployment

Disable source and destination checks on the NGFW instance.

Before you begin

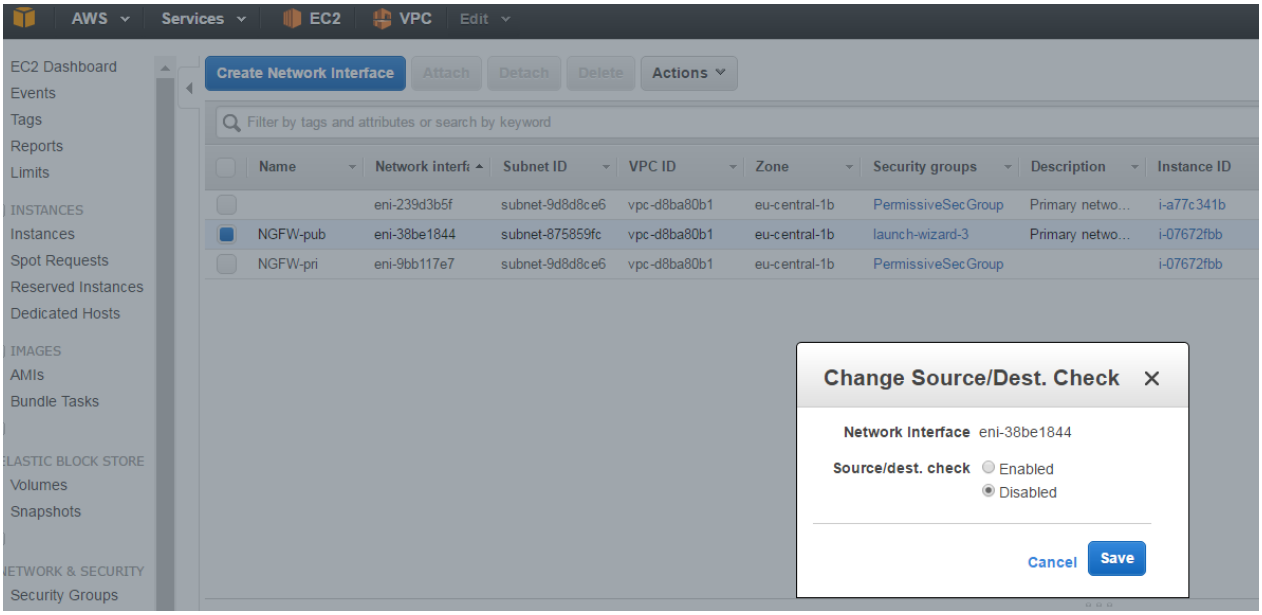
Associate an Elastic IP address with your NGFW instance.

Each EC2 instance performs source and destination checks by default. These checks require that the instance is the source or destination of any traffic it sends or receives. However, an NGFW instance must be able to send and receive traffic when the source or destination is not itself.

Steps

1) To disable source and destination checks for all NGFW interfaces, select **EC2** > **Network Interfaces** > **Actions** > **Change Src/Dst. Check**.

2) Select **Disabled**, then click **Save**.



Next steps

Log on to the NGFW Engine using SSH.

Log on to the NGFW Engine using SSH in the example deployment

You can log onto the engine with the configured key pair.

Before you begin

Disable source and destination checks.

For information about connecting to the engine using PUTTY, see https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html?icmpid=docs_ec2_console.

Steps

1) Check the Elastic IP address for the NGFW instance from **EC2** > **running instances**.

2) Log on with the user name aws.

3) If the AMI does not support the use of sudo without a password, enter the following command to set a sudo password for the aws user:

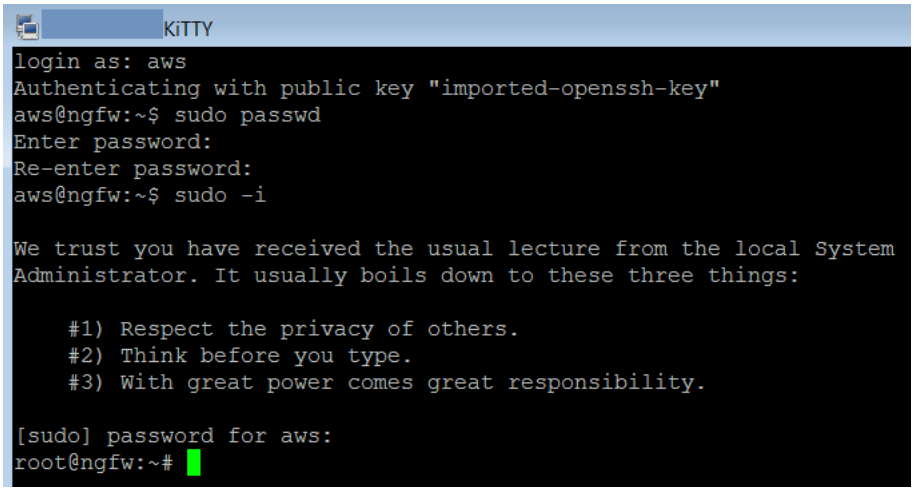
```
sudo passwd
```

After you have set the password for the aws user, the aws user has sudo privileges.

4) To become root, enter the following command:

```
sudo -i
```

5) If required, enter the password for the aws user.



6) If the NGFW Engine did not connect to the Management Server when the NGFW Engine started up, run the sg-reconfigure wizard.

Next steps

Test connectivity through the NGFW Engine.

Test connectivity through the NGFW Engine in the example deployment

To test that traffic is going through the NGFW Engine and that logs are being received, add a Linux host behind the NGFW Engine.

Before you begin

Log on to the NGFW Engine using SSH.

