Barracuda

Barracuda NextGen Firewall F-Series

NGF
Barracuda NextGen Firewall

Implementation Guide - NextGen Firewall in AWS

BarracudaCampus

Table of Contents

AWS Implementation Guide

AWS Reference Architecture

Table of Contents

Step-by-Step Guides

Table of Contents

AWS Implementation Guide

1.1 Implementation Guide - NextGen Firewall in AWS

Amazon Web Services follows the shared security responsibility model. AWS is responsible for security of the cloud. This includes physical security, servers, networking hardware, and the hypervisor. The customer, on the other hand, is responsible for everything running in the cloud, such as securing and managing the operating system, network configuration, data, and connections to the cloud.

The Barracuda NextGen Firewall F is a next generation firewall built to integrate seamlessly with the AWS cloud platform. The flexibility of the NextGen Firewall allows cloud architects to easily select a reference architecture by the intended use case and size of the workload. A CloudFormation template is supplied with each reference architecture, making it easy to deploy or integrate with your current cloud resources. The NextGen Firewall adds network layer security controls, visibility, and connectivity to your cloud network. Depending on the use case, NextGen Firewall deployment is selected to satisfy the correct balance among the following criteria:

- Support for required firewall features such as firewalling or VPN
- High availability
- Scalability
- Cost optimization
- Failover or recovery times

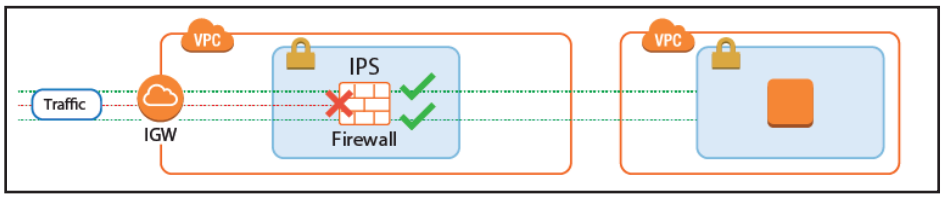
1.1.1 Barracuda NextGen Firewall F Common Use Cases

- Edge Firewall
- Secure Remote Access
- Office to Cloud / Hybrid Cloud
- Segmentation

1.1.2 Edge Firewall

Common use cases:

- Network security enforcement with firewall and IPS.
- Default (outbound) gateway for cloud resources in the same VPC.

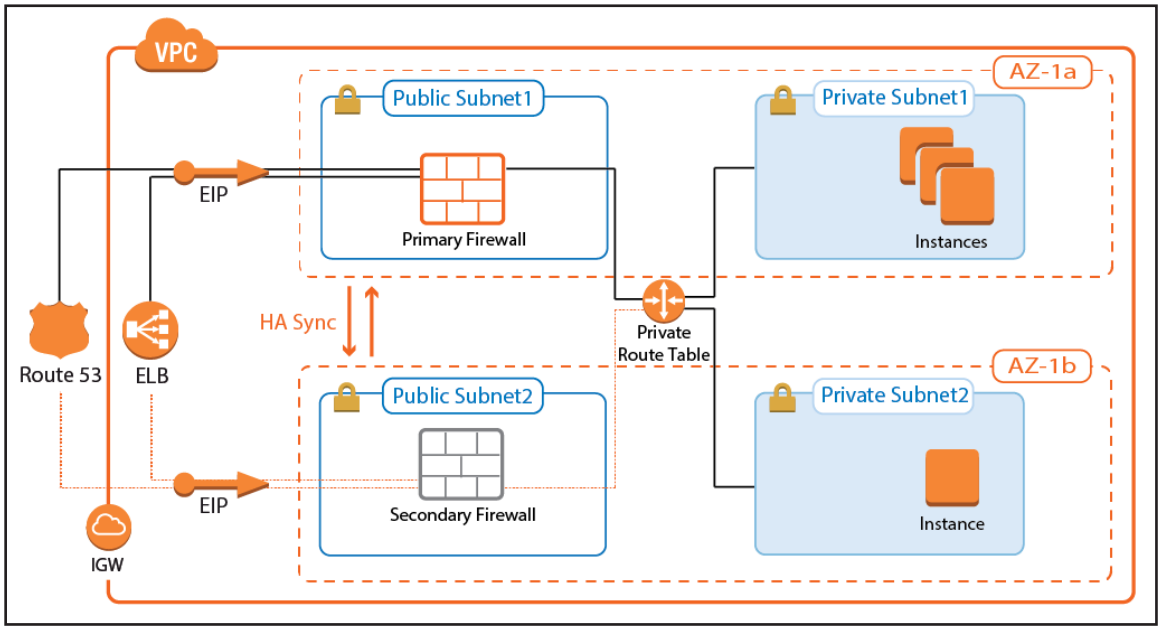


The NextGen Firewall secures access to the AWS cloud resources from the Internet by enforcing granular firewall access policies and scanning incoming traffic for malware and exploits. The next generation firewall features replace or extend the native AWS security groups and NACLs by:

- Protecting against network-based attacks and exploits with the built-in IPS
- Virus scanning and Advanced Threat Protection (ATP) (BYOL only)
- Geolocation-based access control
- Traffic Shaping (QoS) to protect business-critical traffic.

NextGen Firewall High Availability Cluster with Route Shifting

- **High Availabilty** – Yes
- **Failover / Recovery time** – Seconds to minutes, depending on the AWS API
- **Auto Scaling** – No
- **Default Gateway for instances in the VPC** – Yes

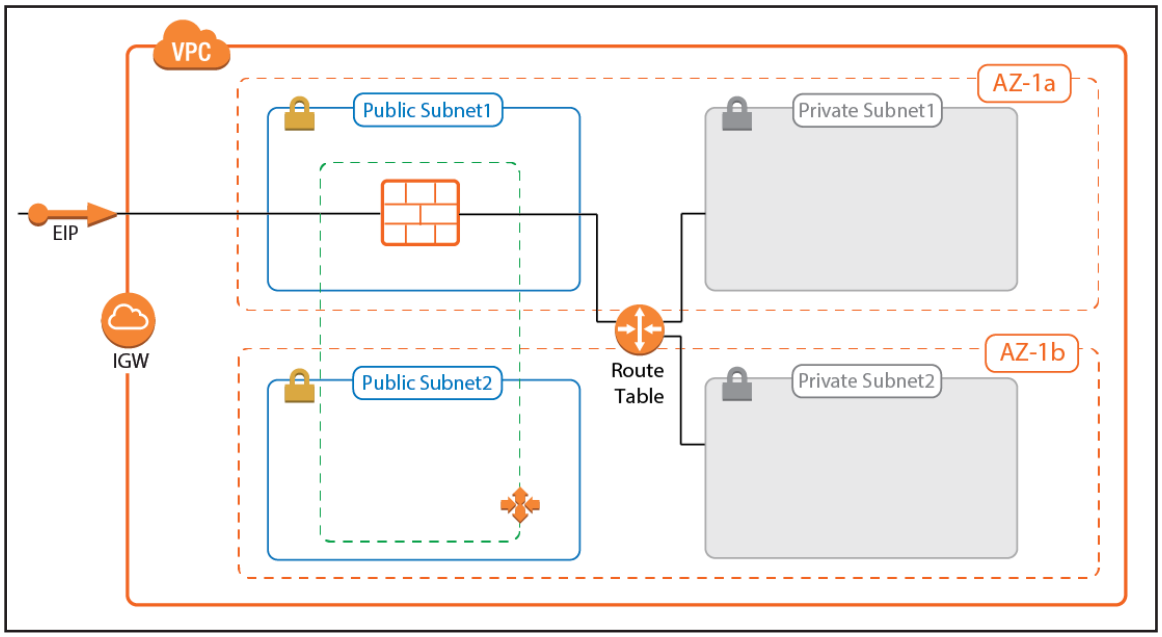


The NextGen Firewall High Availability Cluster supports all firewalling and the default gateway features required to act as an edge firewall. The firewalls are in an active-passive cluster that syncs session information and configurations. All outgoing traffic from the private subnets is routed over the active firewall. In the event of a failover, the passive firewall takes over and connects to the cloud fabric to rewrite all routes to use the now-active firewall in the High Availability Cluster as the target. Routes added after deployment that use the firewall as the gateway are automatically detected and, in the case of a failover, are also rewritten.

For more information, see 2.1 NextGen Firewall High Availability Cluster with Route Shifting (page 21)

NextGen Firewall Cold Standby Cluster

- **High Availabilty** – No
- **Failover / Recovery time** – Multiple minutes
- **Auto Scaling** – No
- **Default Gateway for instances in the VPC** – Yes, with manual changes required for new routes.

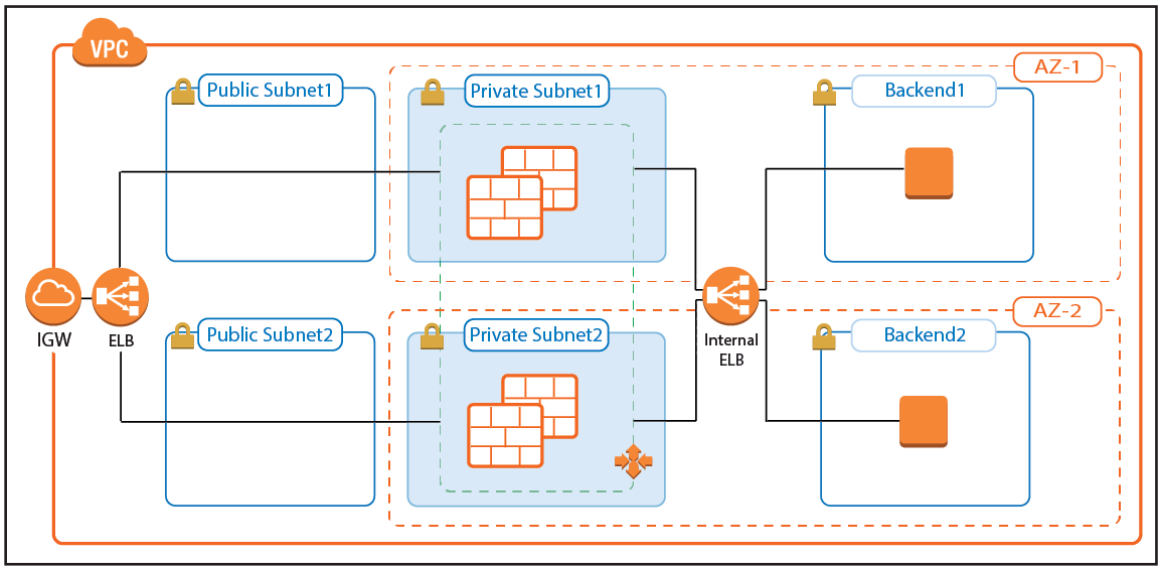


The Cold Standby Cluster is a cost-effective solution that offers the full range of next generation firewall features. In case the firewall instance becomes unresponsive, the firewall instance is automatically replaced. Routes for the private subnets are rewritten, but must be adjusted manually in the CloudFormation template to match your architecture. Route tables are not monitored automatically; additional routes or changes to existing routes must be completed by first updating the template and then updating the CloudFormation stack.

For more information, see 2.3 NextGen Firewall Cold Standby Cluster (page 49)

NextGen Firewall Auto Scaling Cluster

- **High Availabilty** – Yes
- **Failover / Recovery time** – Instant
- **Auto Scaling** – Yes
- **Default Gateway for instances in the VPC** – No, source NAT is required for inbound traffic.



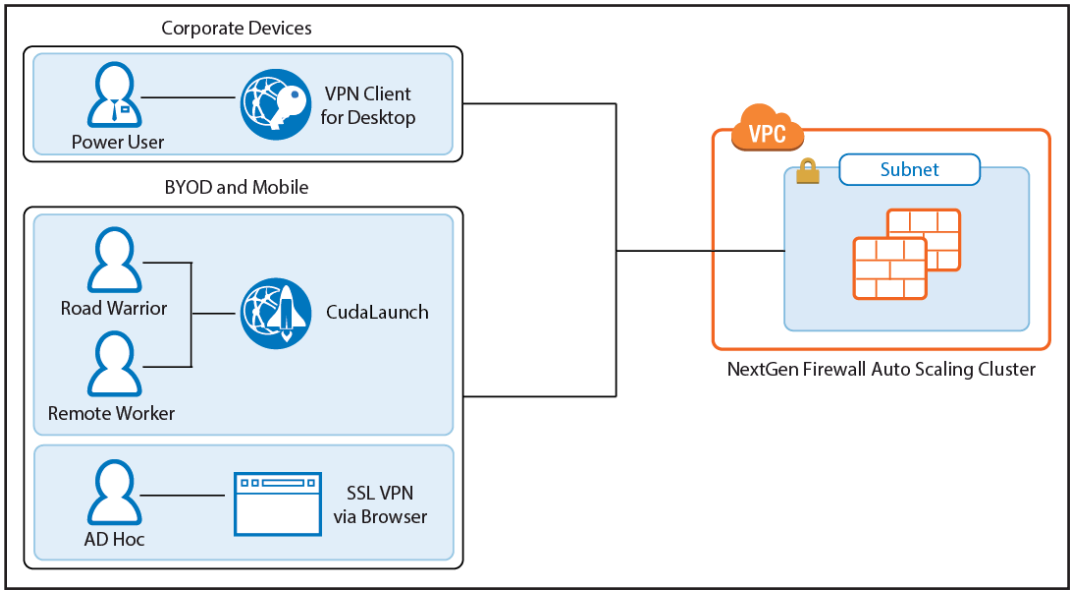
Sizing the firewall for highly dynamic traffic can be difficult. You can easily incur unnecessary costs for instances that are too large, or else you can run the risk of creating bottlenecks in your architectures if the firewall cannot keep up with current demand. The NextGen Firewall Auto Scaling Cluster scales automatically to match your workload. One or more Elastic Load Balancers distribute traffic over the firewall instances in the Auto Scaling group. Custom Firewall metrics collected by CloudWatch allow custom-tailored scaling policies that match your cloud applications. Since the source IP address must be rewritten on the firewall, the NextGen Firewall Auto Scaling Cluster cannot be used as a default gateway for outbound traffic for instances in the private networks.

For more information, see 2.2 NextGen Firewall Auto Scaling Cluster (page 29)

1.1.3 Secure Remote Access

Common use cases:

- Remote access for unknown or highly dynamic workloads.
- Remote access for predictable workloads.



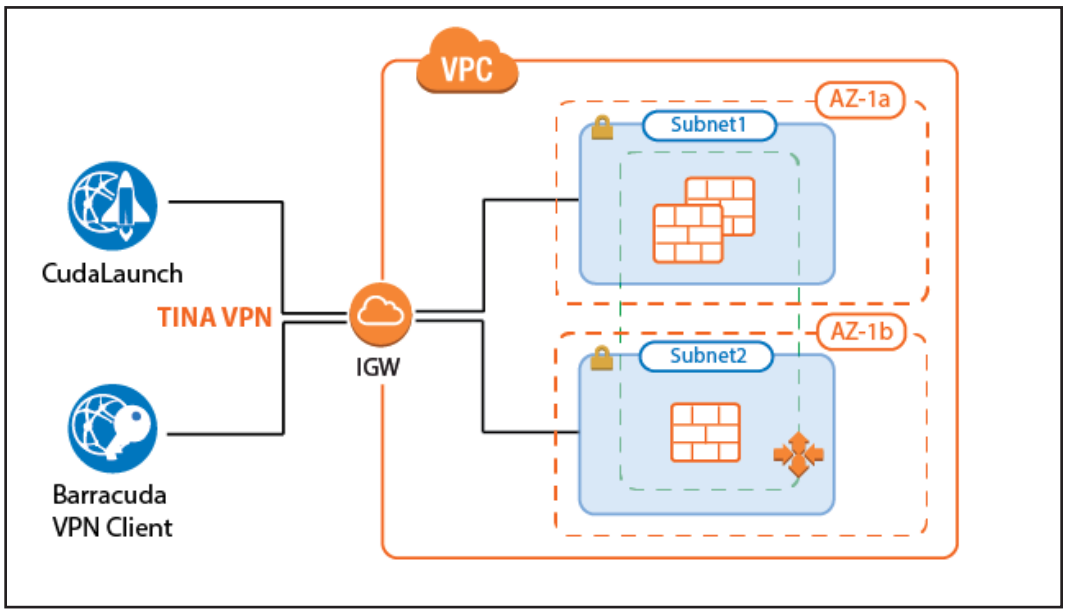
Remote access features offer remote users secure access to their organization's cloud applications and resources from virtually any device. Depending on the workload, full client-to-site VPN or SSL VPN are available, with CudaLaunch offering a richer level of remote access spanning both client-to-site and SSL VPN.

For power users, or users with centrally managed corporate devices, the client-to-site VPN offers transparent access to the corporate network. The Barracuda VPN client uses the TINA VPN protocol, specifically designed for robust VPN connections. VPN clients can be authenticated through client certificates, external and internal authentication schemes, or a combination thereof.

The SSL VPN service provides seamless integration without having to install a client app. CudaLaunch works with the SSL VPN service to provide more advanced SSL VPN features such as SSL tunneling or native app support. The number of simultaneous users using the SSL VPN is limited only by the performance of the AWS instances.

NextGen Firewall Auto Scaling Cluster

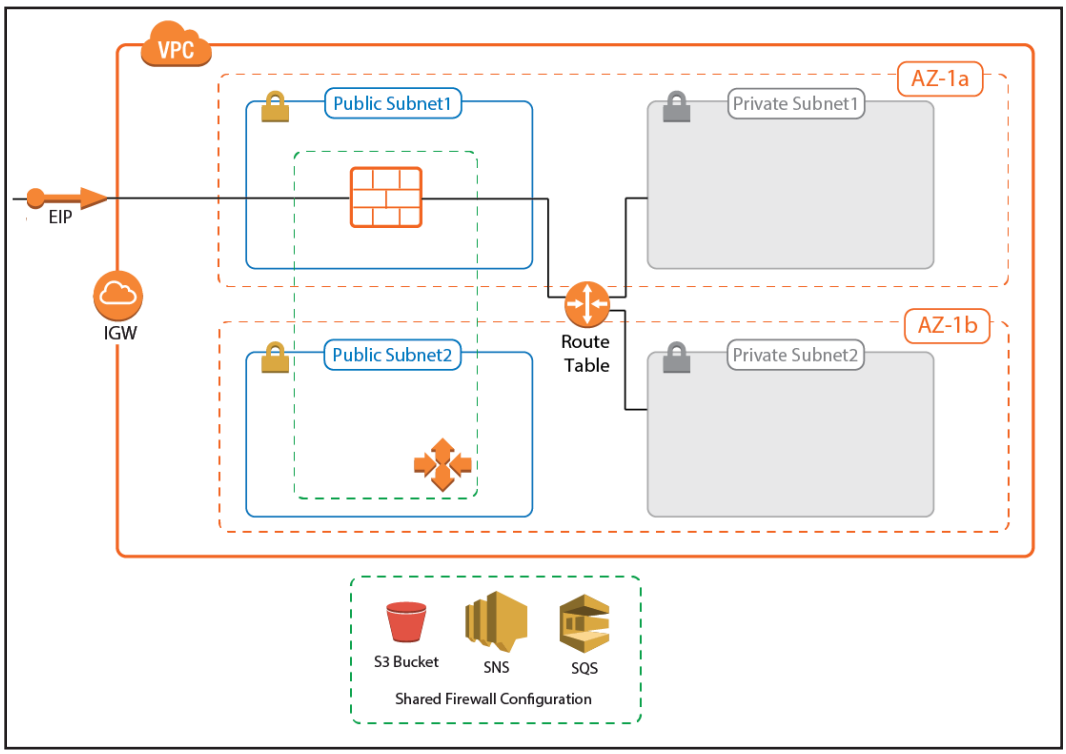
Remote access workloads tend to be cyclical in nature. Remote workers sign in with their VPN clients in the morning, and disconnect at the end of their work day. By using a NextGen Firewall Auto Scaling Cluster, the number of firewalls is scaled automatically to meet the current demand. The cluster can also be scaled according to a schedule, depending on how predictable the workload is. The firewall instances are automatically deployed into two or more Availability Zones. Custom firewall and VPN metrics collected by AWS CloudWatch allow the admin to configure customized scaling policies. Auto Scaling is limited to the PAYG images of the Barracuda NextGen Firewall F.



For more information, see 2.2 NextGen Firewall Auto Scaling Cluster (page 29)

NextGen Firewall Cold Standby Cluster

For a small number of remote users with predictable traffic patterns, the Cold Standby Cluster is a very cost-effective remote access solution. The single firewall running is automatically replaced within minutes after a failure. The configuration is stored on an S3 bucket and can optionally be fetched from a NextGen Control Center. Using a Control Center allows for the use of BYOL pool licenses for the instance. For single firewalls, the PAYG image is used. Cold Standby Clusters must be scaled up manually to meet increased demand.

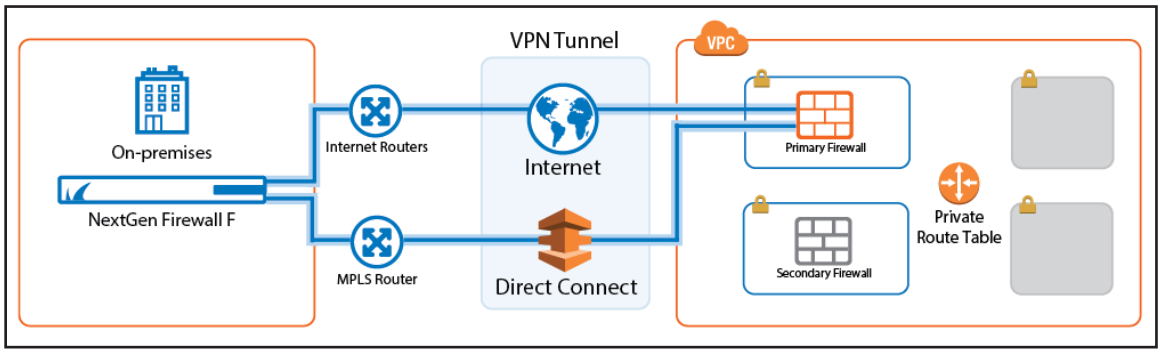


For more information, see 2.3 NextGen Firewall Cold Standby Cluster (page 49)

1.1.4 Office to Cloud / Hybrid Cloud

Common use cases:

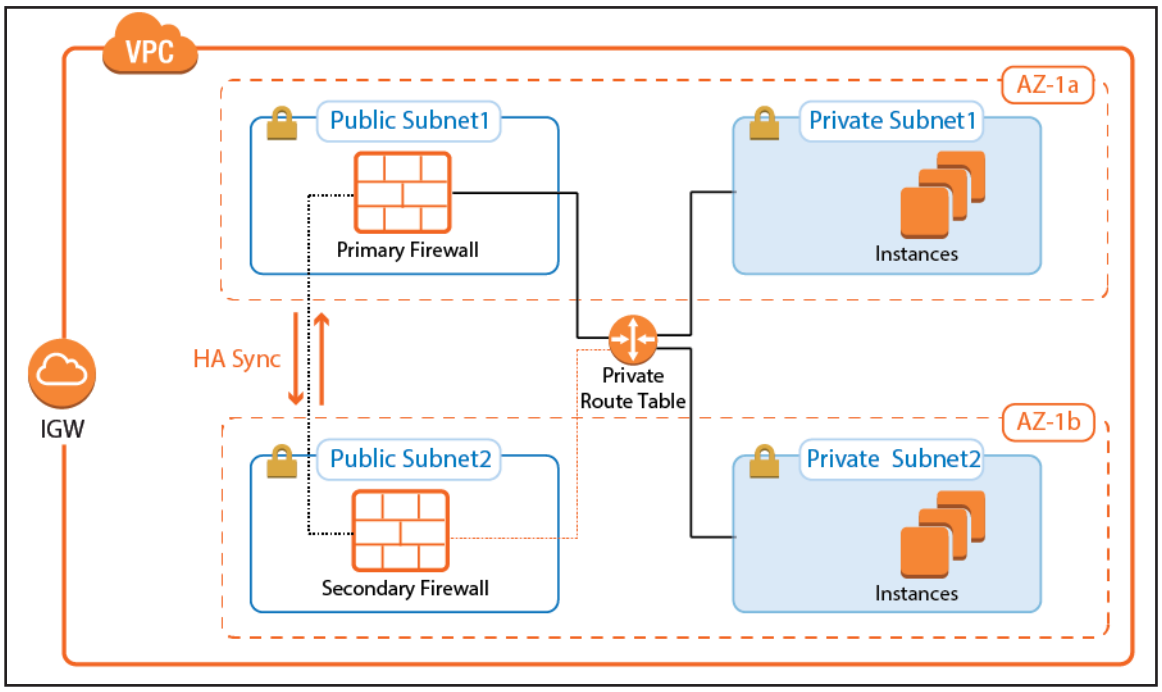
- Hybrid cloud using site-to-site VPN.
- Default (outbound) gateway for cloud resources.
- Secure traffic on the Direct Connect MPLS line.



Create site-to-site VPN connections to transparently connect your on-premises networks with your applications and services hosted in the cloud. For VPN tunnels using the proprietary TINA VPN protocol, Traffic Intelligence allows you to split a VPN tunnel into up to 24 VPN transports, each using a different WAN connection to the firewall in the cloud. For the user, this happens completely transparently. In addition, Traffic Intelligence also allows you to route traffic dynamically based on bandwidth or latency requirements. Offloading traffic to cheaper connections allows to you use smaller bandwidth Direct Connect connections, or to increase the quality for business-critical or latency-sensitive information.

NextGen Firewall High Availability Cluster

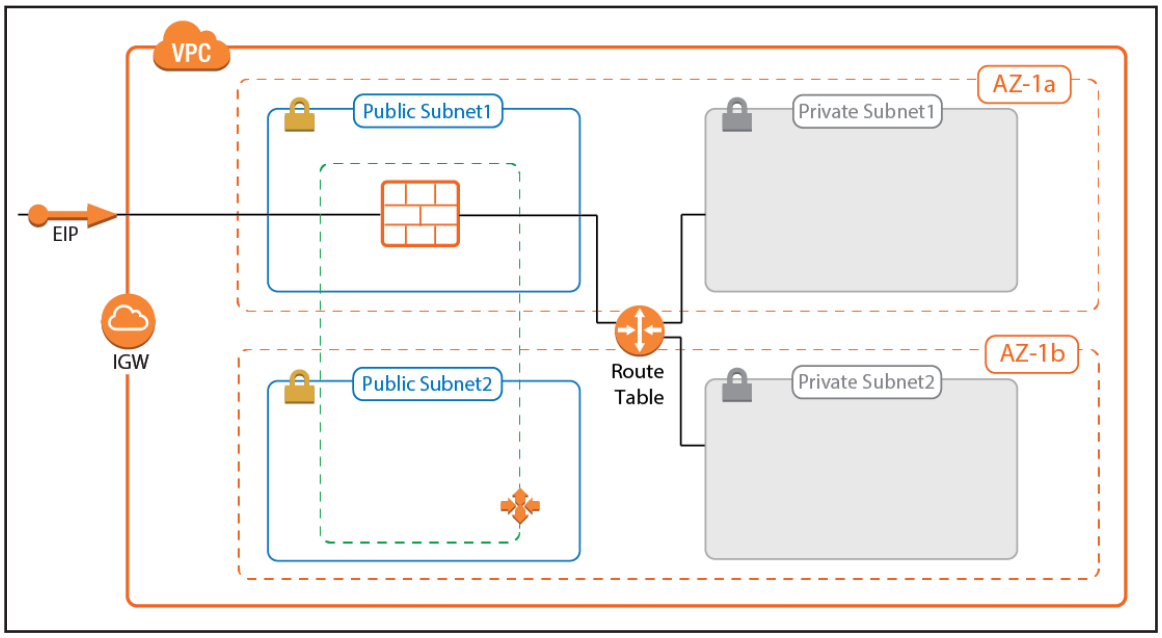
The NextGen Firewall High Availability Cluster supports both TINA and IPsec IKEv1 and IKEv2 site-to-site VPN tunnels. For IPsec tunnels, Route 53 must be used for incoming traffic since the Elastic Load Balancer does not support UDP. Optionally, a NextGen Control Center can be used to retrieve and manage the firewall configuration and to monitor the remote firewalls in one central location. If only TINA VPN tunnels are used, no incoming load balancing is required since TINA VPN tunnels can be configured to use two public IP addresses as the VPN endpoint. NextGen Firewall High Availability Clusters must be scaled up manually if the workload increases.



For more information, see 2.1 NextGen Firewall High Availability Cluster with Route Shifting (page 21)

NextGen Firewall Cold Standby Cluster

The NextGen Firewall Cold Standby Cluster supports the same VPN features as the High Availability Cluster. The single firewall instance runs in an Auto Scaling group of one with the firewall configuration stored on an S3 bucket. In case the firewall becomes unavailable, it is automatically replaced. By default, only PAYG licenses are supported. However, it is possible to use a NextGen Control Center to manage the firewall. This allows for the use of BYOL pool licenses. The Cold Standby Cluster must be sized to meet peek demand because it does not scale dynamically.



For more information, see 2.3 NextGen Firewall Cold Standby Cluster (page 49)

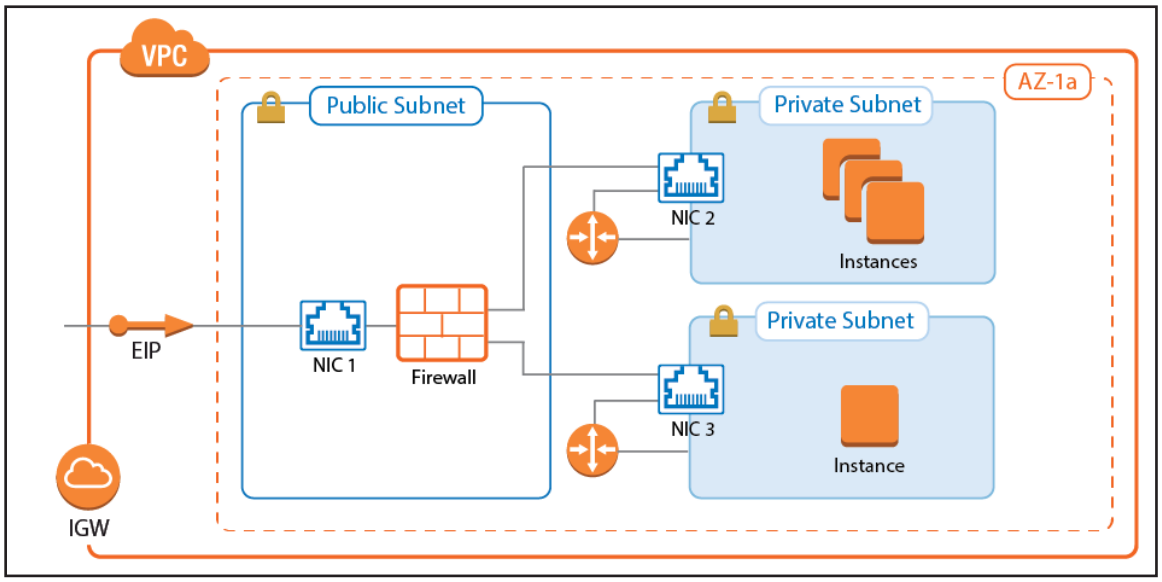
1.1.5 Segmentation

Common use cases:

- Provide network segmentation (INS) in the cloud.

Segmentation Firewall for Single AZ VPCs

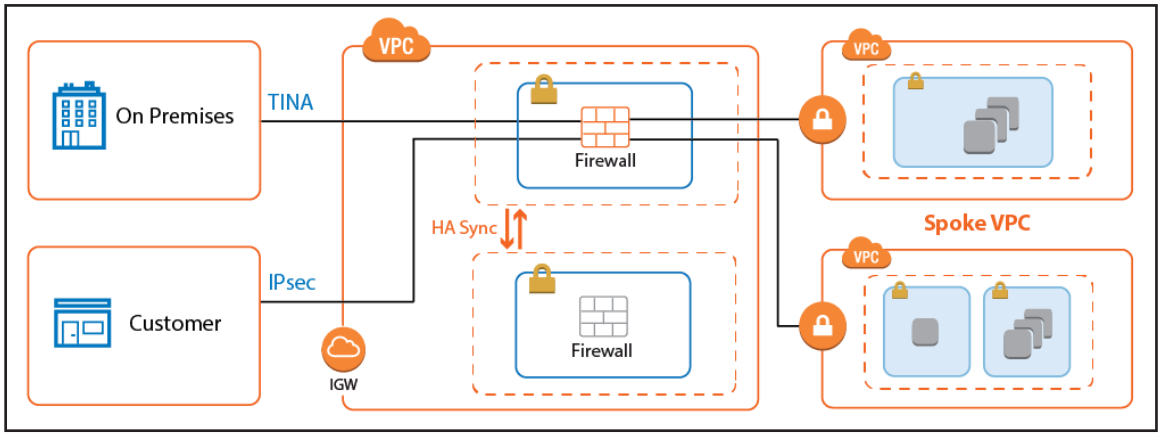
Traditional network security approaches rely heavily on network segmentation to secure the network with internal firewalls that allow only defined traffic between the different services and networks. When these on-premises applications are migrated to the cloud, the internal firewall is replaced by a NextGen Firewall with multiple network interfaces. This allows the application to be moved to the cloud without a costly and time-consuming revamp of the architecture. Firewall access rules and the next generation firewall capabilities provide fine-grained security policies and real-time traffic visibility. Since the Elastic Network Interfaces attached to the firewall instance must be in the same Availability Zone, this solution is limited to single AZ applications.



For more information, see 2.5 Segmentation Firewall for Single AZ VPCs (page 71)

Transit VPC

For cloud-native applications to take full advantage of the AWS cloud platform, each application is hosted in a dedicated VPC. This allows the application to be the logical context for segmentation. To organize and secure these highly dynamic VPCs, connect them in a hub and spoke architecture, with a firewall cluster in the central Transit VPC. The Transit VPC architecture is very flexible: it can be combined with High Availability Clusters, Cold Standby Clusters, or Auto Scaling Clusters, depending on the workload and predominant use case.

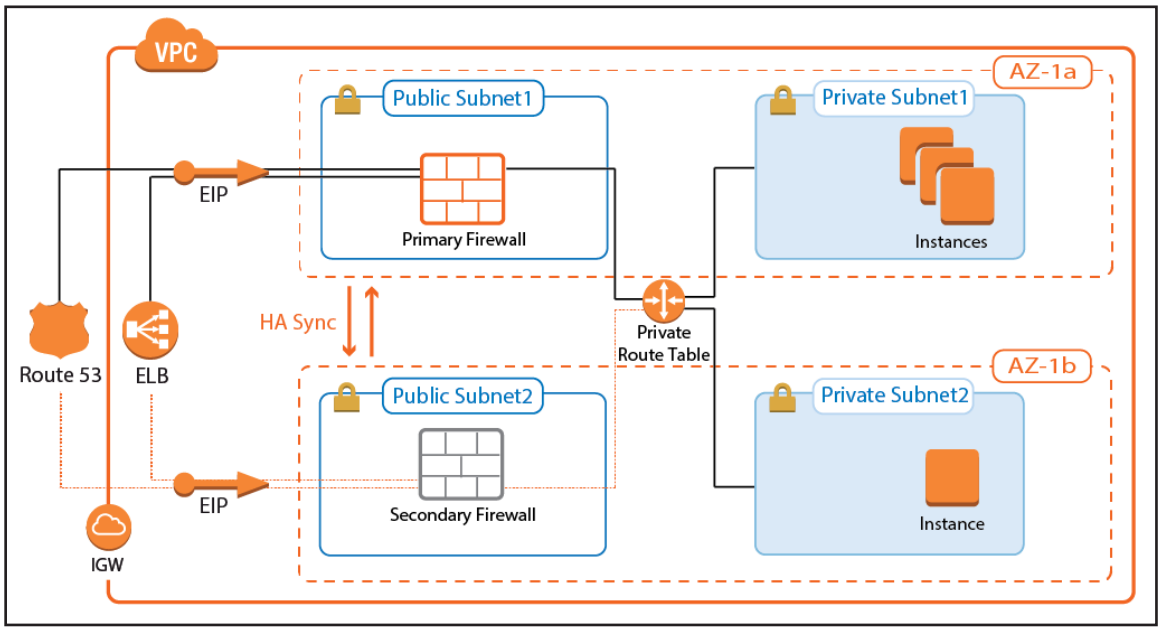


For more information, see 2.4 Transit VPC using NextGen Firewall (page 61)

2.1 NextGen Firewall High Availability Cluster with Route Shifting

To build highly available services in AWS, each layer of your architecture should be redundant over multiple Availability Zones. Each AWS region is made up of at least two isolated Availability Zones. In case one Availability Zone goes down, your application continues to run in the other datacenter without interruption or even minimal failover time. For the Barracuda NextGen Firewall, this means deploying two firewall instances to two public subnets, each in a different Availability Zone. The firewalls are in an active-passive cluster. Both firewalls share a virtual server containing such services as the Forwarding Firewall or VPN service. Should the primary firewall become unavailable, the virtual server is immediately started on the secondary firewall. The now-active secondary firewall connects to the underlying cloud platform and rewrites the routes in the AWS route table to use the now-active firewall as the gateway device for the backend instances. After the route table is rewritten, normal operations are resumed, even if one of the two Availability Zones is experiencing an outage. Failing over the virtual server, although fast, is not transparent to the user. Existing connections will time out.

High Availability Clusters must be sized for the expected peak load. If the expected workload is dynamic in nature and a default gateway is not required, use a NextGen Firewall Auto Scaling cluster instead.



2.1.1 Use Cases for a NextGen Firewall High Availability Cluster

- **Site-to-Site VPN** – One way on-premises to AWS, TINA, and IPsec site-to-site VPN tunnels.
- **Edge Firewall** – Scan for malicious traffic using the built-in IPS and handle access to resources via access rules.
- **Secure Remote Access** – Client-to-site VPN, CudaLaunch, and SSL VPN using TINA, SSL VPN, and IPsec VPN protocols.

2.1.2 Deploying a High Availability Firewall Cluster via CloudFormation Template

It is recommended to deploy the High Availability Cluster via a CloudFormation template. The template deploys two firewalls that are automatically joined into the High Availability Cluster in the public subnets. The route table associated with the private subnets is configured to use the active firewall as the outbound gateway.

1. Create an IAM role for the firewall cluster. For step-by-step instructions, see

   3.1 How to Create an IAM Role for an F-Series Firewall in AWS (page 79)

   Download the **NGF_HA.json** template and parameter file from the Barracuda Network GitHub account:

   https://github.com/barracudanetworks/ngf-aws-templates.
2. Accept the Software Terms for the **Barracuda NextGen Firewall PAYG** or **BYOL** image in the AWS Marketplace.
3. Create a parameter template file containing your parameters values.
4. Deploy the **NGF_HA.json** CloudFormation template via AWS CLI or AWS console.

```
aws cloudformation create-stack  --stack-name "YOUR_STACK_NAME"
--template-body YOUR_S3_BUCKET/NGF_HA.json --parameter YOUR_S3_
BUCKET/NGF_HA_parameters.json
```

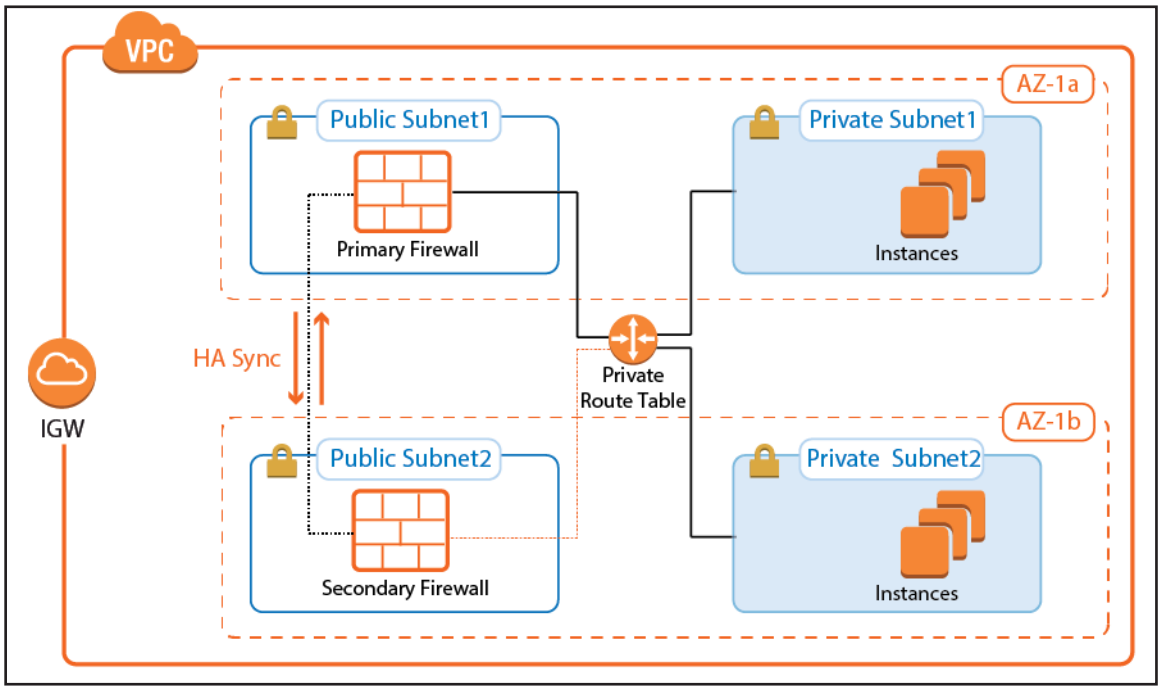
During deployment, the following resources are created by the template:

- Two public and two private subnets in a VPC. The subnets are spread out over multiple Availability Zones.
- Two NextGen Firewall (PAYG or BYOL) instances joined together into a High Availability Cluster.
- One Elastic Load Balancer.

For step-by-step instructions, see 3.10 How to Deploy an F-Series Firewall in AWS via CloudFormation Template (page 139)

2.1.3 (Alternative) Deploying a High Availability Firewall Cluster via AWS Console

To deploy a NextGen Firewall High Availability Cluster via AWS Console, follow these basic steps:



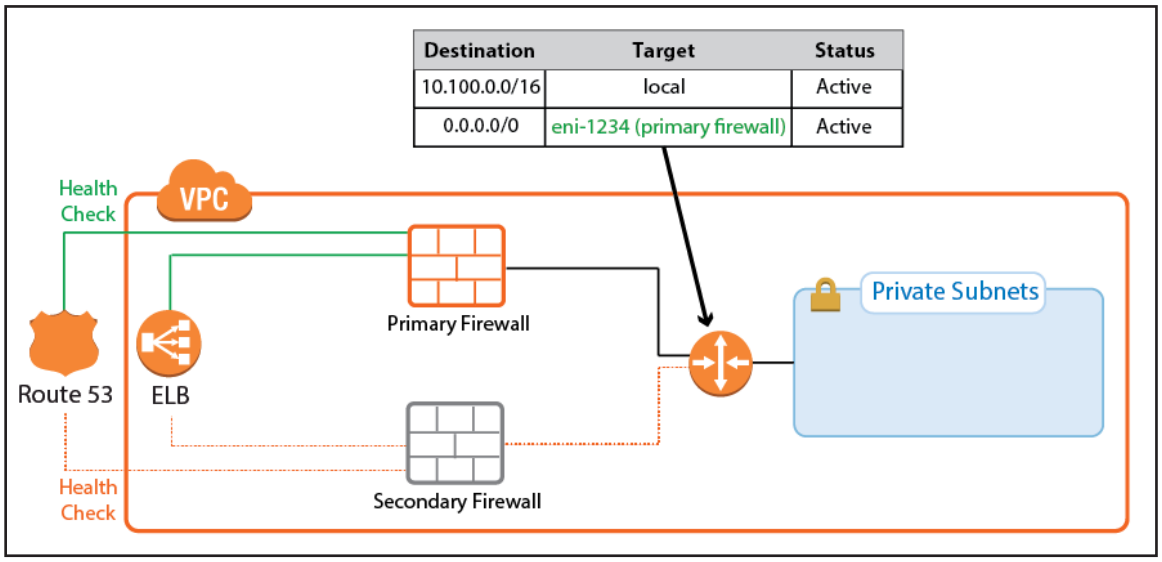
1. Create an IAM role for your firewall instances.
2. Create a VPC and add two public and private subnets in two Availability Zones.
3. Attach an Internet gateway and associate one route table with the public subnets, the second with the private subnets.
4. Launch one firewall instance into each public subnet. Both firewalls require public IP addresses.
5. Disable the source/destination check for each firewall.
6. Add routes to the route table to allow the public subnets Internet access and the private subnets to route over the active firewall instance.
7. Join the two firewalls into an High Availability Cluster.
8. Add an Elastic Load Balancer or configure Route 53.

For step-by-step instructions, see How to Configure a Multi-AZ High Availability Cluster in AWS using the Web Portal and How to Set Up a High Availability Cluster.

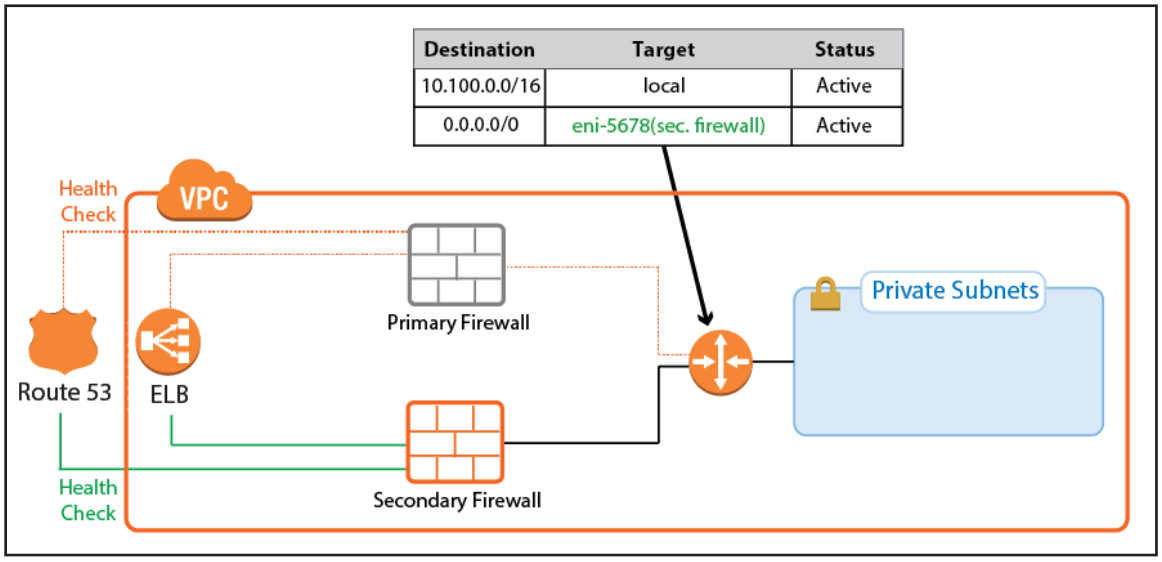
2.1.4 Cloud Integration for Route Shifting

Cloud Integration allows the firewall instance to use API calls to the underlying cloud platform authenticating by using the IAM role supplied during deployment. Cloud Integration is used to populate the cloud information element in the NextGen Admin dashboard and, more importantly, to rewrite AWS route tables. Rewriting the VPC route tables is necessary every time the virtual server fails over. During the failover, the now-active firewall rewrites the target of every route to use the active firewall running the virtual server services. This works for all route tables in the VPC. The active firewall continues to poll the route tables to ensure that the active firewall is always used.

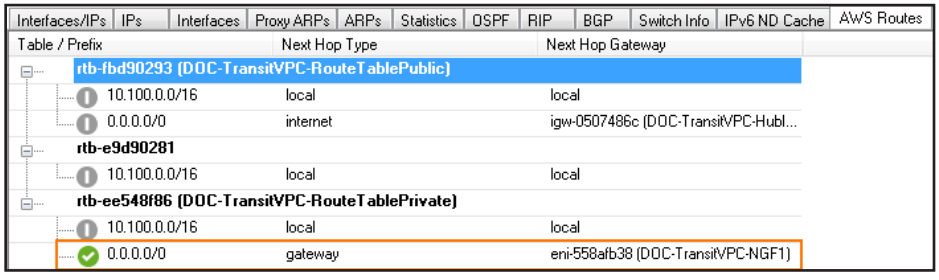
Primary Firewall Active



Secondary Firewall Active



On the firewall, go to **CONTROL > Network > AWS Routes**. All the route tables for the VPC are listed. Routes that use one of the firewalls are shown with a green icon. During takeover, the icon temporary turns red to indicate that a failover is in progress. After the route table rewrite, the network interface ID (eni-123456) matches the now-active firewall.



For step-by-step instructions, see Cloud Integration for AWS.

2.1.5 Single Endpoint for Incoming Traffic: Route 53 or Elastic Load Balancer

Using two public IP addresses for the active-passive High Availability Cluster may not always be possible. To use a single FQDN that always sends traffic over the active firewall, you can use either a classic Elastic Load Balancer or Route 53. Both services are similar in that they use health checks and send traffic to the healthy destination. For TCP-only services, either service can be used. For UDP-based services, such as IPsec, use Route 53.

Classic Elastic Load Balancer

The classic Elastic Load Balancer is a managed layer 4 TCP load balancer. The load balancer can only be addressed by the DNS name associated with it. It is not possible to work with the IP address the hostname resolves to directly because the underlying load balancing instances may change at any time.

The Elastic Load Balancer is responsible for distributing traffic to all healthy instances it is associated to. To make sure that traffic is sent only to the active firewall, define the health check for a service on the virtual service. For example, use TCP:691 as the health check target if a VPN service is running on the virtual server. The load balancer continuously polls the VPN service and considers the instance healthy if the TCP connection succeeds. Since the virtual server is running only on the active firewall, the health check always fails for the passive firewall. The passive firewall is considered unhealthy, and no traffic is forwarded to this instance by the load balancer.

Traffic passing through an Elastic Load Balancer rewrites the source IP address to that of the load balancer instance. If your application requires the public IP address of the client, use Route 53 instead.

For step-by-step instructions, see

3.14 How to Configure an AWS Elastic Load Balancer for F-Series Firewalls in AWS (page 165)

Route 53

Route 53 is an authoritative DNS service by AWS. Route 53 allows you to monitor endpoints and change the returned record set according to the state of the health check. Create a health check for a service running on the virtual server of your High Availability Cluster. Create two record sets using a failover routing policy and attach the health check to the primary firewall. No distinct health check is created for the secondary firewall. If everything fails, it is better to attempt to reach at least one firewall in the cluster than to return nothing at all. The secondary firewall is also a better choice as a fail-safe because the default behavior of a High Availability Cluster favors the secondary firewall. For example, if both the primary and secondary firewall start the virtual server at the same time, the secondary firewall continues to run while the primary firewall shuts the virtual server down.

For step-by-step instructions, see 3.15 How to Configure Route 53 for F-Series Firewalls in AWS (page 169)

2.1.6 Control Center-Managed NextGen Firewall High Availability Cluster

The NextGen Control Center is a central management appliance for the F-Series Firewall that can be deployed as a virtual appliance on-premises or in the cloud. Managing the High Availability Cluster with a NextGen Control Center separates the firewall configuration and monitoring from deployment and integration with other AWS services. This is especially useful for highly specialized or large departments with dedicated network security teams and multiple developer teams using automatic deployments. Managed firewalls are preconfigured on the Control Center. During provisioning of the firewall instance, the firewall configuration and, optionally, licenses are automatically retrieved from the Control Center. To use BYOL licenses, pool licenses bound to Control Center are used instead of single BYOL licenses bound to the EC2 instance of the firewall. Pool licenses are available in multiples of 5.

For step-by-step instructions, see

3.6 How to Modify CloudFormation Templates to Retrieve the PAR File from a Control Center (page 105)

For more information, see NextGen Control Center and Central Management.

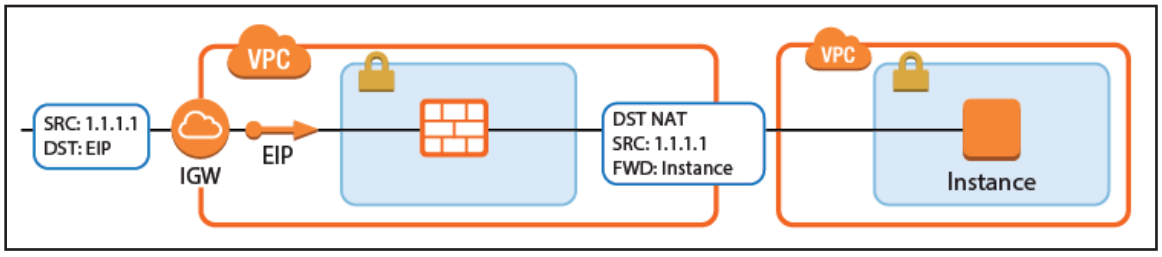
2.1.7 Create Access Rules

By default, the Forwarding Firewall service blocks all traffic. To allow traffic through the firewall, you must create access rules with an allow action, such as Pass or Dst NAT. When creating the rules, make sure you create them so they will match the same type of traffic independent of which virtual server the firewall service is running on. For Dst NAT and App Redirect rules, enter both the management IP address of the primary and secondary firewalls, or use the **All Firewall IPs**.

For step-by-step instructions, see Access Rules.

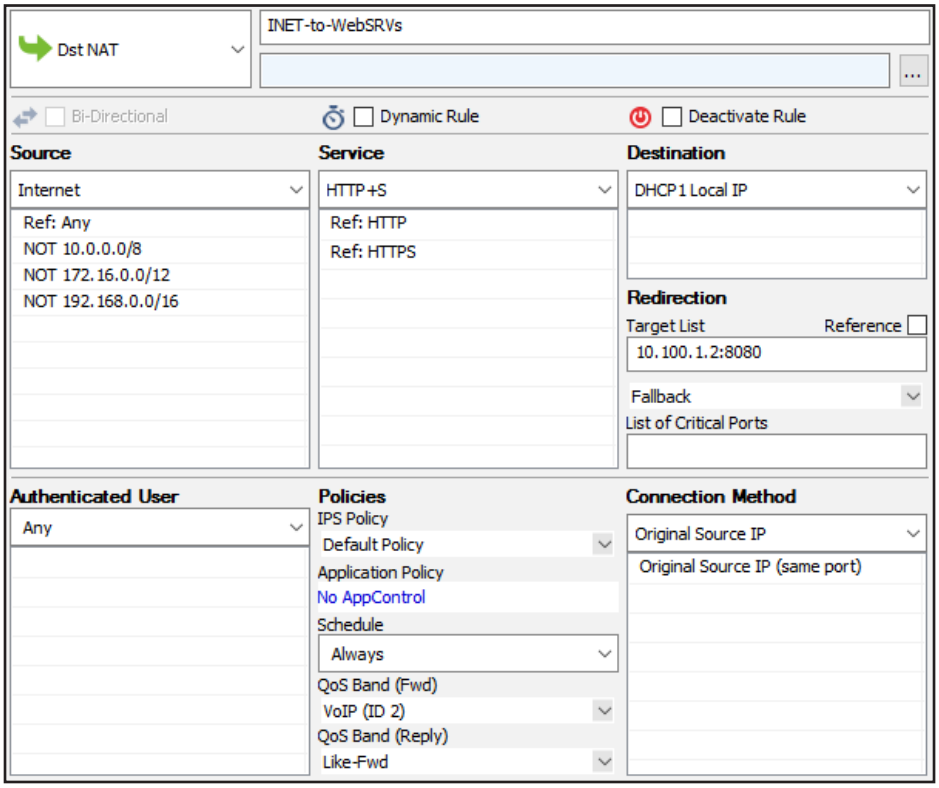
Internet to Backend Services Using the Firewall as the Default Gateway

Create the following access rule to forward traffic from the Internet to an internal web server, where the web server uses the firewall as the default gateway.



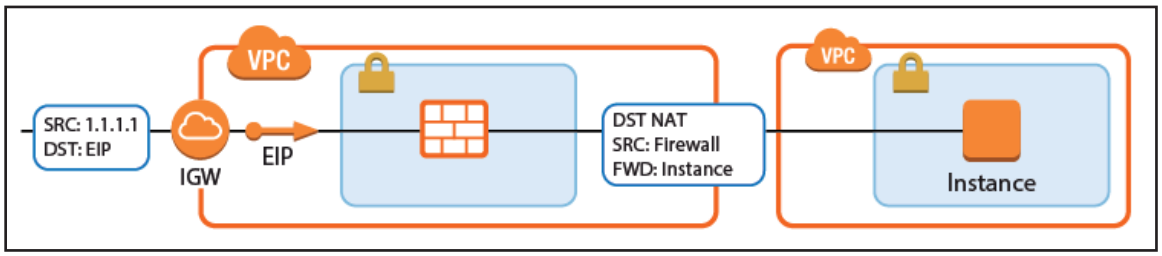
- **Action** – Select **Dst NAT**.
- **Source** – Select the source depending on how traffic is routed to the firewall:
  - **Through an ELB** – Select **Any** or the network object containing the networks the ELB is deployed in.
  - **Through Route 53 / Elastic IP** – Select **Internet**.

- **Destination** – Select **DHCP1 Local IP**.
- **Connection Method** – Select **Original Source IP**.
- **Redirection Target** – Enter the IP address of the backend service. Optionally, append the port number to redirect to a different port. e..g, `10.100.1.2` or `10.100.1.2:8080`



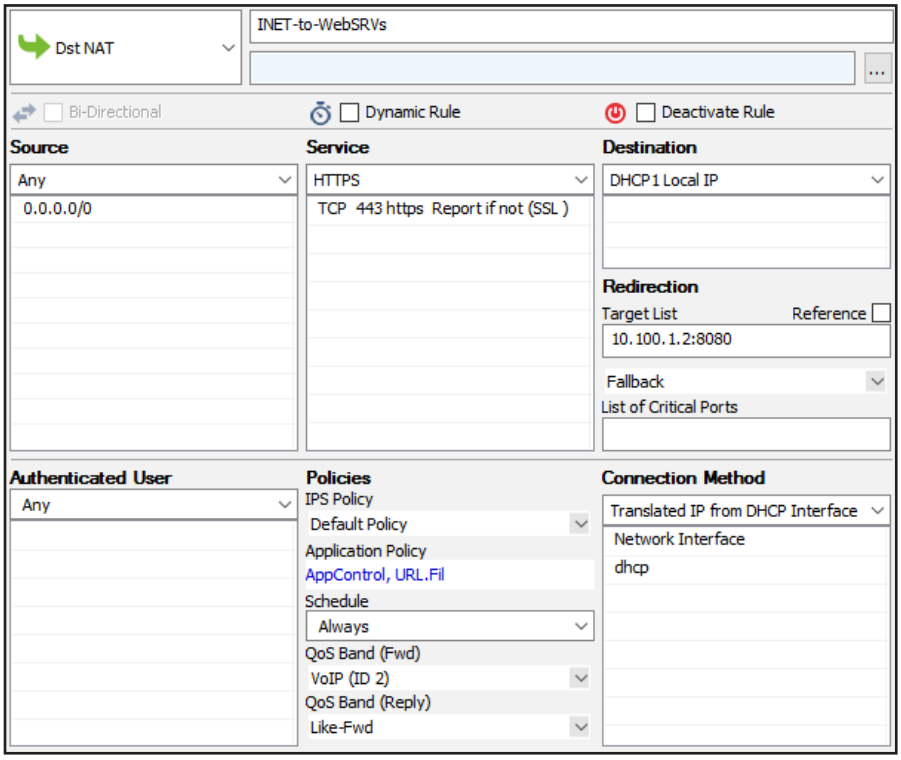
Internet to Backend Services not Using the Firewall as the Default Gateway

Create the following access rule to forward traffic from the Internet to an internal web server.



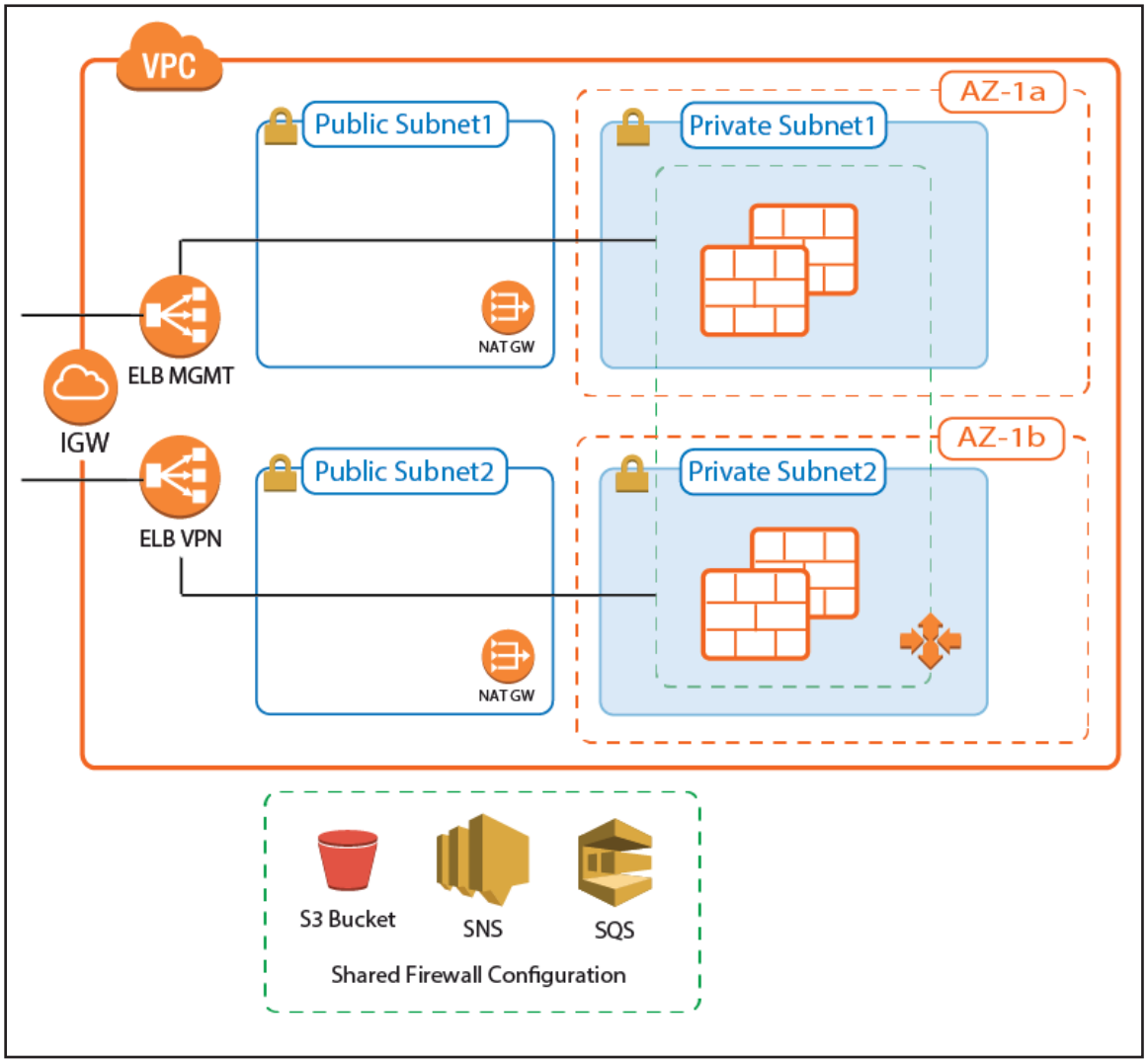
- **Action** – Select **Dst NAT**.
- **Source** – Select the source depending on how traffic is routed to the firewall:
  - **Through an ELB** – Select **Any** or the network object containing the networks the ELB is deployed in.
  - **Through Route 53 / Elastic IP** – Select **Internet**.
- **Service** – Select the service. e.g., **HTTP+S**.
- **Destination** – Select **DHCP1 Local IP**.
- **Connection Method** – Select **Dynamic NAT** or **Translated from DHCP Interface**.

- **Redirection Target** – Enter the IP address of the backend service. Optionally, append the port number to redirect to a different port. e..g, `10.100.1.2` or `10.100.1.2:8080`



2.2 NextGen Firewall Auto Scaling Cluster

Protecting highly dynamic AWS resources with a static firewall setup is neither efficient nor economical. A NextGen Firewall Auto Scaling Cluster scales with demand, thereby creating a cost-effective, robust solution for securing and connecting to your cloud resources. The firewall cluster can be deployed either to integrate with existing resources in an AWS region, or as part of an auto scaling application. Both options offer an integrated Barracuda Web Application Firewall (WAF) as a second security tier. The firewall cluster integrates tightly with AWS services and APIs. Configuration changes are synchronized securely over the AWS backend, with all instances sharing the same configuration. The admin can configure the changes like a single firewall instance. The firewall cluster is highly available and scalable over multiple AWS Availability Zones, without any single point of failure such as additional management or worker node instances. The firewall cluster uses the PAYG image of the Barracuda NextGen Firewall in the AWS Marketplace. This allows you to quickly deploy without the need for long-term licensing commitments. NextGen Firewall clusters cannot be managed by a NextGen Control Center.



2.2.1 Use Cases for a NextGen Firewall Cold Standby Cluster

- **Secure Remote Access** – Client-to-site VPN, CudaLaunch, and SSL VPN using the TINA VPN protocol.
- **Edge Firewall** – Scan for malicious traffic using the built-in IPS and handle access to resources via access rules.

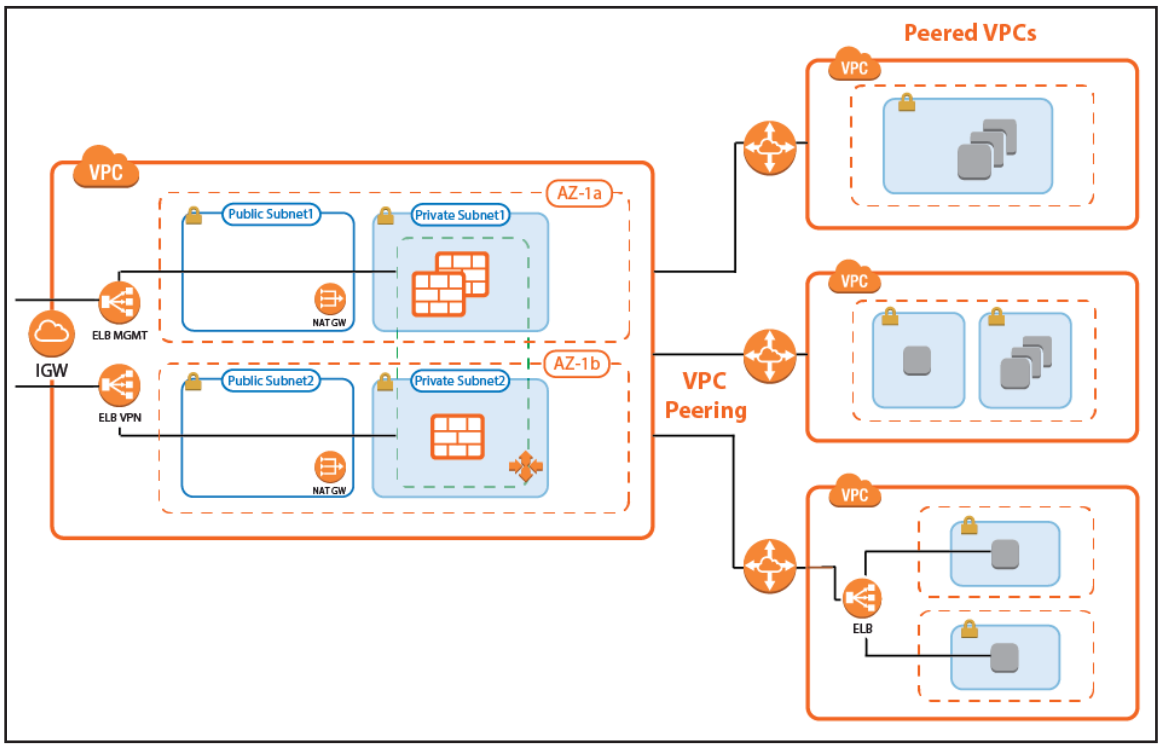
2.2.2 AWS Architectures for NextGen Firewall Auto Scaling Clusters

Since there are no external dependencies, the NextGen Firewall cluster can either be used as a drop-in solution to protect your existing applications in the same AWS region, or it can be included as part of the architecture of your application.

Transit VPC with VPC Peering

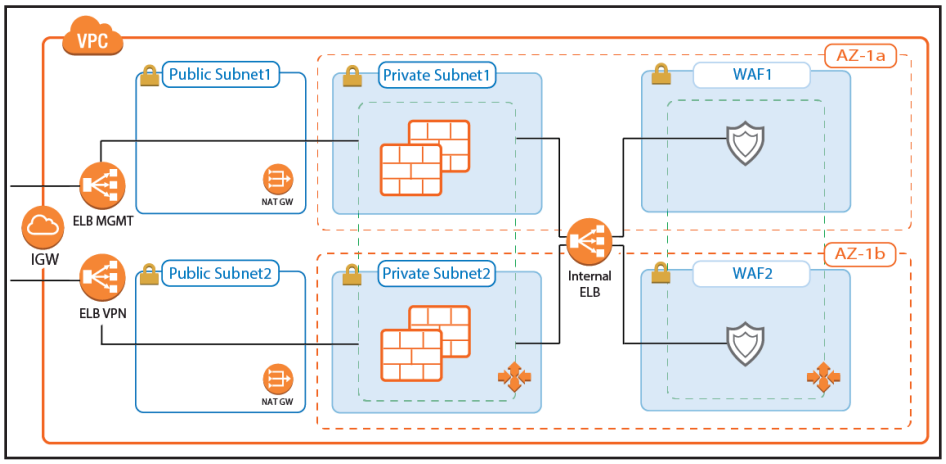
The firewall cluster is used in a Transit VPC configuration. The firewall VPC acts as a hub securing all traffic in and out of the peered VPCs. Two peered VPCs must be in the same AWS region, but can be in different AWS accounts. Transitive peering is not possible; therefore, resources in two VPCs both peered with the Transit VPC cannot communicate with each other. Incoming traffic is handled via access rules allowing access to the backend resources based on the access rule matching criteria, such as source, user, or time. Since the VPC for the firewall cluster is separated from the VPCs containing the applications, rapid iteration of the applications is possible without requiring changes to the firewall cluster. For example, in a typical scenario with production, engineering, and development VPCs, granular access rules allow the firewall admin to separate users based on their role:

- Traffic to production VPCs is secured by IPS and, optionally, forwarded to a Web Application Firewall cluster.
- QA and developers log in via client-to-site VPN. The firewall uses the user information to allow access only to their respective VPCs.
- Admins are in a special group, allowing them backend access to production and QA VPCs.



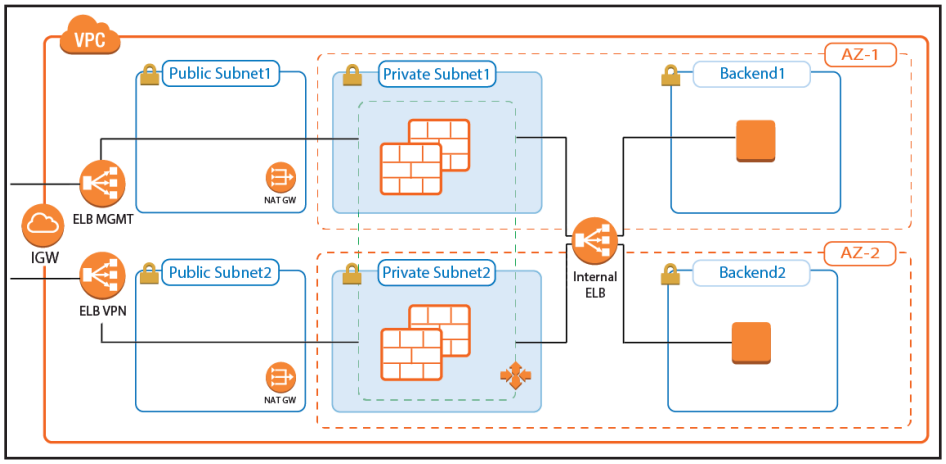
Transit VPC with VPC Peering and Barracuda Web Application Firewall Auto Scaling Cluster

A variation of the transit VPC includes an additional Web Application Firewall cluster behind the NextGen Firewall cluster. The Barracuda Web Application Firewall and NextGen Firewall F can work in tandem to block IP addresses from which malicious activity was detected. Whereas the WAF is very good at detecting application layer attacks, the NextGen Firewall is more efficient on the network layer. Connections blocked by the firewall IPS are never forwarded to the WAF, thereby freeing resources that would otherwise have to be used to block known-bad connections.



Integration into AWS Architecture

You can integrate the firewall cluster into your existing architecture. Use the default CloudFormation template as reference. To be able to reuse the configuration, configure the NextGen Firewall cluster one time via NextGen Admin, and then replicate the S3 bucket to reuse the configuration.



2.2.3 Deploying a NextGen Firewall Auto Scaling Cluster

The firewall cluster must be deployed via CloudFormation template. The template deploys a VPC with public and private subnets in two Availability Zones. In the private subnets, the firewall cluster is deployed. In the public subnets, the Elastic Load Balancer (ELB) and two NAT gateways are deployed (one for each Availability Zone). The NAT gateways are required for the firewalls to be able to access the AWS backend. APIs are required to enable the secure configuration sync over the AWS backend.

Create an IAM role for the firewall cluster. For step-by-step instructions, see 3.1 How to Create an IAM Role for an F-Series Firewall in AWS (page 79)

Download the **NGF_Autoscaling.json** template and parameter file from the Barracuda Network GitHub account: https://github.com/barracudanetworks/ngf-aws-templates.

Accept the Software Terms for the **Barracuda NextGen Firewall PAYG** image in the AWS Marketplace.

Create a parameter template file containing your parameters values.

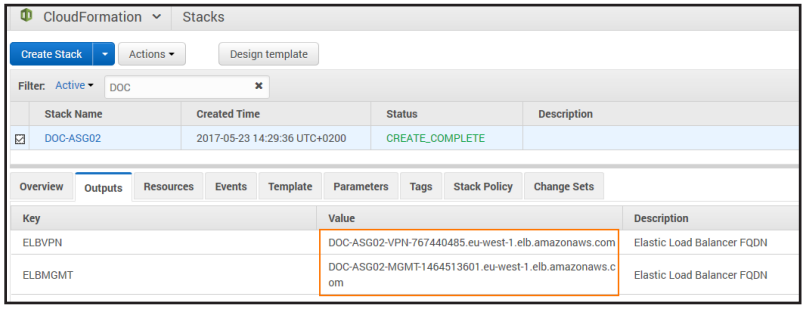
Deploy the **autoscale.json** CloudFormation template via AWS CLI or AWS console.

```
aws cloudformation create-stack  --stack-name "YOUR_STACK_NAME" --template-body
YOUR_S3_BUCKET/NGF_Autoscaling.json --parameter YOUR_S3_BUCKET/NGF_Autoscaling_
parameters.json
```

During deployment, the following resources are created by the template:

- VPC with private and public subnets in two Availability Zones.
- Two ELBs: one for management connections, the other for VPN and SSL VPN services.
- One S3 bucket.
- Automatically created SNS and SQS queues.
- Two NAT gateways.
- A Launch Configuration and Auto Scaling group for the firewall. The Barracuda NextGen Firewall PAYG image must be used.
- Scaling policies using the number of client-to-site VPN tunnels.

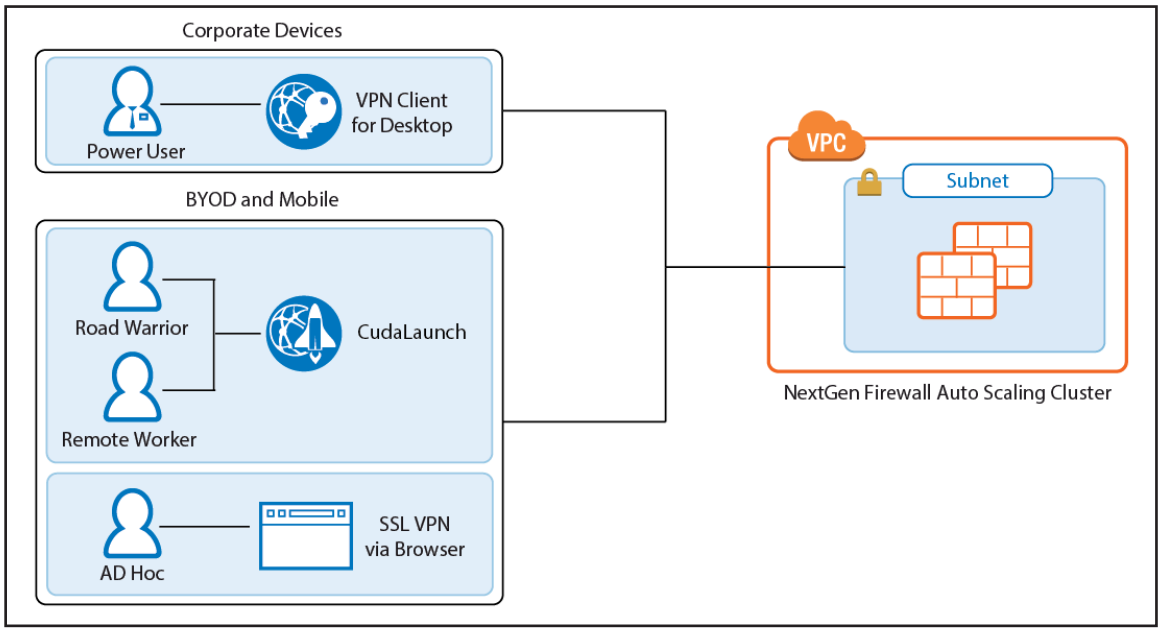
After stack creation is complete, the FQDN of the ELB are listed in the **Output** tab.



For step-by-steps instructions, see 3.12 How to Deploy a NextGen Firewall Auto Scaling Cluster in AWS (page 155)

2.2.4 Remote Access

Remote Access features offer remote users secure access to their organization's cloud applications and resources from virtually any device. Depending on the type of access users require, they can choose between the full client-to-site VPN or the SSL VPN web portal.



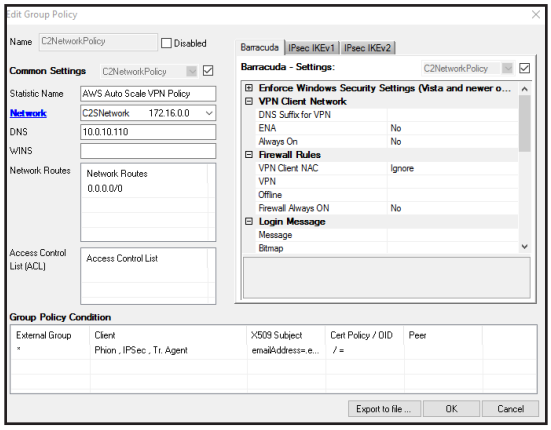
Client-to-Site VPN

The client-to-site VPN uses the TINA VPN protocol on TCP port 691 to connect to the firewall cluster. TINA is designed to overcome limitations imposed by the IPsec protocol and offers immunity to NAT devices or proxies, heartbeat monitoring, and fast failover support. VPN clients can be authenticated through client certificates, external and internal authentication schemes, or a combination thereof. Supported VPN clients are:

- **Barracuda VPN / NAC Client** for Windows, macOS, Linux, and OpenBSD.
- **CudaLaunch** for Windows, macOS, iOS, and Android version 2.3.0 or higher.

On the NextGen Firewall Auto Scaling Cluster, configure the VPN service for client-to-site connections by adding one or more VPN group policies. Incoming client-to-site connections are matched to a VPN group policy based on the group policy condition. The first matching VPN group policy is chosen. VPN group policy conditions allow you to define the following criteria:

- Group patterns from external authentication schemes
- X.509 certificate conditions
- VPN clients
- Source IP address or network for the VPN clients



For step-by-step instructions, see 3.16 How to Configure a Client-to-Site VPN Group Policy for a NextGen Firewall Auto Scaling Cluster in AWS (page 175)

SSL VPN and CudaLaunch

The SSL VPN service provides seamless integration without having to install a client app. For a richer level of remote access, CudaLaunch works with the SSL VPN service to provide more advanced SSL VPN features such as SSL tunneling or native app support. The number of simultaneous users using the SSL VPN is limited only by the performance and number of firewall instances in the Auto Scaling group. Since the SSL VPN service is not designed to share session information between the members of the Auto Scaling group, the ELB must be configured to use sticky sessions and SSL offloading to ensure that the individual client will always be redirected to the same firewall instance. SSL VPN resources can be accessed by the following clients:

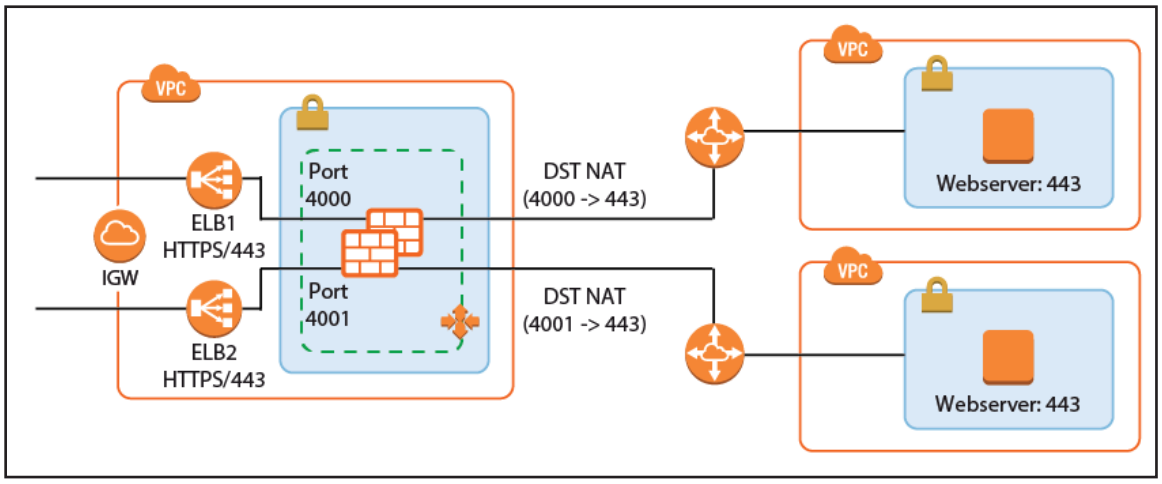
- **CudaLaunch**
- **SSL VPN web interface** – All modern browsers.

For step-by-step instruction, see 3.17 How to Configure the SSL VPN Services for AWS Auto Scaling Clusters (page 183)

2.2.5 Firewall and IPS

The firewall cluster secures incoming and outgoing traffic from your AWS resources. This can be traffic from AWS instances in peered VPCs or instances in the private networks of the VPC. If enabled on the access rule matching the traffic, the IPS engine on the firewall continuously compares the packet stream with the internal signatures database for malicious code patterns. If malicious packets are identified, traffic is either reported or dropped, depending on the configuration of the IPS. To ensure that the latest patterns are used, the IPS patterns are updated automatically from the Barracuda download servers.

When used in combination with a Barracuda Web Application Firewall cluster, the IPS and access rules block network layer attacks, saving processing power on the WAF for layer 7 attacks. For traffic to be able to flow through the firewall cluster and back, all access rules must use both source and destination IP address translation (NAT). This ensures that traffic will go back over the same firewall. The NextGen Firewall Auto Scaling Cluster cannot be used as the default gateways for your AWS resources.



2.2.6 Configuration and Monitoring

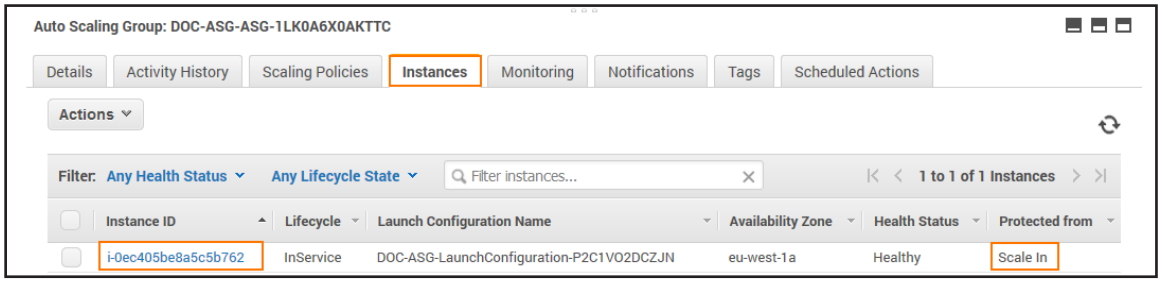
Barracuda NextGen Admin is a stand-alone, multi-administrator Microsoft Windows application used to administer NextGen Firewalls. Managing the configuration of the firewall cluster is very similar to managing a single firewall. Connect to the cluster through the ELB with a listener on TCP 807. This instance now transparently redirects the connections to other instances in the cluster as needed. Information on some tabs, such as the **CONFIGURATION** and **VPN** tabs, are aggregated by combining the data from all firewalls in the cluster. The **FIREWALL > History** page also displays connection data from all firewalls in the cluster. All other tabs and configuration elements display only the information of the firewall instance NextGen Admin is currently connected to.

- **Aggregated tabs** – All pages in the **CONFIGURATION** tab.
- **Aggregated pages** – **VPN > Client-to-Site, VPN > Site-to-Site** and **FIREWALL > History**.
- **Aggregated dashboard elements** – **Updates** element on the **General** dashboard.

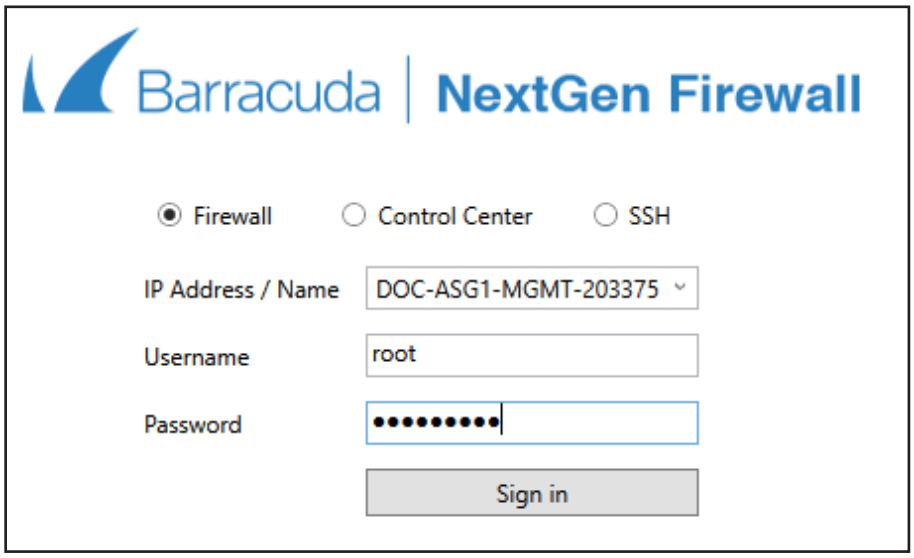
On pages that display aggregated data from all firewall instances in the cluster, use the **Instance ID** column to filter or group the information by instance.

Login and Default Password

Connect to the firewall cluster via NextGen Admin using the FQDN of the ELB with a listener on TCP 807. The default password is the instance ID of the first instance in the Auto Scaling group. Go to the **Instance** tab of the Auto Scaling group and locate the instance that is protected from scale in to identify the first instance.



- **IP Address /Name** – Enter the DNS name of the management ELB in front of the firewall cluster.
- **User** – Enter `root`.
- **Password** – The default password is the instance ID of the first instance.



After logging in the first time, you are prompted to change your password.

Log Streaming to AWS CloudWatch

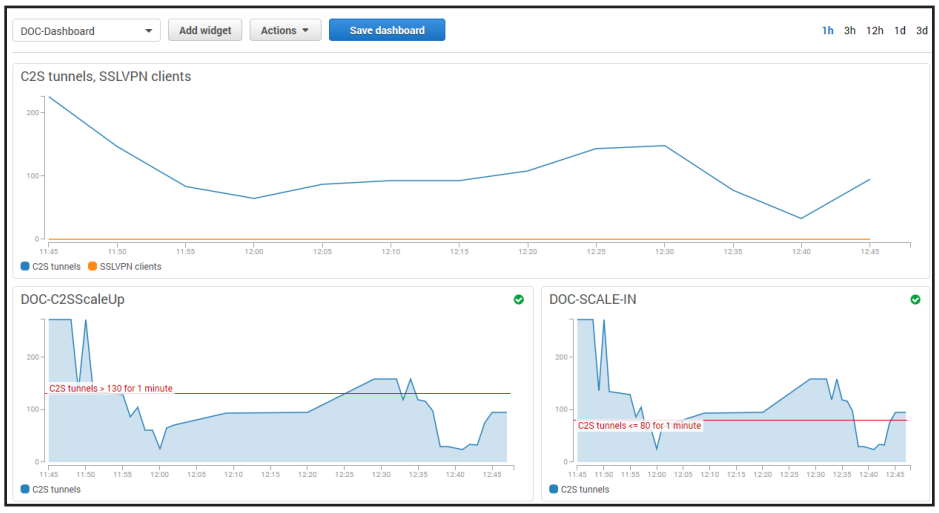
Log files stored on the firewall instances themselves are ephemeral. As soon as an instance is terminated, the log files are deleted with it. To keep the log files for later analysis, troubleshooting, or regulatory reasons, use syslog streaming on the firewalls to send them to AWS CloudWatch. There, the logs can be placed in groups, filtered, or processed further.

For step-by-step instructions, see 3.2 How to Configure Log Streaming to AWS CloudWatch (page 87)

Monitoring and Statistics through AWS CloudWatch

Each firewall in the cluster sends both basic and custom firewall metrics to AWS CloudWatch. Using AWS CloudWatch, you can monitor and visualize these metrics through CloudWatch alarms and dashboard widgets. Monitoring alarms through the dashboard widgets allows the admin to see why auto scaling policies were applied and offers the data necessary to make improvements. The granularity at which the metrics are published can be changed. By default, metrics are published every 5 minutes. Enable detailed monitoring to lower the granularity to 1 minute. This can be configured in the template by adding this parameter to the AutoScalingGroup (ASG) resource in the template.

```
"ASG": {
            "Type": "AWS::AutoScaling::AutoScalingGroup",
            "Properties": {
            [....]
                "MetricsCollection": [{
                        "Granularity" : "1Minute"
                    }],
            [....]
```



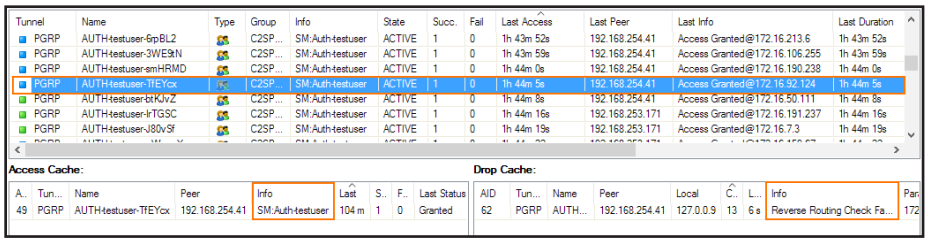
2.2.7 Monitoring via NextGen Admin

For remote access and firewalling workloads, the firewall cluster NextGen Admin provides more detailed, up-to-date information than is accessible through CloudWatch.

When logged in via NextGen Admin, client-to-site and SSL VPN tunnels are listed on the **VPN > Client-to-Site** and **VPN > Status** pages. The data in the **VPN** tab is aggregated from all firewall instances in the ASG. The pages list all available client-to-site and SSL VPN tunnels. On the **VPN > Status** page, the status is indicated by a colored icon In the **Tunnel** column:

- **Blue** – The client is currently connected.
- **Green** – The VPN tunnel is available, but currently not in use.
- **Grey** – The VPN tunnel is currently disabled. To enable the tunnel, right-click it and select **Enable Tunnel**.

To troubleshoot individual connections, click the client-to-site tunnel in the list and then see the error messages in the **inf** columns of the **Drop Cache** and **Access Cache** tabs.



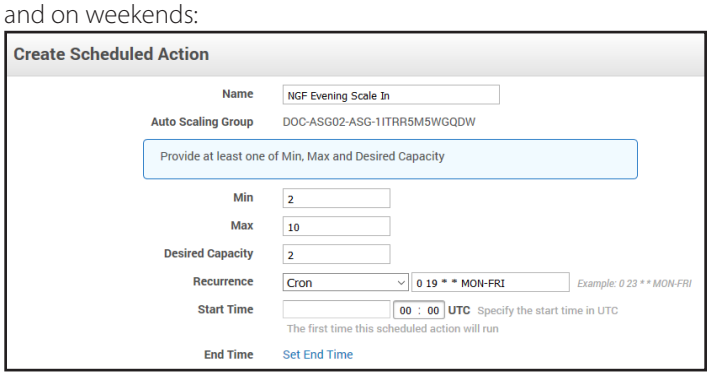
For more information about the **VPN > Client-to-Site** page, see VPN Tab.

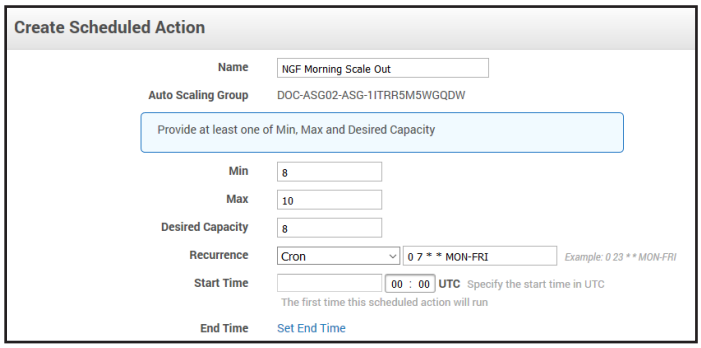
For more information about the **FIREWALL > History** page, see History Page.

2.2.8 Scaling Policies - Scheduled Actions

The cluster scales to a predefined number of instances according to the time of day or date. Unhealthy or terminated instances are automatically replaced. Use scheduled scaling for predictable workloads, or to reduce the number of instances overnight when the load is low. Since one instance in the cluster is always protected from scale-in, it is not possible to completely shut down the firewall cluster. To retain high availability, at least two instances are needed. Scheduled actions are configurable in the **Scheduled Actions** tab of the Auto Scaling group settings.