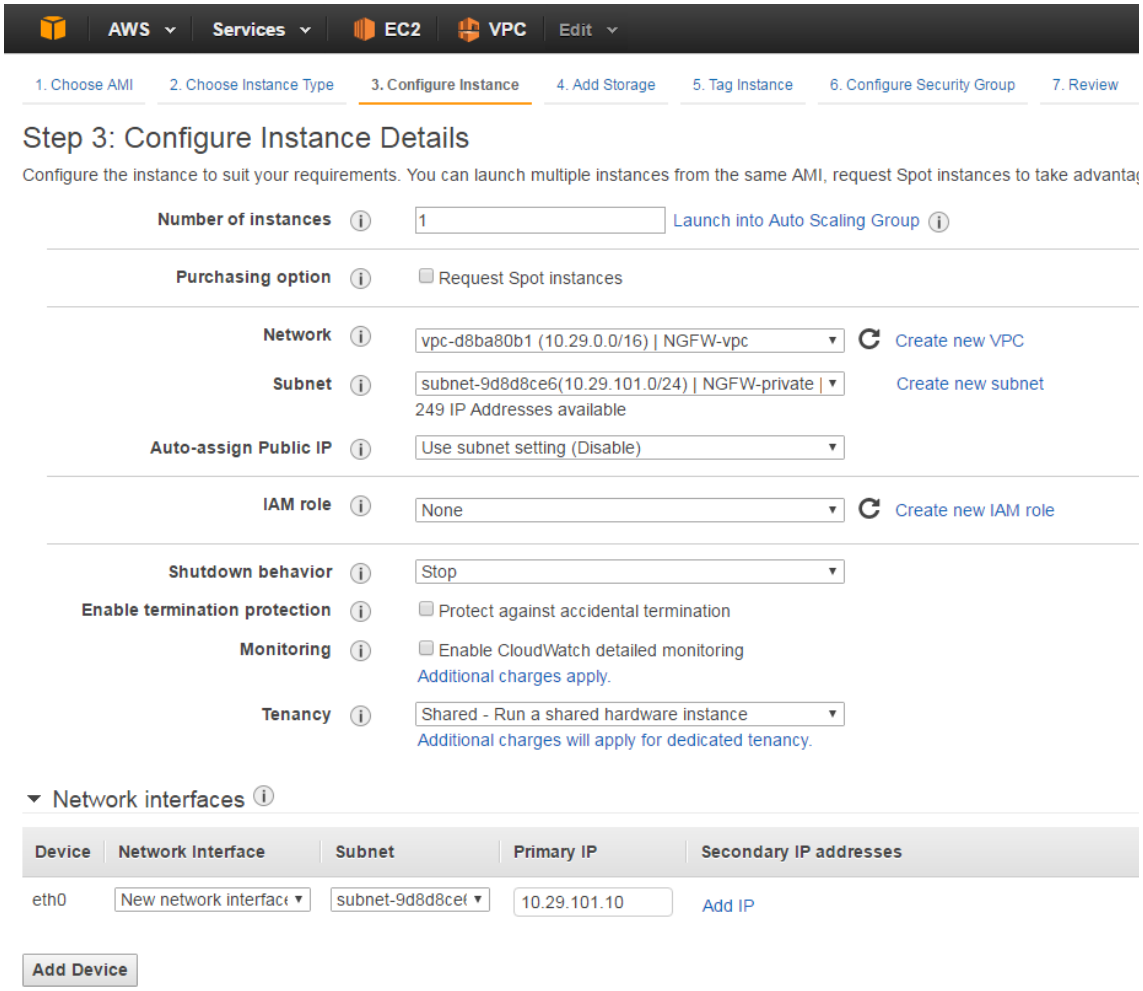
In this example, the following connections are used to test connectivity:

- An SSH connection to the Linux host that is protected by the NGFW Engine
- A ping connection to a Google server.

Steps

1) Deploy a new AMI.
   This example uses Amazon Linux as the operating system for the instance.
   a) Select **EC2** > **Launch instance**, then select an AMI that meets your needs.
   b) Select the internal subnet, then define an IP address from that network for the eth0 network interface.

This example uses the 10.29.101.0/24 internal subnet. The IP address is 10.29.101.1.10.

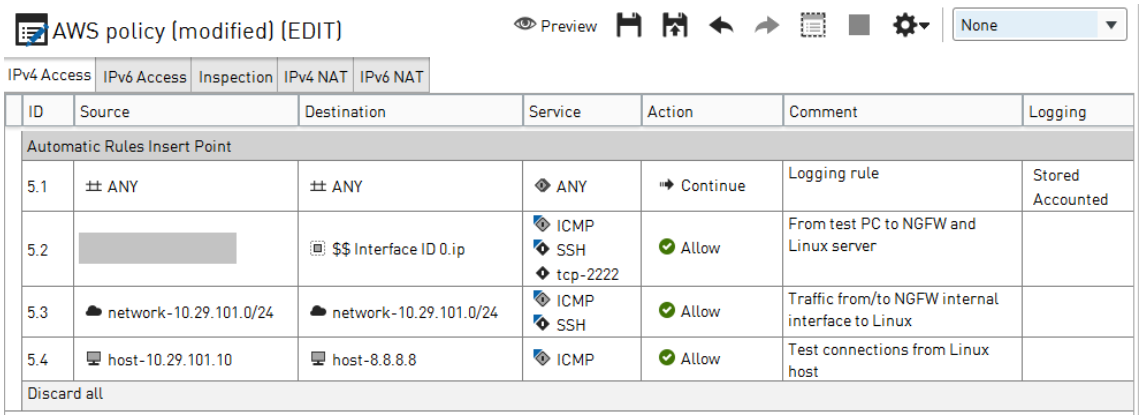


c) Launch the instance.

2) In the Management Client component of the SMC, add rules to the Firewall policy.

a) Add Access rules to allow the following traffic:

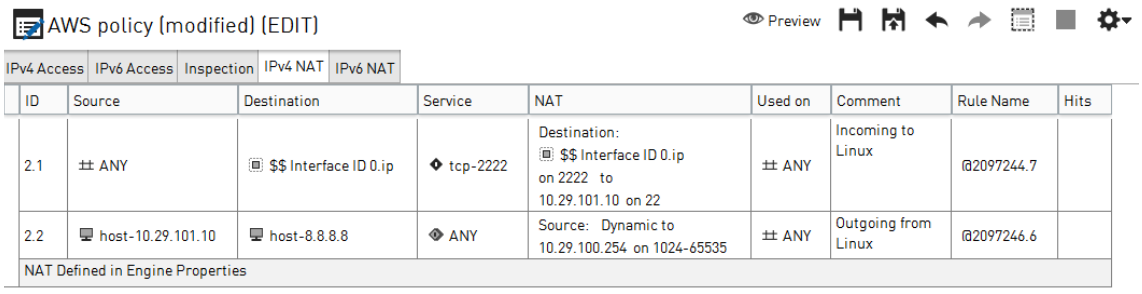
- SSH from your client computer to the NGFW Engine and to the Linux host.
- ICMP and SSH from the Linux host to the NGFW Engine, and from the NGFW Engine to the Linux host.
- Ping from the Linux host to the IP address of a Google server (8.8.8.8).