Example scheduled action that scales up the cluster during the day Monday through Friday and scales back during the night and on weekends:



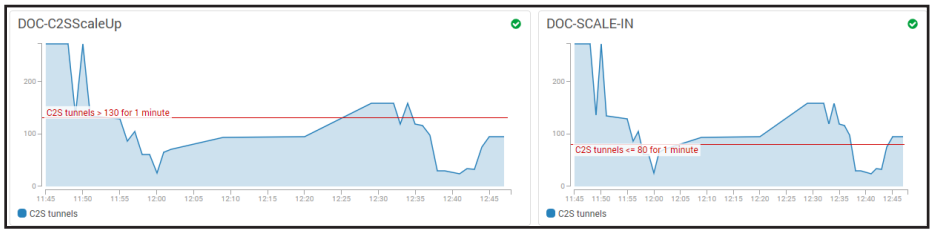


2.2.9 Scaling Policies - Dynamic Scaling

To optimize the firewall cluster to your workload, you need to select the metrics, and know how to interpret the values and the resulting action. Scaling metrics should be selected based on the use case. For example, a firewall cluster with a lot of client-to-site VPN connections should scale on the C2S Tunnels metric. If firewalling is the biggest part of the workload, it can also scale on the throughput or number of sessions, or number of dropped sessions. To achieve high availability, the firewall cluster must always use a minimum of two instances in two different Availability Zones.

Select the Relevant Metrics

To scale your firewall cluster dynamically, you must first select the metrics upon which you are going to the scale. The individual data points of the metrics can be averaged (relative performance metrics) or summed up (absolute performance metric) over a time period when creating the CloudWatch alarms. If you are adding up the data points (SUM), make sure to set the time period to match the metric collection granularity: One minute for detailed monitoring, five minutes for normal monitoring. For each metric, define upper and lower limits at which the cluster is scaled out and in. Make sure to leave enough room between the scale-out and scale-in thresholds to avoid the cluster from scaling in and out too frequently, resulting in additional cost and lower performance of the cluster. To avoid additional scaling actions before the previous action has taken effect, configure the ASG to use a cool-down period of at least 10 minutes (600 seconds). Use the CloudWatch widgets to visualize your alarms. This helps you to adjust the values to fit your workload.



It may also help to think about how the data points collected from the firewall cluster are used in the CloudWatch alarm:

- **Averaged metrics (default)** – Use average values over a time period that uses multiple metrics. Use longer time periods for the threshold to be more inert; shorten the time period to be more responsive. Setting the time period to a value that

is too short causes the cluster to scale out and in too often, causing unnecessary cost. If in doubt use relative metrics.

- **Absolute metrics** – Set the CloudWatch alarm to add the metrics collected from the firewall cluster in the same time period that the metrics are published in. For detailed monitoring, select 1 minute, for standard monitoring 5 minutes. Using absolute values are a good choice if you want to define exact correlations between the value of a metric and the number of instances. For example: 500 client-to-site tunnels always equals 2 firewall instances; 1000 client-to-site tunnels always equals 5 instances, and so on. Absolute metrics are a good choice when the number of instances needed is non-linear.

Simple or Step Scaling

The next thing to define is if the scaling policy should always scale the same amount every time the alarm is triggered, or if there are different steps depending on by how much the value differs from the threshold value set in the alarm. Using a simple one-step scaling policy does not cope well with quickly increasing demand. By the time the scaling action has finished, the demand may have outpaced the number of available instances, forcing to scale multiple times to achieve the desired performance. This effect can be mitigated by scaling up multiple instances each time the alarm is triggered, potentially overshooting the required number of instances and incurring costs for the extra instance until the scale-in policy removes it.

For a more efficient scaling policy that covers both the slowly rising demand and quick changes, create a policy containing multiple steps. This allows you to immediately scale to the correct number of instances, thereby improving the efficiency of the cluster. Depending on the size of the step, increase the cool-down period after scaling to avoid scale-in policies from removing instances too quickly before the increased number of instances take effect.

Add Instances or Go to Exact Capacity when Scaling

The last decision before you can put your scaling policy into action is whether to simply add capacity when the alarm is triggered, or to use exact capacity numbers to match. For alarms using averaged metrics, add capacity; for absolute (SUM) metrics, set the exact capacity. Exact capacity is mainly used for non-linear workloads, whereas adding capacity is more flexible and requires less testing because the scaling threshold values are the same no matter how many instances there are in the cluster.

For step-by-step instructions, see

3.13 How to Configure Scaling Policies for a NextGen Firewall Auto Scaling Cluster (page 161)

Force Reconnect on Scaling Action for Client to Site VPN

A custom parameter in the firewall can force a redistribution of all client-to-site connections on each scaling event of the cluster. All clients automatically reconnect, thereby evening out the load. Depending on the type of traffic going through

the client-to-Site VPN, this action may not be transparent to the remote users because active sessions may time out and, therefore, require the user to reconnect to the backend service.

For step-by-step instructions, see

3.16 How to Configure a Client-to-Site VPN Group Policy for a NextGen Firewall Auto Scaling Cluster in AWS (page 175)

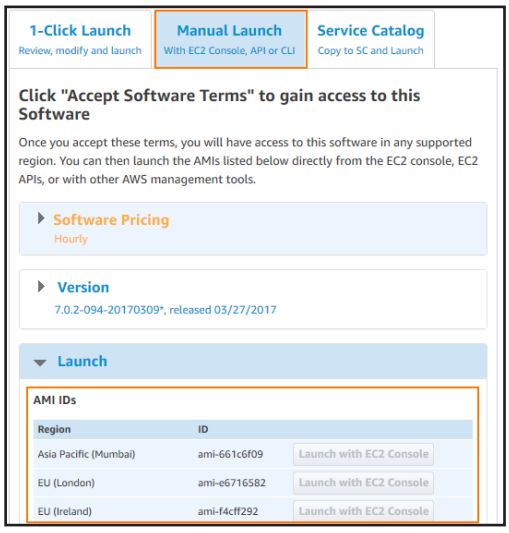
2.2.10 Installing Hotfixes

Hotfixes are published by Barracuda Networks when an issue requires immediate attention, such as newly discovered security vulnerabilities or critical bugs in the firmware. Hotfixes are installed through the **Firewall** dashboard in NextGen Admin. New instances in the cluster automatically install the same hotfixes when the cluster scales. Most hotfixes reboot the firewall. Consider this when setting the cool-down value for scaling actions because it will take longer for an instance to be ready when multiple hotfixes need to be installed.

For step-by-step instructions, see How to Install Updates via NextGen Admin.

2.2.11 Firmware Update via CloudFormation Stack Update

Although possible, it is not recommended to install firmware updates like hotfixes through the **Update** element on the NextGen Admin dashboard. Instead, replace the AMI in the parameter file of your template and update the CloudFormation stack. The AMI for the new firmware version is listed in the **Manual Launch** tab of the listing for the Barracuda NextGen Firewall PAYG image in the AWS Marketplace.



If your templates are stored in an S3 bucket, enter this AWS CLI command to update CloudFormation stack:

```
aws cloudformation update-stack  --stack-name "YOUR_STACK_NAME" --template-body YOUR_S3_BUCKET/autoscale.json --parameter YOUR_S3_BUCKET/autoscale_parameters.json
```

```
PS C:\Users\mzoller\Documents\AWS_NGF_Official_Templates>
PS C:\Users\mzoller\Documents\AWS_NGF_Official_Templates> aws cloudformation update-stack  --stack-name "DOC-ASG02" --template-body https://s3-eu-west-1
.amazonaws.com/campus.deploytemplates/autoscale.json --parameter https://s3-eu-west-1.amazonaws.com/campus.deploytemplates/autoscale_parameters.json
{
    "StackId": "arn:aws:cloudformation:eu-west-1:726256585710:stack/DOC-ASG02/7bb93280-3fb3-11e7-b21d-500c3cb898d2"
}
PS C:\Users\mzoller\Documents\AWS_NGF_Official_Templates>
```

After updating the stack, scale down to one instance and manually terminate the instance protected from scale-in through the AWS CLI or EC2 web portal. All new instances that are launched now use the new AMI in the updated launch configuration. If the instance that is protected from scale-in is not terminated manually, the firewall cluster will be in an inconsistent state.

2.2.12 Backup / Restore

Creating a backup and restoring the firewall configuration is analog to a stand-alone NextGen Firewall. To avoid overwriting the PAYG instance, the license must be saved prior to restoring the configuration.

To automate deployment, it also possible to modify the template to use an existing S3 bucket with a previous firewall configuration. Change the template so it does not create a new S3 bucket, and replace all references to use the existing bucket. Each firewall cluster requires a dedicated S3 bucket; it is not possible to share the configuration over multiple clusters.

For step-by-step instructions, see 3.3 How to Restore a Configuration on a PAYG Firewall in the Public Cloud (page 93)

2.2.13 Building Access Rules

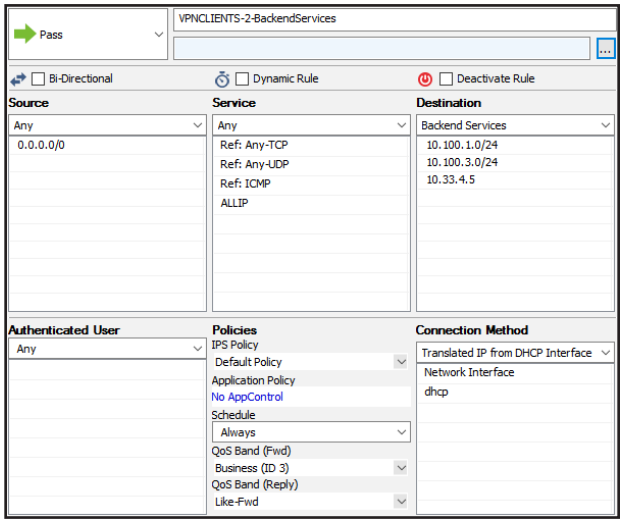
By default, the firewall blocks all traffic. Only traffic matching an access rule with an allowed policy is allowed to pass. For the traffic flow to always use the same firewall, all access rules must translate the source IP address to the IP address of the DHCP interface of the firewall. It is recommended to create a custom service and network object matching your setup. This allows for easy reuse and access rules that are human-readable. Although Dst NAT access rule support basic load balancing, it recommended to use AWS ELBs instead.

For step-by-step instructions, see Access Rules, Network Objects, and Service Objects.

VPN Clients to Backend Services

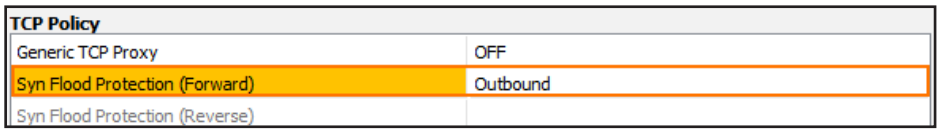
Each VPN client is assigned an IP address in the VPN client network on the firewall the client is connected to. Since all firewalls use the same VPN client network, the source IP address must be rewritten to the IP address of the firewall instance.

- **Action** – Select **Pass**.
- **Source** – Select **Any**.
- **Service** – Select the services remote users are allowed to use, or select **Any** to allow all.
- **Destination** – Select a network object containing the backend services or networks remote users are allowed to access.
- **Connection Method** – **Dynamic NAT** or **Translated IP from DHCP Interface**.



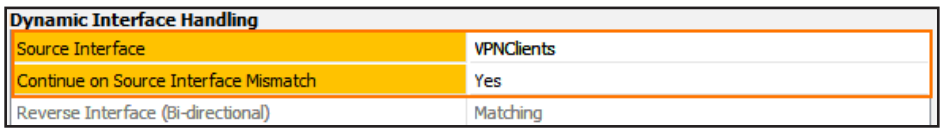
In the **TCP Policy** section of the **Advanced** access rule settings:

- **Syn Flood Protection (Fwd)** – Select **Outbound**.



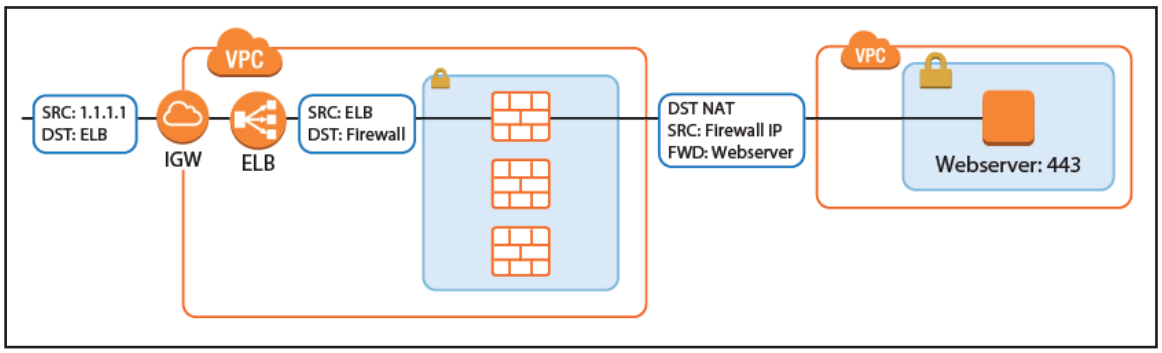
In the **Dynamic Interface Handling** section of the **Advanced** access rule settings:

- **Source Interface** – Select **VPN Clients**.
- **Continue on Source Interface Mismatch** – Select **Yes**.

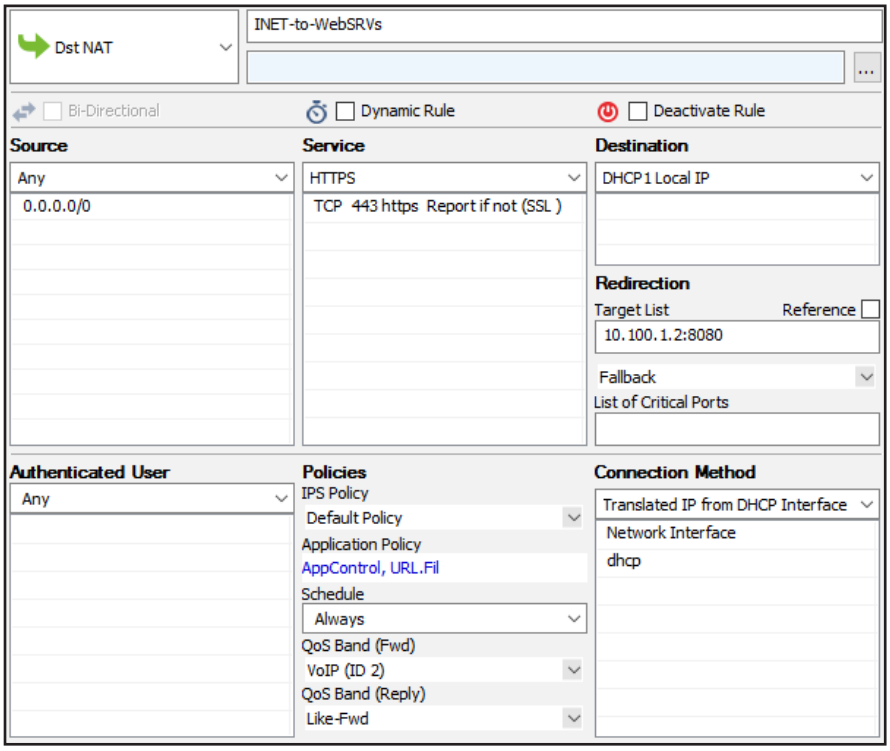


Internet to Backend Services

Create the following access rule to forward traffic from the Internet to an internal web server.

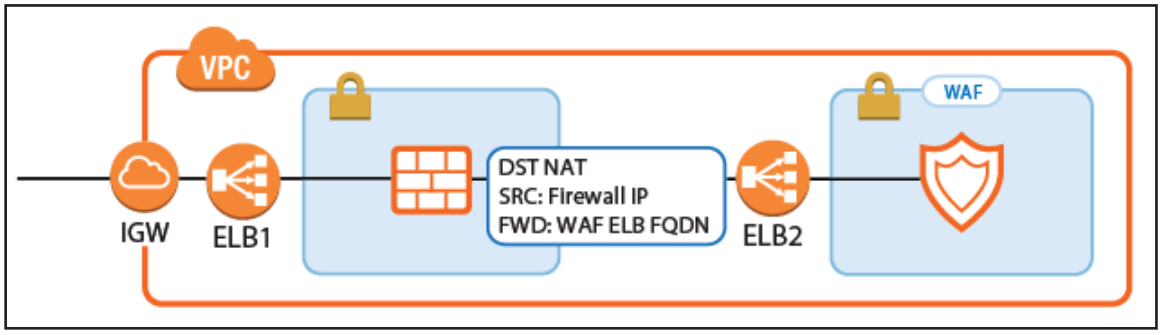


- **Action** – Select **Dst NAT**.
- **Source** – Select **Any** or a network object containing the networks the ELB is deployed in.
- **Service** – Select the service. e.g., **HTTP+S**.
- **Destination** – Select **DHCP1 Local IP**.
- **Connection Method** – Select **Dynamic NAT** or **Translated from DHCP Interface**.
- **Redirection Target** – Enter the IP address of the backend service. Optionally, append the port number to redirect to a different port. e..g, `10.100.1.2` or `10.100.1.2:8080`

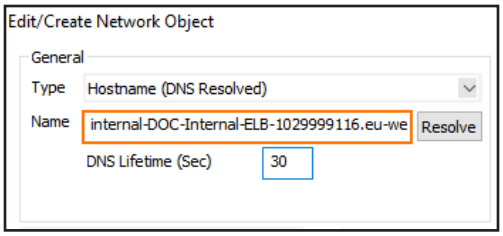


Redirect Traffic through a WAF Cluster or Other Service Behind an Internal ELB

Services behind an internal ELB can also be forwarded via Dst NAT access rule.

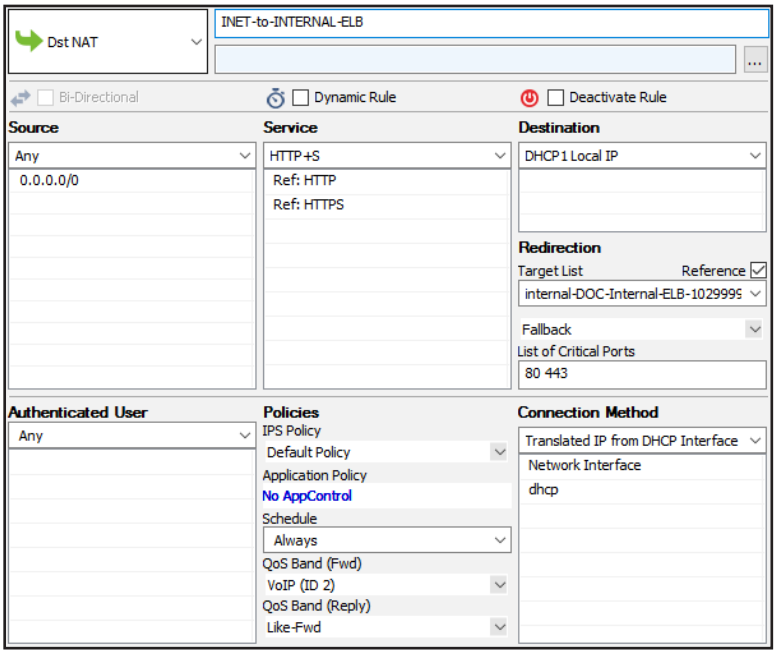


1. Create a hostname network object for the internal DNS name of the ELB, set the **DNS Lifetime** to 30 seconds, and click **Send Changes**.



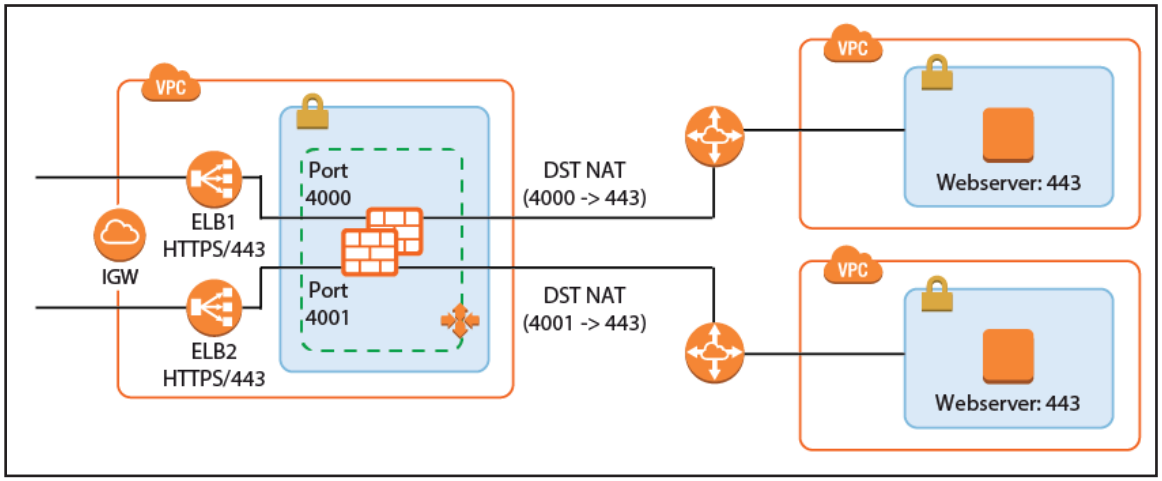
2. Create the access rule:

- **Action** – Select **Dst NAT**.
- **Source** – Select **Any** or a network object containing the networks the ELB is deployed in.
- **Service** – Select the service. e.g., **HTTP+S**.
- **Destination** – Select **DHCP1 Local IP**.
- **Connection Method** – Select **Dynamic NAT** or **Translated from DHCP Interface**.
- **Redirection Target** – Click **Reference** and select the network object for the ELB.

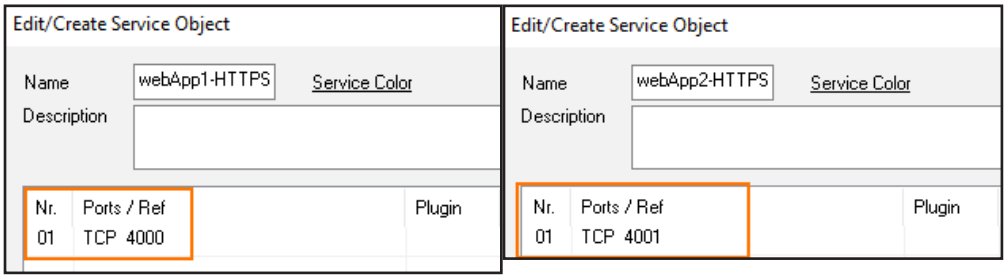


Multiple Backend Services Using the Same Port

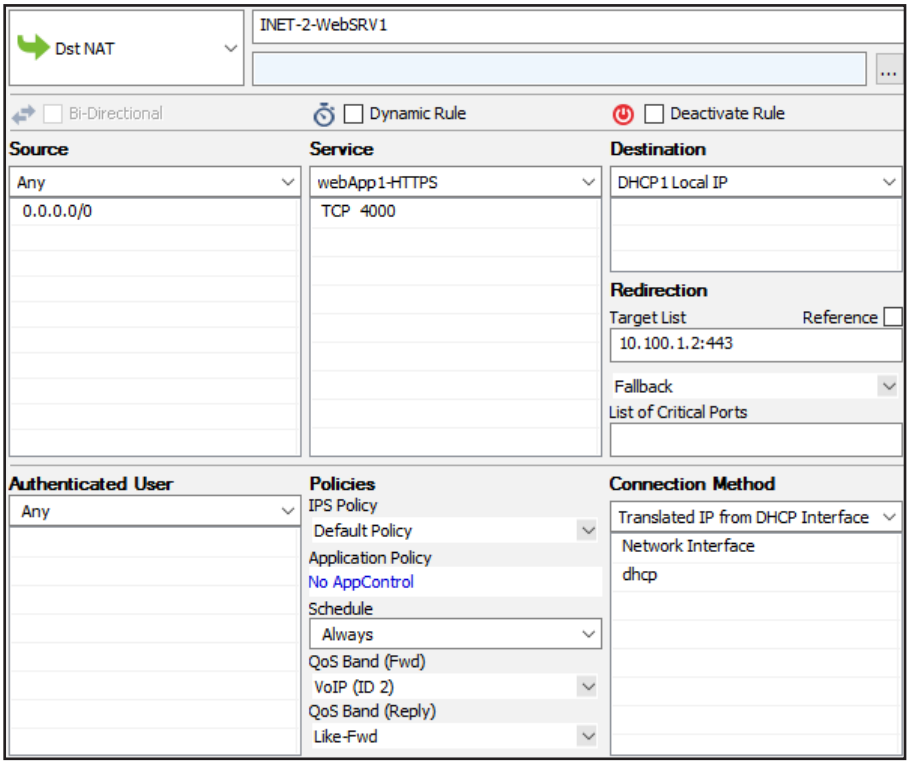
A variation of the same rule, only this time two services running on the same port must be accessed. The ELBs in front of the firewall cluster map the service to different ports on the firewall. The firewall forwards the traffic to the correct instance or internal ELB and maps it back to the correct port.



3. Add one ELB per service to the firewall cluster. Map the external port to a unique internal port. e.g., ELB1: 443 -> 4000 and ELB2 443 -> 4001
4. Create service objects for the internal ports on the firewall. Optionally, add Port Protocol Detection.

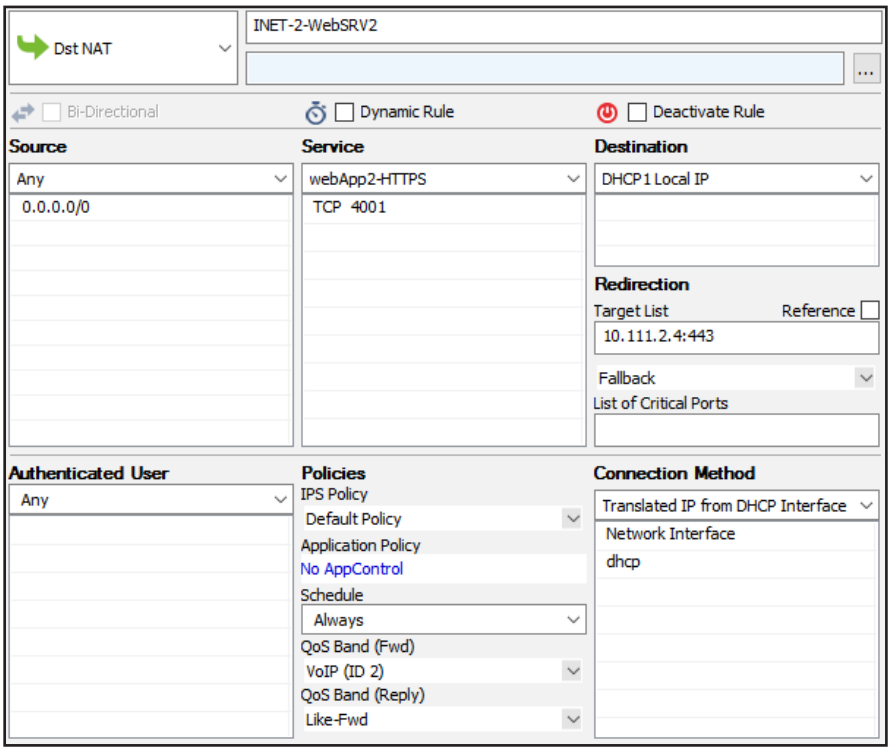


5. Create the Dst NAT access rule for the first backend service:
   - **Action** – Select **Dst NAT**.
   - **Source** – Select **Any** or a network object containing the networks the ELB is deployed in.
   - **Service** – Select the service object for the first service. e.g, `webApp1-HTTPS`
   - **Destination** – Select **DHCP1 Local IP**.
   - **Connection Method** – Select **Dynamic NAT** or **Translated from DHCP Interface**.
   - **Redirection Target** – Enter the IP address and port of the first backend service. e.g., `10.100.1.2:443`



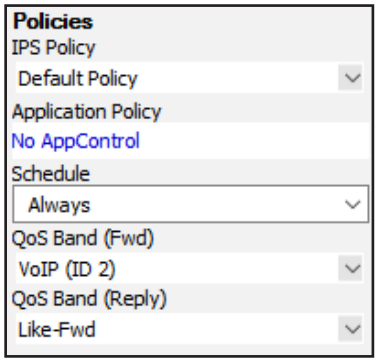
6. Create the Dst NAT access rule for the second backend service

- **Action** – Select **Dst NAT**.
- **Source** – Select **Any** or a network object containing the networks the ELB is deployed in.
- **Service** – Select the service object for the first service. e.g, `webApp2-HTTPS`
- **Destination** – Select **DHCP1 Local IP**.
- **Connection Method** – Select **Dynamic NAT** or **Translated from DHCP Interface**.
- **Redirection Target** – Enter the IP address and port for the second backend service e.g., 10.111.2.4:443



Enabling IPS per Access Rule

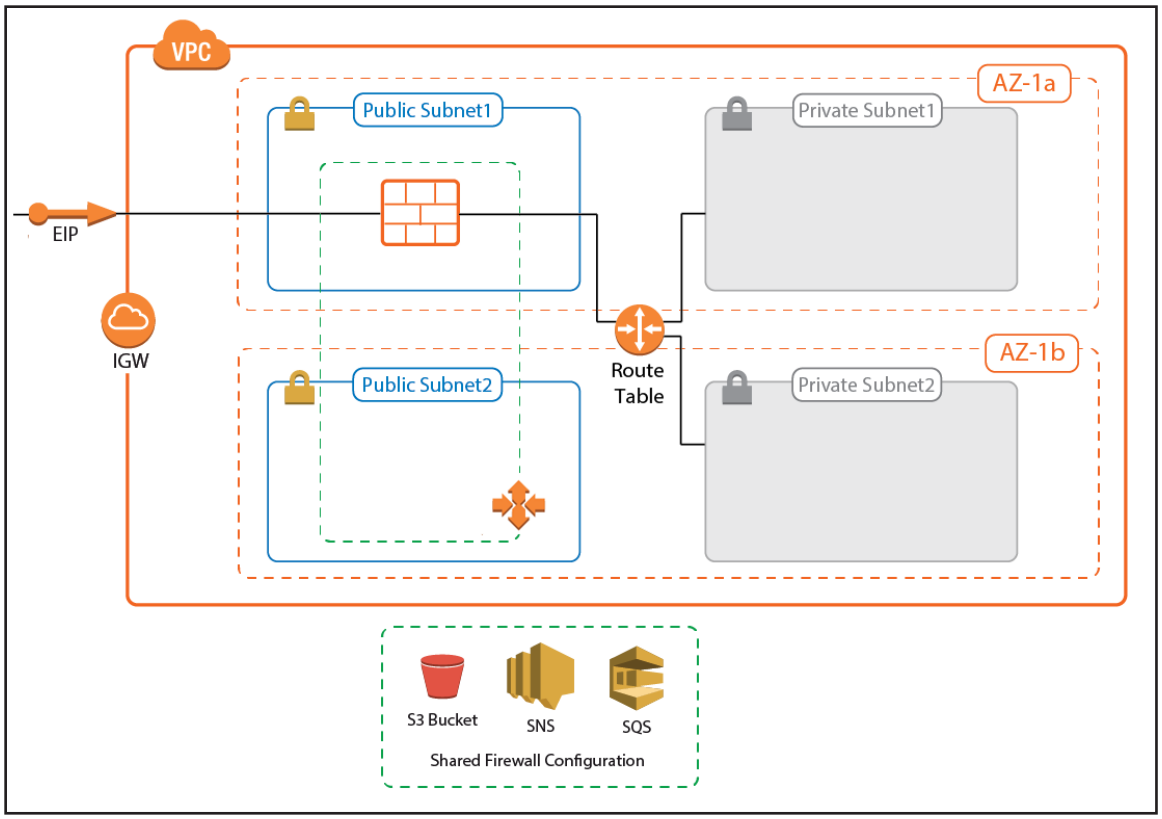
For the Intrusion Prevention System to scan packets matching an access rule, select the **IPS Policy** in the Pass or Dst NAT access rule. Depending on the configuration of the IPS, malicious traffic patterns are now blocked or reported. For SSL encryption, it is recommended to use SSL offloading on the ELB to allow the IPS to analyze the decrypted traffic. This saves processing power on the firewall. If end-to-end encryption is required for regulatory reasons, enable SSL Interception on the access rule and in the IPS configuration to also scan SSL-ecrypted traffic.



For step-by-step instructions, see Intrusion Prevention System (IPS).

2.3 NextGen Firewall Cold Standby Cluster

A NextGen Firewall Cold Standby Cluster is a low-cost architecture for AWS deployments that minimizes downtime to only a few minutes in case of failure of the underlying hypervisor hardware, the firewall instance, or the complete Availability Zone. From a technical standpoint, the Cold Standby Cluster is a NextGen Firewall Auto Scaling Cluster with the size set to one. The firewall configuration is securely stored and synchronized through AWS backend services. Replacing the Elastic Load Balancer used in the NextGen Auto Scaling Cluster with a floating Elastic IP allows the use of both TCP- and UDP-based services on the firewall. The default template uses hourly PAYG licensing, but can be modified to use pool licenses for Control Center-managed instances.



2.3.1 Use Cases for a NextGen Firewall Cold Standby Cluster

The NextGen Firewall Cold Standby Cluster is used to secure access to resources in the private networks of its own VPC, or to deploy in a Transit VPC as a part of a larger cloud infrastructure.

- **Site-to-Site VPN** – One way on-premises to AWS, TINA, and IPsec site-to-site VPN tunnels.
- **Edge Firewall** – Scan for malicious traffic using the built-in IPS and handle access to resources via access rules.
- **Secure Remote Access** – Client-to-site VPN, CudaLaunch, and SSL VPN using TINA, SSL VPN, and IPsec VPN protocols.

2.3.2 Deploying a NextGen Firewall Auto Scaling Cluster

The Cold Standby Cluster must be deployed via a CloudFormation template. The template deploys a VPC with public and private subnets in two Availability Zones. The Auto Scaling Cluster is deployed in the public subnets. Instances placed in the private subnets are automatically routed over the active firewall instance.

Create an IAM role for a NextGen Firewall in an Auto Scaling group. For step-by-step instructions, see 3.1 How to Create an IAM Role for an F-Series Firewall in AWS (page 79)

Download the **NGF_ColdStandby.json** template and parameter file from the Barracuda Network GitHub account: https://github.com/barracudanetworks/ngf-aws-templates.

Accept the Software Terms for the **Barracuda NextGen Firewall PAYG** image in the AWS Marketplace.

Create a parameter template file containing your parameters values.

Deploy the **coldstandby.json** CloudFormation template via AWS CLI or AWS console.

```
aws cloudformation create-stack  --stack-name "YOUR_STACK_NAME" --template-body
YOUR_S3_BUCKET/NGF_ColdStandby.json --parameter YOUR_S3_BUCKET/NGF_ColdStandby_
parameters.json
```

During deployment, the following resources are created by the template:

- VPC with private and public subnets in two Availability Zones.
- One S3 bucket.
- Automatically created SNS and SQS queues.
- A Launch Configuration and Auto Scaling group for the firewall. The Barracuda NextGen Firewall PAYG image must be used.

After stack creation is complete, wait for one firewall instance to spin up and finish provisioning.

For step-by-step instructions, see 2.3 NextGen Firewall Cold Standby Cluster (page 49)

2.3.3 Control Center-Managed NextGen Firewall Cold Standby Cluster

The NextGen Control Center is a central management appliance for F-Series Firewall, that can be deployed as a virtual appliance on-premises or in the cloud. Managing the Cold Standby Cluster with a NextGen Control Center separates the firewall configuration and monitoring from deployment and integration with other AWS services. This is especially useful for highly specialized or large departments with dedicated network security teams and multiple developer teams using automatic deployments. Managed firewalls are preconfigured on the Control Center. During provisioning of the firewall instance, the configuration is retrieved from the Control Center. Optionally, pool licenses can be used that are bound to the Control Center license instead of the EC2 instance of the firewall. Pool licenses are available in multiples of 5.

For step-by-step instructions, see 3.6 How to Modify CloudFormation Templates to Retrieve the PAR File from a Control Center (page 105)

Modifying the Default CloudFormation Template

To fetch the configuration from the Control Center, the default template must be edited to invoke the getpar command with the information required to be able to connect to the Control Center. If the PAYG images are used, the licenses are sent to the Control Center before retrieving the firewall configuration. For a Cold Standby Cluster, only one firewall configuration is required on the Control Center.

For step-by-step instructions, see 2.3 NextGen Firewall Cold Standby Cluster (page 49)

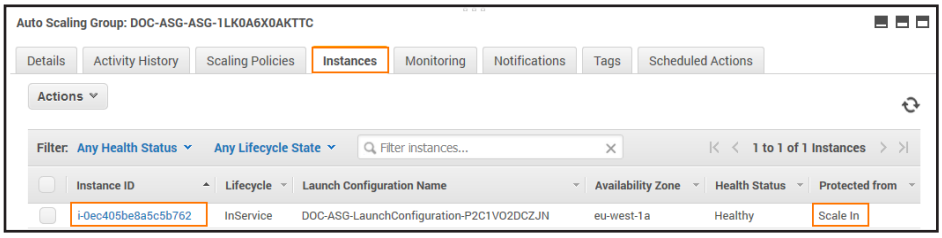
Managing Firewall Configuration at Scale with the Control Center

Managing multiple similar firewall configurations is greatly simplified by using a Control Center. Configuration nodes and cluster-based services, such as the distributed firewall service, are shared across multiple firewall instances. The Control Center also handles pattern updates and hotfixes centrally.

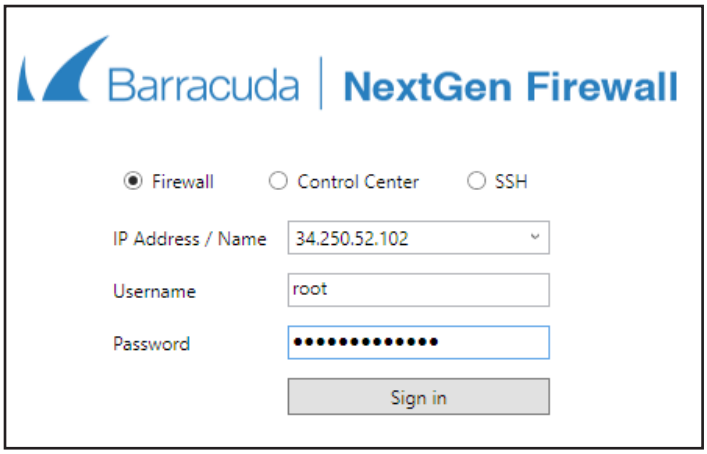
For more information, see 2.3 NextGen Firewall Cold Standby Cluster (page 49)

2.3.4 Login and Default Password

1. Connect to the firewall cluster via NextGen Admin using the Elastic IP. The default password is the instance ID of the firewall.
2. Go to the **Instance** tab of the Auto Scaling group and use the instance ID from the one instance that is running.



- **IP Address /Name** – Enter the Elastic IP address associated with the firewall.
- **User** – Enter `root`.
- **Password** – The default password is the instance ID of the first instance.

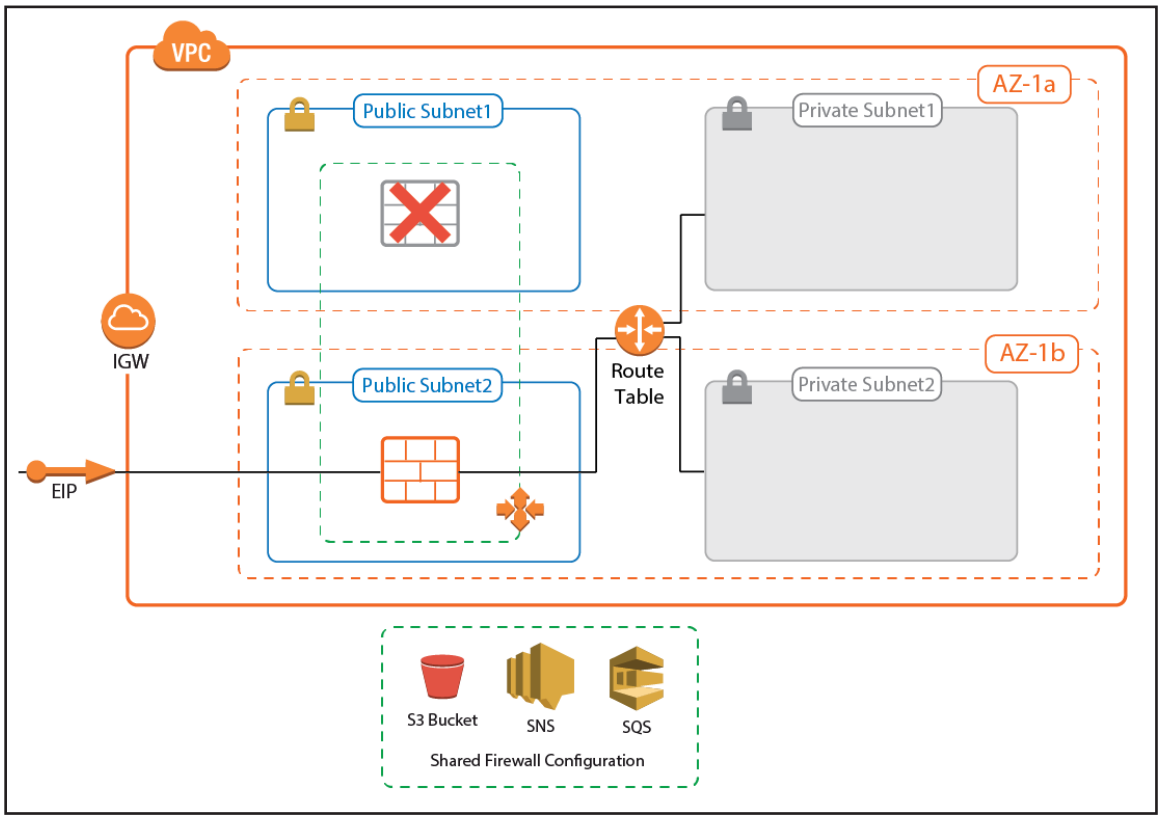


2.3.5 Cold Standby Failover

A failover occurs when the firewall instance is terminated, either due to a cloud-level incident such as emergency maintenance on the hardware, or if the EC2 instance health checks fail for the instance, causing it to be terminated and replaced by the Auto Scaling group. The failover process follows these steps:

1. Health check fails for firewall instance.
2. The unhealthy firewall instance is terminated.
3. The Auto Scaling group launches a replacement firewall instance.
4. Provisioning of the new firewall instance:
   - Configuration from S3 bucket is used.
   - The instance is associated with the Elastic IP.
   - Routes pointing to the firewall instance are rewritten to use the new firewall instance.
   - Hotfixes are installed on the firewall. This may cause the firewall to reboot.
   - (BYOL only) The firewall automatically fetches the configuration from a NextGen Control Center.
5. The new firewall instance is now provisioned.

Note that only routes for which there are AWS CLI commands in the user data section of the templates are changed. The Cold Standby Cluster does not monitor the route tables in the VPC. If this is required, use a NextGen Firewall High Availability Cluster instead.



2.3.6 Scaling Up or Scaling Down

Scale the Cold Standby Cluster up by replacing the instance type in the template with a larger instance size and then update the stack. After the stack update is complete, terminate the running instance using the old instance type. The Auto Scaling group now automatically replaces the firewall with an instance using the new instance type.

2.3.7 Installing Hotfixes

Hotfixes are published by Barracuda Networks when an issue requires immediate attention, such as newly discovered security vulnerabilities or critical bugs in the firmware. Hotfixes are installed through the **Firewall** dashboard in NextGen Admin. New instances in the cluster automatically install the same hotfixes when the cluster scales. Most hotfixes reboot the firewall. Consider this when setting the health check grace period because it will take longer for an instance to be ready when multiple hotfixes need to be installed. If the health check is resumed too early, while the firewall is still installing the hotfix or rebooting, the instance is terminated because it is deemed unhealthy. This results in a loop of starting and terminating instances.

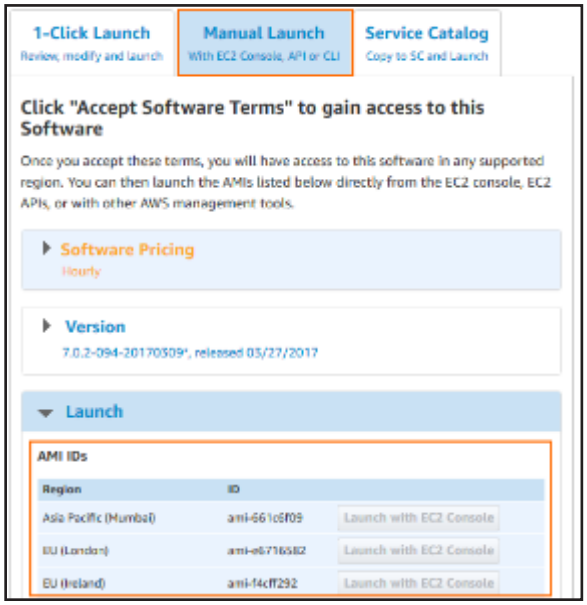
To change the health check grace period, modify the **HealthCheckGracePeriod** parameter in the template and update the stack.

```
    "ASG": {
      "Type": "AWS::AutoScaling::AutoScalingGroup",
      "Properties": {
        "VPCZoneIdentifier": [ { "Ref": "PublicSubnetA" }, { "Ref" :"PublicSubnetB" } ],
        "LaunchConfigurationName": { "Ref": "LaunchConfiguration" },
        "MinSize": 1,
        "DesiredCapacity": 1,
        "MaxSize": 1,
        "HealthCheckGracePeriod": 3600,
        "Tags": [ { "Key": "Name", "Value": { "Ref": "ASGName" }, "PropagateAtLaunch": "True" } ]
      },
      "DependsOn": [ "IGWAttachment" ]
    }
},
```

For step-by-step instructions, see 2.3 NextGen Firewall Cold Standby Cluster (page 49)

2.3.8 Firmware Update via CloudFormation Stack Update

Although possible, it is not recommended to install firmware updates like hotfixes through the **Update** element on the NextGen Admin dashboard. Instead, replace the AMI in the parameter file of your template and update the CloudFormation stack. The AMI for the new firmware version is listed in the **Manual Launch** tab of the listing for the Barracuda NextGen Firewall PAYG image in the AWS Marketplace.



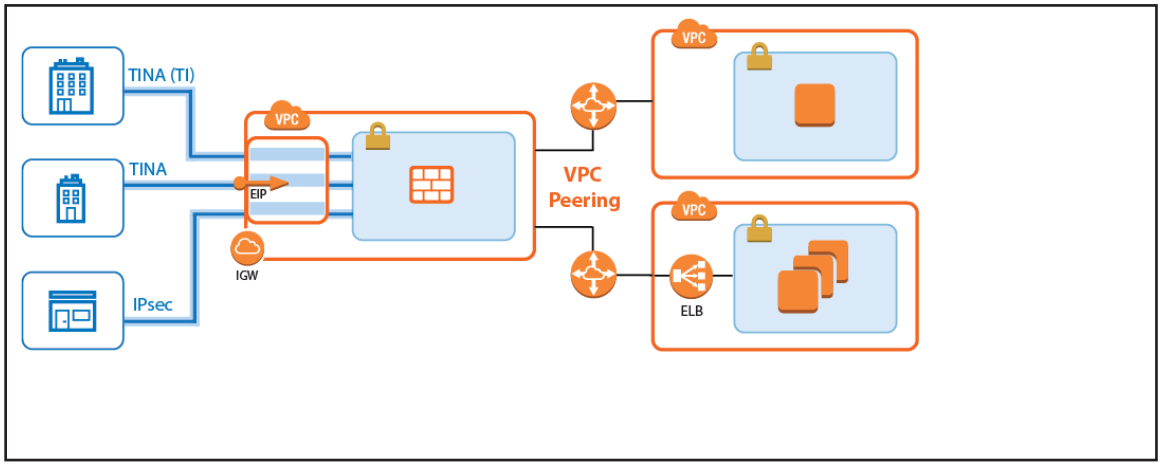
If your templates are stored in an S3 bucket, enter this AWS CLI command to update CloudFormation stack:

```
aws cloudformation update-stack  --stack-name "YOUR_STACK_NAME" --template-body YOUR_S3_BUCKET/coldstandby.json --parameter YOUR_S3_BUCKET/coldstandby_parameters.json
```

After updating the stack, manually terminate the firewall instance. The replacement instance using the new firmware is automatically launched. If the firewall is managed by a Control Center, the cluster the firewall configuration is stored in might have to be migrated after updating the AMI to a new major release, such as 7.1, 7.2, etc...

2.3.9 Site-to-Site VPN Tunnels for Cold Standby Clusters

Site-to-site VPN tunnels transparently connect on-premises networks. The NextGen Firewall supports TINA, IPsec IKEv1, and IKEv2 VPN protocols. Since ESP is not supported, IPsec VPN tunnels must use NAT-T. It is recommended to configure the NextGen Firewall to be the active VPN endpoint. For all instances in the private subnets using the firewall as the default gateway and all instances in VPCs connected via the AWS VPN gateway, the site-to-site VPN tunnel is fully transparent in both directions. If VPC peering is used and the firewall cannot be configured to be the default gateway for the instance, the source IP address for the traffic leaving the tunnel must be rewritten to the address of the DHCP interface of the firewall. Resources in peered VPCs cannot connect directly to the remote networks.



For step-by-step instructions, see 3.7 How to Create a TINA VPN Tunnel between F-Series Firewalls (page 111)

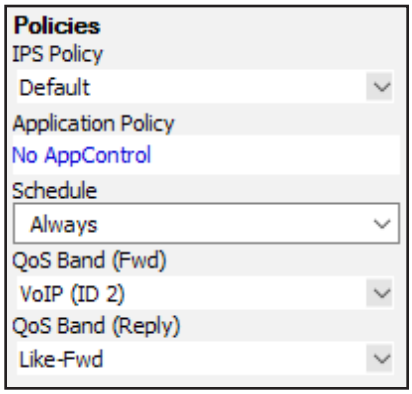
2.3.10 Access Rules

By default, the firewall blocks all traffic. Only traffic matching an access rule with an allowed policy is allowed to pass. If the destination instance is using the firewall as the default gateway, the source IP address does not have to be rewritten; otherwise, the source NAT must be used. Although Dst NAT access rules support basic load balancing, it is recommended to use internal AWS ELBs instead.

For more information, see Access Rules.

Enabling IPS per Access Rule

For the Intrusion Prevention System to scan packets matching an access rule, select the **IPS Policy** in the Pass or Dst NAT access rule. Depending on the configuration of the IPS, malicious traffic patterns are now blocked or reported. For SSL-encrypted traffic, it is recommended to use SSL offloading on the ELB to allow the IPS to analyze the decrypted traffic. This saves processing power on the firewall. If end-to-end encryption is required for regulatory reasons, enable SSL Interception on the access rule and in the IPS configuration to also scan SSL-encrypted traffic.

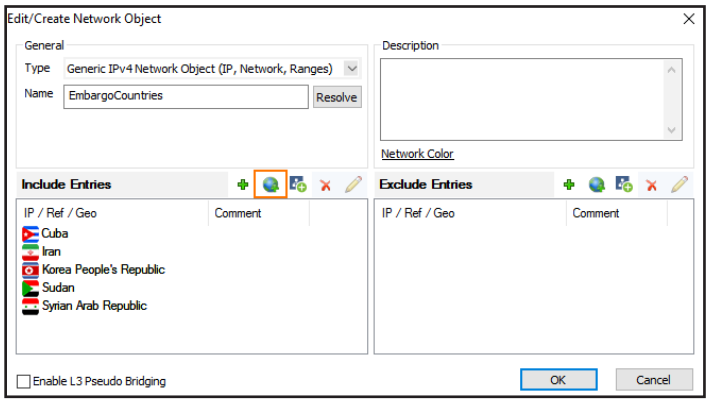


For more information, see Intrusion Prevention System (IPS).

Block Traffic Based on Geographic Location of Source IP Address

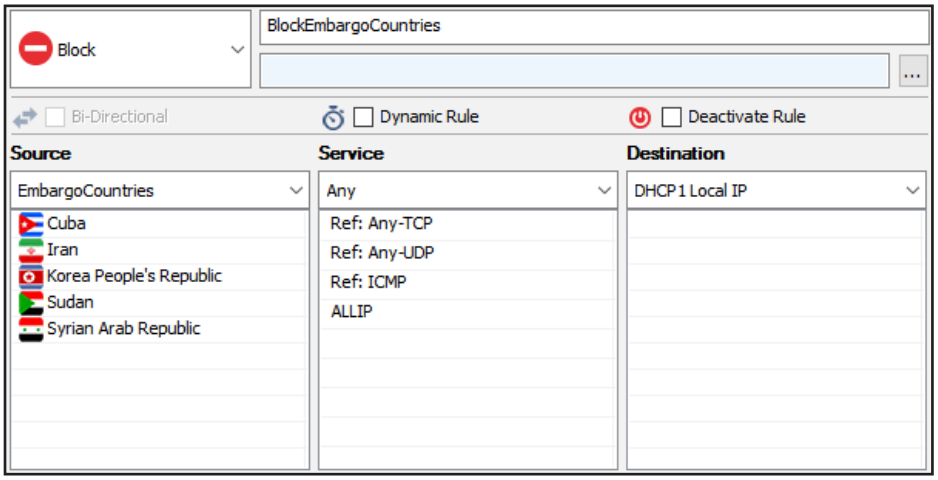
Create a network object containing the countries you want to block.

For more information, see 3.8 How to Create a Geo Location based Network Object (page 119)



Create a Block access rule using the geolocation network object as the source matching criteria:

- **Action** – Select **Block** or **Deny**.
- **Source** – Select the network object containing the countries you want to block.
- **Service** – Select **Any**.
- **Destination** – Select **DHCP1 Local IP**.

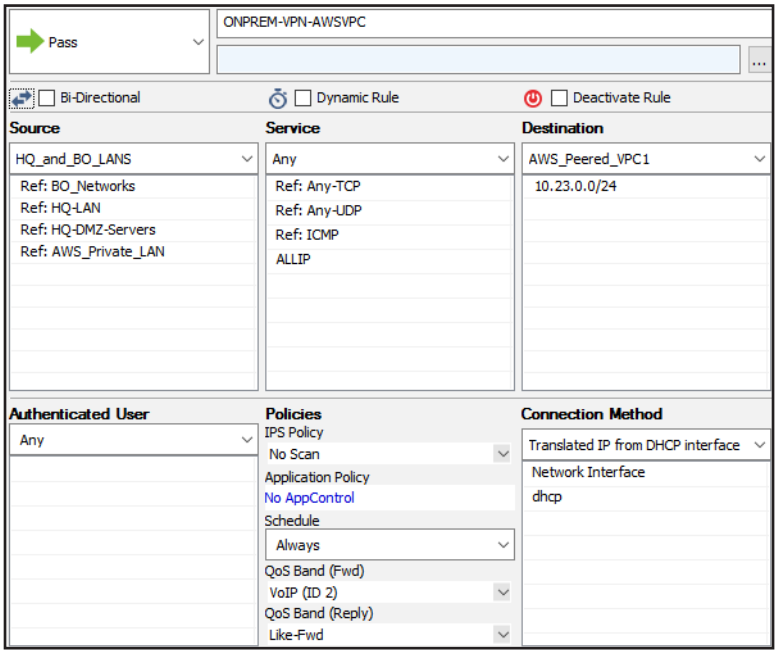


Site-to-Site VPN Tunnel to Backend Services in a Peered VPC

When connecting to services running in a VPC that is peered to the firewall VPC through a site-to-site VPN tunnel, the site-to-site tunnel can only be used one-way from on-premises to the AWS resource. Since the firewall is not the default gateway for the AWS instances running in the private subnets, the source IP address must be rewritten to match the firewall's IP address when exiting the VPN tunnel.

- **Action** – Select **Pass**.

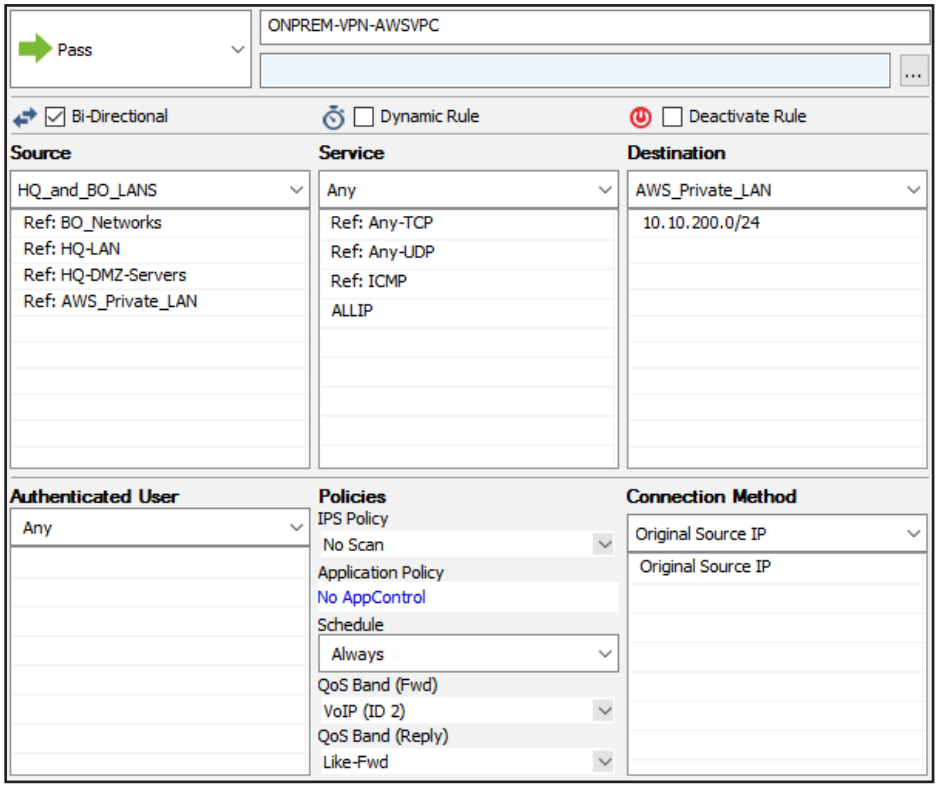
- **Source** – Select a network object containing the on-premises networks. These networks must be configured as the remote network for the site-to-site VPN tunnels.
- **Service** – Select the services, or select **Any** to allow all.
- **Destination** – Select a network object containing the backend networks and/or IP addresses in AWS. These networks must be configured as the local network for the site-to-site VPN tunnels.
- **Connection Method** – **Translated IP from DHCP Interface**.



Site-to-Site VPN Tunnel to Backend Services using the Firewall as the Default Gateway

For EC2 instances using the firewall as the default gateway, the site-to-site VPN tunnels do not require source or destination NAT.

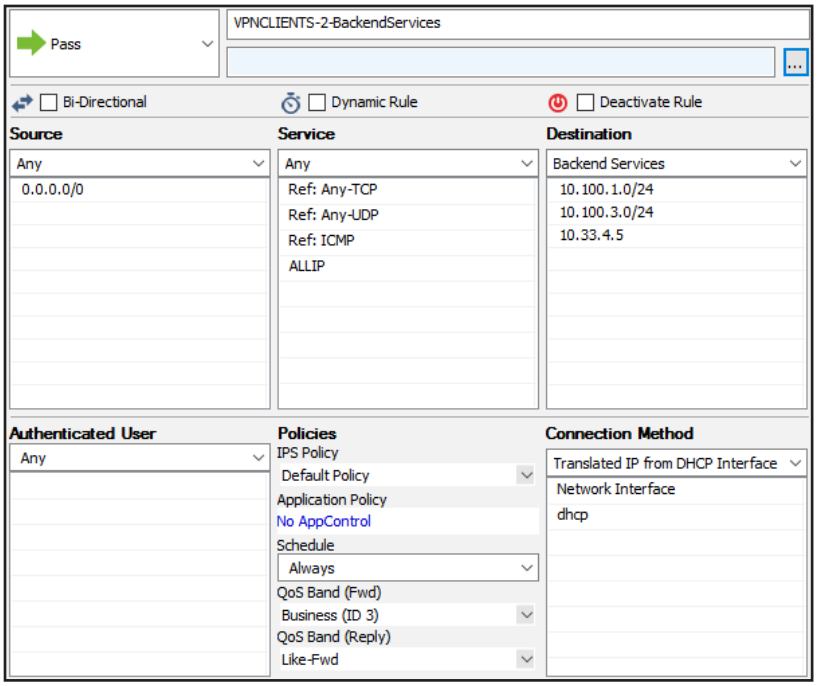
- **Action** – Select **Pass**.
- **Source** – Select a network object containing the on-premises networks. These networks must be configured as the remote network for the site-to-site VPN tunnels.
- **Service** – Select the services, or select **Any** to allow all.
- **Destination** – Select a network object containing the backend networks and/or IP addresses in AWS. These networks must be configured as the local network for the site-to-site VPN tunnels.
- **Bi-Directional** – Select to allow traffic in both directions.
- **Connection Method** – **Original Source IP**.



VPN Clients to Backend Services

Each VPN client is assigned an IP address in the VPN client network on the firewall the client is connected to.

- **Action** – Select **Pass**.
- **Source** – Select **Any**.
- **Service** – Select the services remote users are allowed to use, or select **Any** to allow all.
- **Destination** – Select a network object containing the backend services or networks remote users are allowed to access.
- **Connection Method** – **Translated IP from DHCP Interface** or **Original Source IP** depending on the destination.



In the **TCP Policy** section of the **Advanced** access rule settings:

- **Syn Flood Protection (Fwd)** – Select **Outbound**.

TCP Policy	
Generic TCP Proxy	OFF
Syn Flood Protection (Forward)	Outbound
Syn Flood Protection (Reverse)	

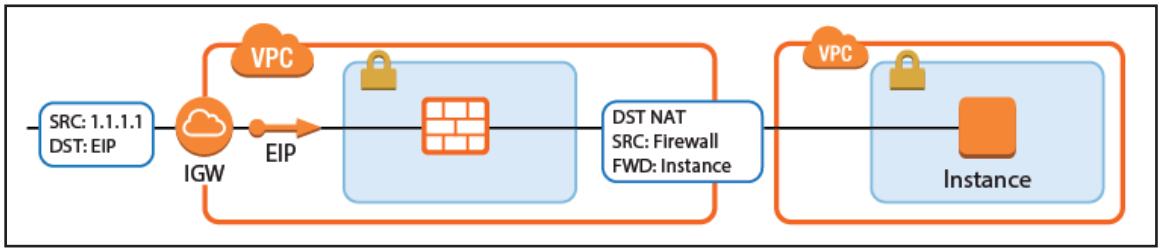
In the **Dynamic Interface Handling** section of the **Advanced** access rule settings:

- **Source Interface** – Select **VPN Clients**.
- **Continue on Source Interface Mismatch** – Select **Yes**.

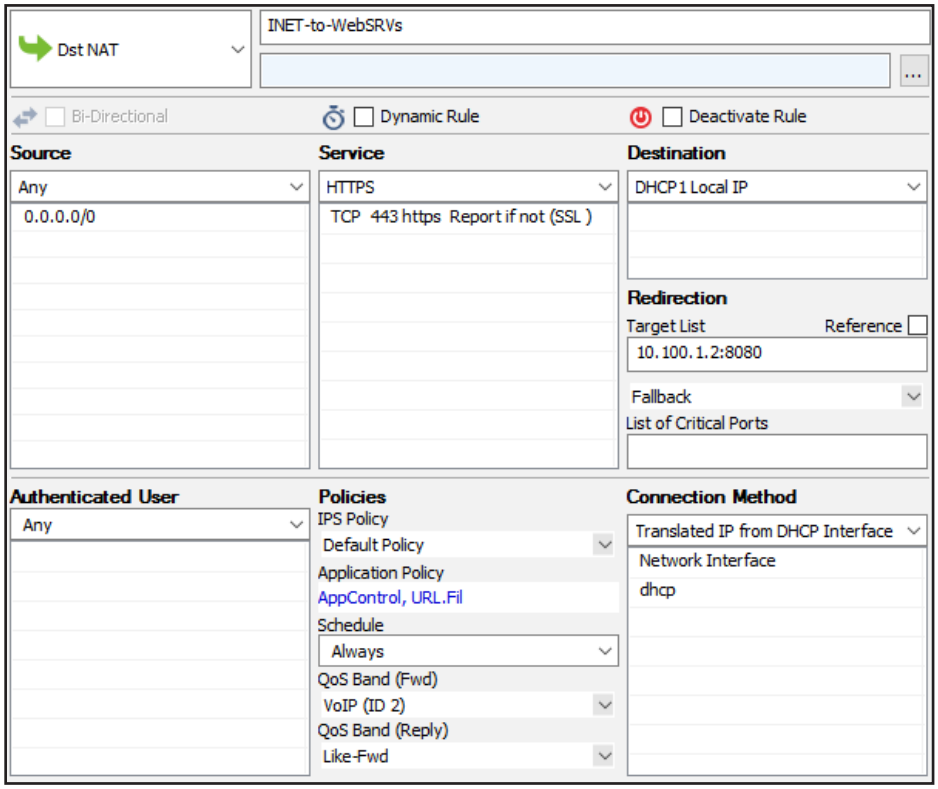
Dynamic Interface Handling	
Source Interface	VPNClients
Continue on Source Interface Mismatch	Yes
Reverse Interface (Bi-directional)	Matching

Internet to Backend Services not Using the Firewall as the Default Gateway

Create the following access rule to forward traffic from the Internet to an internal web server.



- **Action** – Select **Dst NAT**.
- **Source** – Select **Any** or a network object containing the networks the ELB is deployed in.
- **Service** – Select the service. e.g., **HTTP+S**.
- **Destination** – Select **DHCP1 Local IP**.
- **Connection Method** – Select **Dynamic NAT** or **Translated from DHCP Interface**.
- **Redirection Target** – Enter the IP address of the backend service. Optionally, append the port number to redirect to a different port. e..g, `10.100.1.2` or `10.100.1.2:8080`

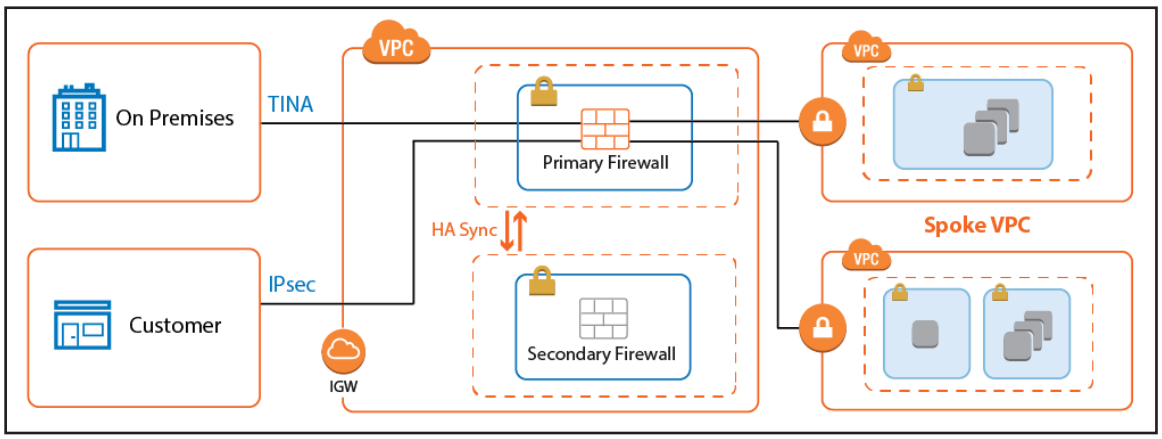

Dst NAT
INET-to-WebSRVs
Bi-Directional
Dynamic Rule
Deactivate Rule
Source
Any
0.0.0.0/0
Service
HTTPS
TCP 443 https Report if not (SSL)
Destination
DHCP1 Local IP
Redirection
Target List
Reference
10.100.1.2:8080
Fallback
List of Critical Ports
Authenticated User
Any
Policies
IPS Policy
Default Policy
Application Policy
AppControl, URL.Fil
Schedule
Always
QoS Band (Fwd)
VoIP (ID 2)
QoS Band (Reply)
Like-Fwd
Connection Method
Translated IP from DHCP Interface
Network Interface
dhcp

2.4 Transit VPC using NextGen Firewall

Connecting multiple VPCs to multiple locations, such as your datacenter or customer offices, can cause significant configuration overhead, especially if VPCs are frequently added and removed. For example, adding a new VPC requires configuration changes to each on-premises location. A second weak point is the communication between the VPCs. To share common resources, VPCs must be peered if they are in the same region; otherwise, the traffic must be routed through your datacenter.

To reduce the number of VPN connections required by each device participating in the network, use a central VPC as a Transit VPC and arrange the VPCs in a hub and spoke topology. The Transit VPC uses a NextGen Firewall High Availability Cluster or a NextGen Firewall Cold Standby Cluster as the VPN hub for all site-to-site VPN tunnels.

Shared services used by all spoke VPCs can be located in the Transit VPC or in a separate VPC peered to the Transit VPC. The service VPC can also host replicated on-premises services to save bandwidth to the datacenter.



2.4.1 Use Cases for a NextGen Firewall Transit VPC

The Transit VPC is a very versatile and flexible architecture that can be combined with the other reference architectures, except multi-NIC Segmentation, to create a central firewall hub for all your cloud resources.

2.4.2 Deploying a Transit VPC via CloudFormation Templates

It is recommended to deploy the Transit VPC via a CloudFormation template. The template deploys a NextGen High Availability Cluster in the Transit VPC and two spoke VPCs with VPN gateways. The firewalls are automatically joined into the High Availability Cluster, but failing over the Elastic IP addresses requires manual configuration steps.

To configure the site-to-site VPN from the VPN gateways:

Create an IAM role for the firewall cluster. For step-by-step instructions, see 3.1 How to Create an IAM Role for an F-Series Firewall in AWS (page 79)

Download the **NGF_TransitVPC.json** template and parameter file from the Barracuda Network GitHub account:

https://github.com/barracudanetworks/ngf-aws-templates.

Accept the Software Terms for the **Barracuda NextGen Firewall PAYG** or **BYOL** image in the AWS Marketplace.

Create a parameter template file containing your parameters values.

Deploy the **transit_vpc.json** CloudFormation template via AWS CLI or AWS console.

```
aws cloudformation create-stack  --stack-name "YOUR_STACK_NAME"--
template-body YOUR_S3_BUCKET/NGF_TransitVPC.json --parameter YOUR_S3_
BUCKET/NGF_TransitVPC_parameters.json
```

During deployment, the following resources are created by the template:

- One Transit VPC with a NextGen Firewall High Availability Cluster.
- Two Elastic IP addresses for the firewall cluster.
- Two spoke VPCs with VPN gateways.

After deploying the template, the following manual configuration steps are required to finish the setup:

- Configure site-to-site VPN tunnels and BGP routing for each VPN gateway.
- Configure Elastic IP addresses to fail over with the virtual server.

For step-by-step instructions on how to deploy a CloudFormation template, see 3.10 How to Deploy an F-Series Firewall in AWS via CloudFormation Template (page 139)

Configure Elastic IP Address Transfer

Since the AWS VPN gateway can only be configured to use one IP address, the same elastic IP address must always be associated with the active firewall in the cluster. Configure the virtual server on the firewall to execute an AWS CLI command that reassigns the Elastic IP addresses every time the virtual server fails over. Write down the Elastic IP addresses associated with the primary and secondary firewalls:

- **Primary Firewall** – Elastic IP address for the active firewall.
- **Secondary Firewall** – Elastic IP address for the passive firewall.

1. Log into the primary firewall with NextGen Admin.
2. Go to **CONFIGURATION > Configuration Tree > Box > Virtual Servers > S1 > Server Properties**.
3. Click **Lock**.
4. In the left menu, select **Custom Scripts**.
5. Enter the **Start Script** AWS CLI command to re-associate the active Elastic IP address when the virtual server starts.

```
/opt/aws/bin/aws ec2 associate-address --instance-id $(/usr/bin/
curl -s http://169.254.169.254/latest/meta-data/instance-id)
--allocation-id ACTIVE_ELASTIC_IP_ID --allow-reassociation
```

6. In the **Stop Script**, enter the AWS CLI command to re-associate the passive Elastic IP address when the virtual server shuts down.

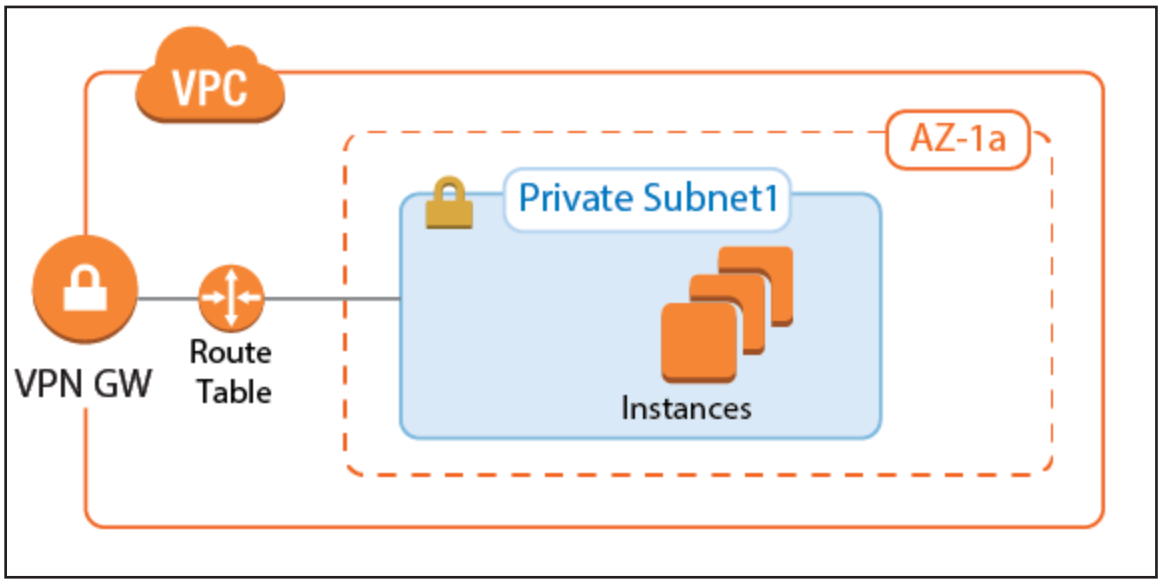
```
/opt/aws/bin/aws ec2 associate-address --instance-id $(/usr/bin/
curl -s http://169.254.169.254/latest/meta-data/instance-id)
--allocation-id PASSIVE_ELASTIC_IP_ID --allow-reassociation
```

7. Click **Send Changes** and **Activate**.

AWS VPN Gateway

The AWS VPN gateway connects the EC2 instances in the VPC to the Transit VPC via VPN connections. The customer gateway is configured for the Elastic IP address associated with the active firewall. Each VPN connection to the AWS VPN gateway is made up of two parallel IPsec IKEv1 tunnels. BGP is configured on the firewall to prefer the first tunnel and to use the secondary tunnel in case the primary is down.

The routing between the Transit VPC and the spokes is handled by BGP. The spoke VPCs learn the default route from the firewall and send all traffic through the VPN gateway and the Transit VPC high availability firewall cluster. The firewall learns the spoke VPC networks propagated by the VPN gateway. When a spoke VPC is added or removed, BGP automatically propagates the changes to all connected networks.



For step-by-step instructions, see Step 1 in 3.9 How to Configure an IKEv1 IPsec VPN to an AWS VPN Gateway with BGP (page 121)

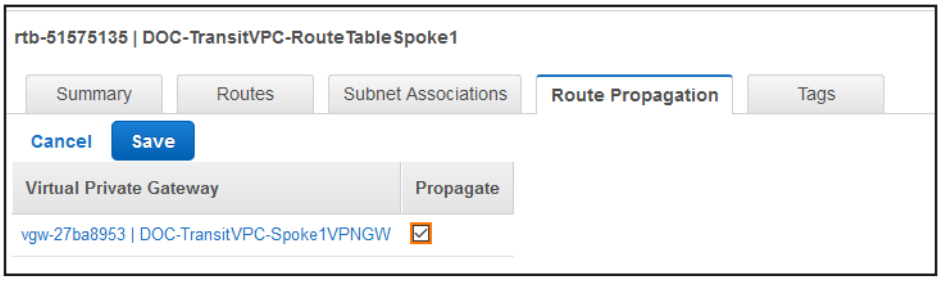
AWS Route Tables

The AWS route tables can be configured with static routes over the VPN gateway, or they can be configured to learn the routes via BGP. Using BGP has the advantage of being able to control all routing in the firewall's BGP service. However, whether static or dynamic, it is recommended to configure the default route through the VPN gateway. This ensures that all traffic for the VPC passes through the firewalls and that the security policies can be applied in one central location.

Configure the AWS route table for the spoke VPCs to learn the routes propagated by the firewall BGP service. To send all traffic through the Transit VPC, propagate the default route to the spoke VPCs. If propagated routes in the AWS route tables overlap with the local route of the VPC, the local route is always preferred. This applies not only to the local route, but also to all static routes. Static routes are preferred over the learned routes.

Enabling Route Propagation for AWS Route Tables

1. Log into the AWS console.
2. Click **Services**, and select **VPC**.
3. In the **Virtual Private Cloud** section of the left menu, click **Route Tables**.
4. (optional) Filter the list using the VPC ID.
5. Select the route table for the spoke VPC.
6. In the lower half of the page, click the **Route Propagation** tab.
7. Click **Edit**.
8. Select the VPN gateway and click **Save**.



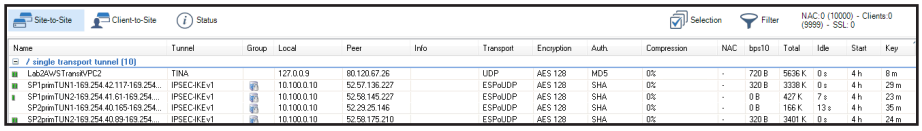
Configure the IPsec Tunnels on the Transit VPC Firewalls

To connect the spoke VPC to the Transit VPC, configure two IPsec tunnels: two parallel IPsec tunnels to the Elastic IP of the active firewall. AWS defines a /30 intermediary network for each IPsec tunnel. The IP addresses in this intermediary network are used by BGP. Define BGP neighbors for each next-hop address as per the instructions provided by AWS.

The VPN connection information is unique for each VPN connection and can be downloaded by right-clicking the VPN connection. In addition to the encryption settings in the AWS configuration file, the following settings are supported:

- **Encryption** – AES, AES256
- **Hash** – SHA1, SHA256

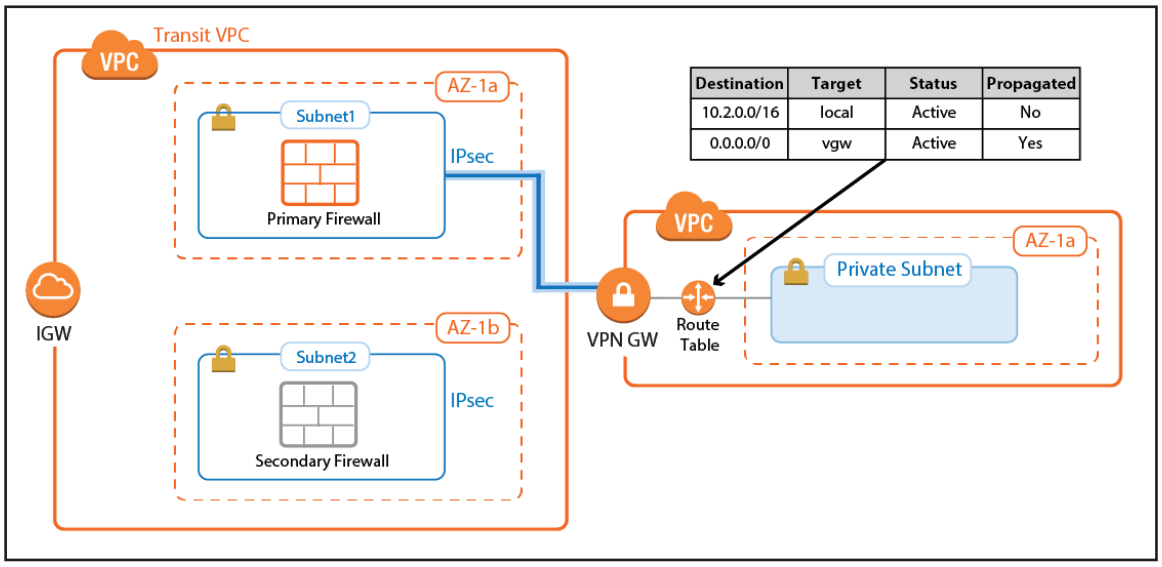
- **Phase 1 DH-Group** – Group 2 and Group 14-18
- **Phase 2 DH-Group** – Group 1, 2, 5 and Group 14-18

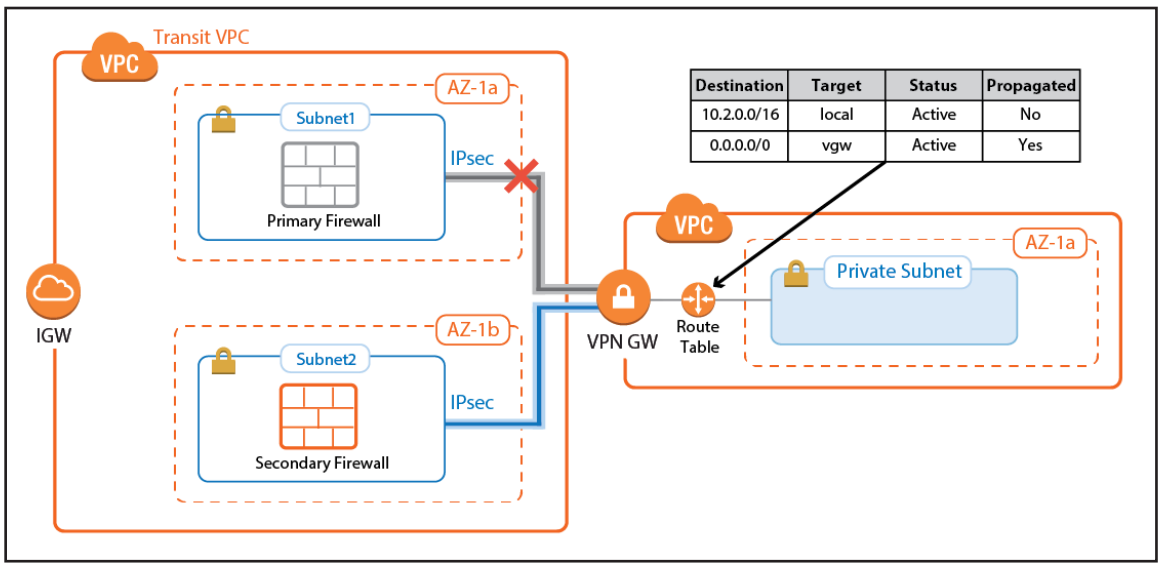


For step-by-step instructions, see Step 2 in 3.9 How to Configure an IKEv1 IPsec VPN to an AWS VPN Gateway with BGP (page 121)

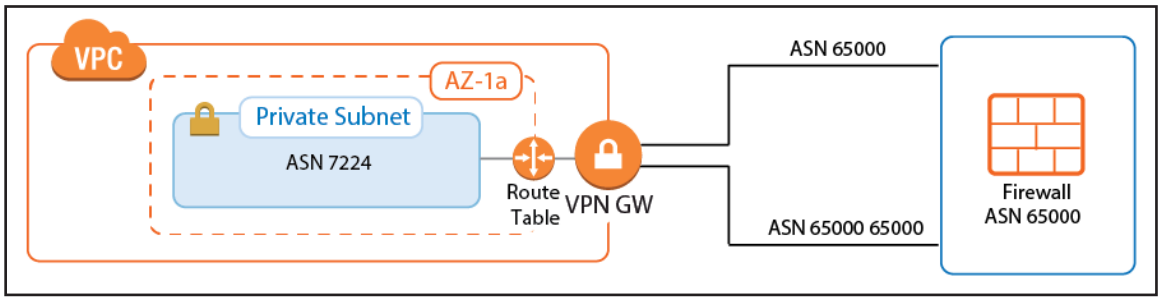
Configure BGP on the Transit VPC Firewalls

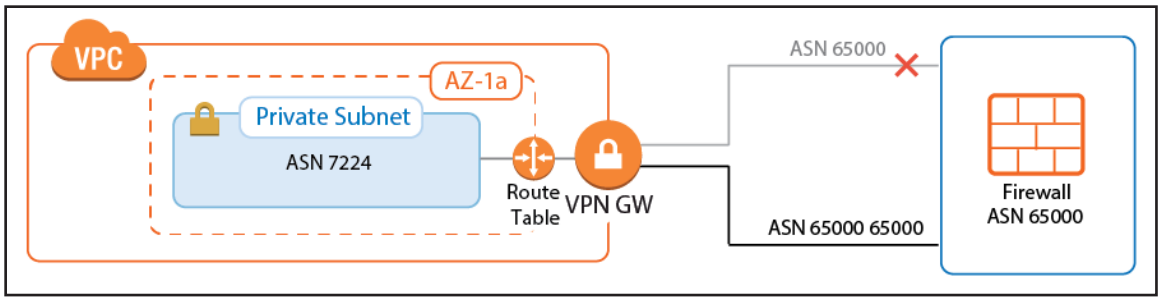
The BGP service on the firewall learns and propagates the routes from each location. Create a BGP neighbor configuration for each IPsec tunnel and each on-premises network connected to the Transit VPC. If you are not using a static route in the spoke VPCs routing table, propagate the default route to the BGP neighbor for each spoke VPC. The VPN gateway automatically propagates the VPC network via BGP. Since spoke VPCs are always connected by two parallel IPsec tunnels, the route over one IPsec tunnel should be preferred over the other.





Configure the BGP service on each firewall to exchange information with the BGP service on the other side of the VPN tunnels. Using **Route Maps**, modify the routes learned for the second of the parallel IPsec connections. By lengthening the AS PATH of the IPsec tunnels, traffic is sent through the first tunnel at all times, unless the tunnel is down.





For step-by-step instructions, see Step 3 in

3.9 How to Configure an IKEv1 IPsec VPN to an AWS VPN Gateway with BGP (page 121)

Create Access Rules to Allow Traffic

By default, the Forwarding Firewall service blocks all traffic not explicitly allowed by an access rule. Since all traffic is routed through the Transit VPC, create access rules to allow access for individual services and/or entire networks. Access rules allowing traffic through the AWS VPN gateway IPsec tunnels must set the following advanced access settings:

- **Force MSS (Maximum Segment Size)** – Set to `1387`.
- **Clear DF Bit** – Set to **yes**.
- **Reverse Interface (Bi-directional)** – If you are using two parallel IPsec tunnels per firewall, set this to **Any**. This allows the traffic to use either IPsec tunnel.

Be sure to sync the access rules on both firewalls to make sure that the behavior is identical no matter which firewall the traffic is sent through.

For step-by-step instructions, see Step 4 in 3.9 How to Configure an IKEv1 IPsec VPN to an AWS VPN Gateway with BGP (page 121)

Launching EC2 Instances in Spokes

If your Transit VPC is created with spokes in a single CloudFormation template, the instances will not have Internet access during launch. Use NAT gateways or VPC endpoints in the spoke VPC to access AWS services before the VPN connection and BGP routing to the firewall is configured.

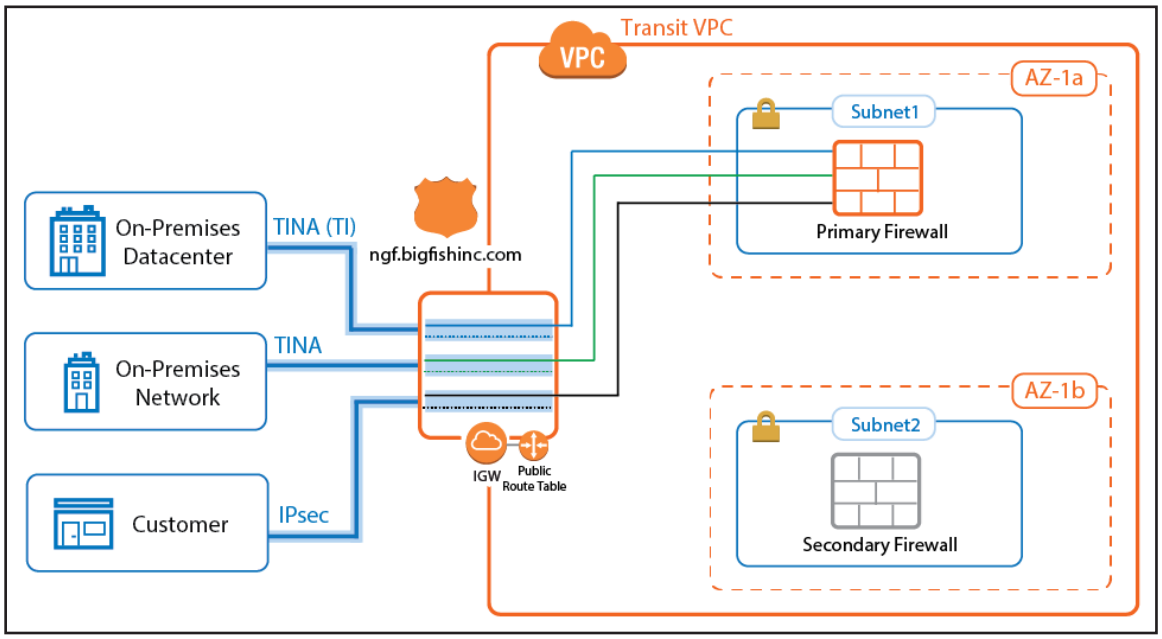
If your spoke is already connected, verify that access rules are in place that allow the new instance access to all resources required during the provisioning process. If unsure, log into the active firewall and use the **Firewall > History** page in NextGen Admin to check if traffic from the instance was blocked.

For more information, see NextGen Admin History Page.

2.4.3 Connecting to On-Premises Networks

To be able to forward traffic between your AWS VPC and your on-premises networks, create site-to-site VPN tunnels between the High Availability Cluster in the Transit VPC and the VPN gateways in each remote location.

The networks of the spoke VPC are propagated via BGP over the VPN tunnels. BGP is used to propagate the AWS VPC networks to your on-premises locations. Depending on the remote device, you can use either Barracuda's proprietary TINA VPN or the industry standard IPsec VPN protocol. Failover and preference of the VPN tunnel to the primary firewall is handled by BGP.



TINA Site-to-Site VPN Tunnels to F-Series Firewalls

If the remote location uses an F-series Firewall, you can take advantage of the TINA VPN protocol. TINA offers many enhancements not featured in the standard IPsec protocol, such as Traffic Intelligence, Traffic Compression, and WAN Optimization. Traffic Intelligence is a logical layer used to manage multiple parallel VPN tunnels (transports) in one VPN tunnel configuration. So if your remote location has multiple Internet connections (perhaps in combination with AWS Direct

Connect), all connections can be combined into one VPN tunnel. Traffic Intelligence patterns in the connection object of the access rule determine how the traffic is distributed over the VPN transports and failover behavior. WAN Optimization and Compression reduces the amount of traffic sent through the tunnel by using data deduplication .

For more information, see How to Configure BGP Routing over a TINA VPN Tunnel, Traffic Intelligence, and WAN Optimization.

IPsec Site-to-Site VPN Tunnels to Third-Party Devices

Third-party VPN gateways can be connected via IPsec IKEv1 or IKEv2 VPN tunnels. The remote device must support routing BGP over IPsec tunnels to be able to learn the routes. Create one IPsec tunnel from the active EIP for each on-premises location.

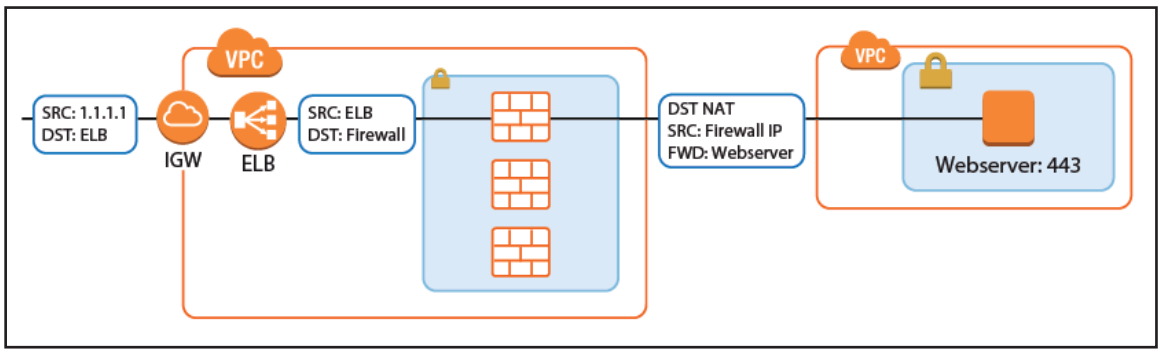
For more information, see 3.9 How to Configure an IKEv1 IPsec VPN to an AWS VPN Gateway with BGP (page 121)

Create Access Rules for On-Premises Networks

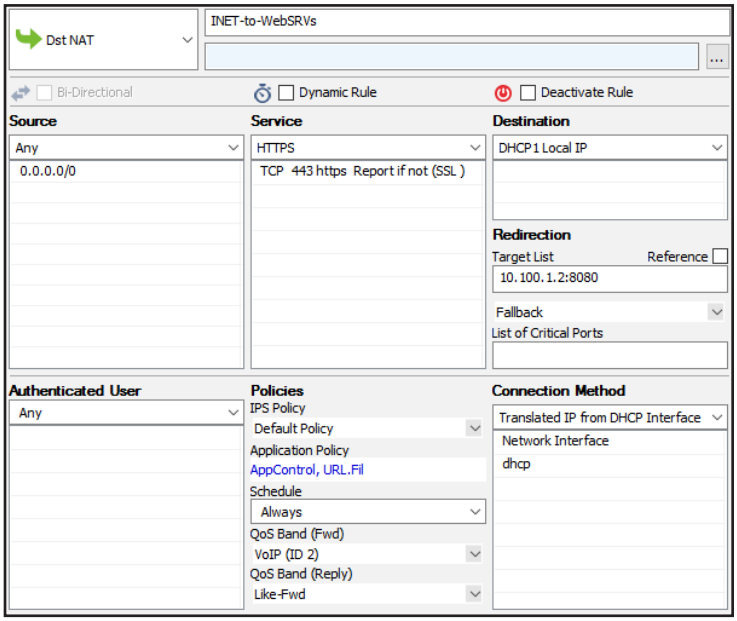
Just like when connecting the spoke VPCs, the firewall blocks all traffic by default. To allow connections to the networks learned via BGP, create pass access rules on both firewalls. These rules must be the same on both firewalls to ensure that if the connection fails over to the secondary firewall, the same policies are applied. Access rules to cloud services connected to the Transit VPC via VPC peering must translate the source IP address to the IP address of the DHCP interface of the firewall to satisfy the AWS restriction on peering that transitive VPCs are not allowed.