ID	Source	Destination	Service	Action	Comment	Logging
Automatic Rules Insert Point						
5.1	ANY	ANY	ANY	Continue	Logging rule	Stored Accounted
5.2		$$ Interface ID 0.ip	ICMP SSH tcp-2222	Allow	From test PC to NGFW and Linux server	
5.3	network-10.29.101.0/24	network-10.29.101.0/24	ICMP SSH	Allow	Traffic from/to NGFW internal interface to Linux	
5.4	host-10.29.101.10	host-8.8.8.8	ICMP	Allow	Test connections from Linux host	
Discard all						

b) Add the following NAT rules:

- Destination translation for the public IP address of the NGFW Engine on port 2222 to port 22 on the Linux host.
- Source translation for connections from the Linux host to the public IP address of the NGFW Engine.



ID	Source	Destination	Service	NAT	Used on	Comment	Rule Name	Hits
2.1	ANY	$$ Interface ID 0.ip	tcp-2222	Destination: $$ Interface ID 0.ip on 2222 to 10.29.101.10 on 22	ANY	Incoming to Linux	@2097244.7	
2.2	host-10.29.101.10	host-8.8.8.8	ANY	Source: Dynamic to 10.29.100.254 on 1024-65535	ANY	Outgoing from Linux	@2097246.6	
NAT Defined in Engine Properties								

c) Install the policy.

After the policy has been successfully installed, the status of the NGFW Engine is shown as green in the Home view of the Management Client component of the SMC.

3) Test connectivity.

a) In a terminal program, make an SSH connection on TCP port 2222 to the public IP address of the NGFW Engine.

b) Log on using the key pair exported from AWS.

The default user for Amazon Linux is ‘ec2-user’.

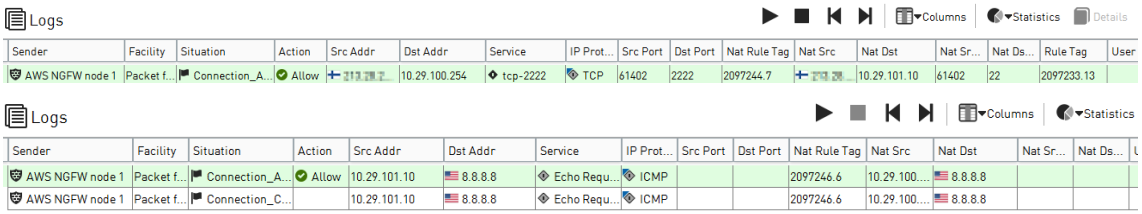
c) When logged in, ping 8.8.8.8.

```
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"
Last login: Wed Sep  7 09:35:39 2016 from [redacted].bb.dnainternet.fi

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2016.03-release-notes/
[ec2-user@ip-10-29-101-10 ~]$ ping 8.8.8.8
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=47 time=7.78 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=47 time=7.76 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=47 time=7.78 ms
^C
--- 8.8.8.8 ping statistics ---
[ec2-user@ip-10-29-101-10 ~]$ ved, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 7.762/7.778/7.787/0.072 ms
```

d) In the Management Client component of the SMC, open the Logs view, then check to see that the connection was allowed.



Result

The example deployment is now complete.

- The NAT operation that translates the Elastic IP address of the NGFW Engine to the internal IP address of the NGFW Engine is applied on the Internet gateway before the packet reaches the NGFW Engine. For this reason, the public IP address of the NGFW Engine is not shown in the logs.
- AWS reserves the .1 IP address for its router. The routing table on the Linux host shows that the next-hop subnet gateway is 10.29.101.1. However, the NGFW Engine has been configured as the next-hop subnet gateway and its IP address is 10.29.101.254. AWS uses 10.29.101.1 as the default gateway and applies internal NAT to 10.29.101.254. It is important to keep this internal NAT operation in mind, especially when troubleshooting.

```
[ec2-user@ip-10-29-101-10 ~]$ ip route
default via 10.29.101.1 dev eth0
10.29.101.0/24 dev eth0  proto kernel  scope link  src 10.29.101.10
[redacted] dev eth0
[ec2-user@ip-10-29-101-10 ~]$
```

Configuring VPC ingress routing for an Internet gateway

VPC ingress routing can direct all traffic from an edge location, such as the Internet or a VPN gateway, through the Forcepoint NGFW Engine before reaching its final destination. These instructions describe how to configure VPC ingress routing for an Internet gateway.

When you use VPC ingress routing, you do not need to configure NAT rules for the NGFW Engine to direct connections to the public IP address of the host to the private IP address of the host.

Configuring VPC ingress routing consists of these general steps:

1) Create a route table for the VPC.

2) Define public IP addresses for hosts in the VPC.

Begin by creating a route table for the VPC.

Create a route table for VPC ingress routing

Create a route table, define the routes, then associate the Internet gateway with the route table.

Steps

1) Open the Amazon VPC console.

2) Create a new route table.
   a) In the navigation pane, select **Route Tables**.
   b) Click **Create route table**.
   c) In the **Name tag** field, enter a unique name.
   d) In the **VPC** field, select the VPC in which the Forcepoint NGFW Engine is deployed.
   e) Click **Create**.
   f) Click **Close**.

3) Define a route to the network interface of the Forcepoint NGFWEngine.
   a) Select the route table, then select **Actions** > **Edit routes**.
   b) From the **Target** drop-down list, select the network interface of the Forcepoint NGFW Engine.
   c) Click **Save routes**.

d) Click **Close**.

4) Associate the Internet gateway with the route table.
   a) Select the route table, then select **Actions** > **Edit edge associations**.
   b) Select**Internet gateways**, then select the Internet gateway.
   c) Click **Save**.