Internet to Backend Services

Create the following access rule to forward traffic from the Internet to an internal web server.

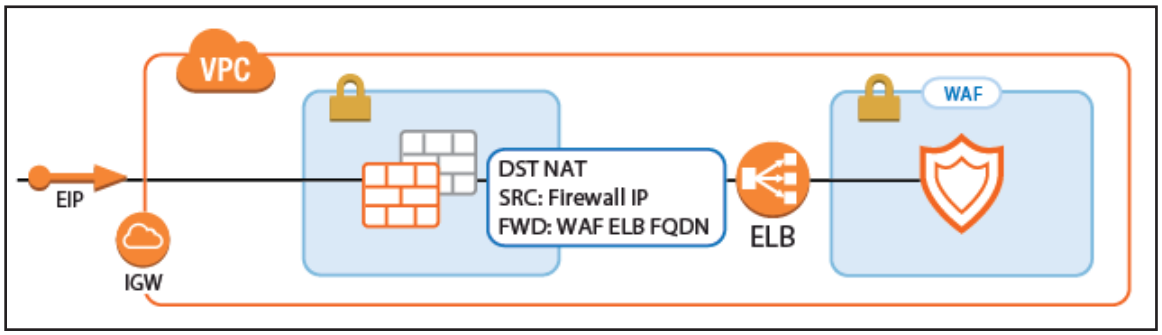


- **Action** – Select **Dst NAT**.
- **Source** – Select **Any** or a network object containing the networks the ELB is deployed in.
- **Service** – Select the service. e.g., **HTTP+S**.
- **Destination** – Select **DHCP1 Local IP**.
- **Connection Method** – Select **Dynamic NAT** or **Translated from DHCP Interface**.
- **Redirection Target** – Enter the IP address of the backend service. Optionally, append the port number to redirect to a different port. e..g, `10.100.1.2` or `10.100.1.2:8080`

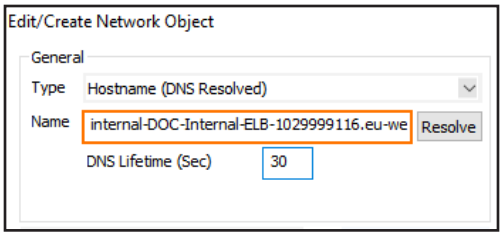


Redirect Traffic through a WAF Cluster or Other Service Behind an Internal ELB

Services behind an internal ELB can also be forwarded via Dst NAT access rule.

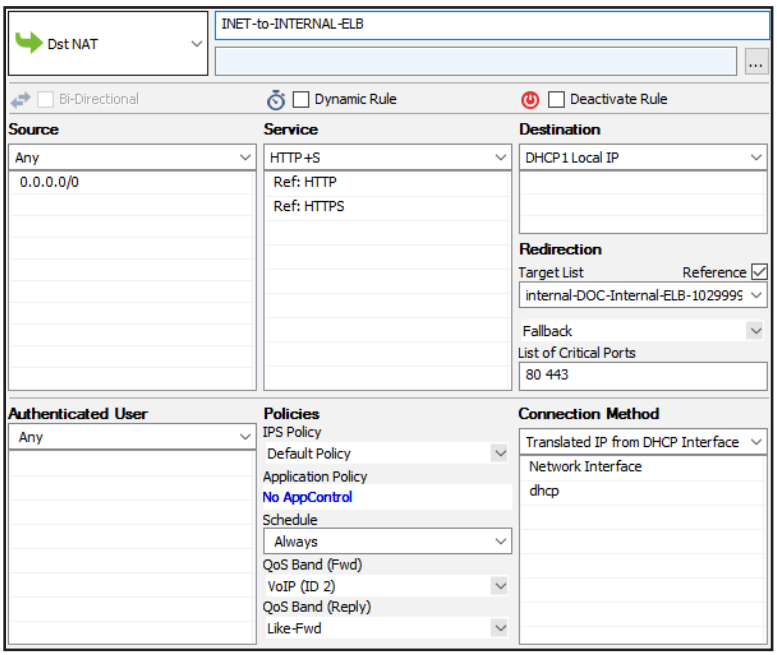


1. Create a hostname network object for the internal DNS name of the ELB, set the **DNS Lifetime** to 30 seconds, and click **Send Changes**.



2. Create the access rule:
   - **Action** – Select **Dst NAT**.
   - **Source** – Select **Any** or a network object containing the networks the ELB is deployed in.
   - **Service** – Select the service. e.g., **HTTP+S**.
   - **Destination** – Select **DHCP1 Local IP**.

- **Connection Method** – Select **Dynamic NAT** or **Translated from DHCP Interface**.
- **Redirection Target** – Click **Reference** and select the network object for the ELB.

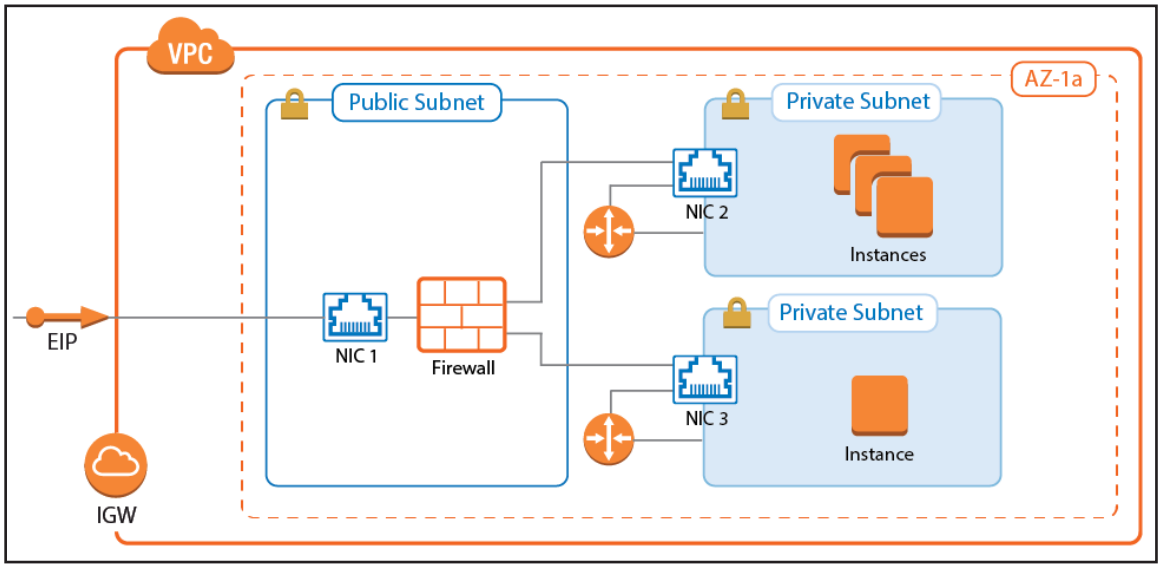


2.5 Segmentation Firewall for Single AZ VPCs

A NextGen Firewall F with multiple network interfaces can be used as a segmentation firewall for your private subnets in the VPC. Traffic passing between the private subnets is routed through the firewall, where you can apply security policies and visualize traffic in real time between the subnets. To be able to route the traffic over the firewall, the standard route for internal VPC traffic must be circumvented. By default, all traffic within the VPC is routed over the default gateway. This route cannot be overridden by other more specific routes, nor can it be changed to use the firewall as the gateway instead. Using a combination of a firewall instance with multiple network interfaces and adding a route on the client instances allows you to use the F-Series Firewall as a segmentation firewall in AWS.

Use a segmentation firewall to enforce access policies and monitor traffic passing between the subnets. When compared with an AWS native solution, a NextGen Firewall is vastly superior regarding the depth at which both traffic can be inspected and security policies applied. In addition, NextGen Admin also provides real-time traffic visibility, and the Firewall Live and History pages allow quick, fine-grained access to all the traffic currently passing through the firewall.

For the firewall, select the instance type according to the number of network interfaces. The number of network interfaces is the number of private subnets plus one for the public subnet. At least three network interfaces are required. The instance type must support at least three network interfaces: one for the public subnet and two for the private subnets.



2.5.1 Use Cases for a Multi-NIC Segmentation Firewall

A NextGen Firewall Segmentation is deployed like an internal firewall for applications moved to AWS using lift-and-shift migrations.

2.5.2 Limitations

- All resources must be in a single Availability Zone.
- The number of private subnets is limited by the number of network interfaces supported by the instance type. So if the firewall supports three network interfaces, two private subnets can be connected. The primary network interface is used for external connectivity.
- A route must be added to the client instances in the private subnets. The default route over the gateway in the subnet bypasses the firewall. This can be stopped via Security Groups.
- Cannot be deployed as a High Availability Cluster.
- Connecting to subnets in other Availability Zones requires use of source NAT on the matching access rule.

2.5.3 Deploying a Segmentation Firewall via CloudFormation Template

It is recommended to deploy the Segmentation Firewall via a CloudFormation template. The template deploys one firewall that is automatically joined into the High Availability Cluster in the public subnets. The route table associated with the private subnets is configured to use the active firewall as the outbound gateway. This template only deploys the AWS infrastructure. The NextGen Firewall must be configured manually.

1. Create an IAM role for the firewall cluster. For step-by-step instructions, see 3.1 How to Create an IAM Role for an F-Series Firewall in AWS (page 79)
2. Download the **NGF_Segmentation.json** template and parameter file from the Barracuda Network GitHub account: https://github.com/barracudanetworks/ngf-aws-templates.
3. Accept the Software Terms for the **Barracuda NextGen Firewall PAYG** or **BYOL** image in the AWS Marketplace.
4. Create a parameter template file containing your parameters values.
5. Deploy the template via AWS CLI or AWS console.

```
aws cloudformation create-stack  --stack-name "YOUR_STACK_NAME"
--template-body YOUR_S3_BUCKET/NGF_Segmentation.json --parameter YOUR_
S3_BUCKET/NGF_Segmentation_parameters.json
```

During deployment, the following resources are created by the template:

- One VPC with one public and two private subnets in the same AZ.
- One Barracuda NextGen Firewall with three ENIs.

For step-by-step instructions on how to deploy a CloudFormation template, see 3.10 How to Deploy an F-Series Firewall in AWS via CloudFormation Template (page 139)

2.5.4 (Alternative) Deploying a Segmentation Firewall via AWS Console

Complete the following configuration steps to deploy the NextGen Firewall F as a segmentation firewall. For more detailed descriptions, follow the links for step-by-step instructions.

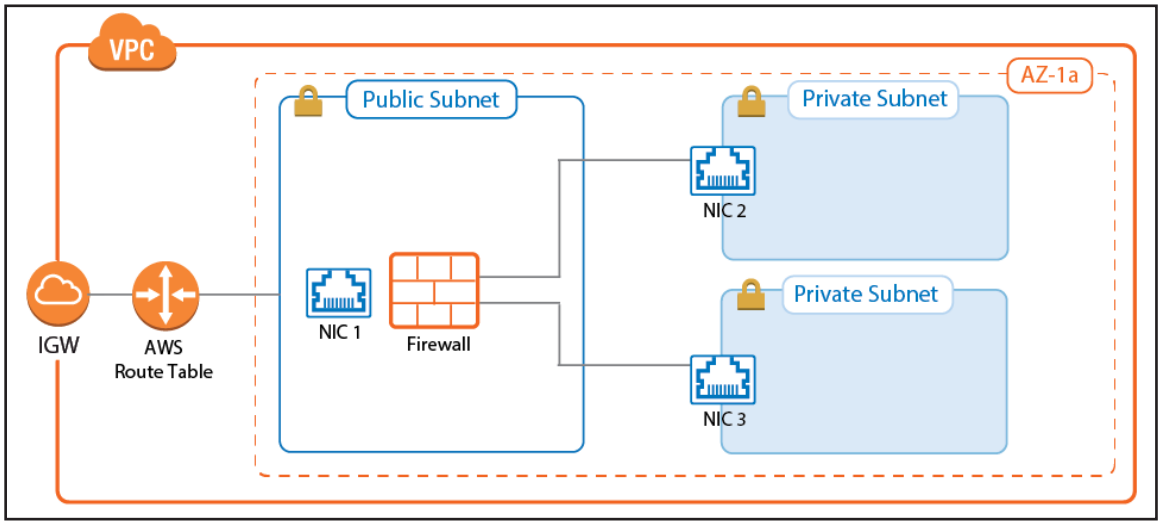
Create a VPC with the public and private subnets all in one Availability Zone.

- Launch a NextGen Firewall instance into the public subnet.
- Add an additional ENI per private subnet.

For step-by-step instruction, see 3.11 How to Deploy an F-Series Firewall in AWS via Web Portal (page 143)

2.5.5 Adding Additional Network Interfaces for Each Private Subnet

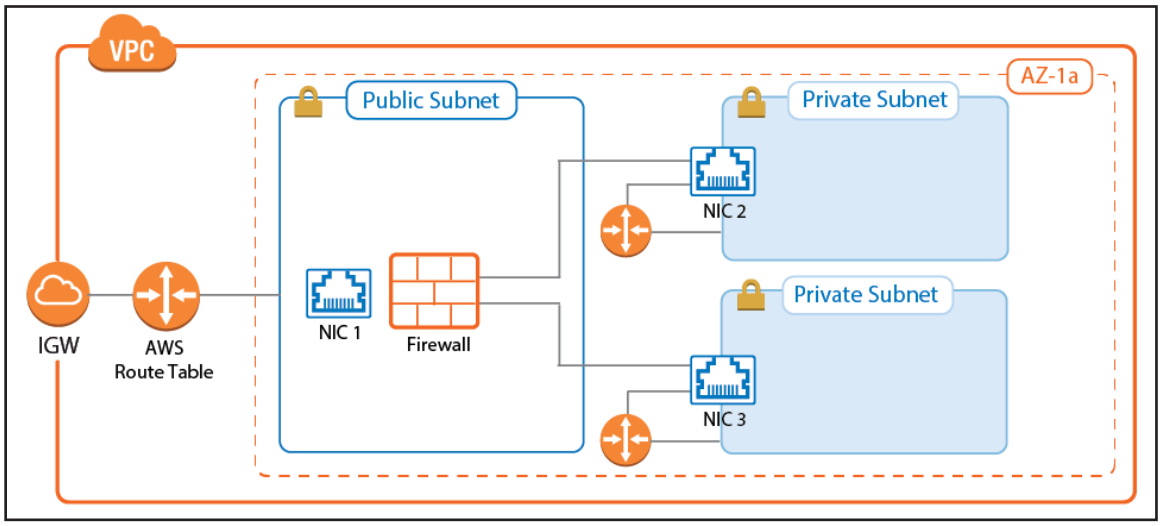
The firewall must have a network interface in each private subnet. Create an AWS elastic network interface (ENI) for each private subnet in your VPC. The private IP address must be set explicitly to be able to configure the network interface statically. Also, disable the source/destination check for each interface to be able to process traffic with a destination address not matching the private IP of the network interface. Before attaching the ENIs to the firewall, shut the firewall instance down. Attach the network interfaces. After starting the firewall, configure the new network interfaces and add the required direct attached routes and virtual server IP addresses.



For step-by-step instructions, see 3.4 How to Add AWS Elastic Network Interfaces to a Firewall Instance (page 95)

Route Table for Private Subnets

For each private subnet, a dedicated AWS route table handles all traffic with destinations outside the VPC. Associate the subnet with the route table and create a default route with the network interface of the firewall in this subnet as the target.



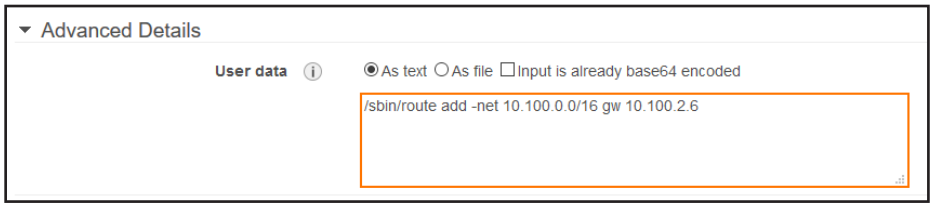
For step-by-step instructions, see 3.5 How to Configure AWS Route Tables for Firewalls with Multiple Network Interfaces (page 101)

2.5.6 Deploying Instances to Use the Firewall as Default Gateway

It is not currently possible to configure the AWS route table to send traffic between two subnets through the firewall instance. By default, each route table includes a static route for the VPC pointing to the AWS gateway of the subnet. This route cannot be overridden by a more specific route, nor can it be deleted. To send traffic via the firewall, add a route directly on the instance. The route can be added either manually after the instance has been deployed, or automatically in the **User data** section.

AWS Console (Linux Instances Only)

Add the routes to **User data** field of the **Advanced Details** section.



CloudFormation (Linux Instances Only)

Add the definition for the routes in the **UserData** section of the CloudFormation template. If multiple private subnets are used, more than one route may be required.

```
            "UserData": {
                                "Fn::Base64": {
                                    "Fn::Join": [
                                        "", [
                                            "#!/bin/bash\n\n",
                                            "/sbin/route add -net 10.100.1.0/16 gw
        10.100.2.6",
                                                    "\n" ]
                                    ]
                                }
                            },
```

Manually (Linux Instances Only)

Log into the instance via SSH, and with root privileges enter:

```
root@ip-10-100-2-10:/home/ubuntu# route add -net 10.100.0.0/16 gw 10.100.2.6
```

The route is now in the route table. Enter `route -n` to list the routes:

```
root@ip-10-100-2-10:/home/ubuntu# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.100.2.1      0.0.0.0         UG    0      0        0 eth0
10.100.0.0      10.100.2.6      255.255.0.0     UG    0      0        0 eth0
10.100.2.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
root@ip-10-100-2-10:/home/ubuntu#
```

Firewall Service Configuration

Now that the routing and setup in AWS is complete, access rules must be configured to apply your security policies to the traffic passing between the VPC subnets:

- **Network objects** – Create network objects for the VPC, for each subnet, and for individual instances. For more information, see Network Objects.
- **Access rules** – By default, all connections are blocked. Create access rules for each service the instances are allowed to access. Use the **FIREWALL > Live** and **FIREWALL > History** pages to verify which rule matches and which traffic is blocked. For more information, see Live Page and History Page.

Access rules allowing the backend instances access to the Internet must use the **Dynamic NAT** connection objects to rewrite the source IP of the packets to the IP address of the firewall.

Step-by-Step Guides

3.1 How to Create an IAM Role for an F-Series Firewall in AWS

IAM roles are the preferred method for NextGen Firewall instances in AWS to authenticate against AWS APIs. For each feature that requires direct access to AWS resources, a customized IAM policy must be created. These policies are then attached to the IAM role assigned to the instance during deployment. It is not possible to add a role to an existing instance. It is possible, however, to change the IAM policies attached to the IAM role on the fly. If an Access Key ID and Secret Access Key are configured in AWS cloud integration, they take precedence over the IAM role attached to the instance. In order to use all firewall features, the following IAM security policies must be created and attached to the IAM role:

- Cloud Information element
- Route shifting (includes Cloud Information dashboard element)
- AWS CloudWatch streaming
- AWS Auto Scaling or cold standby S3 bucket access

Step 1. Create IAM Policy for Route Shifting

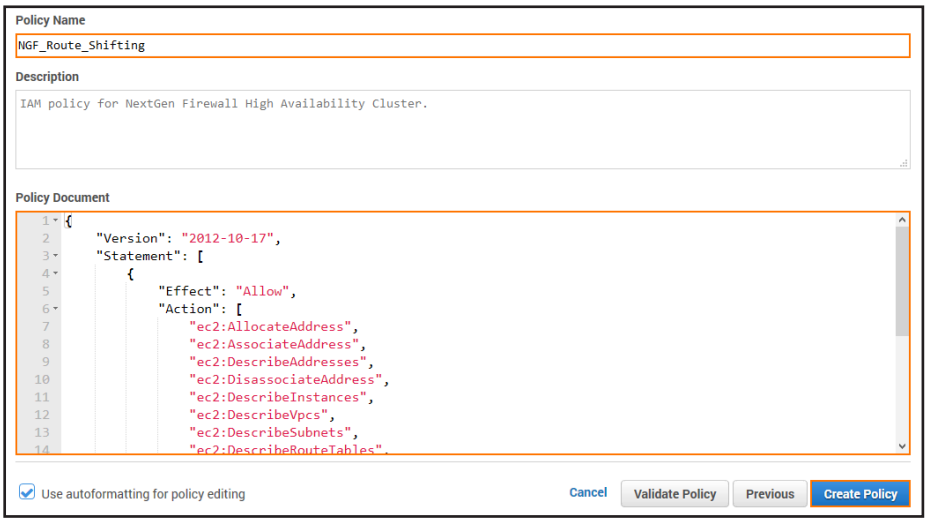
Create an IAM policy to allow route shifting.

1. Log into the AWS console.
2. Click **Services** and select **IAM**.
3. In the left menu, click **Policies**.
4. Click **Create Policy**.

5. Next to **Create Your Own Policy**, click **Select**.

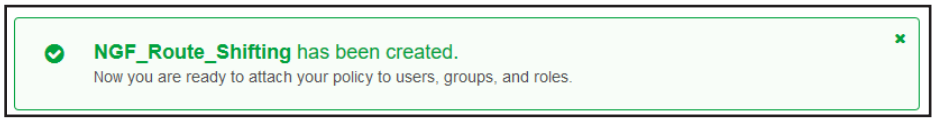
6. Configure the IAM policy:
   - **Policy Name** – Enter a name for the policy.
   - **(optional) Description**
   - **Policy Document** – Copy and paste the following policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AllocateAddress",
                "ec2:AssociateAddress",
                "ec2:DescribeAddresses",
                "ec2:DisassociateAddress",
                "ec2:DescribeInstances",
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets",
                "ec2:DescribeRouteTables",
                "ec2:DeleteRoute",
                "ec2:CreateRoute",
                "ec2:DescribeNetworkInterfaces"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```



7. Click **Create Policy**.

The IAM policy for route shifting is now available to be assigned to an IAM role for the NextGen Firewall.



Step 2. Create IAM Policy for the Cloud Information Dashboard Element

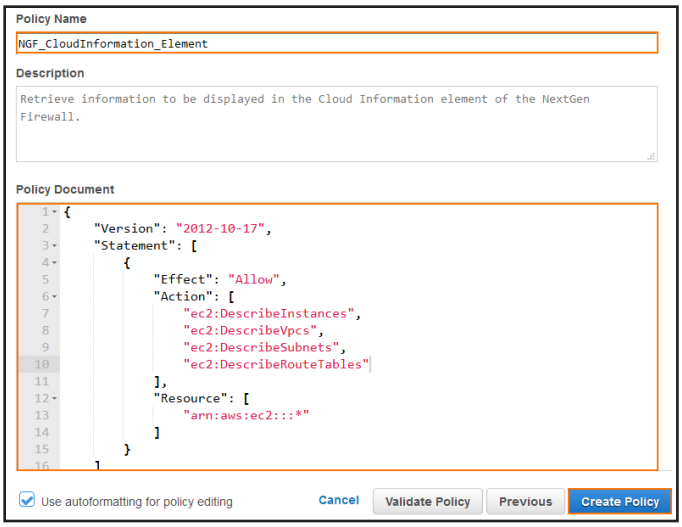
Create this policy only if you are not using the route shifting IAM policy. The route shifting IAM policy includes all permissions necessary for the Cloud Information element.

1. Log into the AWS console.
2. Click **Services** and select **IAM**.
3. In the left menu, click **Policies**.
4. Click **Create Policy**.
5. Next to **Create Your Own Policy**, click **Select**.

   Configure the IAM policy:

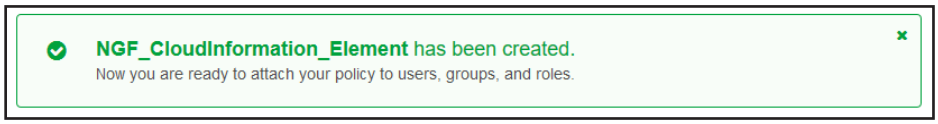
   - **Policy Name** – Enter a name for the policy.
   - **(optional) Description**
   - **Policy Document** – Copy and paste the following policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets",
                "ec2:DescribeRouteTables"
            ],
            "Resource": [
                "arn:aws:ec2:::*"
            ]
        }
    ]
}
```



6. Click **Create Policy**.

The IAM policy for the Cloud Information element is now available to be assigned to an IAM role for the NextGen Firewall.

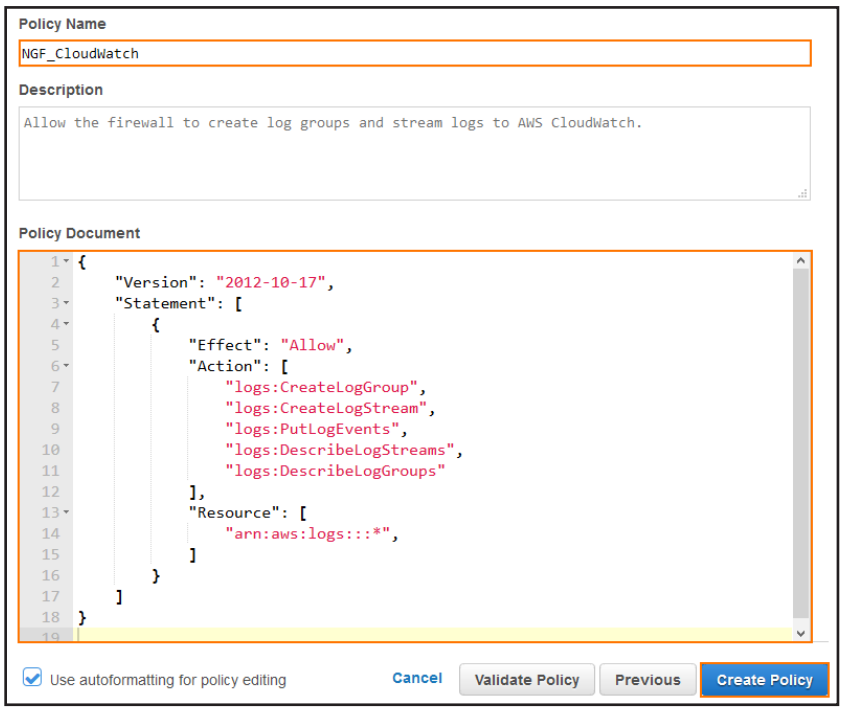


Step 3. Create IAM Policy for Log Streaming to AWS CloudWatch

This IAM policy grants the firewall the necessary permissions to stream logs to AWS CloudWatch.

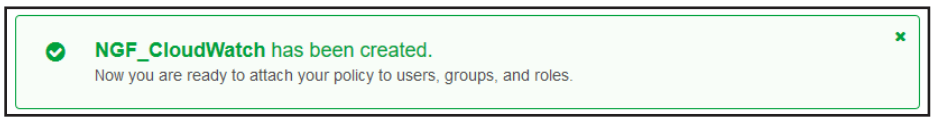
1. Log into the AWS console.
2. Click **Services** and select **IAM**.
3. In the left menu, click **Policies**.
4. Click **Create Policy**.
5. Next to **Create Your Own Policy**, click **Select**.
6. Configure the IAM policy:
   - **Policy Name** – Enter a name for the policy.
   - **(optional) Description**
   - **Policy Document** – Copy and paste the following policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:DescribeLogStreams",
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "arn:aws:logs:*:*:*"
            ]
        }
    ]
}
```



7. Click **Create Policy**.

The IAM policy for streaming logs to AWS CloudWatch is now available to be assigned to an IAM role for the NextGen Firewall.



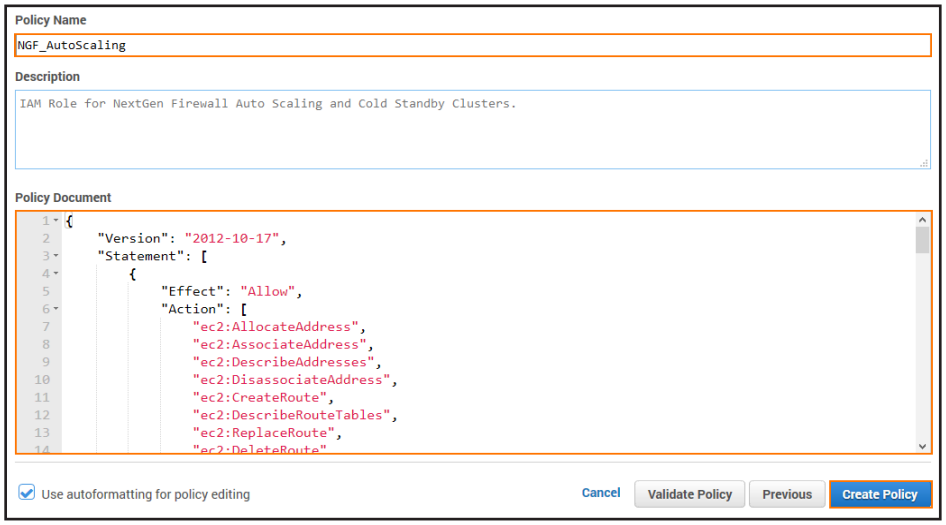
Step 4. Create IAM Policy for AWS Auto Scaling Group Deployments

This IAM policy grants the necessary permissions for Auto Scaling and cold standby architectures for the NextGen Firewall.

1. Log into the AWS console.
2. Click **Services** and select **IAM**.
3. In the left menu, click **Policies**.
4. Click **Create Policy**.
5. Next to **Create Your Own Policy**, click **Select**.
6. Configure the IAM policy:
   - **Policy Name** – Enter a name for the policy.
   - **(optional) Description**
   - **Policy Document** – Copy and paste the following policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AllocateAddress",
                "ec2:AssociateAddress",
                "ec2:DescribeAddresses",
                "ec2:DisassociateAddress",
                "ec2:CreateRoute",
                "ec2:DescribeRouteTables",
                "ec2:ReplaceRoute",
                "ec2:DeleteRoute",
                "ec2:CreateTags",
                "ec2:DescribeInstances",
                "ec2:DeleteTags",
                "ec2:DescribeTags",
                "ec2:ModifyInstanceAttribute"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "autoscaling:CreateOrUpdateTags",
                "autoscaling:DeleteTags",
                "autoscaling:DescribeAutoScalingGroups",
                "autoscaling:DescribeAutoScalingInstances",
                "autoscaling:DescribeTags",
                "autoscaling:SetInstanceProtection"
            ],
            "Resource": "*"
        },
        {
            "Action": [
                "sqs:CreateQueue",
                "sqs:DeleteMessage",
                "sqs:DeleteQueue",
                "sqs:GetQueueAttributes",
                "sqs:ReceiveMessage",
                "sqs:SetQueueAttributes",
                "sqs:GetQueueUrl"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:sqs:*"
        },
        {
            "Action": [
                "sns:CreateTopic",
                "sns:Publish",
                "sns:Subscribe",
                "sns:Unsubscribe",
                "sns:ListSubscriptionsByTopic"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:sns:*"
        },
        {
            "Action": [
```

```
                "cloudwatch:PutMetricData"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "sts:GetCallerIdentity"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:ListBucket",
                "s3:PutBucketVersioning",
                "s3:PutObject",
                "s3:GetBucketVersioning",
                "s3:ListBucketVersions",
                "s3:GetObject",
                "s3:GetObjectVersion",
                "s3:DeleteObjectVersion"
            ],
            "Resource": "arn:aws:s3:::*"
        }
    ]
}
```



7. Click **Create Policy**.

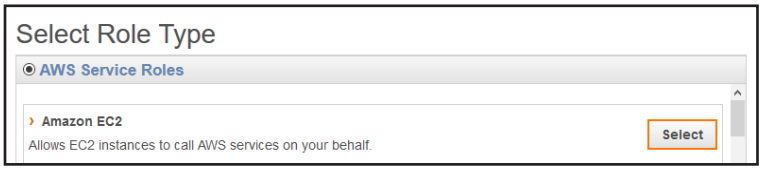
The IAM policy for AWS Auto Scaling and cold standby architectures is now available to be assigned to an IAM role for the NextGen Firewall.

Step 5. Create the IAM Role

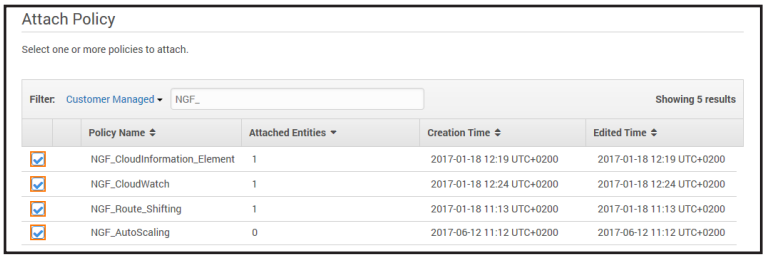
Create the IAM role and assign the IAM policies for all NextGen Firewall Cloud Integration features used by the firewall Instance.

1. Log into the AWS console.
2. Click **Services** and select **IAM**.
3. In the left menu, click **Roles**.
4. Click **Create New Role**.

5. Enter the **Role Name**.
6. Click **Next Step**.
7. In the **AWS Service Roles** section, next to **Amazon EC2** click **Select**.



8. Select the IAM firewall policies you just created.
9. Select the policies only for features that will be used in the deployed firewall instance. You can change the attached IAM policies later if required.



10. Click **Next Step**.
11. Review the settings and click **Create Role**.
12. Assign this role to the NextGen Firewall instance during deployment.

3.2 How to Configure Log Streaming to AWS CloudWatch

To stream log data from your firewall to AWS CloudWatch, you must configure AWS Cloud Integration and configure syslog streaming on the firewall. The destination is AWS CloudWatch. The configured log group is automatically created, and the logs are placed into a folder using either the instance ID or the hostname as the name.

3.2.1 Before You Begin

The firewall must be deployed with an IAM role that allows access to AWS CloudWatch. For more information, see

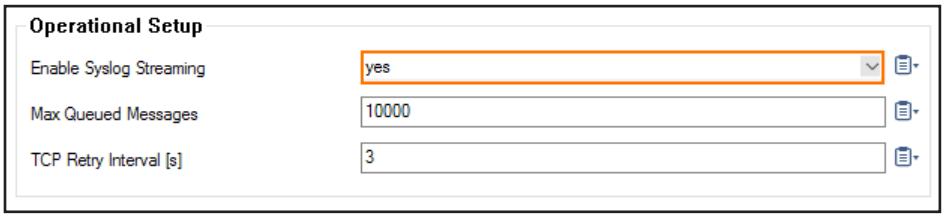
3.1 How to Create an IAM Role for an F-Series Firewall in AWS(page 79)

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:DescribeLogStreams",
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "arn:aws:logs:*:*:*"
            ]
        }
    ]
}
```

Step 1. Enable Syslog Streaming

Enable syslog streaming and, optionally, configure the AWS region if it is different from the region of the firewall instance.

1. Go to **CONFIGURATION > Full Configuration > Box > Infrastructure Services > Syslog Streaming**.
2. Click **Lock**.
3. Set **Enable Syslog Streaming** to **yes**.



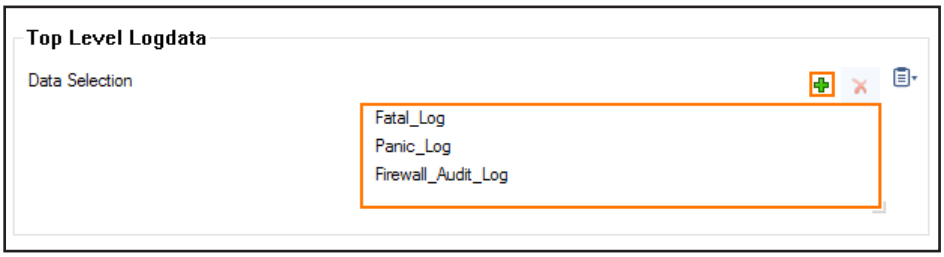
4. In the left menu, expand the **Configuration Mode** section and click **Switch to Advanced View**.

5. (optional) Enter the AWS CloudWatch region. e.g., `eu-west-1`
6. Click **Send Changes** and **Activate**.

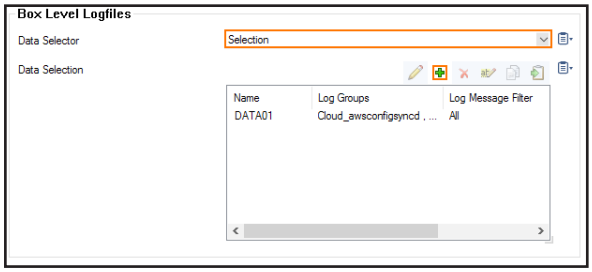
Step 2. Configure Logdata Filters

Define profiles specifying the log file types to be transferred / streamed. Log file are classified into top level, box level, and service level log data sources.

1. Go to **CONFIGURATION > Full Configuration > Box > Infrastructure Services > Syslog Streaming**.
2. In the left menu, select **Logdata Filters**.
3. Click **Lock**.
4. In the **Filters** table, click **+** to add a new filter. The **Filters** window opens.
5. Enter a **Name**.
6. Click **OK**.
7. In the **Data Selection** table, add the **Top Level Log Files** log files to be streamed. You can select:
   - **Fatal_log**
   - **Firewall_Audit_Log**– The firewall audit log must be enabled and configured, and **Audit Delivery** must be set to **Syslog Proxy**. For more information, see How to Enable the Firewall Audit Log Service. Alternatively, the firewall audit log can also be streamed as a part of the firewall service logs.
   - **Panic_log**



8. Configure the **Box Level Logfile** filters:
   a. From the **Data Selector** list, select which files for this category are streamed:
      - **All** – All box level logs are streamed.
      - **None** – Box level logs are not streamed.
      - **Selection** – Only box level log files defined in the **Data Selection** list are streamed.



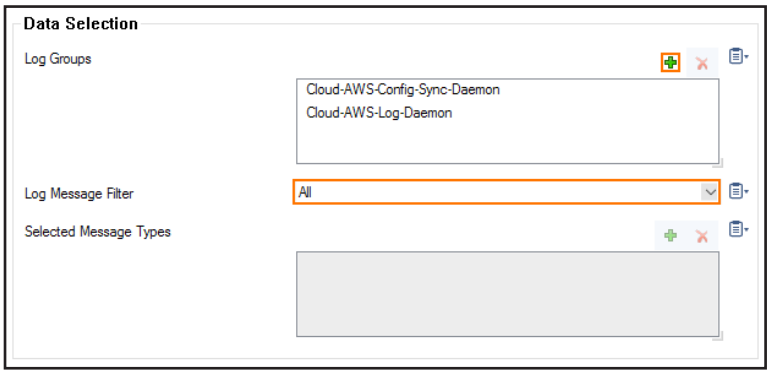
b. (**Selection** only) Click **+** to add custom filters to the **Data Selection** table.

i. In the **Log Groups** table, click **+**.

ii. Select the box level log files, or select **Other** to enter a **user defined log group pattern** to stream log files matching this pattern.

iii. (optional) From the **Log Level Filter** list, select the message types from the log group that are streamed.

iv. (**Selection** only) In the **Selected Messages Types** table, click **+** to add message types.



9. Configure the **Service Level Logfile** filters:

a. From the **Data Selector** list, select which files for this category are streamed:

- **All** – All service logs are streamed.
- **None** – Service level logs are not streamed.
- **Selection** – Only service level log files defined in the **Data Selection** list are streamed.

b. (**Selection** only) Click **+** to add custom filters to the **Data Selection** table.

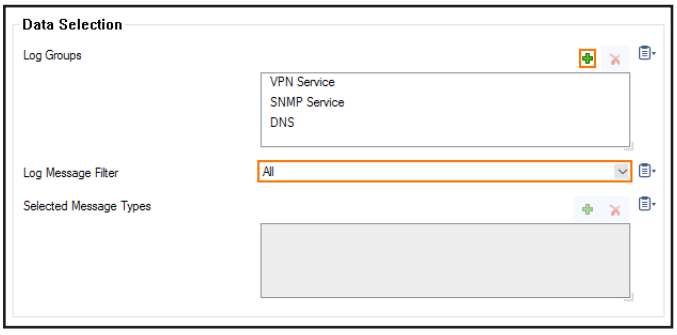
i. In the **Log Groups** table, click **+**.

ii. Select the box level log files, or select **Other** to enter a **user defined log group pattern** to stream log files matching this pattern.

iii. (optional) From the **Log Level Filter** list, select the message types from the log group that are streamed.

iv. (**Selection** only) In the **Selected Messages Types** table, click **+** to add message types.

v. Click **OK**.

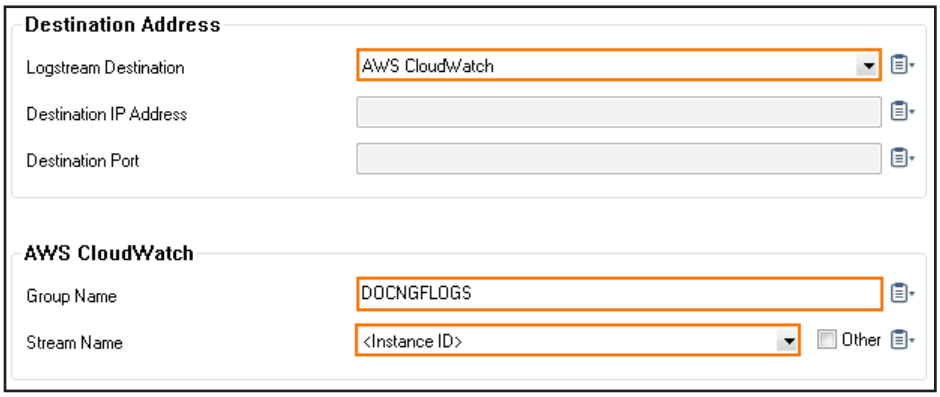


10. Click **Send Changes** and **Activate**.

Step 3. Configure AWS CloudWatch as the Logstream Destination

Configure the firewall to send the syslog stream to AWS CloudWatch. The AWS CloudWatch log group name is created automatically, with one stream per firewall.

1. Go to **CONFIGURATION > Full Configuration > Box > Infrastructure Services > Syslog Streaming**.
2. In the left menu, select **Logstream Destinations**.
3. Click **Lock**.
4. In the **Destinations** table, click **+** to add a new filter. The **Destinations** window opens.
5. Enter a **Name**.
6. Click **OK**.
7. From the **Logstream Destination** list, select **AWS CloudWatch**.
8. In the **AWS CloudWatch** section, enter the name of the AWS CloudWatch log **Group Name**.
9. (optional) Select the **Stream Name** from the drop-down list, or select **Other** and enter the stream name. The stream name must be unique in the AWS CloudWatch group.



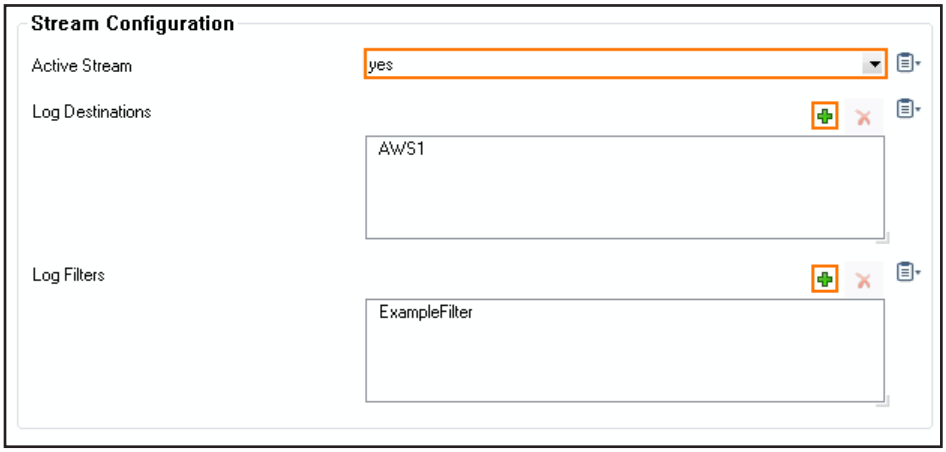
10. Click **OK**.
11. Click **Send Changes** and **Activate**.

Step 4. Configure the Logdata Streams to AWS CloudWatch

Combine the logdata filters and logstream destination to a logdata stream.

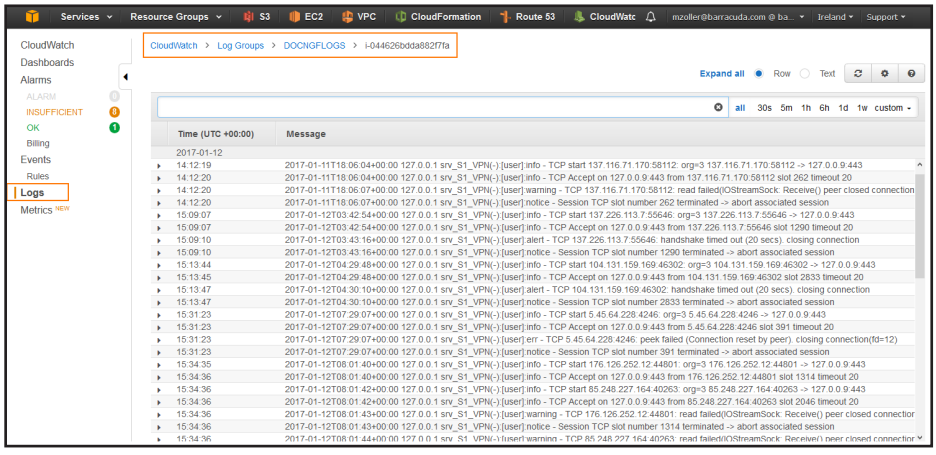
1. Go to **CONFIGURATION > Full Configuration > Box > Infrastructure Services > Syslog Streaming**.
2. In the left menu, select **Logdata Streams**.
3. Click **Lock**.
4. In the **Streams** table, click **+** to add a new syslog stream. The **Streams** window opens.
5. Enter a **Name**.
6. Click **OK**.
7. Set **Active Stream** to **yes**.

8. In the **Log Destinations** table, click **+** and select the logstream destination configured in step 3.
9. In the **Log Filters** table, click **+** and select the logdata filter configured in step 2.



10. Click **OK**.
11. Click **Send Changes** and **Activate**.

All logs covered by the logdata filter are now streamed to AWS CloudWatch. It might take up to 30 minutes for logs to be displayed in the console.