Define public IP addresses for hosts in the VPC

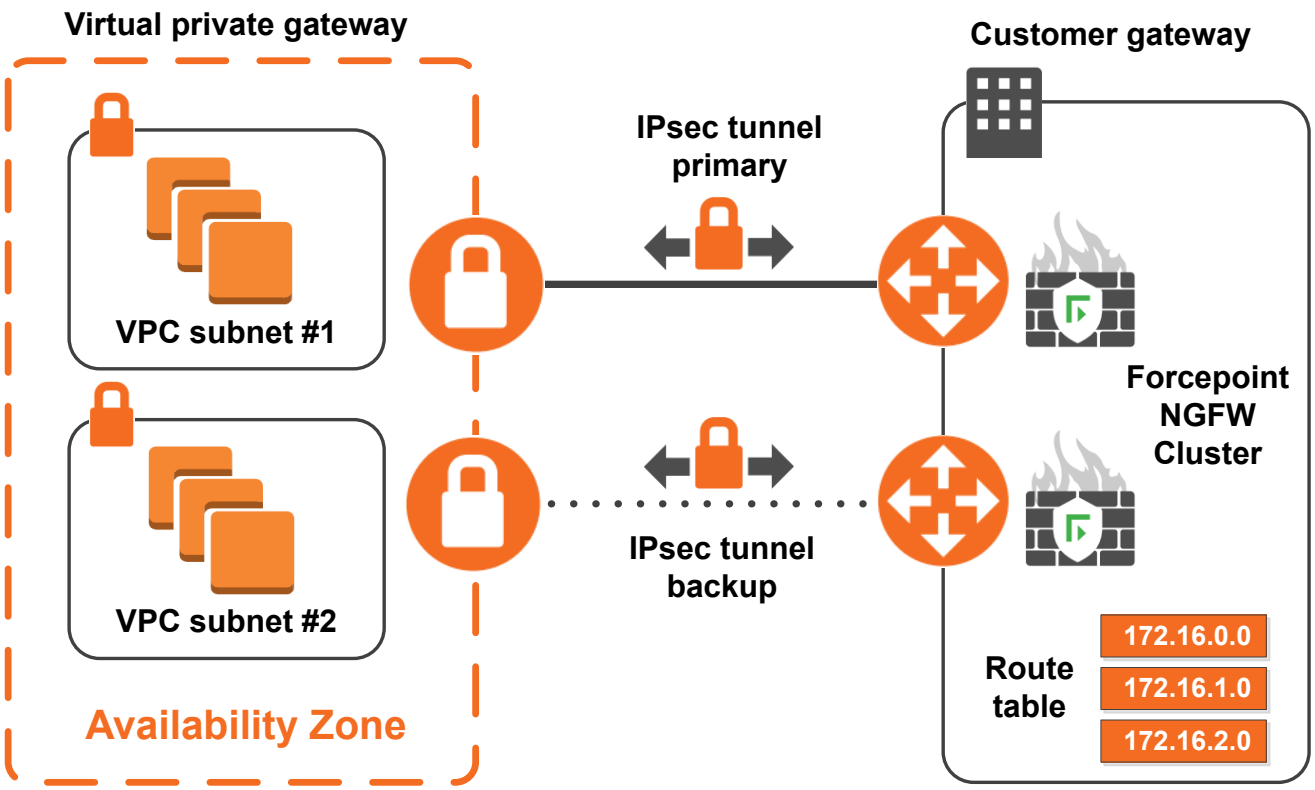
Create public IP addresses and associate the IP addresses with hosts in the VPC.

Steps

1) Open the Amazon VPC console.
2) Create a public IP address.
   For more information, see https://docs.aws.amazon.com/vpc/latest/userguide/vpc-ip-addressing.html
3) Associate the IP address with the host instance.
   a) Click the IP address.
   b) Select **Actions** > **Associate address**.
   c) From the **Instance** drop-down list, select the host instance.
   d) From the **Private IP** drop-down list, select the private IP address of the host.
   e) Click **Associate**.
4) Click **Close**.

Configuring a route-based VPN to AWS with BGP

The configuration for this scenario includes a virtual private cloud (VPC) with a public subnet and private subnets. A virtual private gateway enables communication with your own on-premises network over an IPsec VPN tunnel. All routing configuration is done using BGP.



Configure the VPN settings in AWS

Follow these steps to configure the VPN settings in AWS.

Steps

1) Create the Customer Gateway.

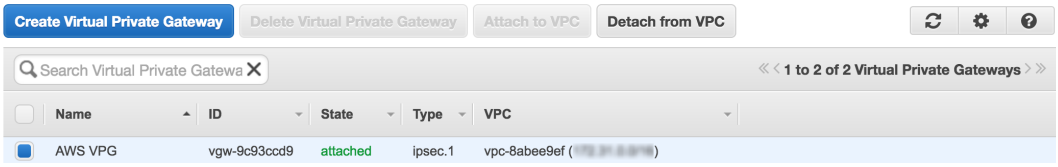
a) Browse to **VPN Connections** > **Customer Gateways**.

b) Click **Create Customer Gateway**.

c) In the **IP Address** field, enter the public IP address of the NGFW Engine.

d) Click **Yes, Create**.

2) Create the Virtual Private Gateway and attach it to the VPC.



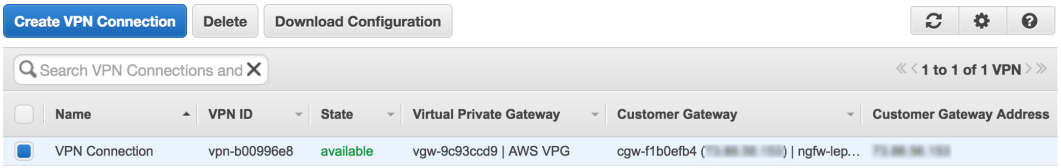
a) Browse to **VPN Connections** > **Virtual Private Gateways**.

b) Click **Create Virtual Private Gateway**.

c) Configure the settings, then click **Yes, Create**.

d) Right-click the virtual private gateway, select **Attach to VPC**, then select the VPC.

3) Create the VPN Connection.



a) Browse to **VPN Connections** > **VPN Connections**.

b) Click **Create VPN Connection**.

c) For **Routing Options**, select **Dynamic**, then specify BGP.

d) Click **Yes, Create**.

4) Download the VPN Connection configuration.

a) Click **Download Configuration**.

b) In the **Download Configuration** dialog box, select **Generic** as the vendor type.

c) Click **Yes, Download**.

d) Save the file that contains the VPN Connection configuration.