3.3 How to Restore a Configuration on a PAYG Firewall in the Public Cloud

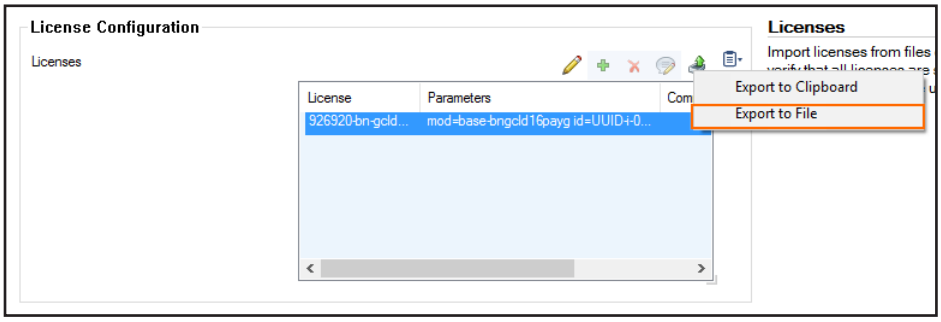
PAYG licenses are generated only once during the first boot. To avoid the PAYG license to be overwritten save the license before restoring the firewall configuration from a PAR file. Then import the license before activating the configuration.

3.3.1 Before You Begin

You must have a working PAR file of the previous configuration. For information on how to back up and restore configurations, see Backups and Recovery.

Step 1. Save the PAYG Firewall License

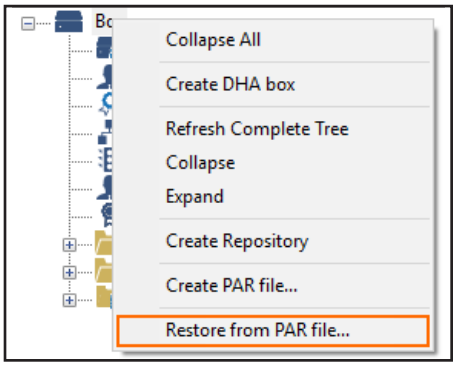
1. Go to **CONFIGURATION > Configuration Tree > Box > Box Licenses**.
2. Click **Lock**.
3. Click on the PAYG license in the **Licenses** list.
4. Click the export icon, and select **Export to File**.



5. Enter a name for the license, and save the .lic file.

Step 2. Restore the Configuration from the PAR File

1. Go to **CONFIGURATION > Configuration Tree**.
2. Right-click **Box** and select **Restore from PAR file**.



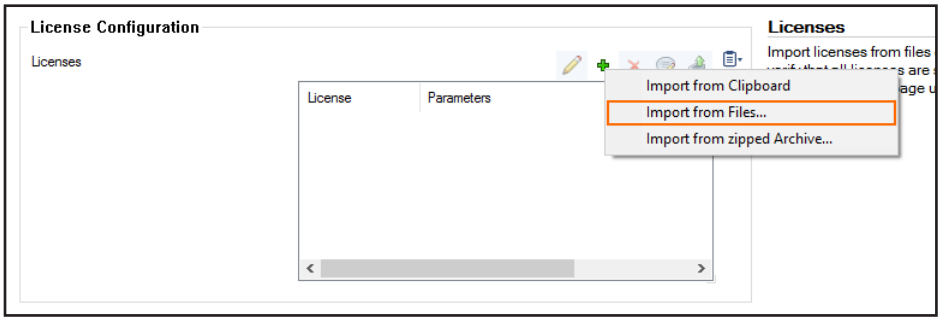
3. Click **OK**.
4. Select the PAR file with the previously configured settings, and click **Open**.

Do not click **Activate**.

Step 3. Restore the License

Remove the license from the box license configuration, and replace it with the PAYG license saved in Step 1.

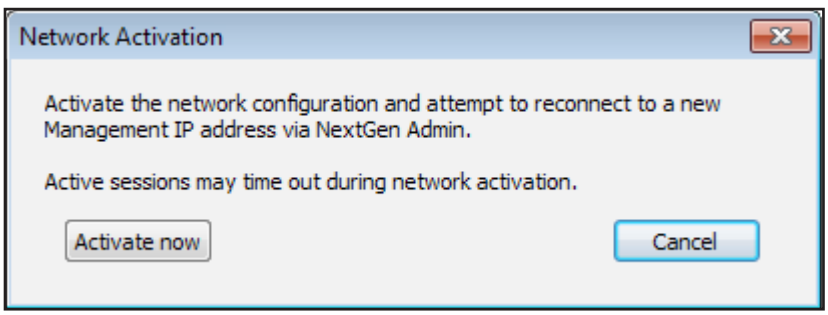
1. Go to **CONFIGURATION > Configuration Tree > Box > Box Licenses**.
2. Click **Lock**.
3. Click the PAYG license in the **Licenses** list.
4. Delete the license.
5. Click the **+** icon, and select **Import from Files**.



6. Select the license file created in Step 1.
7. Click **Open**.
8. Click **Send Changes** and **Activate**.

Step 4. Activate the Network Configuration

1. Go to **CONTROL > Box**.
2. In the left menu, expand **Network** and click **Activate new network configuration**.
3. Click **Activate now**. The **Activation Succeeded** message is displayed after the network configuration has been activated.



3.4 How to Add AWS Elastic Network Interfaces to a Firewall Instance

To make traffic between subnets visible in the firewall, you must add one network interface per subnet. The number of network interfaces you can add to your instance is limited by the instance type. Firewall instances with multiple network interfaces cannot be deployed in a high availability configuration.

3.4.1 AWS Reference Architectures

This article is used in the following AWS reference architectures:

2.5 Segmentation Firewall for Single AZ VPCs(page 71)

3.4.2 Before You Begin

Deploy a firewall instance in the public subnet of the VPC. For more information, see

3.11 How to Deploy an F-Series Firewall in AWS via Web Portal(page 143)

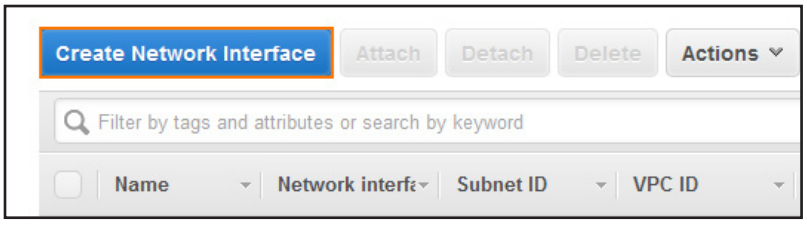
Verify that the Elastic IP address is associated with the elastic network interface (ENI) of the firewall instance and not with the instance itself.

Stop the firewall instance. Additional network interfaces cannot be attached to a running system.

Step 1. Add an Elastic Network Interface

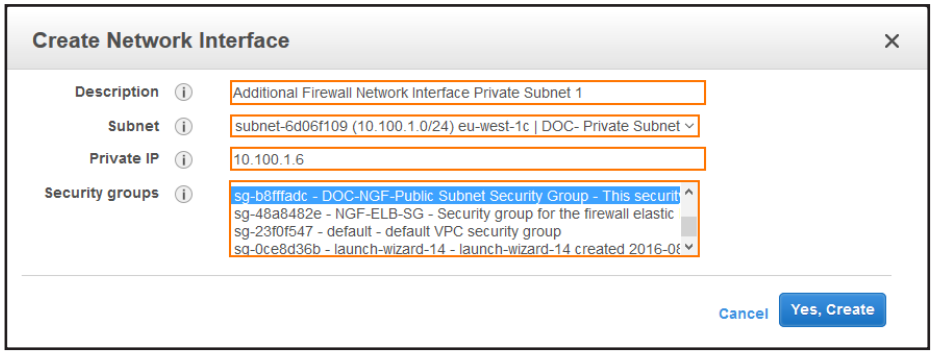
Create an elastic network interface. This interface will then be attached to the instance later.

1. Log into the AWS console.
2. Click **Services** and select **EC2**.
3. In the **Network & Services** section of the left menu, click **Network Interfaces**.
4. Click **Create Network Interface**. The **Create Network Interface** popover opens.



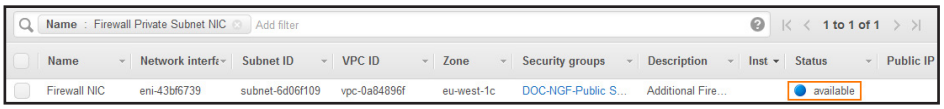
5. Configure the network interface:
   - **Description** – Enter a description for the network interface.
   - **Subnet** – Select the private subnet in the VPC for the network interface. The subnet must be in the same Availability Zone as the firewall instance.
   - **Private IP** – Enter a free IP address in the subnet. The first three IP addresses in the subnet are reserved by AWS.

- **Security groups** – Select the security group assigned to the firewall instance.



6. Click **Yes, Create**.

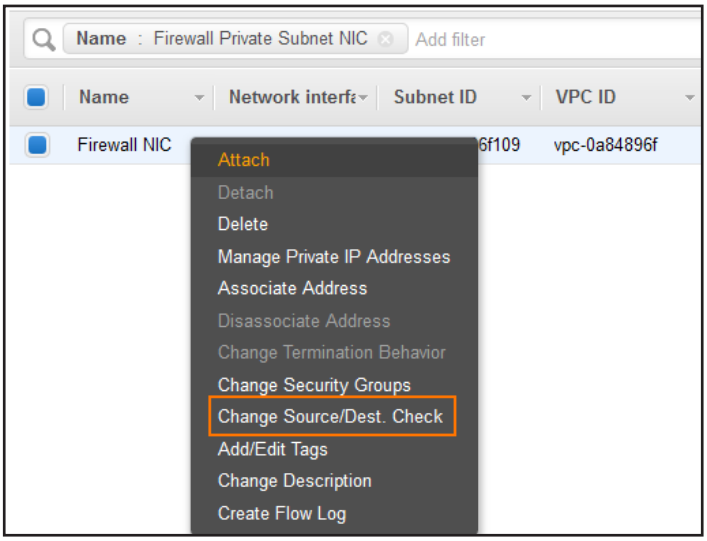
The elastic network interface is now listed with the **Status** column showing **Available**.



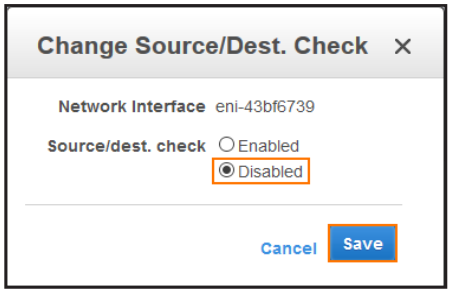
Step 2. Disable Source/Destination Check

To be able to perform NAT operations, the source/destination check must be disabled.

1. Log into the AWS console.
2. Click **Services** and select **EC2**.
3. In the **Network & Services** section of the left menu, click **Network Interfaces**.
4. Right-click on the network interface created in step 1 and click **Change Source/Dest. Check**. The **Change Source/Dest. Check** popover opens.



5. Select **Disabled**.
6. Click **Save**.

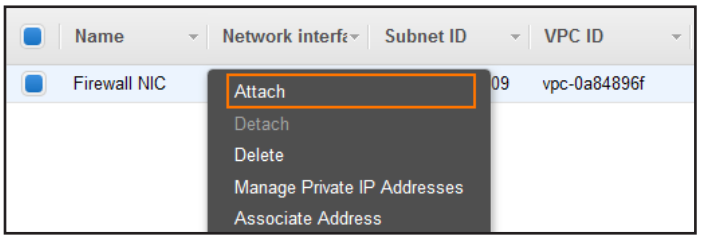


The network interface is now able to handle traffic with destination IP addresses that do not match its own private IP address.

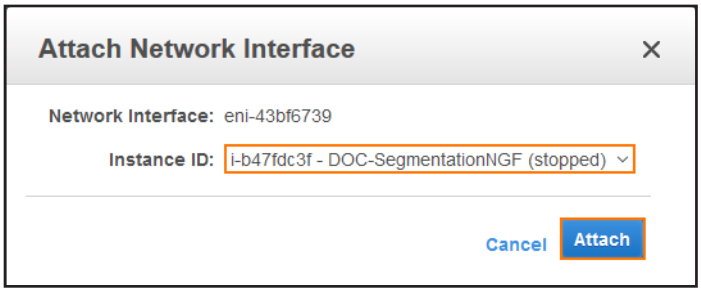
Step 3. Attach the Network Instance to the Firewall Instance

Verify that the firewall instance is shut down, and then add the network interface to the instance.

1. Log into the AWS console.
2. Click **Services** and select **EC2**.
3. In the **Network & Services** section of the left menu, click **Network Interfaces**.
4. Right-click on the network interface created in step 1 and click **Attach**. The **Attach Network Interface** popover opens.

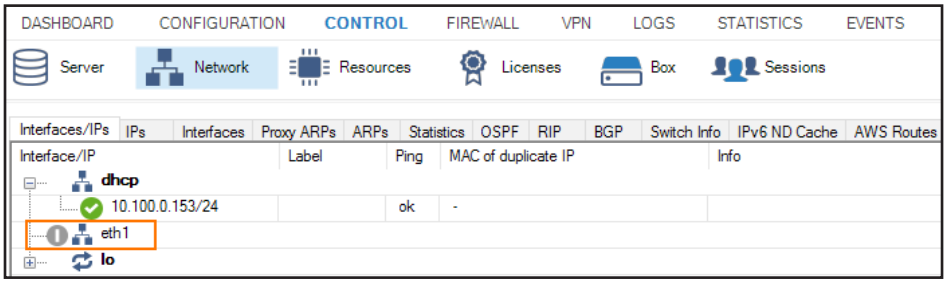


5. In the **Instance ID** list, select the firewall instance.
6. Click **Attach**.



Step 4. Start the Firewall Instance

1. Log into the AWS console.
2. Click **Services** and select **EC2**.
3. In the **Instances** section of the left menu, click **Instances**.
4. Right-click the firewall instance, select **Instance State**, and click **Start**. Wait for the firewall instance to start.
5. Log into the firewall.
6. Go to **CONTROL > Networking**.
7. Verify that the network interface you attached in step 4 is listed.

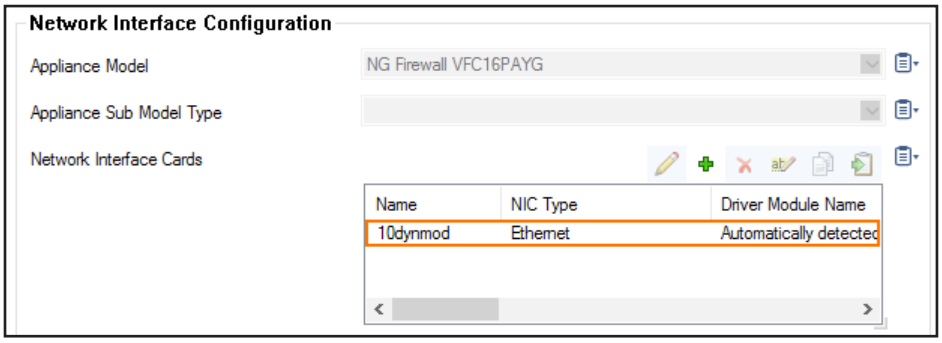


Step 5. Add the Network Interface in the Firewall Configuration

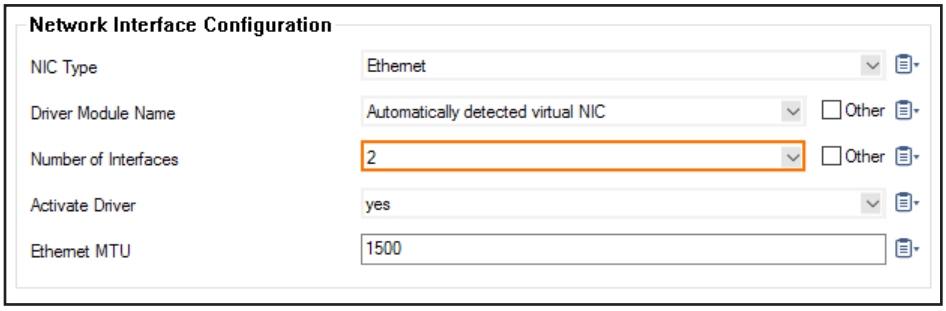
The network interface must be added and configured in the firewall configuration.

Step 5.1 Add the Network Interface

1. Log into the firewall.
2. Go to **CONFIGURATION > Configuration Tree > Box > Network**.
3. Click **Lock**.
4. In the left menu, click **Interfaces**.
5. In the **Network Interface Cards** table, double-click the **10dynmod** entry. The **Network Interface Cards: 10dynmod** window opens.



6. From the **Number of Interfaces**, select the number of network interfaces attached to the firewall instance.
7. Click **OK**.

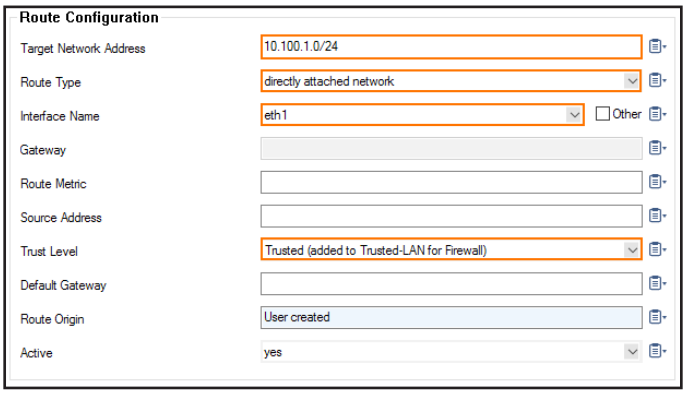


8. Click **Send Changes** and **Activate**.

Step 5.2 Add a Direct Attached Route for the Network Interface

Add the subnet the network interface is in as a direct attached route.

1. Go to **CONFIGURATION > Configuration Tree > Box > Network**.
2. Click **Lock**.
3. In the left menu, click **Routing**.
4. Click **+** in the **IPv4 Routing Table** to add an attached route.
   - **Target Network Address** – Enter the network of the subnet in CIDR format.
   - **Route Type** – Select **direct attached network**.
   - **Interface Name** – Select the interface used to connect to the network. e..g, **eth1**
   - **Trust Level** – Select **Trusted**.



5. Click **OK**.
6. Click **Send Changes** and **Activate**.

Step 5.3 Activate the Network Configuration

1. Go to **CONTROL > Box**.
2. In the **Network** section of the left menu, click **Activate new network configuration**. The **Network Activation** window opens.
3. Click **Failsafe**.

The route is now pending in **CONTROL > Network.**

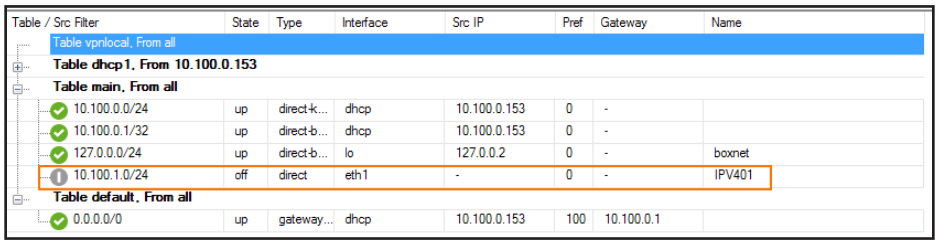
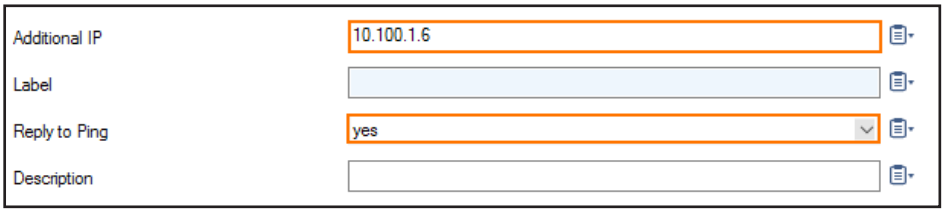


Table / Src Filter	State	Type	Interface	Src IP	Pref	Gateway	Name
Table vpnlocal, From all							
Table dhcp1, From 10.100.0.153							
Table main, From all							
10.100.0.0/24	up	direct-k...	dhcp	10.100.0.153	0	-	
10.100.0.1/32	up	direct-b...	dhcp	10.100.0.153	0	-	
127.0.0.0/24	up	direct-b...	lo	127.0.0.2	0	-	boxnet
10.100.1.0/24	off	direct	eth1	-	0	-	IPV401
Table default, From all							
0.0.0.0/0	up	gateway...	dhcp	10.100.0.153	100	10.100.0.1	

Step 5.4 Add a Virtual Server IP

Add the private IP address assigned to the network interface as a virtual server IP address.

1. Go to **CONFIGURATION > Configuration Tree > Box > Virtual Servers > your virtual server > Server Properties**.
2. Click **Lock**.
3. Click **+** in the **Additional IP** table. The **Additional IP** window opens.
4. Configure the additional virtual server IP:
   - **Additional IP** – Enter the private IP address configured for the network interface in step 1.
   - **Reply to Ping** – Select **yes**.



5. Click **OK**.
6. Click **Send Changes** and **Activate**.

The route is now active and the virtual server IP reachable for all clients in the subnet.

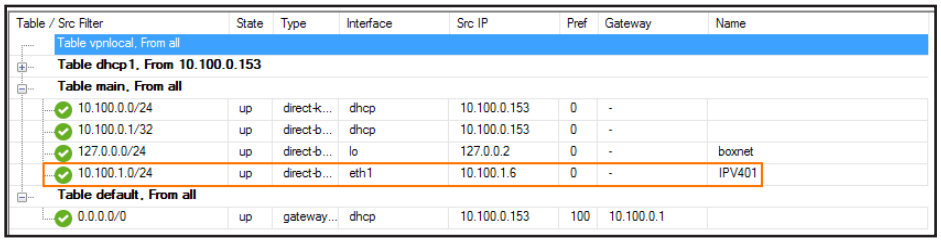


Table / Src Filter	State	Type	Interface	Src IP	Pref	Gateway	Name
Table vpnlocal, From all							
Table dhcp1, From 10.100.0.153							
Table main, From all							
10.100.0.0/24	up	direct-k...	dhcp	10.100.0.153	0	-	
10.100.0.1/32	up	direct-b...	dhcp	10.100.0.153	0	-	
127.0.0.0/24	up	direct-b...	lo	127.0.0.2	0	-	boxnet
10.100.1.0/24	up	direct-b...	eth1	10.100.1.6	0	-	IPV401
Table default, From all							
0.0.0.0/0	up	gateway...	dhcp	10.100.0.153	100	10.100.0.1	

3.4.3 Next Steps

- Configure the AWS route table to use the network interface as the default route for all clients in this subnet.

To send traffic between two subnets over the firewall, the firewall must have a network interface in each subnet. A gateway route must be added on the clients with the private IP address of the firewall used as the gateway. For more information, see

2.5 Segmentation Firewall for Single AZ VPCs(page 71)

3.5 How to Configure AWS Route Tables for Firewalls with Multiple Network Interfaces

For instances in a private subnet to send traffic through the network interface of the firewall in this subnet, you must create an AWS route table for each private subnet. Add a default route using the elastic network interface as the target device. Traffic leaving the VPC is now sent via the network interface of the firewall in the same subnet. However, internal VPC traffic is not sent through the firewall. For more information, see 2.5 Segmentation Firewall for Single AZ VPCs(page 71)

3.5.1 AWS Reference Architectures

This article is used in the following AWS reference architectures:

2.5 Segmentation Firewall for Single AZ VPCs(page 71)

3.5.2 Before You Begin

- Deploy a firewall instance in the public subnet of the VPC.
- The public and private subnets must be in the same Availability Zone.

Add a network interface in the private subnet to the firewall instance. For more information, see

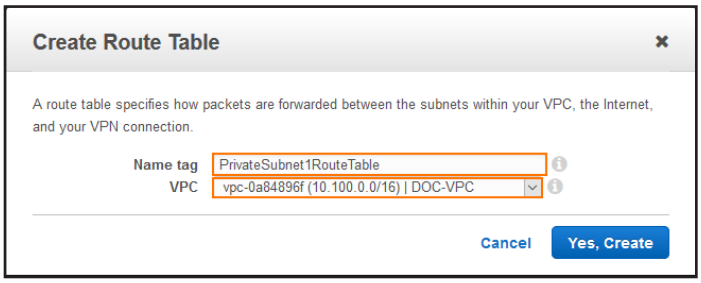
3.4 How to Add AWS Elastic Network Interfaces to a Firewall Instance(page 95)

Step 1. Create an AWS Route Table

Create an AWS route table for each private subnet.

1. Log into the AWS console.
2. Click **Services** and select **VPC**.
3. In the **Virtual Private Cloud** section of the left menu, click **Route Tables**.
4. Click **Create Route Table**. The **Create Route Table** popover opens.

5. Configure the route table:
   - **Name tag** – Enter the name for the route table.
   - **VPC** – Select the VPC from the list.

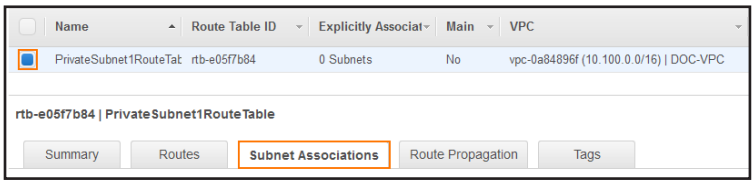


6. Click **Yes, Create**.

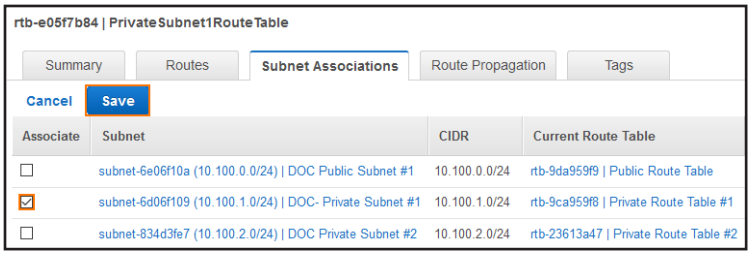
Step 2. Associate the Private Subnet with the Route Table

If the subnet is not explicitly associated with a route table, the main route table for the VPC is used.

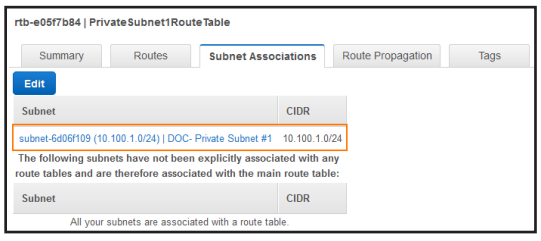
1. Log into the AWS console.
2. Click **Services** and select **VPC**.
3. In the **Virtual Private Cloud** section of the left menu, click **Route Tables**.
4. Select the route table created in step 1.
5. In the lower half of the screen, click on the **Subnet Associations** tab.



6. Click **Edit**.
7. Select the subnet you want to associate with this route table.
8. Click **Save**.

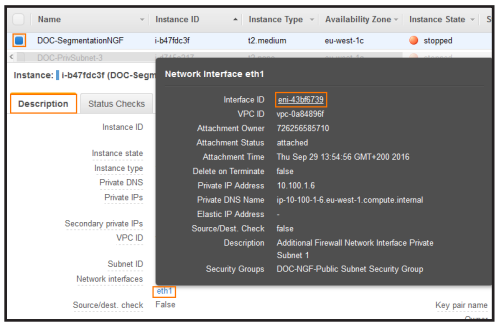


The private subnet is now associated with the route table.

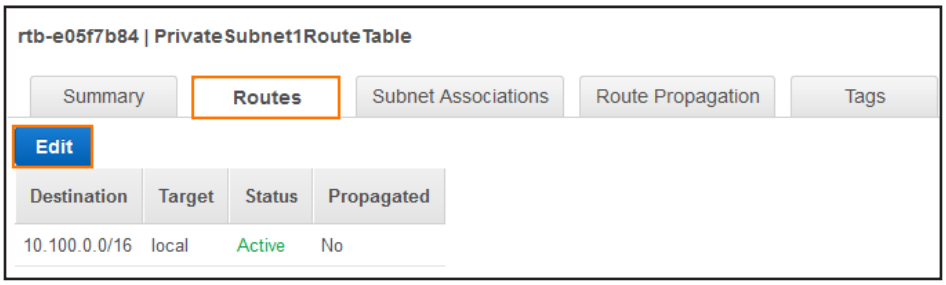


Step 3. Add a Default Route with the Network Interface of the Firewall as the Target

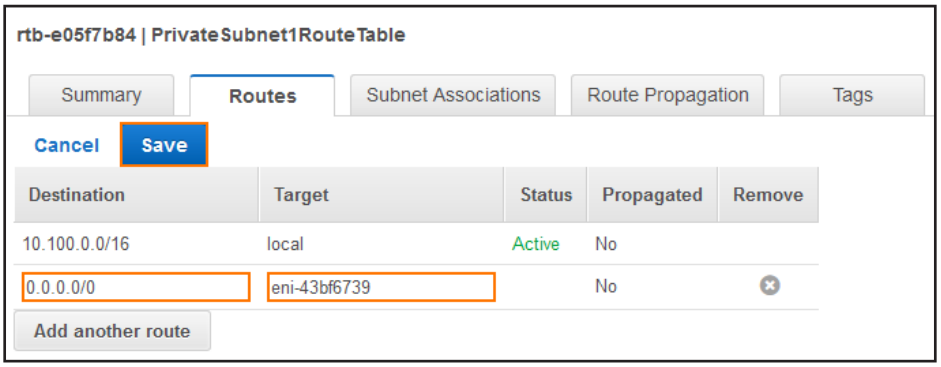
Locate the elastic network interface identifier (eni-12345678) for the network interface in this subnet. Click on the network interface in the **Description** tab of the firewall instance to retrieve the ID.



1. Log into the AWS console.
2. Click **Services** and select **VPC**.
3. In the **Virtual Private Cloud** section of the left menu, click **Route Tables**.
4. Select the route table created in step 1.
5. In the lower half of the screen, click on the **Routes** tab.
6. Click **Edit**.

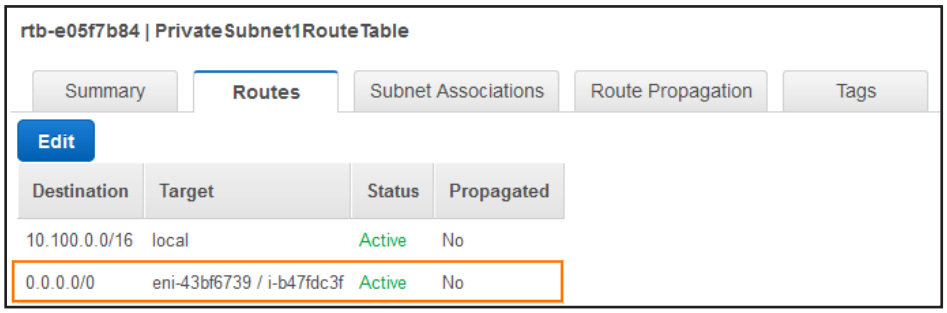


7. Click **Add another route**.
8. Configure the route:
   - **Destination** – Enter 0.0.0.0/0.
   - **Target** – Enter the ID for the firewall network interface located in this subnet.



9. Click **Save**.

All traffic leaving the VPC from the associated subnet is now sent through the firewall. The status of the route must be **Active**.




```
Shared parameters
         "CCIPAddress": {
             "Description": "IP Address or hostname of the Control
Center",
             "Type": "String",
             "Default": "127.0.0.1"
         },
         "Cluster": {
             "Description": "Case sensitive Control Center cluster name",
             "Type": "String"
         },
         "Range": {
             "Description": "Control Center range number",
             "Type": "String"
         },
         "FirewallName": {
             "Description": "Case sensitive name of the Firewall on the
Control Center",
             "Type": "String"
         }
Additional required parameters for Control Center authentication:
         "CCUser": {
             "Description": "CC admin username",
             "Type": "String",
             "Default": ""
         },
         "CCPassword": {
             "Description": "CC admin user password",
             "Type": "String",
             "Default": "",
             "NoEcho": "true"
         },
Additional required parameters for shared key authentication:
          "CCSharedKey": {
             "Description": "shared key to retrieve PAR file",
             "Type": "String",
             "Default": "",
             "NoEcho": "true"
         },
```

3.6 How to Modify CloudFormation Templates to Retrieve the PAR File from a Control Center

If you are using the NextGen Control Center, you can modify your firewall's AWS CloudFormation template to retrieve the PAR file for the new F-Series Firewall Instance from the Control Center. The script authenticates either with CC admin credentials or a shared secret. Licenses that are already installed on PAYG firewall Instances are pushed to the Control Center before retrieving the PAR file. Firewalls using the BYOL images use the licenses configured on the Control Center.

'getpar' Command Line Parameters Usage

- **-a|--address <address>** – Control Center IP address.
- **-u|--username <username>** – CC admin user used to connect to the Control Center
- **-c|--cluster <cluster>** – Cluster name
- **-r|--range <range>** – Range number
- **-b|--boxname <boxname>** – Firewall name.
- **-d|--destination <dest>** – Destination directory and filename for the par file. e.g., /opt/phion/update/box.par
- **-s|–spoe** – Use Single Point of Entry to connect to the Control Center.
- **-l|--pushlic auto|always|never** – Configures if the licenses should be pushed to the Control Center before retrieving the PAR file.

3.6.1 Before You Begin

- Create an AWS CloudFormation template to deploy your F-Series Firewall.

Step 1. Create the Firewall Configuration in the Control Center

Create the F-Series Firewall configuration in the Control Center.

For more information, see How to Add a new F-Series Firewall to the Control Center.

Step 2. Configure Authentication

The newly deployed firewall can authenticate either through a CC Admin account or with a shared key. The shared key is defined on a per-firewall level.

CC Admin Authentication

Create a CC admin and assign it an Administrative role with the following permissions:

- **CC Configuration Permission** – Click the **Get PAR File** check box.

For more information, see Control Center Admins and How to Configure Administrative Roles.

Shared Key Authentication

1. Log in to the Control Center.
2. Go to `your firewall` > **Box Properties**.
3. In the left menu, click **Operational**.
4. In the left menu, expand **Configuration Mode** and click **Switch to Advanced View**.
5. Click **Lock**.
6. Enter the **PAR File Retrieval Shared Key**.
7. Click **Send Changes** and **Activate**.

Step 3. Add the Parameters to the Template

You must add the parameters you need to the **Parameters** section of the template.

1. Add the following mandatory parameters to the **parameter** section of the template:
   - **CCIPAddress** – The IP address of your Control Center if it is directly reachable, or the IP address of the border firewall forwarding the traffic to the Control Center.
   - **Range** – The range number.

Cluster – The cluster name.

- **FirewallName** – The name of the Firewall.
- Add the authentication parameters:
  - **For Control Center Admins:**
    - **CCUser** – The CC admin.
    - **CCPassword** – The password for the CC admin.
  - **For Shared Key Authentication**:
    - **CCSharedKey** – The shared key used to authenticate to the Control Center.

Step 4. Modify the Template to Retrieve the PAR File

Add a script to the userData element of the template. Use the parameters defined above.

1. Locate the **Gateway** section.

Add the getparfile script to the **UserData** parameter with the desired authentication method:

Control Center Admin:

```
"Gateway": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "ImageId": "ami-XXXXXXXX",
    "InstanceType": { "Ref": "InstanceType" },
    "KeyName": { "Ref": "KeyName" },
    "SecurityGroups" : [{ "Ref" : "NGSecurityGroup" }],
    "UserData": {
      "Fn::Base64": {
        "Fn::Join": [
          "", [
            "#!/bin/bash\n\n",
            "echo \"userdata\" >> /tmp/userdata.txt\n",
            "/opt/aws/bin/cfn-init -v --region ",
            { "Ref": "AWS::Region" },
            " -s ",
            { "Ref": "AWS::StackName" },
            " -r ",
            "Gateway\n",
            "/opt/aws/bin/cfn-hup-config -r",
            { "Ref": "AWS::Region" },
            " -s ",
            { "Ref": "AWS::StackName" },
            "\n"
          ]
        ]
      }
    }
  },
  "Metadata": {
    "AWS::CloudFormation::Init": {
      "configSets": {
        "default": [ "getparfile" ]
      },
      "getparfile": {
        "files": {
          "/etc/cfn/hooks.d/test.conf": {
            "content": { "Fn::Join": [ "", [
              "[testhook]\n",
              "triggers=post.add\n",
              "path=Resources.Gateway\n",
              "action=\"echo blabla > /tmp/hook.log\"\n",
              "runas=root"
            ]]},
            "mode": "000644",
            "owner": "root",
            "group": "root"
          }
        },
        "commands": {
          "retrievepar": {
            "command": {
              "Fn::Join": ["", [
                "echo \"",
                { "Ref": "CCPassword" },
                "\" | /opt/phion/bin/getpar -a ",
                { "Ref": "CCIPAddress" },
                " -u ",
                { "Ref": "CCUser" },
                " -c ",
```

```
                        { "Ref": "Cluster" },
                        " -r ",
                        { "Ref": "Range" },
                        " -b ",
                        { "Ref": "FirewallName" },
                        " -d /opt/phion/update/box.par -s",
                        " --verbosity 10 ",
                        " >> /tmp/getpar.log"
                      ]]
                    }
                  }
                }
              }
            }
```

```
          }
        }
```

Shared key authentication:

```
      "Gateway": {
        "Type": "AWS::EC2::Instance",
        "Properties": {
          "ImageId": "ami-XXXXXXXX",
          "InstanceType": { "Ref": "InstanceType" },
          "KeyName": { "Ref": "KeyName" },
          "SecurityGroups" : [{ "Ref" : "NGSecurityGroup" }],
          "UserData": {
            "Fn::Base64": {
              "Fn::Join": [
                "", [
                  "#!/bin/bash\n\n",
                  "echo \"userdata\" >> /tmp/userdata.txt\n",
                  "/opt/aws/bin/cfn-init -v --region ",
                  { "Ref": "AWS::Region" },
                  " -s ",
                  { "Ref": "AWS::StackName" },
                  " -r ",
                  "Gateway\n",
                  "/opt/aws/bin/cfn-hup-config -r",
                  { "Ref": "AWS::Region" },
                  " -s ",
                  { "Ref": "AWS::StackName" },
                  "\n"
                ]
              ]
            }
          }
        },
        "Metadata": {
          "AWS::CloudFormation::Init": {
            "configSets": {
              "default": [ "getparfile" ]
            },
            "getparfile": {
              "files": {
                "/etc/cfn/hooks.d/test.conf": {
                  "content": { "Fn::Join": [ "", [
                    "[testhook]\n",
                    "triggers=post.add\n",
                    "path=Resources.Gateway\n",
```

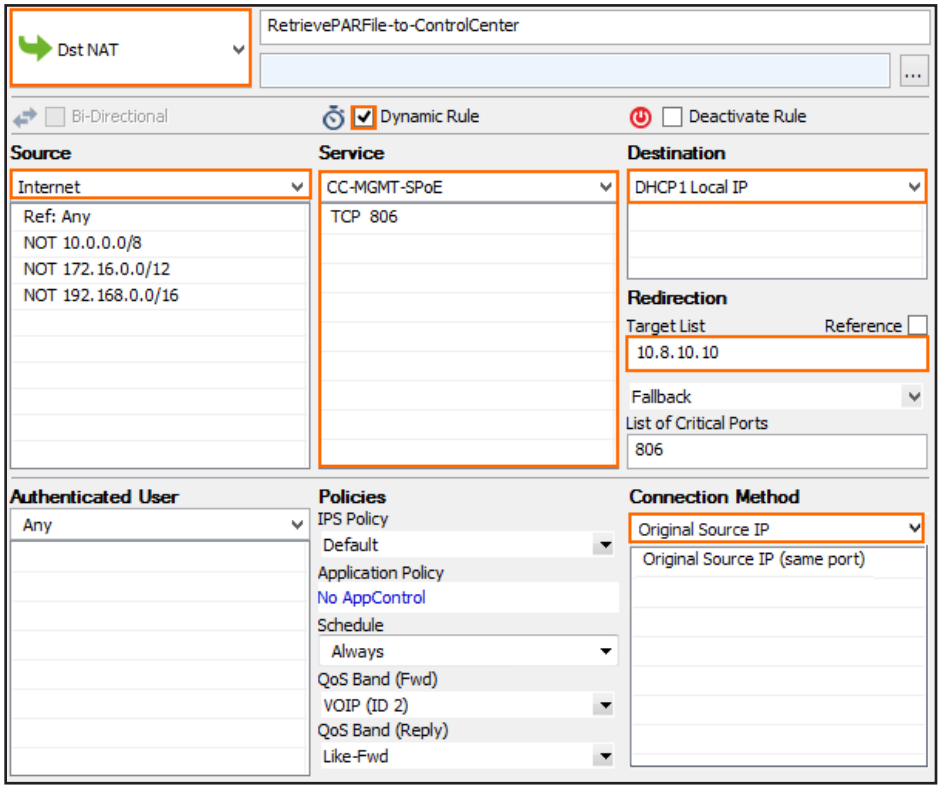
```
                "action=\"echo blabla > /tmp/hook.log\"\n",
                "runas=root"
              ]]},
              "mode": "000644",
              "owner": "root",
              "group": "root"
            }
          },
          "commands": {
            "retrievepar": {
              "command": {
                "Fn::Join": ["", [
                  "echo \"",
                  { "Ref": "CCSharedKey" },
                  "\" | /opt/phion/bin/getpar -a ",
                  { "Ref": "CCIPAddress" },
                  " -c ",
                  { "Ref": "Cluster" },
                  " -r ",
                  { "Ref": "Range" },
                  " -b ",
                  { "Ref": "FirewallName" },
                  " -d /opt/phion/update/box.par -s",
                  " --verbosity 10 ",
                  " >> /tmp/getpar.log"
                ]]
              }
            }
          }
        }
      }
    }
  }

}
```

2. Save the template.

Step 5. (optional) Allow Access to the Control Center

If the firewall VM cannot directly reach the Control Center, you must create a dynamic access rule on the border firewall. Using dynamic rules allows you to enable access only when deploying a new firewall. If SPoE is used, you must open port TCP 806.

- **Action** – Select **Dst NAT**.
- **Source** – If known, enter the public IP address of the Firewall, or select **Internet**.
- **Service** – Create and select a service object for TCP 806. For more information, see Service Objects.
- **Destination** – Enter the **Point of Entry** IP address of the border firewall.
- **Redirect to** – Enter the IP address of the Control Center.
- **Connection Method** – Select **Original Source IP**.



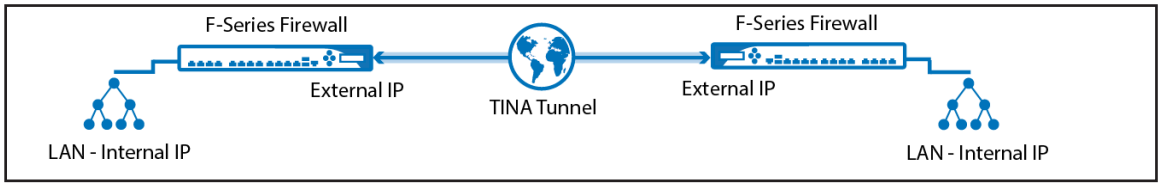
3.6.2 Next Steps

Deploy the firewall via the AWS CloudFormation template.

For more information, see 3.10 How to Deploy an F-Series Firewall in AWS via CloudFormation Template(page 139)

3.7 How to Create a TINA VPN Tunnel between F-Series Firewalls

As the TINA protocol offers significant advantages over IPsec, it is the main protocol that is used for VPN connections between F-Series Firewalls. Many of the advanced VPN features, such as Traffic Intelligence or WAN Optimization, are only supported for TINA site-to-site tunnels.



You must complete this configuration on both the local and the remote Barracuda NextGen Firewall F-Series by using the respective values below:

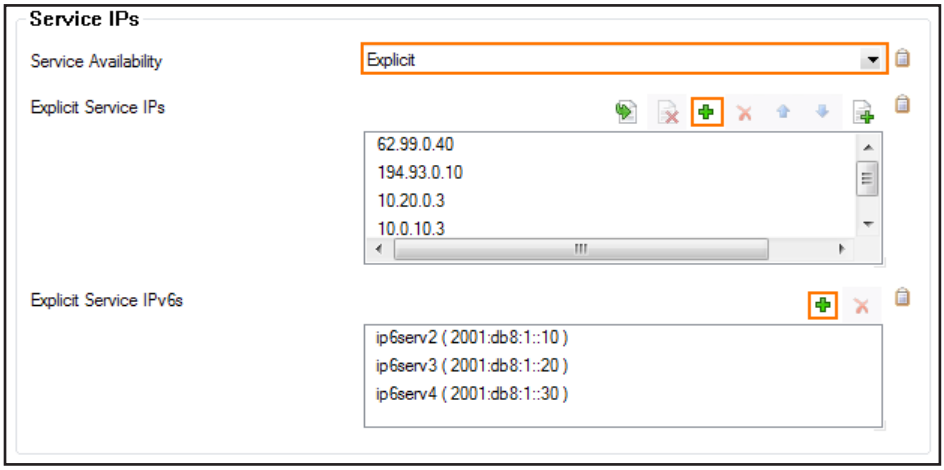
Example values for the local firewall	Example values for the remote firewall	
VPN local networks	10.0.10.0/25	10.0.81.0/24
VPN remote networks	10.0.81.0/24	10.0.10.0/25
External IP address (listener VPN service)	62.99.0.40	212.86.0.10

The following sections use the default transport, encryption, and authentication settings. For more detailed information, see TINA Tunnel Settings.

Step 1. Configure the VPN Service Listeners

Configure the IPv4 and IPv6 listener addresses for the VPN service.

1. Go to **CONFIGURATION > Configuration Tree> Box > Virtual Server > your virtual server > Assigned Services > VPN > Service Properties**.
2. Click **Lock**.
3. From the **Service Availability** list, select the source for the IPv4 listeners:
   - **First+Second-IP** – The VPN service listens on the first and second virtual server IPv4 address.
   - **First-IP** – The VPN service listens on the first virtual server IPv4 address.
   - **Second-IP** – The VPN service listens on the second virtual server IPv4 address.
   - **Explicit** – For each IP address, click **+** and enter the IPv4 addresses in the **Explicit Service IPs** list.
4. Click **+** to add an entry to the **Explicit IPv6 Service IPs**.
5. Select an IPv6 listener from the list of configured explicit IPv6 virtual server IP addresses.

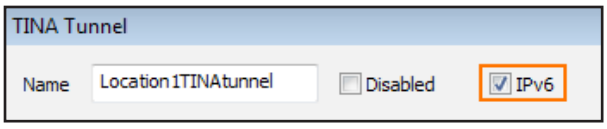


6. Click **Send Changes** and **Activate**.

Step 2. Configure the TINA Tunnel at Location 1

For the firewall at location 1, configure the network settings and export the public key. For more information on specific settings, see TINA Tunnel Settings

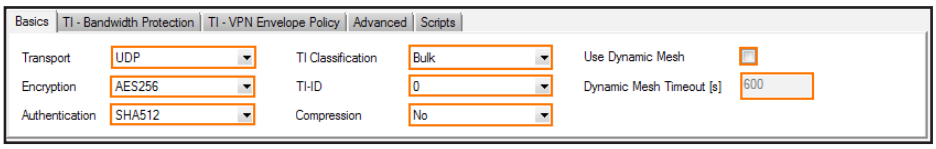
1. Log into the firewall at location 1.
2. Go to **CONFIGURATION > Configuration Tree > Box > Virtual Servers >** `your virtual server` **> Assigned Services > VPN > Site to Site**.
3. Click **Lock**.
4. Click the **TINA Tunnels** tab.
5. Right-click the table, and select **New TINA tunnel**.
6. In the **Name** field, enter the name for the new VPN tunnel.
7. (IPv6 only). Select **IPv6**.