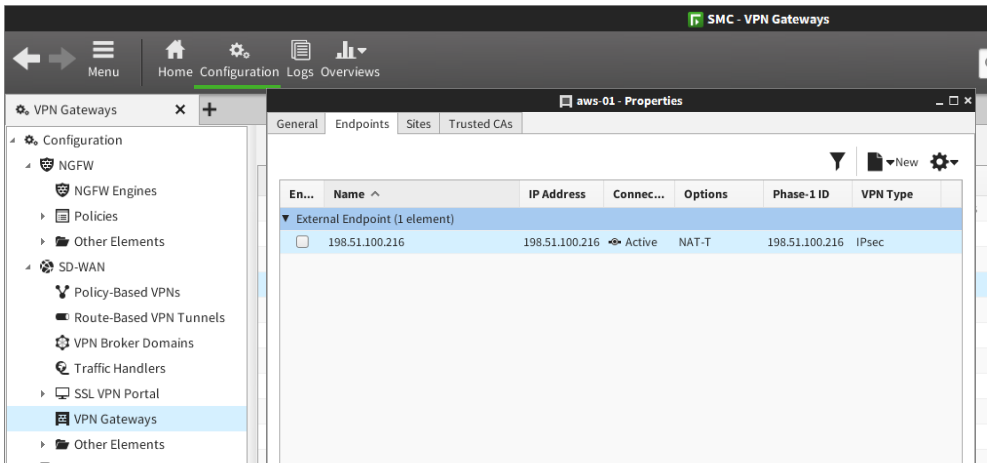
Configure Forcepoint NGFW settings in the SMC

Use the VPN Connection configuration that you downloaded from AWS to configure the remaining NGFW Engine settings.

The VPN Connection configuration provides the tunnel interface IP addresses, the next-hop gateway, autonomous system (AS) numbers, pre-shared keys, and the cryptographic specifications.

Steps

1) In the Management Client, create two External VPN Gateway elements that represent the two AWS endpoints.

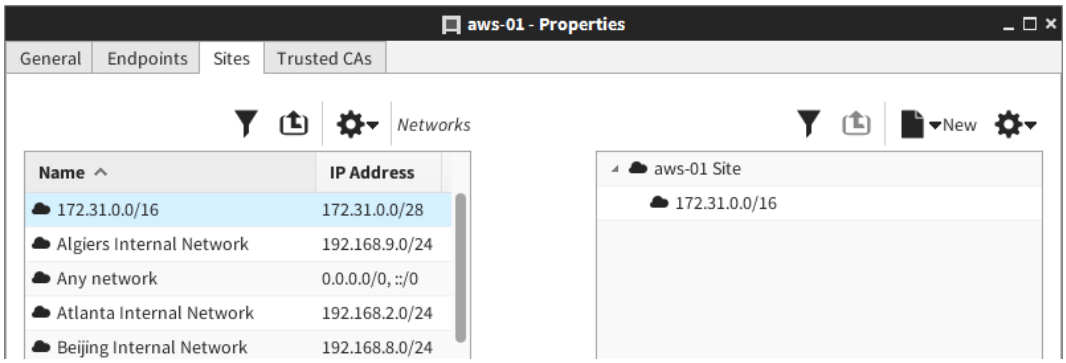


a) Select **Configuration** then browse to **SD-WAN** > **VPN Gateways**.
In SMC 6.4 or lower, select **Configuration** then browse to **VPN** > **Gateways**.

b) Right-click **VPN Gateways**, then select **New External VPN Gateway**.
In SMC 6.4 or lower, right-click **Gateways**, then select **New External VPN Gateway**.

c) On the **Endpoints** tab of each External VPN Gateway element, add the IP address of the AWS endpoint.

d) On the **Sites** tab of each External VPN Gateway element, configure each external gateway site to match the VPC network, then click **OK**.
In this example, the VPC network is 172.31.0.0/16.



2) Add a tunnel interface to the NGFW Engine for each VPN gateway, then add the IP address of the AWS endpoint to each tunnel interface.

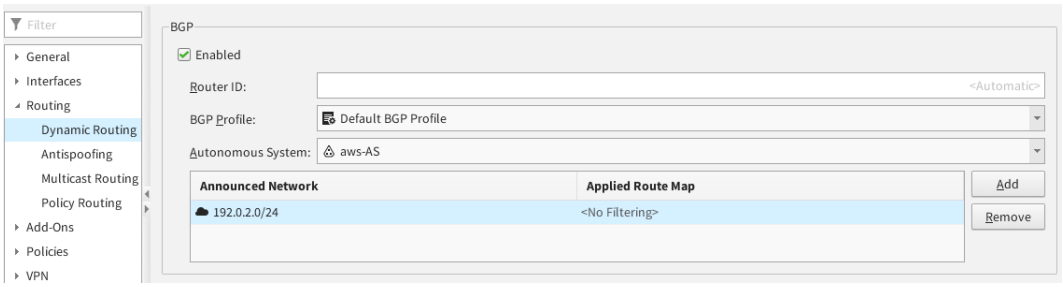
a) Browse to **Configuration** > **NGFW** > **NGFW Engines**.

b) Right-click the NGFW Engine, then select **Edit Single Firewall**.

c) In the Engine Editor, browse to **Interfaces**.

d) Add one tunnel interface for each VPN gateway.

3) Enable BGP in the NGFW Engine properties.



a) In the Engine Editor, browse to **Routing** > **Dynamic Routing**.

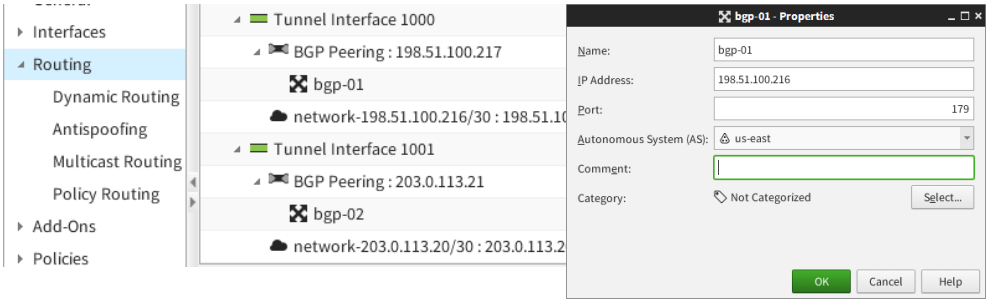
b) In the **BGP** settings, select **Enabled**.

c) In the **Autonomous System** field, create an Autonomous System element that uses the AS number that AWS specified in the configuration.

The default is 65000.

d) Add your protected network to the Announced Network configuration.

4) Edit the routing configuration for the NGFW Engine.



a) In the Engine Editor, browse to **Routing**.

b) Under each tunnel interface, add a BGP Peering element.

c) Right-click the BGP Peering element under each tunnel interface, then select **Add External BGP Peer**.

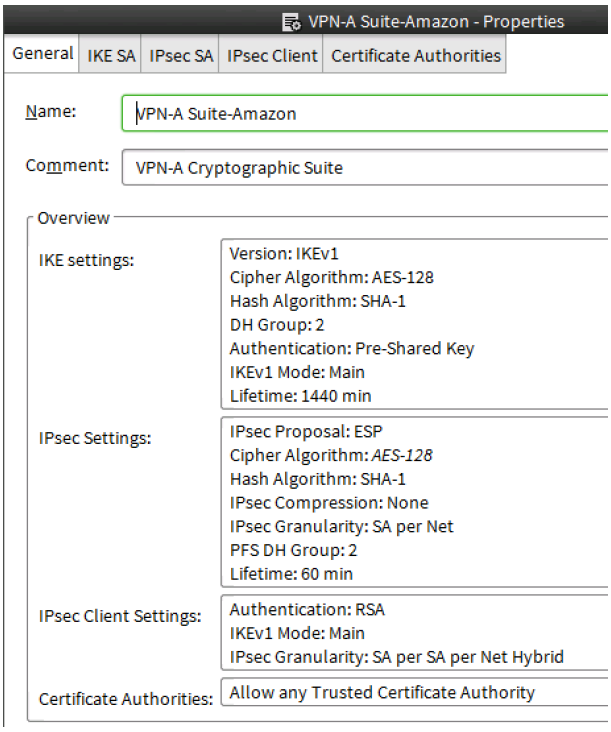
d) Select an AWS gateway for each tunnel interface.

For the **Autonomous System (AS)** field, create an Autonomous System element that uses the AS number provided by AWS.

In this example, the AS number is 7224 for us-east.

e) Click **Save**.

5) Create a VPN Profile that matches the settings required by AWS.



a) Select **Configuration** then browse to **SD-WAN** > **Other Elements** > **Profiles** > **VPN Profiles**.
In SMC 6.4 or lower, select **Configuration** then browse to **VPN** > **Other Elements** > **Profiles** > **VPN Profiles**.

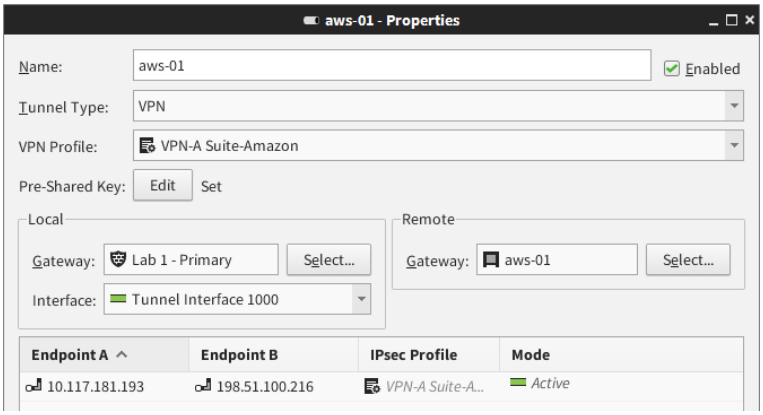
b) Right-click **VPN Profiles**, then select **New VPN Profile**

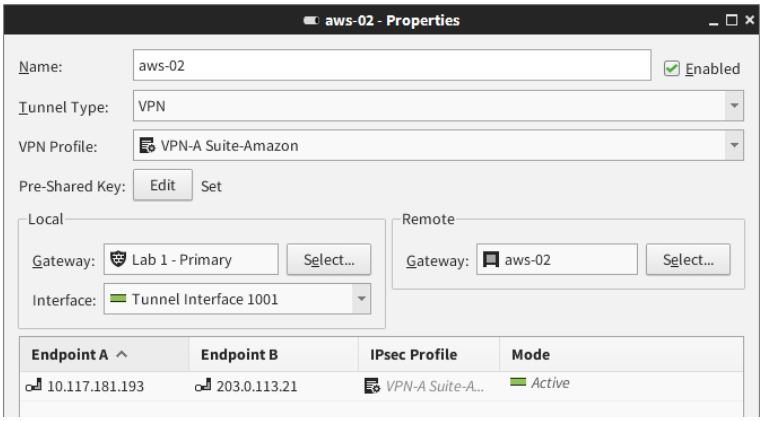
c) Configure the settings to match the settings required by AWS, then click **OK**.

6) Create route-based VPN tunnels for each AWS gateway.

a) Browse to **Configuration** > **SD-WAN** > **Route-Based VPN Tunnels**.
In SMC 6.4 or lower, browse to **Configuration** > **VPN** > **Route-Based VPN Tunnels**.

b) Right-click **Route-Based VPN Tunnels**, then select **New Route-Based VPN Tunnel**.





c) For each tunnel, select the VPN Profile element that you created.

d) For each tunnel, enter the pre-shared key from the AWS VPN Connection configuration.

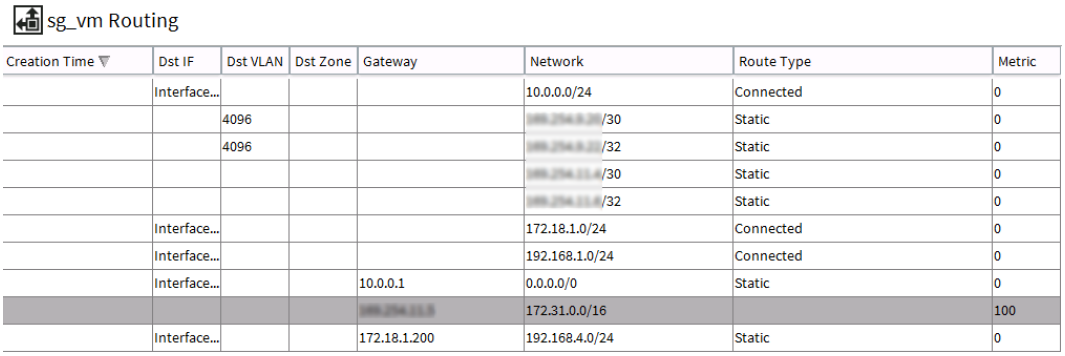
e) In the **Local** settings, select the NGFW Engine, then select a tunnel interface.

f) In the **Remote** settings, select an AWS gateway.
Make sure that you select the correct AWS gateway for each tunnel interface.