Configure the **Basic** TINA tunnel settings. For more information, see TINA Tunnel Settings.

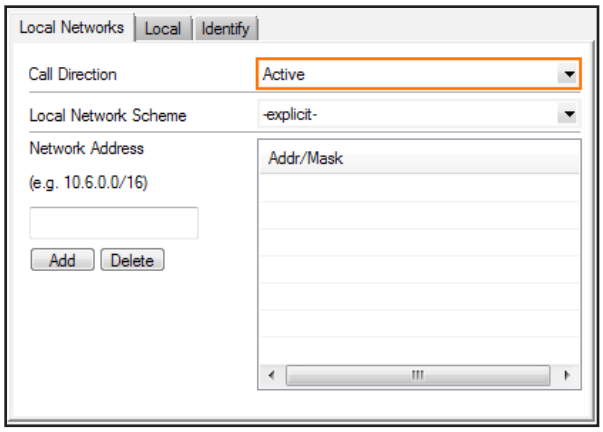
- **Transport** – Select the transport encapsulation: **UDP** (recommended), **TCP**, **TCP&UDP**, **ESP**, or **Routing**.
- **Encryption** – Select the encryption algorithm: **AES**, **AES256**, **3DES**, **CAST**, **Blowfish**, **DES**, or **Null**.
- **Authentication** – Select the hashing algorithm: **MD5**, **SHA**, **SHA256**, **SHA512**, **NOHASH**, **RIPEMD160**, or **GCM**.
- **(optional) TI Classification / TI-ID** – For more information, see Traffic Intelligence.
- **(optional) Compression** – Select **yes** to enable VPN compression. Do not use in combination with WAN Optimization.

- **(optional) Use Dynamic Mesh / Dynamic Mesh Timeout** – For more information, see Dynamic Mesh VPN Networks.

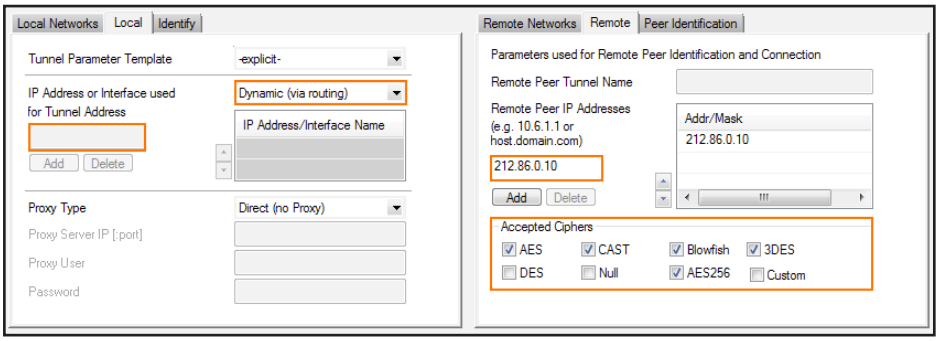


In the **Local Networks** tab, select the **Call Direction**. At least one of the firewalls must be active.

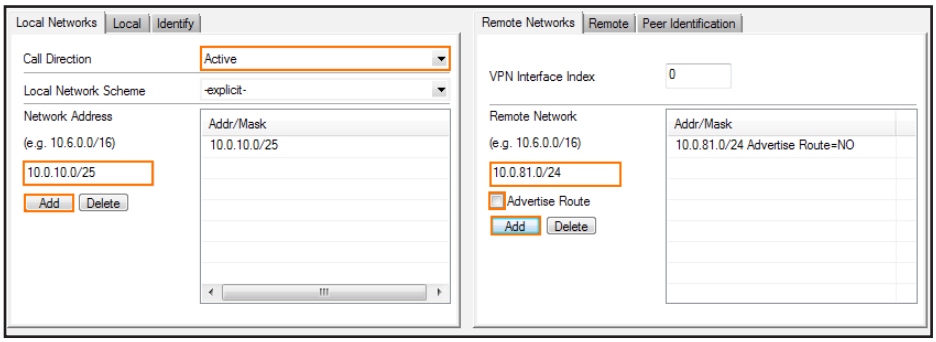
Configure the NextGen Firewall F-Series with a dynamic IP address to be the active peer. If both firewalls use dynamic IP addresses, a DynDNS service must be used. For more information, see How to Configure VPN Access via a Dynamic WAN IP Address.



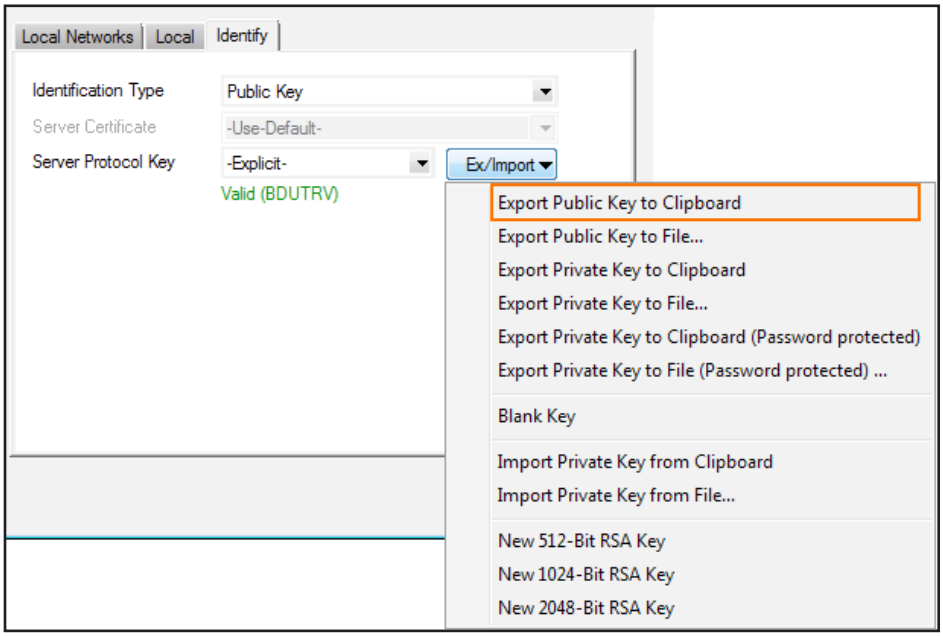
8. Click the **Local** tab, and configure the **IP address or Interface used for Tunnel Address**:
   - **(IPv4 only) First Server IP** – First IP address of the virtual server the VPN service is running on.
   - **(IPv4 only) Second Server IP** – Second IP address of the virtual server the VPN service is running on.
   - **Dynamic (via routing)** – The firewall uses a routing table lookup to determine the IP address.
   - **Explicit List (ordered)** – Enter one or more explicit IP addresses. Multiple IP addresses are tried in the listed order.
   - In the **Remote** tab, enter one or more IPv4 or IPv6 addresses or an FQDN as the **Remote Peer IP Addresses,** and click **Add**



9. In the **Remote** tab, select the **Accepted Ciphers**. To use a cipher, the list must match the **Encryption** settings previously configured.
10. For each local network, enter the **Network Address** in the **Local Networks** tab and click **Add**. e.g., 10.0.10.0/25
11. For each remote network enter the **Network Address** in the **Remote Networks** tab and click **Add**. e.g., 10.0.81.0/24
12. (optional) To propagate the remote VPN network via dynamic routing enable **Advertise Route**.



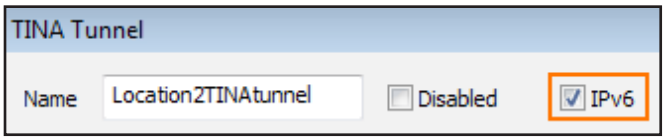
13. Click on the **Identity** tab.
14. From the **Identification Type** list, select **Public Key**.
15. Click **Ex/Import** and select **Export Public Key to Clipboard**.



16. Click **OK**.
17. Click **Send Changes** and **Activate**.

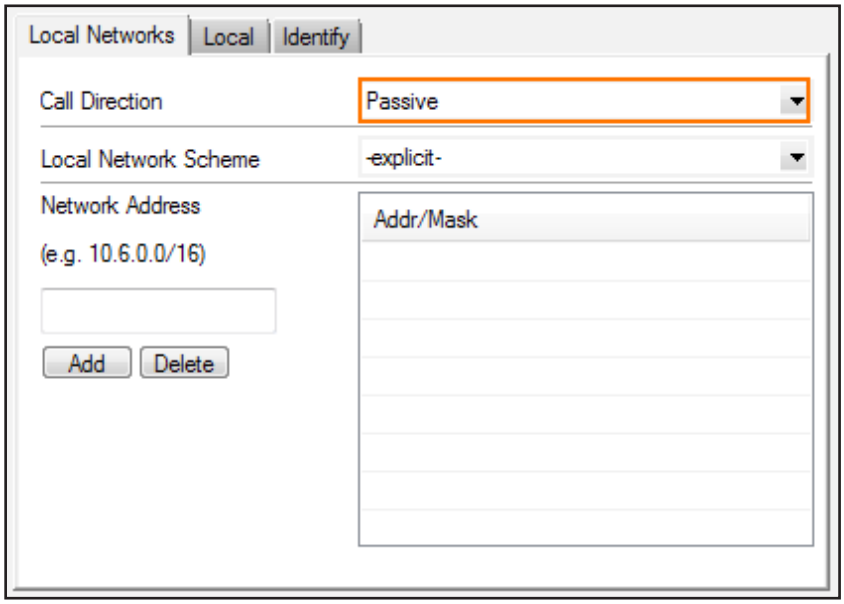
Step 3. Create the TINA Tunnel at Location 2

1. Log into the firewall at location 2.
2. Go to **CONFIGURATION > Configuration Tree > Box > Virtual Servers >** `your virtual server` **> Assigned Services > VPN > Site to Site** .
3. Click **Lock**.
4. Click the **TINA Tunnels** tab.
5. Right-click the table, and select **New TINA tunnel**.
6. In the **Name** field, enter the name for the new VPN tunnel.
7. (IPv6 only) Click the **IPv6** check box.

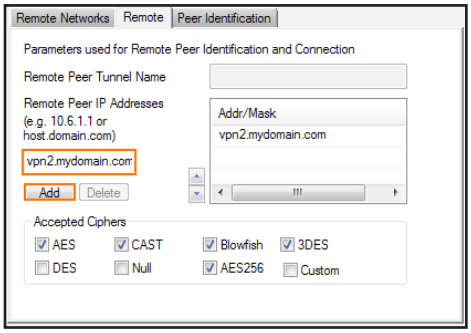


8. Configure the **Basic** TINA tunnel settings to match the settings configured for the Location1

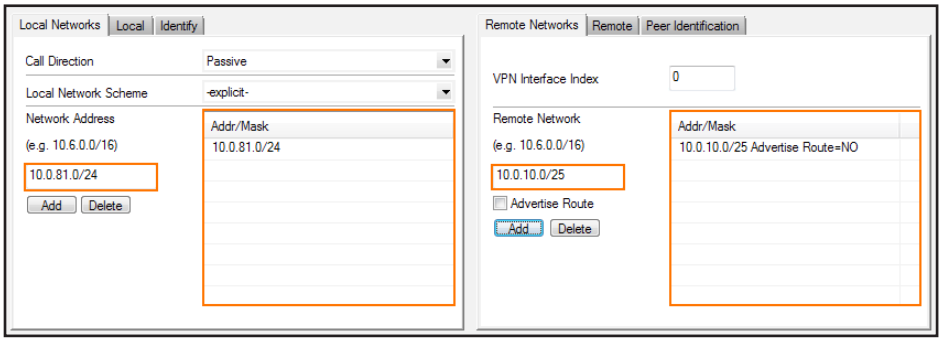
In the **Local Networks** tab, select the **Call Direction**. Make sure that one or both firewalls are set to **active**.



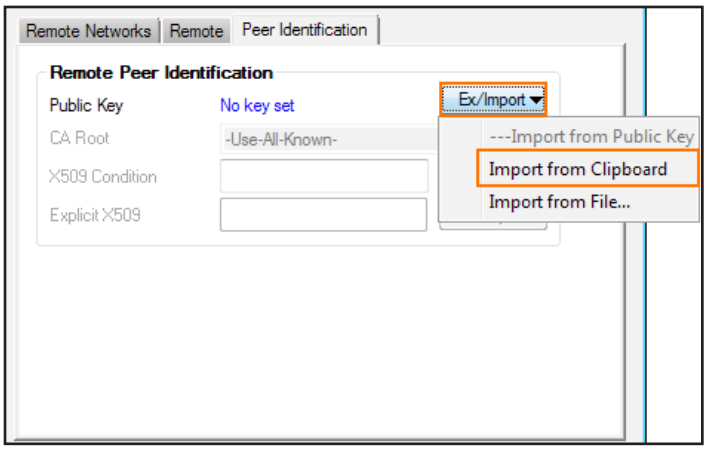
9. Click the **Local** tab, and configure the **IP address or Interface used for Tunnel Address**:
   - **(IPv4 only) First Server IP** – First IP address of the virtual server the VPN service is running on.
   - **(IPv4 only) Second Server IP** – Second IP address of the virtual server the VPN service is running on.
   - **Dynamic (via routing)** – The firewall uses a routing table lookup to determine the IP address.
   - **Explicit List (ordered)** – Enter one or more explicit IP addresses. Multiple IP addresses are tried in the listed order.
10. Click the **Remote** tab, enter one or more IP addresses or a FQDN as the **Remote Peer IP Addresses,** and click **Add**.



11. In the **Remote** tab, select the **Accepted Ciphers**. To use a cipher, the list must match the **Encryption** settings previously configured.
12. For each local network, enter the **Network Address** in the **Local Networks** tab and click **Add**. e.g., 10.0.81.0/24 behind Location 2 NextGen Firewall F-Series.
13. For each remote network, enter the **Network Address** in the **Remote Networks** tab and click **Add**. e.g., 10.0.10.0/25 behind Location1 NextGen Firewall F-Series.



14. Click on the **Peer Identification** tab.
15. Click **Ex/Import** and select **Import from Clipboard**.



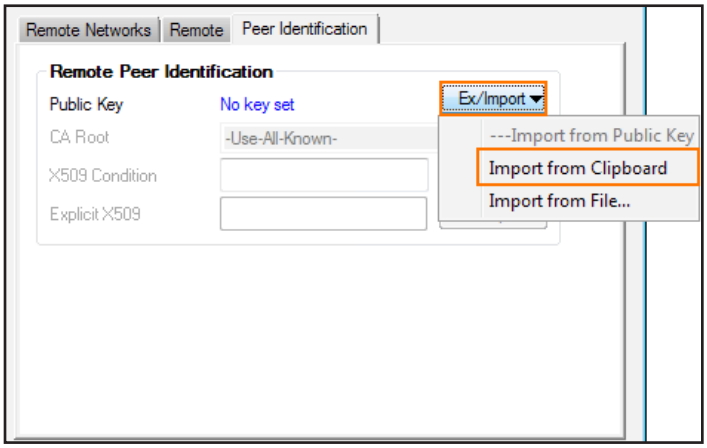
16. Click on the **Identity** tab.
17. From the **Identification Type** list, select **Public Key**.
18. Click **Ex/Import** and select **Export Public Key to Clipboard**.

19. Click **OK**.
20. Click **Send Changes** and **Activate**.

Step 4. Import the Public Key for Location 1

The VPN tunnel is not activated until the public key of location 2 is imported to location 1.

1. Log into the firewall at location 1.
2. Go to **CONFIGURATION > Configuration Tree > Box > Virtual Servers >** `your virtual server` **> Assigned Services > VPN-Service > Site to Site** .
3. Click **Lock**.
4. Open the configuration for the site-to-site tunnel created in Step 1.
5. Click the **Peer Identification** tab.
6. Click **Ex/Import** and select **Import from Clipboard**.



7. Click **OK**.
8. Click **Send Changes** and **Activate**.

After configuring the TINA VPN tunnel on both firewalls, you must also create an access rule on both systems to allow access to the remote networks through the VPN tunnel.

3.7.1 Next Step

Create access rules to allow traffic in and out of your VPN tunnel: How to Create Access Rules for Site-to-Site VPN Access.

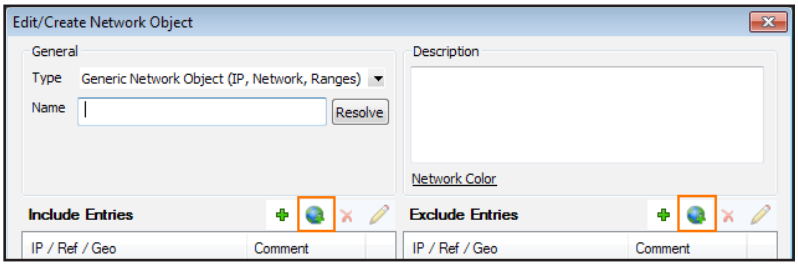
3.8 How to Create a Geo Location based Network Object

The geolocation database included with the F-Series Firewall can match the IP address and network to the country it was issued to. This enables you to create access rules based on the physical location of the source or destination. Lists of countries or regions are combined in a reusable network object. The geolocation database is updated with every firmware release.

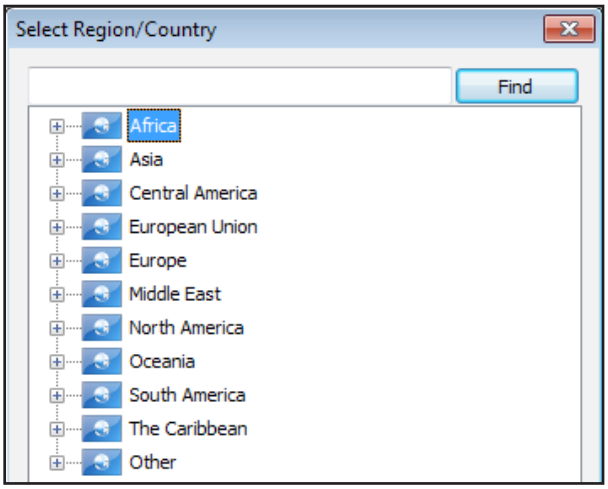
3.8.1 Create a Network Object

Create a network object and include all countries you want to use for your access rule.

1. Go to **CONFIGURATION > Configuration Tree > Box > Virtual Servers >** `your virtual server` **> Assigned Services > Firewall > Forwarding Rules**.
2. In the left menu, click on **Networks**.
3. Right click in the main area and select **New**. The **Edit/Create Network Object** window will open.
4. Enter a **Name**.
5. To include or exclude a region or country:
   a. Click the globe icon either in the **Include** or **Exclude Entries** section.

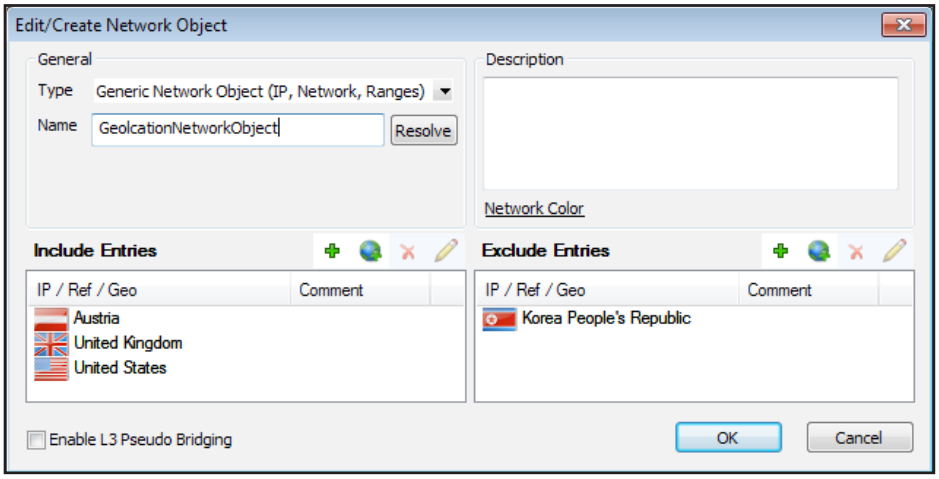


   b. In the **Select Region/Country** window, select the region or country.

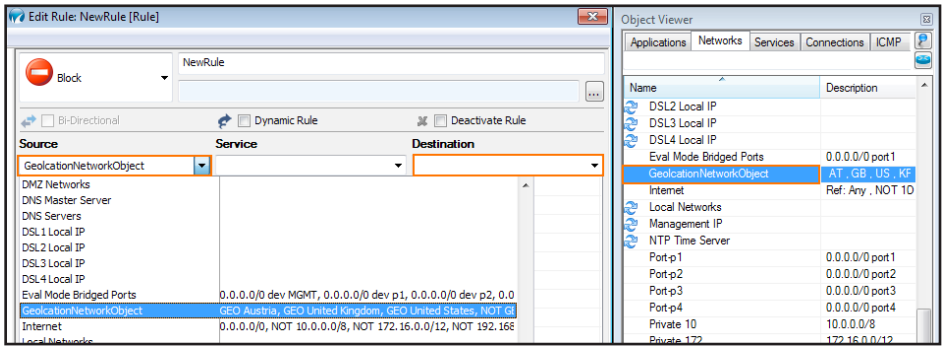


c. Click **OK**.

6. Click **Send Changes** and **Activate**.

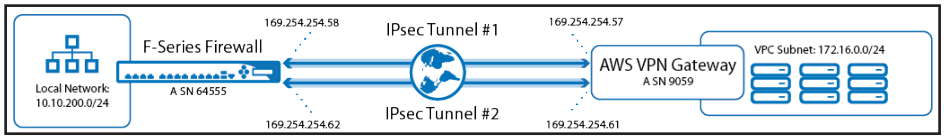


You can now select the geolocation network object you just created from the **Source** and **Destination** dropdown lists when creating access rules. Alternatively you can find the network object icon the Object Viewer in the **Networks > Network Objects** section.



3.9 How to Configure an IKEv1 IPsec VPN to an AWS VPN Gateway with BGP

If you are using the Amazon Virtual Private Cloud, you can transparently extend your local network to the cloud by connecting both networks with a site-to-site IKEv1 IPsec VPN tunnel. The Amazon virtual private gateway uses two parallel IPsec tunnels IKEv1 to ensure constant connectivity. The subnets behind the VPN Gateway are propagated via BGP. Additional Amazon AWS charges apply. For more information, see Amazon's monthly pricing calculator at http://calculator.s3.amazonaws.com/calc5.html.



3.9.1 Before You Begin

Create an Amazon Virtual Private Cloud (VPC).

The local and remote (VPC) subnets must not overlap. e..g, if your local network is 10.0.1.0/24, do not use 10.0.0.0/16 for your VPC.

- Create at least one subnet in the VPC.
- Create and configure the Amazon Routing Table.

Step 1. Create the Amazon VPN Gateway

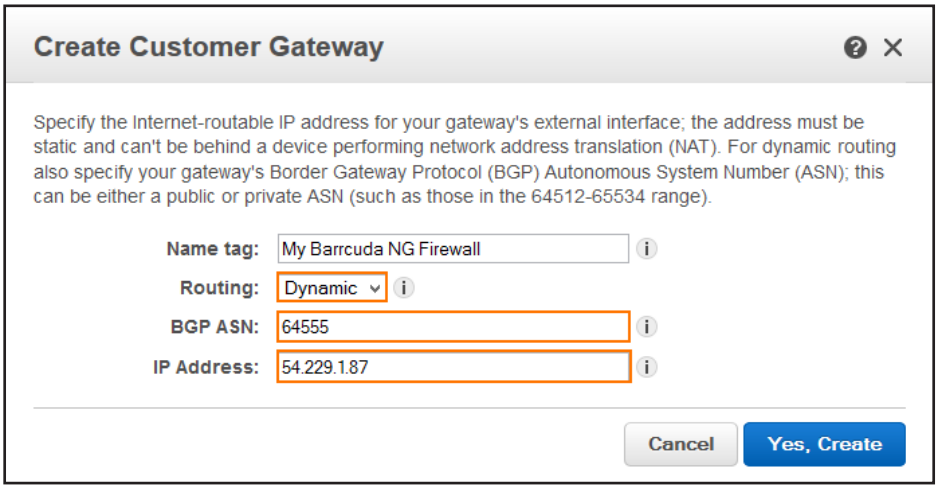
Step 1.1 Create a Virtual Private Gateway

1. The Amazon virtual private gateway is the VPN concentrator on the remote side of the IPsec VPN connection.
2. Go to the Amazon VPC Management Console.
3. In the left menu, click **Virtual Private Gateways**.
4. Click **Create Virtual Private Gateway**.
5. Enter the **Name tag** for the VPN gateway (e.g., `Techlib Virtual Private Gateway`).
6. Click **Yes, Create**.
7. Select the newly created virtual private gateway, and click **Attach to VPC**.
8. Select your VPC from the **VPC** list, and click **Yes, Attach**.
9. The virtual private gateway is now available.

Step 1.2 . Add Your Customer Gateway Configuration

The Amazon customer gateway is your Barracuda NextGen Firewall F-Series on your end of the VPN connection. Specify your external IP address and routing type in the customer gateway configuration:

1. Go to the Amazon VPC Management Console.
2. In the left menu, click **Customer Gateway**.
3. Click **Create Customer Gateway**.
4. Enter the connection information for your Barracuda Firewall:
   - **Name Tag** – Enter a name for your device (e.g., `My Barracuda NextGen Firewall F-Series`).
   - **Routing** – Select **Dynamic**.
   - **IP Address** – Enter your external **IP Address**. To look up your external IP address, go to **CONTROL > Network**.



5. Click **Yes, Create**.

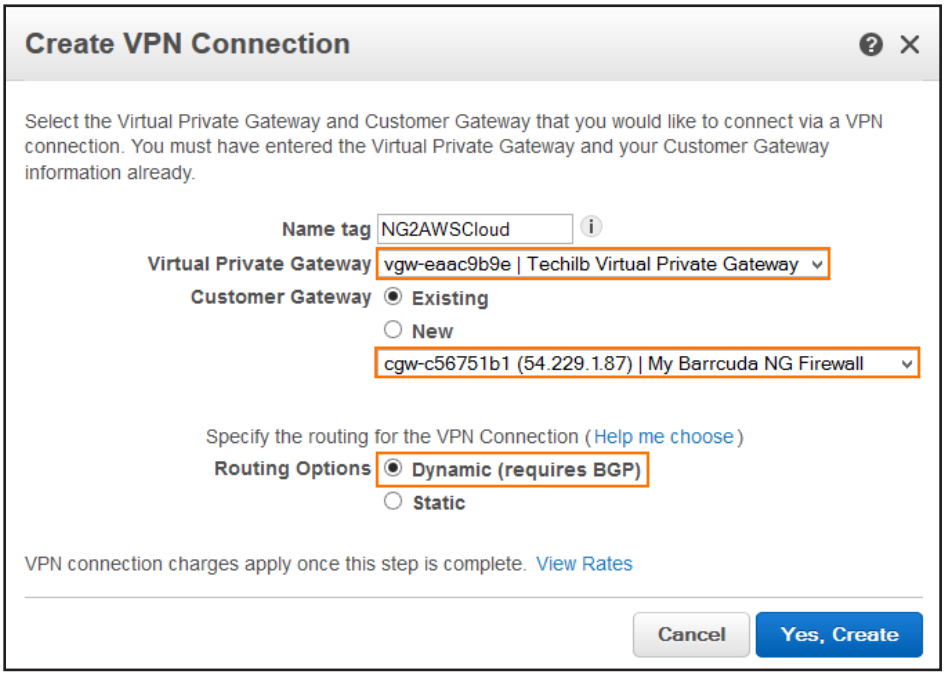
Your Barracuda NextGen Firewall F-Series is now configured in the AWS cloud and can be used to configure VPN connections.

Step 1.3 . Create a VPN Connection

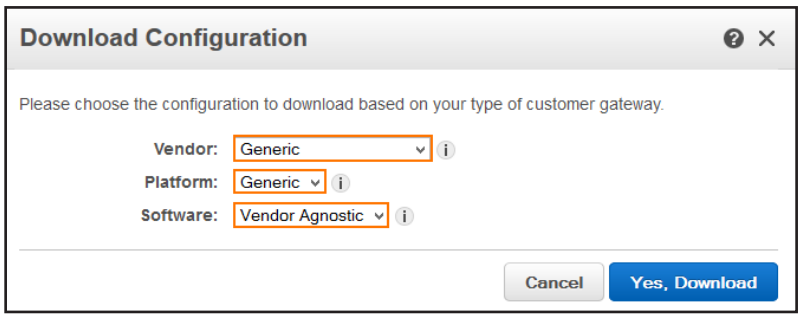
Create a VPN connection with the customer gateway and the virtual private gateway that you just created. Then download the VPN configuration file because it contains all the necessary information for configuring the VPN connection on the Barracuda NextGen Firewall F-Series.

The Amazon VPN configuration file is different for every VPN connection.

1. Go to the Amazon VPC Management Console.
2. In the left menu, click **VPN Connections**.
3. Click **Create VPN Connection**.
4. In the **Create VPN Connection** window, enter the configuration information for your VPN connection:
   - **Name tag** – Enter a name for your VPN connection (e.g., `NG2AWSCloud`).
   - **Virtual Private Gateway** – Select the virtual private gateway created in Step 1.
   - **Routing Options** – Select **Dynamic (requires BGP)**.



5. Click **Yes, Create**.
6. Click **Download Configuration**.
7. Select generic vendor and platform settings for the configuration file:
   - **Vendor** – Select **Generic**.
   - **Platform** – Select **Generic**.
   - **Software** – Select **Vendor Agnostic**.




```
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-YOUR-VPN-CONNECTION-ID
Your Virtual Private Gateway ID          : vgw-YOUR-VIRTUAL-PRIVATE-GATEWAY-ID
Your Customer Gateway ID             : cgw-YOUR-CUSTOMER-GATEWAY-ID

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
  - Authentication Method    : Pre-Shared Key
  - Pre-Shared Key           : YOUR-PRESHARED-KEY
  - 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:
  - 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                : YOUR-EXTERNAL-IP
  - Virtual Private Gateway           : VIRTUAL-PRIVATE-NETWORK-EXTERNAL-IP

Inside IP Addresses
  - Customer Gateway                : 169.254.254.58/30
  - Virtual Private Gateway            : 169.254.254.57/30

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              : 64555 <--- CAN BE REPLACED BY YOUR OWN ASN.
  - Virtual Private  Gateway ASN         : 9059
  - Neighbor IP Address               : 169.254.254.57
  - 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
  - Authentication Method    : Pre-Shared Key
  - Pre-Shared Key           : YOUR-PRESHARED-KEY
  - 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:
  - 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                : YOUR-EXTERNAL-IP
  - Virtual Private Gateway           : EXTERNAL-VIRTUAL-PRIVATE-NETWORK-IP

Inside IP Addresses
  - Customer Gateway                : 169.254.254.62/30
  - Virtual Private Gateway            : 169.254.254.61/30

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             : 64555 <--- CAN BE REPLACED WITH YOUR ASN
  - Virtual Private  Gateway ASN          : 9059
  - Neighbor IP Address              : 169.254.254.61
  - 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.
```

```
Additional Notes and Questions
================================================================================

  - Amazon Virtual Private Cloud Getting Started Guide:
      http://docs.amazonwebservices.com/AmazonVPC/latest/GettingStartedGuide
  - Amazon Virtual Private Cloud Network Administrator Guide:
      http://docs.amazonwebservices.com/AmazonVPC/latest/NetworkAdminGuide
  - XSL Version: 2009-07-15-1119716
```

Step 2. Configure IPsec Tunnels on the Barracuda NextGen Firewall F-Series

For each IPsec tunnel, create a next-hop-interface and then configure two IPsec site-to-site VPN tunnel. Use the IP addresses provided in the Amazon generic VPN configuration file you downloaded at the end of Step 1.

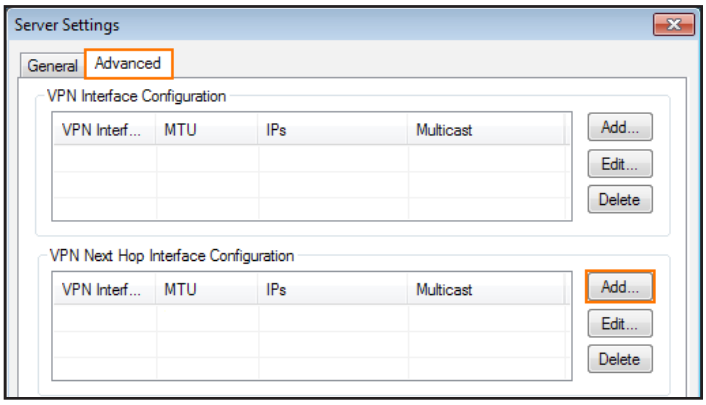
Step 2.1 Create VPN Next-hop Interfaces

For each IPsec tunnel, a VPN next-hop interface must be created. Use the IP addresses provided in the Amazon generic VPN configuration file you downloaded at the end of Step 1.

```
[...]
IPSec Tunnel #1
================================================================================
=========
[...]
#3: Tunnel Interface Configuration
[...]
Inside IP Addresses
  - Customer Gateway                : 169.254.254.58/30
  - Virtual Private Gateway             : 169.254.254.57/30

Configure your tunnel to fragment at the optimal size:
  - Tunnel interface MTU     : 1436 bytes
[...]
IPSec Tunnel #2
================================================================================
[...]
#3: Tunnel Interface Configuration
[...]
Inside IP Addresses
  - Customer Gateway                : 169.254.254.62/30
  - Virtual Private Gateway             : 169.254.254.61/30
Configure your tunnel to fragment at the optimal size:
  - Tunnel interface MTU     : 1436 bytes
[...]
```

1. Go to **CONFIGURATION > Configuration Tree > Box > Virtual Servers >** `your virtual server` **> Assigned Services > VPN-Service > VPN Settings** .
2. Click **Lock**.
3. Click on **Click here for Server Settings**.

4. Click on the **Advanced** tab.

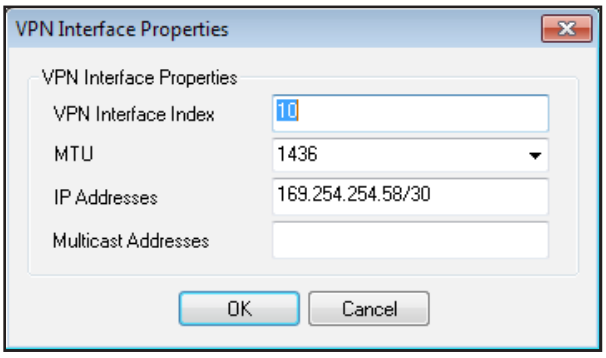


5. Create a VPN next hop interface for each IPsec tunnel by clicking **Add** in the **VPN Next Hop Interface Configuration** section.

   a. In the **VPN Interface Properties** window enter:

   - **VPN Interface Index** – Enter a number between 0 and 99. Each interface index number must be unique. e.g., IPsec tunnel1: `10` and IPsec tunnel: `11`
   - **MTU** – Enter `1436`.
   - **IP Addresses** – Enter the **Inside IP Address** for the **Customer Gateway** provided by Amazon. e..g, IPsec tunnel1: `169.254.254.58/30`, IPsec tunnel 2: `169.254.254.62/30`

   b. Click **OK**.



6. Click **OK**.
7. Click **Send Changes** and **Activate**.

Step 2.2 Configure Two Site-to-Site IPsec Tunnels

Configure two site-to-site IPsec tunnels using the VPN next-hop interfaces. Make sure to use the correct IP addresses and corresponding next-hop interfaces listed in the Amazon generic VPN configuration file for each tunnel.

```
Amazon Web Services
Virtual Private Cloud
[...]

IPSec Tunnel #1
================================================================================
#1: Internet Key Exchange Configuration

Configure the IKE SA as follows
 - Authentication Method : Pre-Shared Key
 - Pre-Shared Key : YOUR-PRESHARED-KEY
 - 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:
 - 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
 [...]
#3: Tunnel Interface Configuration
[...]
Outside IP Addresses:
 - Customer Gateway : YOUR-EXTERNAL-IP-ADDRESS
 - Virtual Private Gateway : AMAZON-VPN-GATEWAY-IP-ADDRESS-TUNNEL-2

[...]
Configure your tunnel to fragment at the optimal size:
 - Tunnel interface MTU : 1436 bytes

[...]

IPSec Tunnel #2
================================================================================
#1: Internet Key Exchange Configuration

Configure the IKE SA as follows
 - Authentication Method : Pre-Shared Key
 - Pre-Shared Key : YOUR-PRESHARED-KEY
 - 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:
 - 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
[...]
#3: Tunnel Interface Configuration
[...]
Outside IP Addresses:
 - Customer Gateway : YOUR-EXTERNAL-IP-ADDRESS
 - Virtual Private Gateway : AMAZON-VPN-GATEWAY-IP-ADDRESS-TUNNEL-2

[...]
Configure your tunnel to fragment at the optimal size:
 - Tunnel interface MTU : 1436 bytes

[...]
```

Services > VPN-Service > Site to Site .

2. Click on the **IPSEC IKEv1 Tunnels** tab.
3. Click **Lock**.
4. For each IPsec tunnel, right-click and click **New IPsec IKEv1 tunnel**.
   a. Enter the IPsec tunnel configurations:
      i. Enter a **Name**. e.g, IPsec Tunnel 1: `IPsecAWSTunnel1` and for IPsec Tunnel 2: `IPsecAWSTunnel2`
      ii. Enter the **Phase 1** and **Phase 2** settings:

	Phase 1	Phase 2
Encryption	AES	AES
Hash Meth.	SHA	SHA
DH-Group	Group2	Group 2
Lifetime(sec)	28800	3600
Perfect Forward Secrecy		Enable

      iii. In the **Local Network s** tab:
      - **Local IKE Gateway** – Enter your external IP address. If you are using a dynamic WAN interface enter `0.0.0.0`
      - **Network Address** – Enter the **Inside IP Address** of the **Customer Gateway** (without the /30) and click **Add**.

        e.g., IPsec tunnel 1 `169.254.254.58` and for IPsec tunnel 2 `169.254.254.62`.

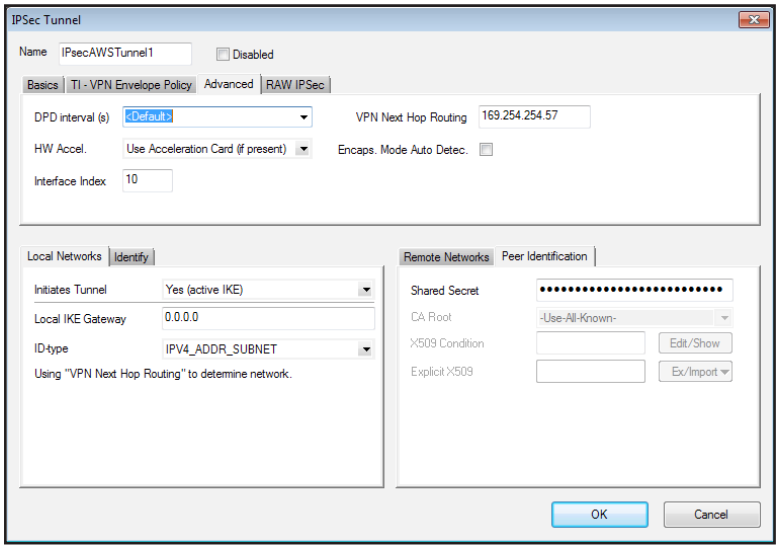
      iv. In the **Remote Networks** tab:
      - **Remote IKE Gateway** – Enter the **Outside IP Address** of the **Virtual Private Gateway .**

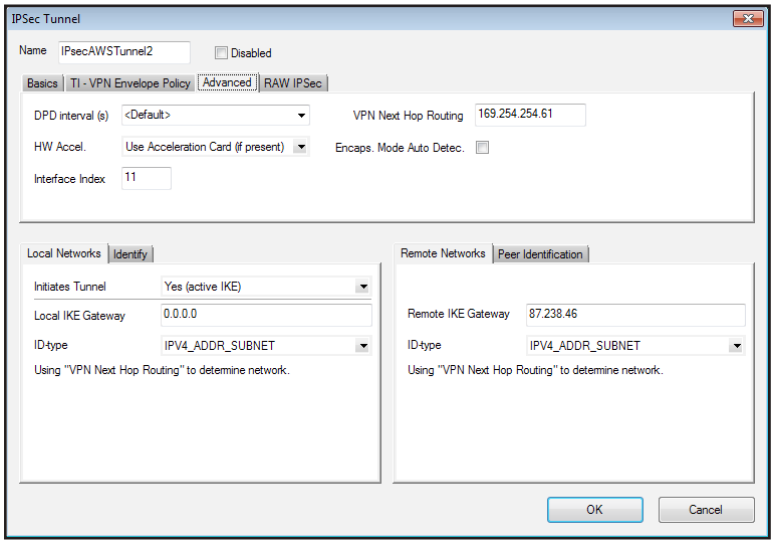
      v. In the **Peer Identification** tab:
      - **Shared Secret** – Enter the Amazon **Pre-Shared Key**.

vi. In the **Advanced** tab:

- **DPD intervals (s)** – Enter `10`.
- **Interface Index** – Enter the **VPN Next Hop Interface index** number you entered in step 1.1. e.g., IPsec tunnel 1 `10` and for IPsec tunnel 2 `11`.
- **VPN Next Hop Routing** – Enter the **Inside IP address** of the **Virtual Private Gateway.** e.g., IPsec tunnel 1 `169.254.254.57` and for IPsec tunnel 2 `169.254.254.61`

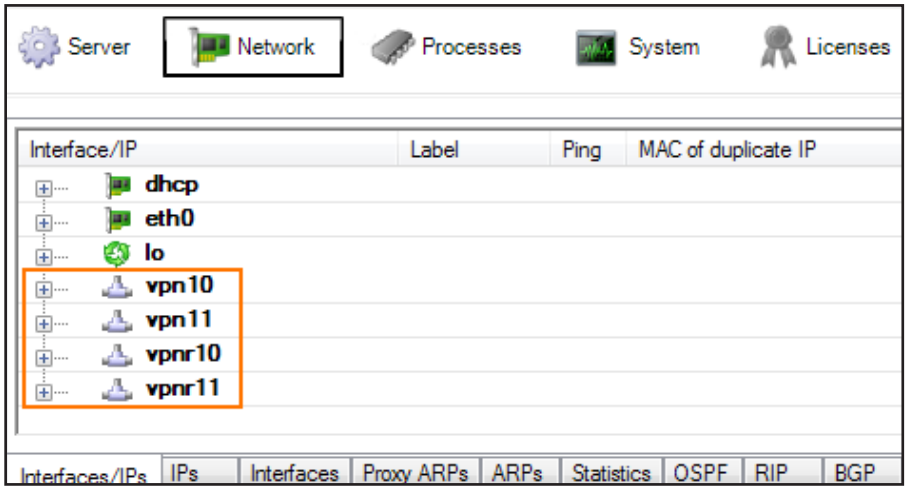
vii. Click **OK**.

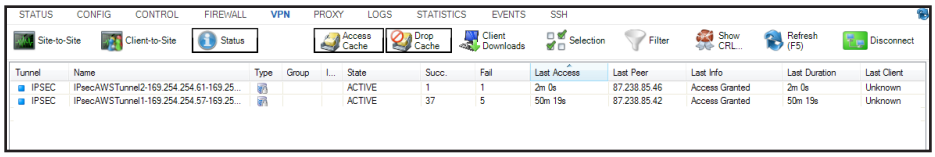




5. Click **Send Changes** and **Activate**.

You now have two VPN next-hop interfaces listed in the **Interfaces/IPs** section on the **CONTROL > Network** page and the VPN tunnels on the **CONTROL > VPN > STATUS**.





Step 3. Configure the BGP Service

Configure BGP routing to learn the subnets on the other side of the VPN tunnels. The BGP route propagated by the second (backup) IPsec tunnel is artificially elongated so traffic is routed per default over the first IP tunnel, as suggested by Amazon.

```
[...]IPSec Tunnel #1
================================================================================
[...]
#4: Border Gateway Protocol (BGP) Configuration:
[...]
BGP Configuration Options:
 - Customer Gateway ASN : YOUR-ASN-NUMBER (e.g., 64555)
 - Virtual Private Gateway ASN : 9059
 - Neighbor IP Address : 169.254.254.57
 - Neighbor Hold Time : 30
 [...]

IPSec Tunnel #2
================================================================================
[...]

#4: Border Gateway Protocol (BGP) Configuration:
[...]
BGP Configuration Options:
 - Customer Gateway ASN : 64555
 - Virtual Private Gateway ASN : 9059
 - Neighbor IP Address : 169.254.254.61
 - Neighbor Hold Time : 30
 [...]
```

Step 3.1 Configure Routes to be Advertised via BGP

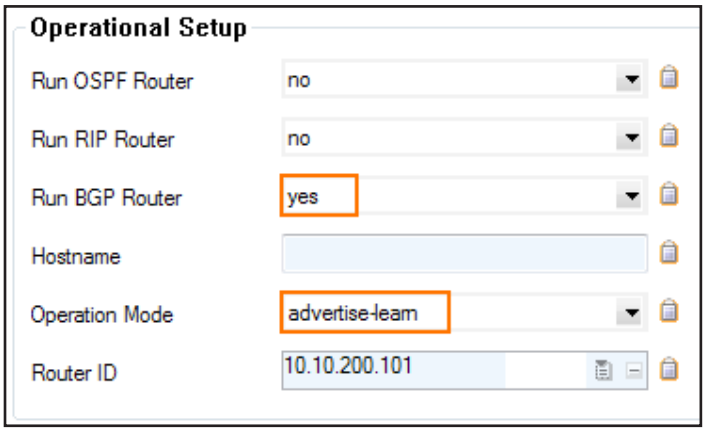
Only routes with the parameter **Advertise** set to **yes** will be propagated via BGP.

1. Go to **CONFIGURATION > Configuration Tree > Box > Network** .
2. Click **Lock**.
3. (optional) To propagate the management network, set **Advertise Route** to **yes**.
4. In the left menu, click on **Routing**.
5. Double-click on the **Routes** you want to propagate, and set **Advertise Route** to **yes**.
6. Click **OK**.
7. Click **Send Changes** and **Activate**.

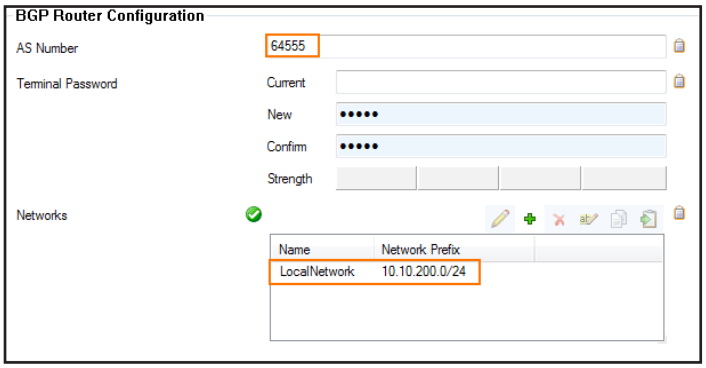
Step 3.2 Configure the BGP Routes

Configure the BGP setting for the BGP service on the Barracuda NextGen Firewall F-Series.