7) Browse to **Configuration** > **Policy** > **Firewall Policy**, then create a Firewall Policy that allows traffic in both directions between the networks.

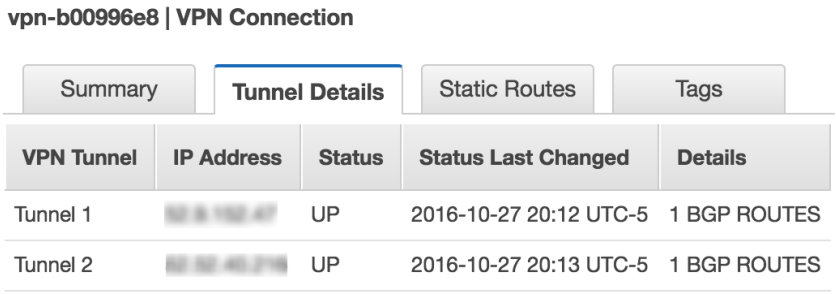
8) To verify that the IPsec tunnel is correctly established, right-click the NGFW Engine, then select **Monitoring** > **VPN SAs**.

9) To verify that BGP correctly propagates routes, select **Home**, right-click the NGFW Engine, then select **Monitoring** > **Routing**.



Creation Time	Dst IF	Dst VLAN	Dst Zone	Gateway	Network	Route Type	Metric
	Interface...				10.0.0.0/24	Connected	0
		4096			[redacted]/30	Static	0
		4096			[redacted]/32	Static	0
					[redacted]/30	Static	0
					[redacted]/32	Static	0
	Interface...				172.18.1.0/24	Connected	0
	Interface...				192.168.1.0/24	Connected	0
	Interface...			10.0.0.1	0.0.0.0/0	Static	0
				[redacted]	172.31.0.0/16		100
	Interface...			172.18.1.200	192.168.4.0/24	Static	0

10) In the AWS console, browse to the **Tunnel Details** tab, then verify that the tunnels are active.



VPN Tunnel	IP Address	Status	Status Last Changed	Details
Tunnel 1	[redacted]	UP	2016-10-27 20:12 UTC-5	1 BGP ROUTES
Tunnel 2	[redacted]	UP	2016-10-27 20:13 UTC-5	1 BGP ROUTES

Troubleshooting the BGP configuration

If necessary, you can troubleshoot the configuration on the command line of the NGFW Engine and in the Management Client.

Troubleshooting on the command line of the NGFW Engine

Connect to the NGFW Engine using SSH, enter `vtysh`, then use the following commands:

```
show ip bgp
show ip bgp neighbors
show ip bgp summary
show ip bgp ? (list all possible command options)
```

Troubleshooting in the Management Client

In the Management Client, you can do the following:

- To view log entries in the Logs view, select **Logs**.
- To check the configuration in Quagga format, right-click an NGFW Engine, then select **Configuration** > **Dynamic Routing** > **Quagga Preview**.

- To restart the dynamic routing process, right-click an NGFW Engine, then select **Configuration** > **Dynamic Routing** > **Restart**.
- Connect to the NGFW Engine using SSH, then ping a tunnel interface of the BGP peer gateway.
- Use tcpdump on the tunnel interfaces to verify that traffic is passing through.

Reference Quagga configuration

```
!Configuration generated by the SMC via DRCFGD
hostname SG-Quagga-Router
!
router bgp 65000
bgp router-id 172.18.1.254
!element979
network 172.18.1.0/24
neighbor 203.0.113.5 remote-as 7224
neighbor 203.0.113.5 update-source 203.0.113.6
neighbor 203.0.113.5 timers 60 180
neighbor 203.0.113.5 timers connect 120
neighbor 203.0.113.5 disable-connected-check
neighbor 203.0.113.5 soft-reconfiguration inbound
neighbor 203.0.113.5 next-hop-self
neighbor 203.0.113.21 remote-as 7224
neighbor 203.0.113.21 update-source 203.0.113.22
neighbor 203.0.113.21 timers 60 180
neighbor 203.0.113.21 timers connect 120
neighbor 203.0.113.21 disable-connected-check
neighbor 203.0.113.21 soft-reconfiguration inbound
neighbor 203.0.113.21 next-hop-self
bgp graceful-restart
distance bgp 20 200 200
```

Reference AWS VPN Connection configuration

```
Amazon Web Services
Virtual Private Cloud
VPN Connection Configuration
================================================================================
AWS utilizes unique identifiers to manipulate the configuration of
a VPN Connection. Each VPN Connection is assigned a VPN Connection Identifier
and is associated with two other identifiers, namely the
Customer Gateway Identifier and the Virtual Private Gateway Identifier.
Your VPN Connection ID : vpn-b00996e8
Your Virtual Private Gateway ID : vgw-9c93ccd9
Your Customer Gateway ID : cgw-f1b0efb4
A VPN Connection consists of a pair of IPSec tunnel security associations (SAs).
It is important that both tunnel security associations be configured.
IPSec Tunnel #1
================================================================================
```

```
#1: Internet Key Exchange Configuration
Configure the IKE SA as follows:
Please note, these sample configurations are for the minimum requirement of AES128, SHA1, and DH
Group 2.
You will need to modify these sample configuration files to take advantage of AES256, SHA256, or
 other
DH groups like 2, 14-18, 22, 23, and 24.
The address of the external interface for your customer gateway must be a static address.
Your customer gateway may reside behind a device performing network address translation (NAT).
To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall !rules to unblock
 UDP
port 4500. If not behind NAT, we recommend disabling NAT-T.
- Authentication Method : Pre-Shared Key
- Pre-Shared Key : [ hidden ]
- Authentication Algorithm : sha1
- Encryption Algorithm : aes-128-cbc
- Lifetime : 28800 seconds
- Phase 1 Negotiation Mode : main
- Perfect Forward Secrecy : Diffie-Hellman Group 2
```

```
#2: IPSec Configuration
Configure the IPSec SA as follows:
Please note, you may use these additionally supported IPSec parameters for encryption like AES256
and other DH groups like 1,2, 5, 14-18, 22, 23, and 24.
- Protocol : esp
- Authentication Algorithm : hmac-sha1-96
- Encryption Algorithm : aes-128-cbc
- Lifetime : 3600 seconds
- Mode : tunnel
- Perfect Forward Secrecy : Diffie-Hellman Group 2
IPSec Dead Peer Detection (DPD) will be enabled on the AWS Endpoint. We
recommend configuring DPD on your endpoint as follows:
- DPD Interval : 10
- DPD Retries : 3
IPSec ESP (Encapsulating Security Payload) inserts additional
headers to transmit packets. These headers require additional space,
which reduces the amount of space available to transmit application data.
To limit the impact of this behavior, we recommend the following
configuration on your Customer Gateway:
- TCP MSS Adjustment : 1387 bytes
- Clear Don't Fragment Bit : enabled
- Fragmentation : Before encryption
```

```
#3: Tunnel Interface Configuration
Your Customer Gateway must be configured with a tunnel interface that is
associated with the IPSec tunnel. All traffic transmitted to the tunnel
interface is encrypted and transmitted to the Virtual Private Gateway.
The Customer Gateway and Virtual Private Gateway each have two addresses that relate
to this IPSec tunnel. Each contains an outside address, upon which encrypted
traffic is exchanged. Each also contain an inside address associated with the tunnel interface.
The Customer Gateway outside IP address was provided when the Customer Gateway
was created. Changing the IP address requires the creation of a new Customer Gateway.
The Customer Gateway inside IP address should be configured on your tunnel interface.
Outside IP Addresses:
- Customer Gateway : 192.0.2.153
- Virtual Private Gateway : 172.16.0.47
Inside IP Addresses
- Customer Gateway : 203.0.113.6/30
- Virtual Private Gateway : 203.0.113.5/30
Configure your tunnel to fragment at the optimal size:
- Tunnel interface MTU : 1436 bytes
```

```
#4: Border Gateway Protocol (BGP) Configuration:
The Border Gateway Protocol (BGPv4) is used within the tunnel, between the inside
IP addresses, to exchange routes from the VPC to your home network. Each
BGP router has an Autonomous System Number (ASN). Your ASN was provided
to AWS when the Customer Gateway was created.
BGP Configuration Options:
- Customer Gateway ASN : 65000
- Virtual Private Gateway ASN : 7224
- Neighbor IP Address : 203.0.113.5
- Neighbor Hold Time : 30
Configure BGP to announce routes to the Virtual Private Gateway. The gateway
will announce prefixes to your customer gateway based upon the prefix you
assigned to the VPC at creation time.
IPSec Tunnel #2
================================================================================
```

```
#1: Internet Key Exchange Configuration
Configure the IKE SA as follows:
Please note, these sample configurations are for the minimum requirement of AES128, SHA1, and DH
Group 2.
You will need to modify these sample configuration files to take advantage of AES256, SHA256, or
 other
DH groups like 2, 14-18, 22, 23, and 24.
The address of the external interface for your customer gateway must be a static address.
Your customer gateway may reside behind a device performing network address translation (NAT).
To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall !rules to unblock
 UDP
port 4500. If not behind NAT, we recommend disabling NAT-T.
- Authentication Method : Pre-Shared Key
- Pre-Shared Key : [ hidden ]
- Authentication Algorithm : sha1
- Encryption Algorithm : aes-128-cbc
- Lifetime : 28800 seconds
- Phase 1 Negotiation Mode : main
- Perfect Forward Secrecy : Diffie-Hellman Group 2
```

```
#2: IPSec Configuration
Configure the IPSec SA as follows:
Please note, you may use these additionally supported IPSec parameters for encryption like AES256
and other DH groups like 1,2, 5, 14-18, 22, 23, and 24.
- Protocol : esp
- Authentication Algorithm : hmac-sha1-96
- Encryption Algorithm : aes-128-cbc
- Lifetime : 3600 seconds
- Mode : tunnel
- Perfect Forward Secrecy : Diffie-Hellman Group 2
IPSec Dead Peer Detection (DPD) will be enabled on the AWS Endpoint. We
recommend configuring DPD on your endpoint as follows:
- DPD Interval : 10
- DPD Retries : 3
IPSec ESP (Encapsulating Security Payload) inserts additional
headers to transmit packets. These headers require additional space,
which reduces the amount of space available to transmit application data.
To limit the impact of this behavior, we recommend the following
configuration on your Customer Gateway:
- TCP MSS Adjustment : 1387 bytes
- Clear Don't Fragment Bit : enabled
- Fragmentation : Before encryption
```

```
#3: Tunnel Interface Configuration
Your Customer Gateway must be configured with a tunnel interface that is
associated with the IPSec tunnel. All traffic transmitted to the tunnel
interface is encrypted and transmitted to the Virtual Private Gateway.
The Customer Gateway and Virtual Private Gateway each have two addresses that relate
to this IPSec tunnel. Each contains an outside address, upon which encrypted
traffic is exchanged. Each also contain an inside address associated with
the tunnel interface.
The Customer Gateway outside IP address was provided when the Customer Gateway
was created. Changing the IP address requires the creation of a new Customer Gateway.
The Customer Gateway inside IP address should be configured on your tunnel interface.
Outside IP Addresses:
- Customer Gateway : 192.0.2.153
- Virtual Private Gateway : 198.51.100.216
Inside IP Addresses
- Customer Gateway : 203.0.113.22/30
- Virtual Private Gateway : 203.0.113.21/30
Configure your tunnel to fragment at the optimal size:
- Tunnel interface MTU : 1436 bytes
```

```
#4: Border Gateway Protocol (BGP) Configuration:
The Border Gateway Protocol (BGPv4) is used within the tunnel, between the inside
IP addresses, to exchange routes from the VPC to your home network. Each
BGP router has an Autonomous System Number (ASN). Your ASN was provided
to AWS when the Customer Gateway was created.
BGP Configuration Options:
- Customer Gateway ASN : 65000
- Virtual Private Gateway ASN : 7224
- Neighbor IP Address : 203.0.113.21
- Neighbor Hold Time : 30
Configure BGP to announce routes to the Virtual Private Gateway. The gateway
will announce prefixes to your customer gateway based upon the prefix you
assigned to the VPC at creation time.
```

Find product documentation

In the Forcepoint Customer Hub, you can find information about a released product, including product documentation, technical articles, and more.

You can get additional information and support for your product in the Forcepoint Customer Hub at https://support.forcepoint.com. There, you can access product documentation, release notes, Knowledge Base articles, downloads, cases, and contact information.

You might need to log on to access the Forcepoint Customer Hub. If you do not yet have credentials, create a customer account. See https://support.forcepoint.com/CreateAccount.

Published 07 October 2022