1. Go to **CONFIGURATION > Configuration Tree > Box > Virtual Servers >** `your virtual server` **> Assigned Services > OSPF-RIP-BGP-Service > OSPF/RIP/BGP Settings** .
2. Select **yes** from the **Run BGP Router** list.
3. Select **advertise-learn** from the **Operations Mode** list.



4. In the left menu, click **BGP Router Setup**.
5. Enter the **AS Number** (e.g., `64555`).
6. In the **Networks** table, add the local network(s)(e.g., `10.10.200.0/24`).

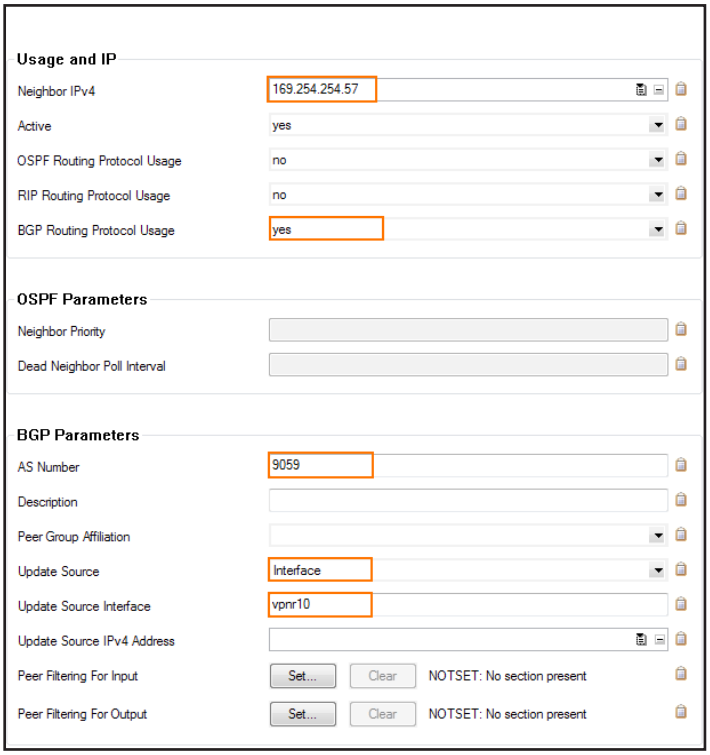


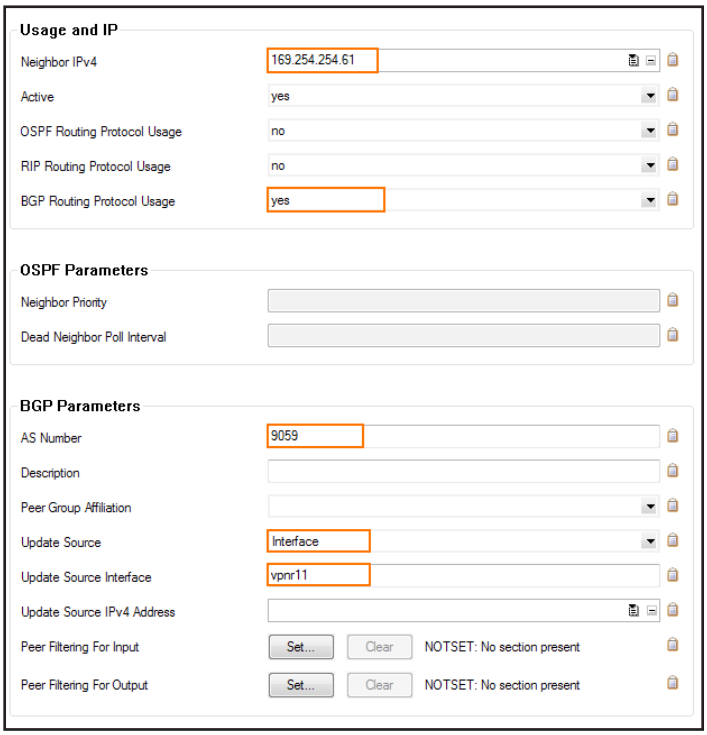
7. In the left menu, expand **Configuration Mode** and click **Switch to Advanced Mode**.
8. Click the **Set** button for the **Advanced Settings.** The **Advanced Settings** window opens.
9. Set the **Hold timer** to `30` seconds.
10. Set the **Keep Alive Timer** to `10` seconds.
11. Click **OK**.
12. Click **Send Changes** and **Activate**.

Step 3.3 . Add a BGP Neighbor for each IPsec Tunnel

To dynamically learn the routing of the neighboring network, set up a BGP neighbor for each VPN next-hop interface.

1. In the left menu of the **OSPF/RIP/BGP Settings** page, click **Neighbor Setup IPv4**.
2. Click **Lock**.
3. For each IPsec tunnel, click the plus sign (**+**) next to the **Neighbors** table to add a new neighbor.
4. Enter a **Name** for the neighbor. e.g., `AWS1` and `AWS2`
5. In the **Neighbors** window, configure the following settings in the **Usage and IP** section:
   - **Neighbor IPv4** – Enter the inside IP Address of the Virtual Private Gateway (remote address for the VPN next hop interface on the NextGen Firewall F-Series) e.g., IPsec Tunnel 1: `169.254.254.57` and for IPsec Tunnel 2 `169.254.254.61`.
   - **OSPF Routing Protocol Usage** – Select **no**.
   - **RIP Routing Protocol Usage** – Select **no**.
   - **BGP Routing Protocol Usage** – Select **yes**.
6. In the **BGP Parameters** section, configure the following settings:
   - **AS Number**: Enter the ASN for the remote network: `9059`
   - **Update Source**: Select **Interface**.vpnr
   - **Update Source Interface**: Enter the vpnr interface for the IPsec tunnels. e.g., IPsec Tunnel 1: `vpnr10` and for IPsec Tunnel 2 `vpnr11`.





7. Click **OK**.
8. Click **Send Changes** and **Activate**.

Step 3.4 . Add an Access List for the Second IPsec Tunnel

1. In the left menu of the **OSPF/RIP/BGP Settings** page, click **Filter Setup IPv4**.
2. In the **Access List IPv4 Filters** section, click **+**.
3. Enter a **Name** for the Access List. e.g., `2ndGWIP` The **Access List IPv4** windows opens.

4. Click **+** to add an access list **Type**. The **Type** window opens.
5. Select **permit** from the **Type** dropdown.
6. Enter the **Inside IP** for the **Virtual Private Gateway** for IPsec Tunnel #2. E.g., `169.254.254.62`
7. Click **OK**.
8. Click **OK**.

Step 3.5 Add a Filter Setup for the Second IPsec Tunnel

To make the route over the first IPsec tunnel the preferred route, we will lengthen the AS-Path of the second tunnel.

1. In the left menu of the **OSPF/RIP/BGP Settings** page, click **Filter Setup IPv4**.
2. Click **Lock**.
3. In the **Route Map IPv4 Filters** section, click on **+**. The **Route Maps IPv4** window opens.
4. In the **BGP Specific Conditions** section, click **+**. The **Route Map Entry** window opens.
5. In the **Route Map Entry** window, specify the following settings:
   - **Sequence Number** – Enter a unique sequence number (e.g., `1`). This sequence number must be unique across all route maps. For additional entries, iterate the sequence numbers.
   - **Type** – Select **permit**.
   - **Match Condition** – Select **Gateway_IP**.
   - **Gateway IP (Access List)** – Select the access list for the listed created in Step 3.4.
   - **Set Action** – Select **AS_Path**.
   - **Set addition to AS-Path** – Enter Amazons ASN number `9059`.
6. Click **OK**.
7. Click **OK**.
8. Click **Send Changes** and **Activate**.

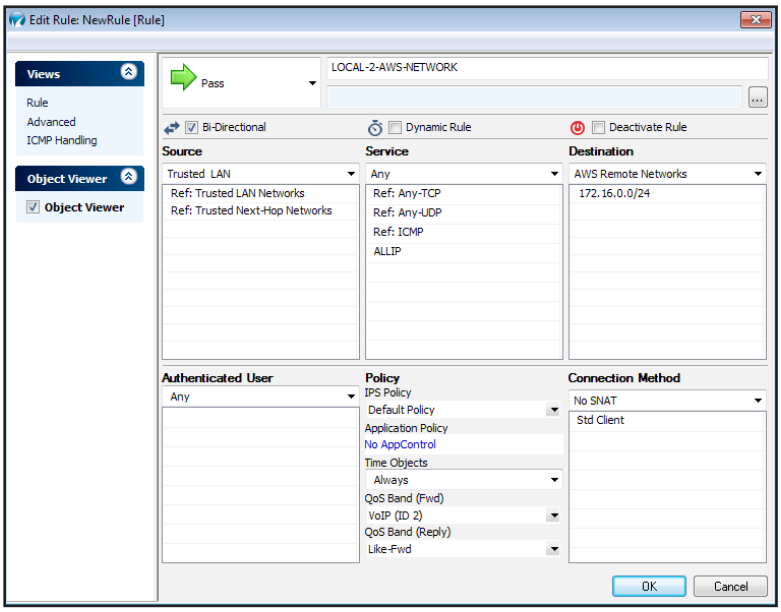
Step 4. Create an Access Rule for VPN Traffic

To allow traffic to and from the VPN networks, a pass access rule is needed. You also need to set the **Clear DF bit** and **Force Maximum Segment Size** settings according to the Amazon configuration file in the advanced firewall rule settings. You also need to set **Reverse Interface (Bi-directional)** to **Any** to allow return traffic using a different VPN tunnel than was used to initiate the connection.

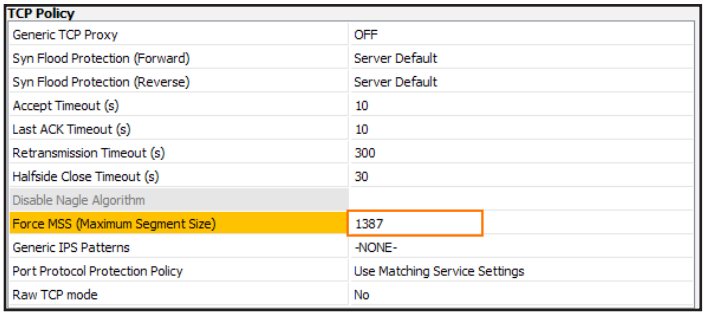
```
[...]

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

[...]
```

Step 5. Create a Pass firewall rule:

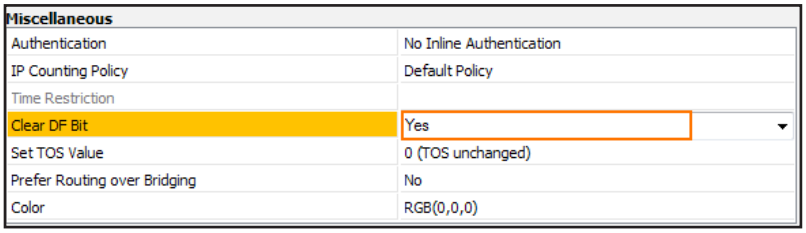
- **Bi-Directional** – Enable.
- **Source** – Select the local network(s) you are propagating via BGP.
- **Service** – Select the service you want to have access to the remote network or **ALL** for complete access.
- **Destination** – Select the remote VPC subnet(s).
- **Connection Method** – Select **Original Source IP**.



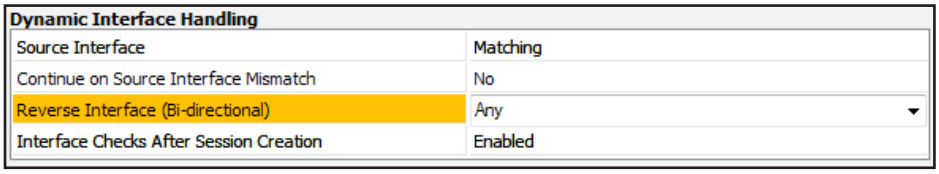
1. In the left navigation, click on **Advanced.**
2. In the **TCP Policy** section, set **Force MSS (Maximum Segment Size)** to `1387`.



3. In the **Miscellaneous** section, set **Clear DF Bit** to **Yes**.

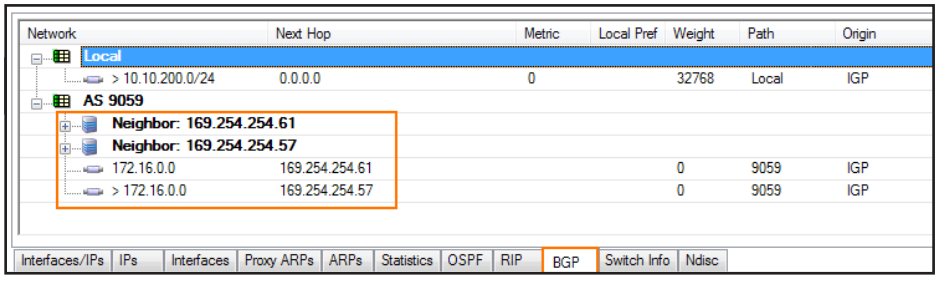


4. In the **Dynamic Interface Handling** section, set **Reverse Interface (Bi-directional)** to **Any**.

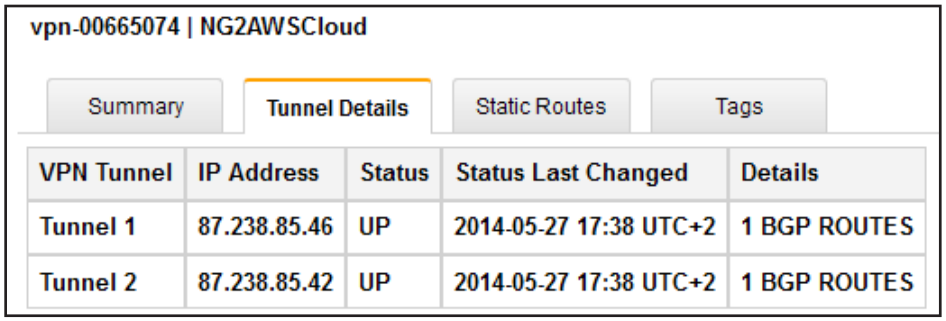


5. Click **OK**.
6. Move the access rule up in the rule list, so that it is the first rule to match the firewall traffic.
7. Click **Send Changes** and **Activate**.

You now have two IPsec VPN tunnels connecting your F-Series firewalls to the Amazon AWS cloud. Per default, the first IPsec tunnel is chosen. It may take some time for BGP to learn the new routes, in case of a failure.



AWS VPN status in the Amazon AWS management interface



3.10 How to Deploy an F-Series Firewall in AWS via CloudFormation Template

CloudFormation templates allow you to automate your deployments in AWS and make them more consistent. You can replicate the deployment multiple times for testing and production, or you can spin up additional environments in other regions.

3.10.1 CloudFormation Templates

CloudFormation templates are available for all our AWS reference architectures in the Barracuda Networks GitHub account: https://github.com/barracudanetworks/ngf-aws-templates.

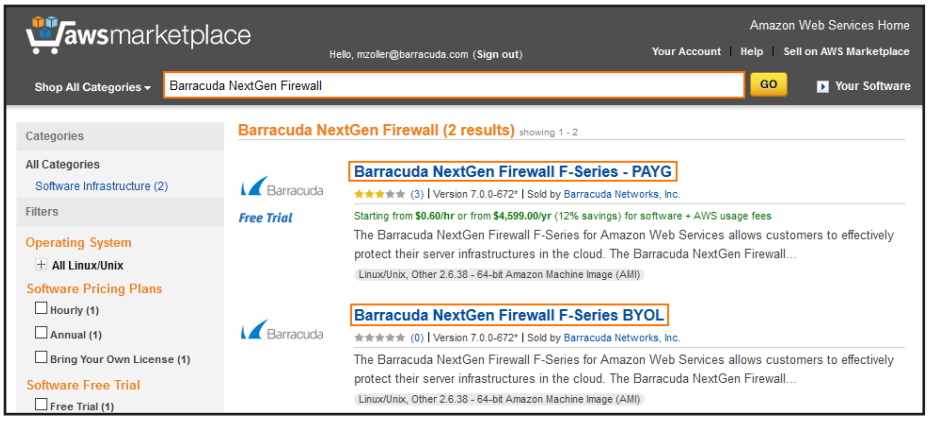
3.10.2 Before You Begin

Verify that the AMI image IDs used in the CloudFormation template match the IDs for the NextGen Firewall image listed in the AWS Marketplace. The AMI disk images change for every released version. Each region has a separate AMI ID.

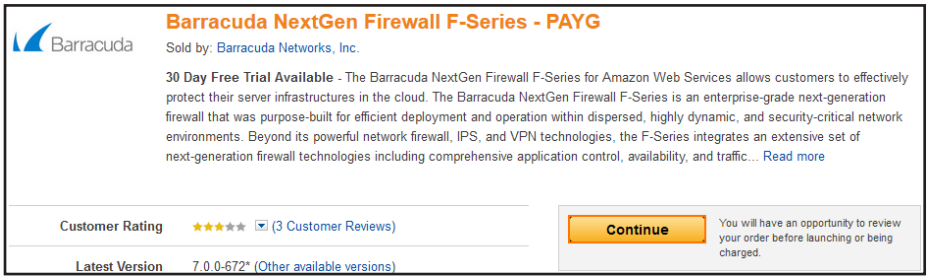
Step 1. Subscribe to NextGen Firewall in AWS Marketplace

To be able to deploy a NextGen Firewall image via the CloudFormation template, you must agree to the **Terms of Service** and subscribe to the image in the AWS Marketplace. You need to do this only once per account, but it must be done separately for PAYG and BYOL images.

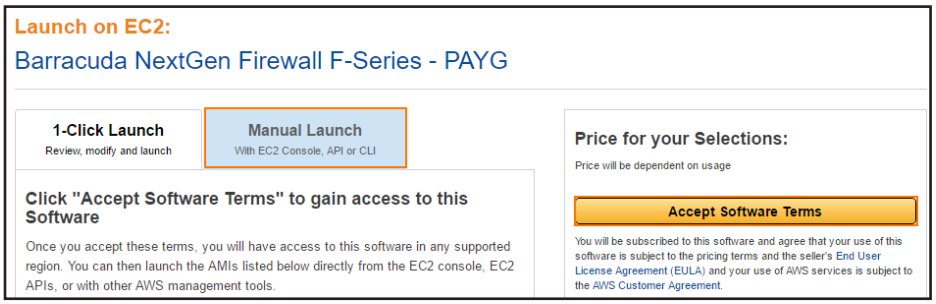
1. Go to the AWS Marketplace: https://aws.amazon.com/marketplace/
2. Search for `Barracuda NextGen Firewall`.
3. Click on the **Barracuda NextGen Firewall F-Series PAYG** or **Barracuda NextGen Firewall F-Series BYOL** image.



4. Click **Continue**.



5. Click on the **Manual Launch** tab.
6. Click **Accept Software Terms**.



You will now receive an email from Amazon confirming your subscription. You can now use the provided AMI in your CloudFormation templates.

Step 2. (BYOL only) Create Stack Policy to Protect Firewall Instance from Stack Updates

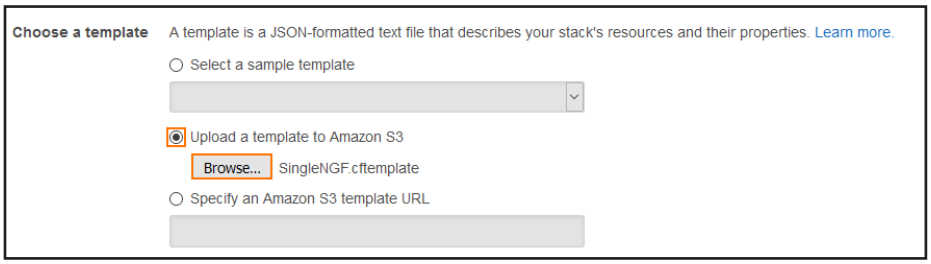
Protect your firewall instances from being replaced during stack updates use a stack policy when deploying the CloudFormation template. Replacing the instance automatically invalidates your license. If your license is invalidated, contact Barracuda technical support during the 15 day grace period to transfer your license to the instance.

Step 3. Deploy the CloudFormation Template

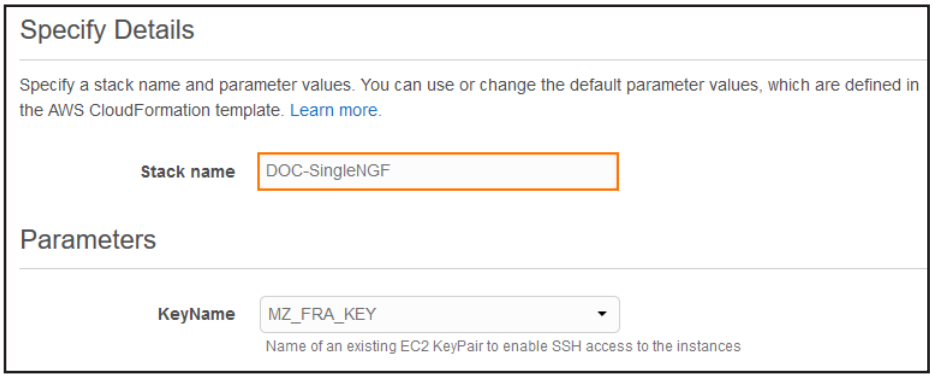
CloudFormation templates can be deployed via the AWS web console, CLI, REST, or PowerShell.

1. Log into the AWS console.
2. Click **Services** and select **CloudFormation**.
3. Click **Create Stack**.

4. Select **Upload a template to Amazon S3**.
5. Click **Browse** and select the template file.



6. Click **Next**.
7. Enter the **Stack name**.
8. (optional) If the template includes parameters, fill in the values in the **Parameters** section.



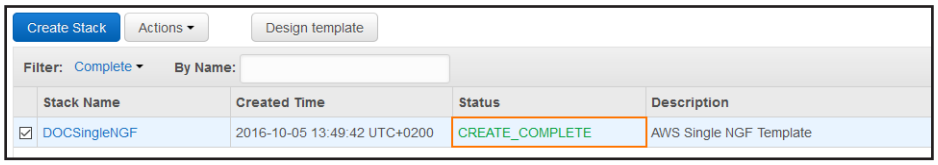
9. Click **Next**.
10. (optional) Enter **Tags** for your stack.
11. In the **Advanced** section, set additional options for your stack:
    - **Notification options**
    - **Timeout** – Set the timeout in minutes.
    - **Rollback on failure** – When set to **yes**, the deployment will be rolled back if any errors are encountered.

Stack policy – For BYOL images, it is highly recommended to protect the firewall instance from stack updates.

Stack updates that require redeploying the firewall instance will invalidate the license for BYOL firewalls.

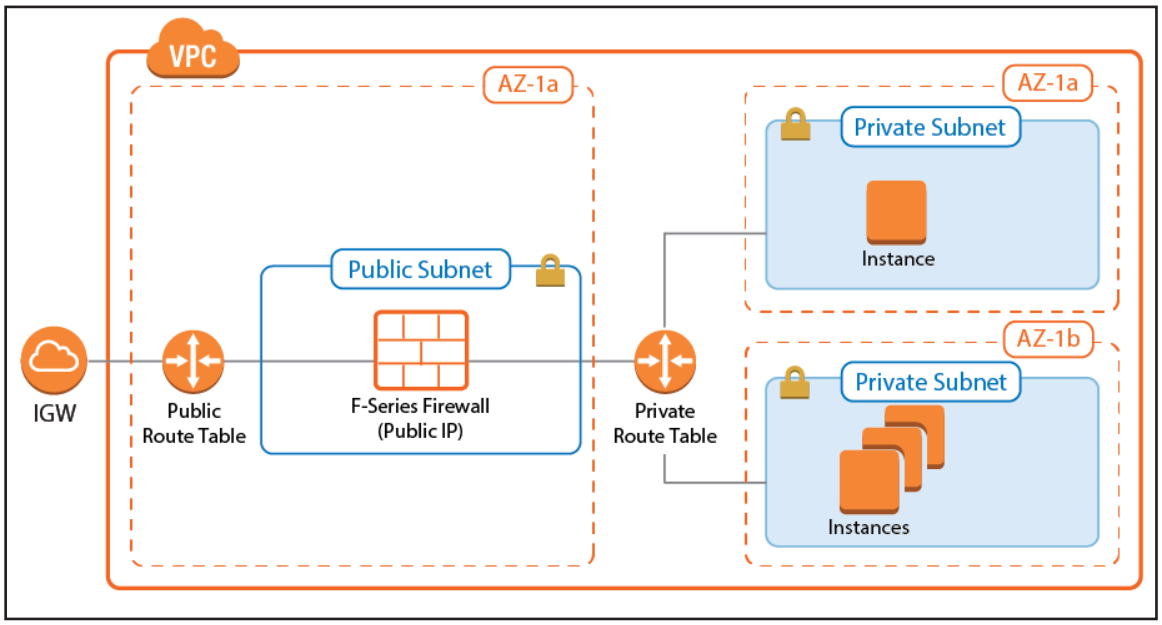
12. Click **Next**.
13. Review the settings and click **Create**.

The resources defined in the template are now deployed. This may take a couple of minutes. When the **Status** column shows **CREATE_COMPLETE**, the template has been deployed successfully. If the firewall fetches a PAR file from a Control Center, it may take a couple of minutes for the firewall to be available.



3.11 How to Deploy an F-Series Firewall in AWS via Web Portal

The Barracuda NextGen Firewall F in AWS secures and connects the services running in your AWS virtual private cloud (VPC). The firewall monitors and secures all traffic between subnets to and from the Internet. It also connects your cloud resources either to your on-premise networks with site-to-site VPN, or to your remote users with client-to-site VPN and SSL VPN. After the deployment the Instance ID is the root password set to log in via NextGen Admin. Logging in via SSH is only possible through certificate file set during the last deployment step.



Step 1. Create an IAM Role for the Firewall

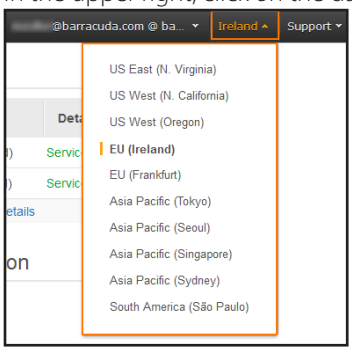
Create an IAM role for your firewall instance. Verify that the all the required IAM policies are attached to the role.

For step-by-step instructions, see

3.1 How to Create an IAM Role for an F-Series Firewall in AWS(page 79)

Step 2. Select the AWS Datacenter

1. Log into the AWS console.
2. In the upper right, click on the datacenter location, and select the datacenter you want to deploy to from the list.

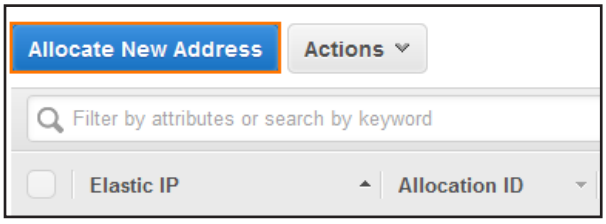


The selected datacenter location is now displayed in the AWS console.

Step 3. Create an Elastic IP

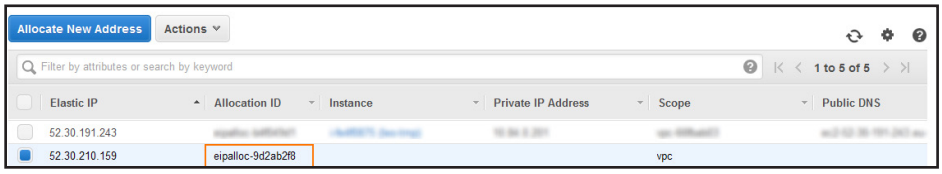
Create an elastic IP address. This is the public IP address that will be used for your firewall instance.

1. Log into the AWS console.
2. Click **Services** and select **EC2**.
3. In the **Network & Security** section of the left menu, click on **Elastic IPs**.
4. Click **Allocate New Address**.



5. Click **Yes, Allocate**.

An unassigned elastic IP is now added to the list. Copy the **Allocation ID** for future use.

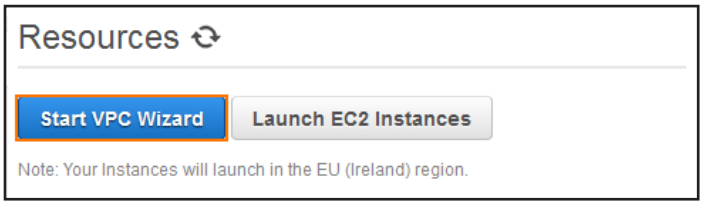


Step 4. Create VPC with VPC Wizard

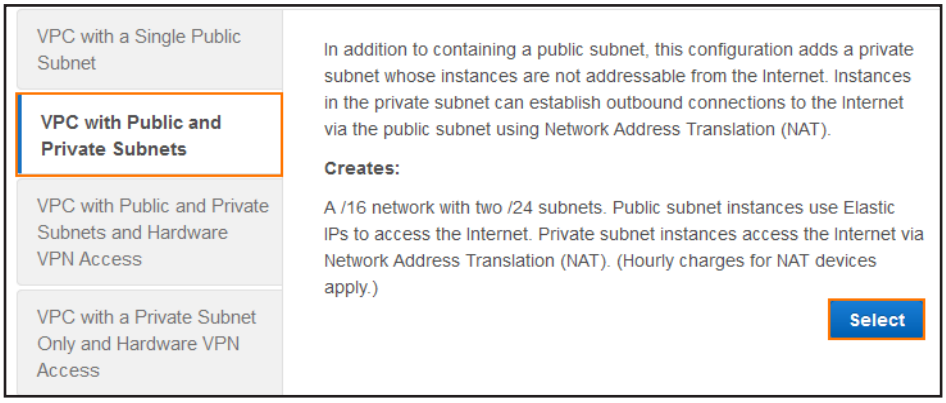
Use the VPC wizard to create a VPC with one public and one private subnet. The firewall will be deployed in the public subnet.

If needed, you can add additional subnets after the deployment.

1. Log into the AWS console.
2. Click **Services** and select **VPC**.
3. Click **Start VPC Wizard**. The VPC wizard opens.

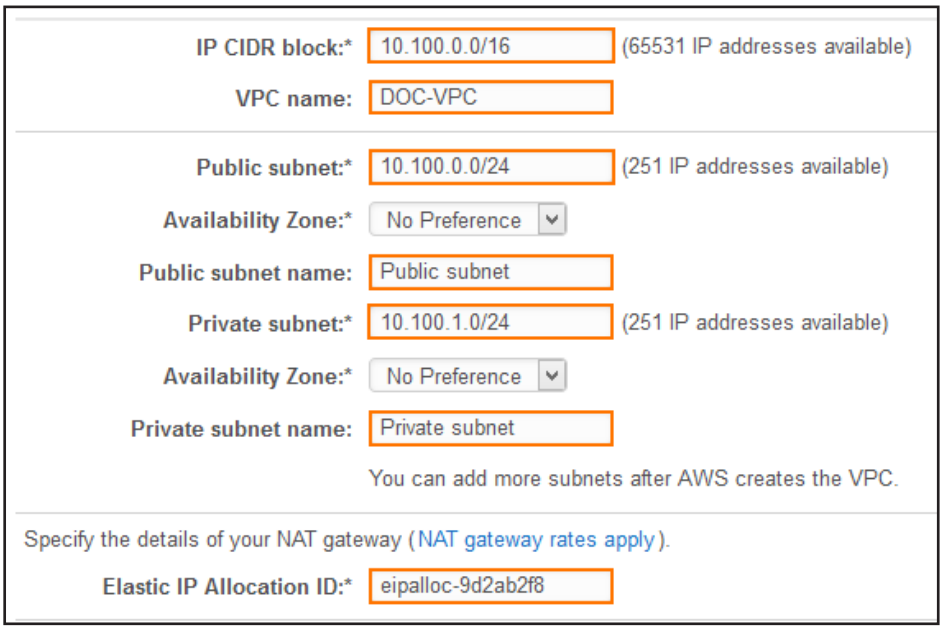


4. Select **VPC with Public and Private Subnets** and click **Select**.

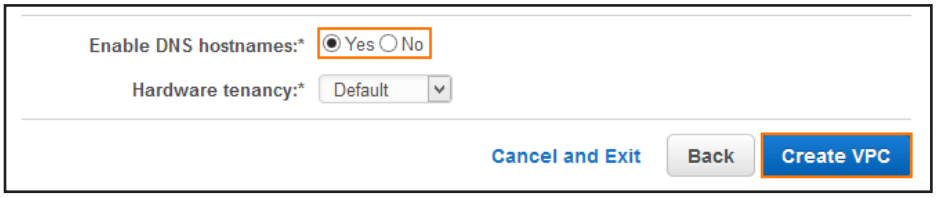


5. On the **VPC with Public and Private Subnets** change the following settings:

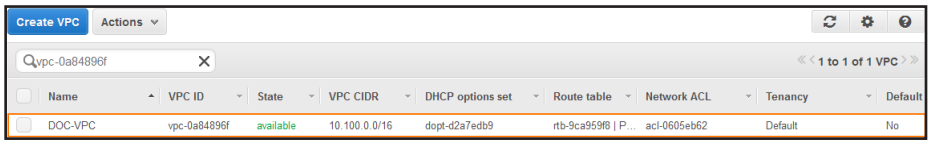
   - **IP CIDR block** – Enter a /16 CIDR block that does not overlap with any of your other networks.
   - **VPC Name** – Enter the name.
   - **Public subnet** – Enter the /24 subnet used for the firewall instance.
   - **Public subnet name** – Enter a name for the public subnet.
   - (optional) **Availability Zone** – Select which availability zone the VPC is created in. Select **No Preference** for AWS to assign it automatically.
   - **Private subnet** – Enter the /24 subnet used for the instances protected by the firewall.
   - **Private subnet name** – Enter a name for the private subnet.
   - **Elastic IP Allocation ID** – Enter the **Allocation ID** for the elastic IP address created in step 1.



6. (optional) Set **Enable DNS hostnames** to **NO** to only use IP addresses to access your VPC.
7. Click **Create VPC**.



The VPC is now listed in the **Your VPCs** list.



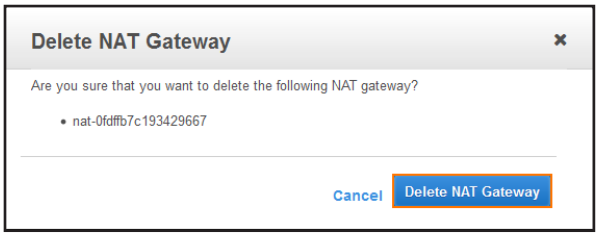
Step 5. Delete the NAT Gateway

Delete the NAT gateway.

The VPC wizard automatically creates a NAT gateway instance. But since the firewall already includes this functionality, the NAT gateway instance must be deleted.

1. Log into the AWS console.
2. Click **Services** and select **VPC**.
3. In the **Virtual Private Cloud** section of the left menu, click on **NAT Gateways**.
4. (optional) Enter the VPC ID in the **search bar**.
5. Select the NAT gateway created for your VPC and click **Delete NAT Gateway**. The **Delete NAT Gateway** pop-over window opens.

6. Click **Delete NAT Gateway**.

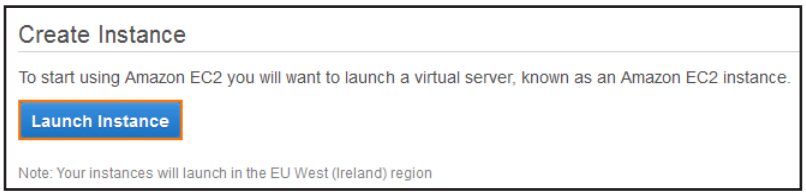


The elastic IP address associated with the NAT gateway is released automatically and is now free to use for the firewall instance.

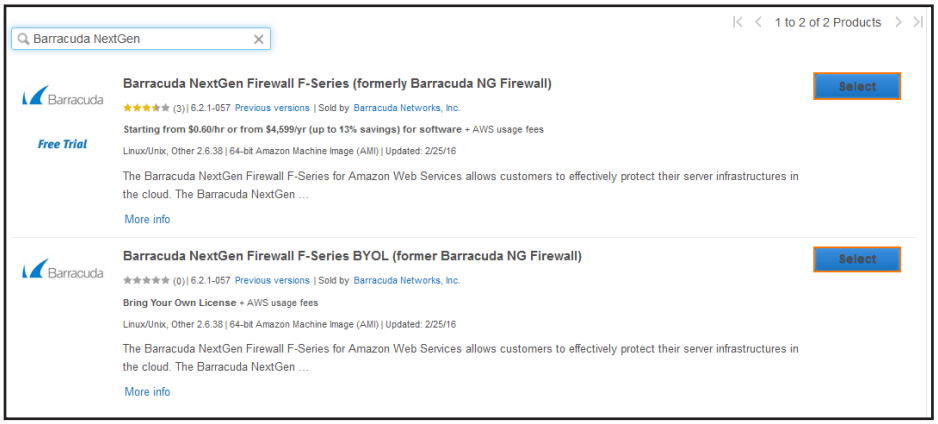
Step 6. Deploy the NextGen Firewall F Instance

You can deploy the NextGen Firewall F instance in two different ways from the AWS Marketplace: BYOL and hourly. The firewall instance is deployed into the public subnet and can be configured to use either a single network interface or one network interface per subnet. The number of network interfaces is limited by the instance size.

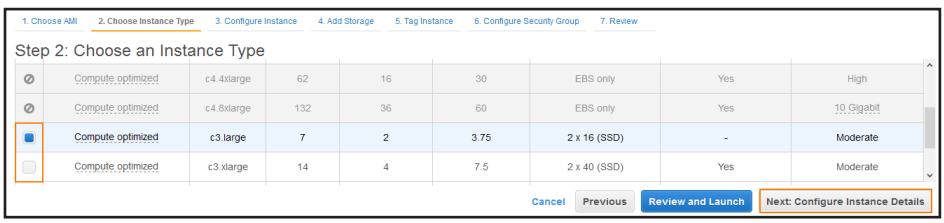
1. Log into the AWS console.
2. Click **Services** and select **EC2**.
3. In the **Create Instance** section, click **Launch Instance**. The **VPC wizard** starts.



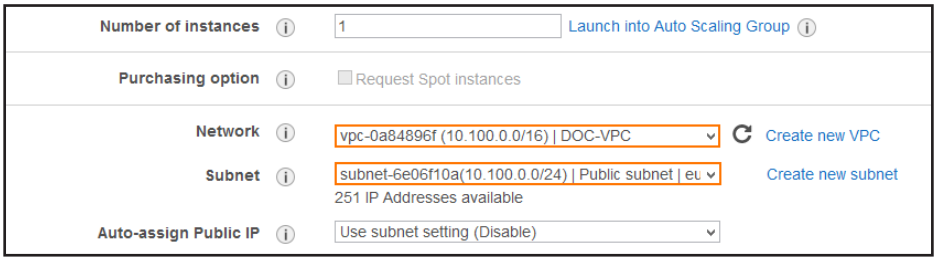
4. In the left menu, click **AWS Marketplace**.
5. Enter `Barracuda NextGen` in the **Search for AWS Marketplace Product** search box.
6. Click **Select** next to the image type you want to deploy: BYOL or hourly.



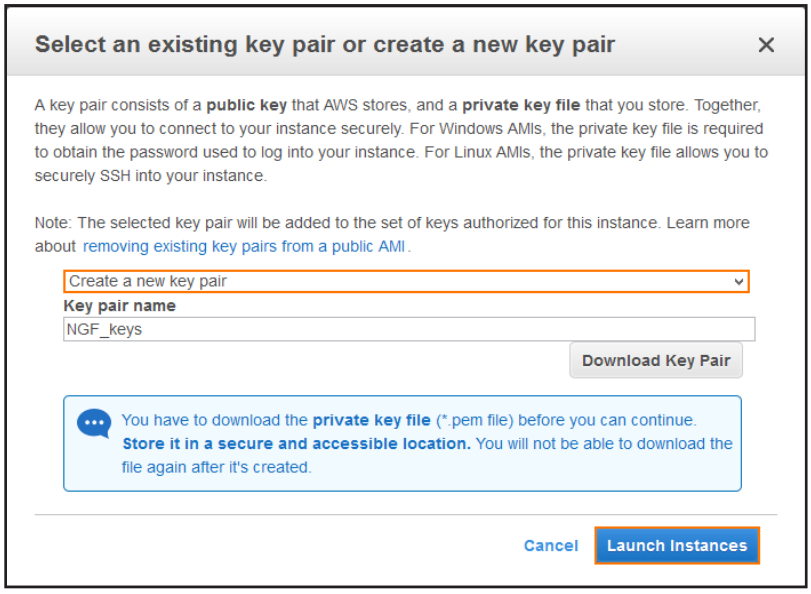
7. Select the **Instance Type**. If you are deploying a BYOL image, verify that the number of CPU cores of the instance matches your license.
8. Click **Next: Configure Instance Details**.



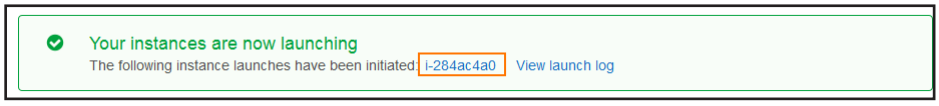
9. Configure the **Instance Details**:
   - **(HA only) Number of instances** – To deploy two instances to create an HA cluster, enter 2. For stand-alone deployments, deploy one instance.
   - **Network** – Select the VPC created in step 2.
   - **Subnet** – Select the public subnet.



10. (optional) Add additional **Network Interfaces**:
    - Click **Add Device**. The device is added to the list.
    - Select the **Subnet** the network interface is connected to.
    - (optional) Enter the **Primary IP** address for this interface. The IP address must be in the subnet selected above.
11. Click **Next:Add Storage**.
12. (optional) Change the **Volume Type** as needed.
13. Click **Next: Tag Instance**.
14. Click **Next: Configure Security Group**.
15. (optional) Click **Add Rule** and add rules for ICMP
    - **Type** – Select **All ICMP**.
    - **Source** – Select **Anywhere**.
16. (optional) Click **Add Rule** and add rules for HTTP
    - **Type** – Select **HTTP**.
    - **Source** – Select **Anywhere**.
17. Click **Review and Launch**.
18. Click **Launch**. The **Select and existing key pair or create a new key pair** pop-over window opens.
19. From the drop-down list, select **Choose an existing key pair** or **Create a new key pair**. The certificate is valid only for SSH logins with the root user. For NextGen Admin the Instance ID is the default password.
20. Click the checkbox to verify that you have access to the selected key or click **Download Key Pair** to download a new key pair.
21. Click **Launch Instances**.



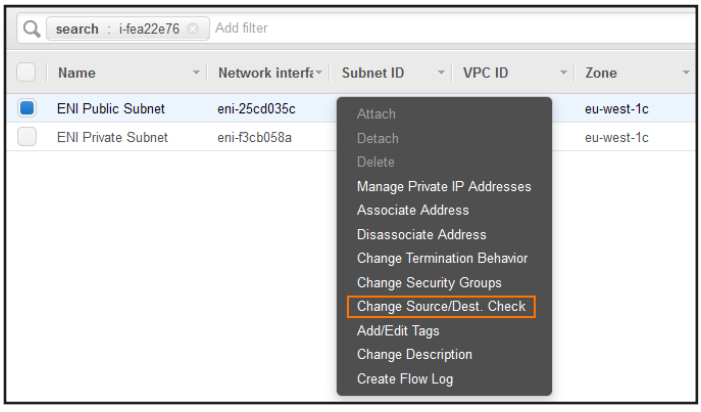
On the **Launch Status** page, locate and copy the **Instance IDs**. This is the default password used to log in via NextGen Admin.



Step 7. Disable Source/Destination Check for the Network Interface

For the interface to be allowed to forward traffic with a destination IP address that is different from the IP addresses assigned to the network interfaces, you must disable the source/destination check.

1. Log into the AWS console.
2. Click **Services** and select **EC2**.
3. In the **Network & Security** section of the left menu, click on **Network Interfaces**.
4. (optional) Filter the list using the Instance ID.
5. Right-click on the network interface, and select **Change Source/Dest. Check**.



a. Set the **Source/dest. check** to **Disabled**.

b. Click **Save**.

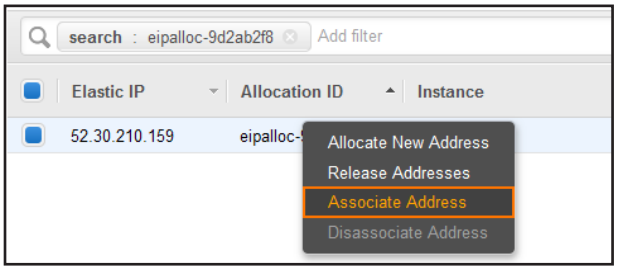
The source/destination check is now disabled for the network interface connected to the firewall instance.

Step 8. Associate the Elastic IP with the Firewall

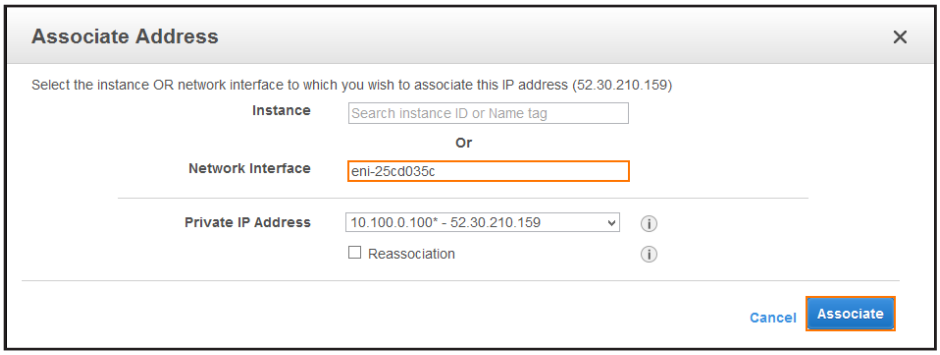
Use the Elastic IP (EIP) as the public IP address for the firewall network interface connected to the public subnet.

1. Log into the AWS console.
2. Click **Services** and select **EC2**.
3. In the **Network & Security** section of the left menu, click on **Network Interfaces**.
4. (optional) Filter the list using the Instance ID.
5. Locate the network interface connected to the public subnet, and copy the **Network interface ID**.

6. In the **Network & Security** section of the left menu, click on **Elastic IPs**.
7. Right-click the EIP created in step 2, and click **Associate Address**.



8. Enter the **Network Interface** ID, and click **Associate**.

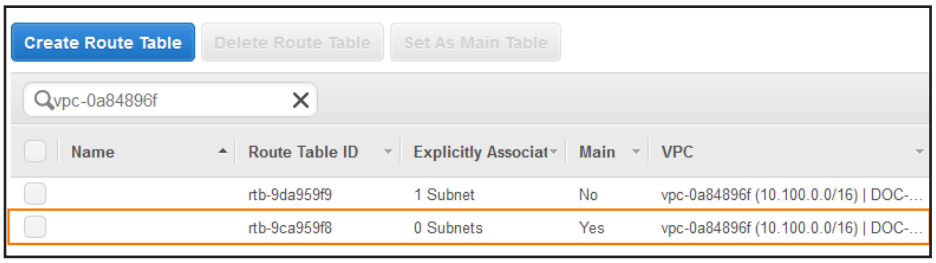


Traffic to the EIP is now automatically forwarded to the network interface attached to the public subnet of the VPC.

Step 9. Adjust the Routing Tables

Adjust the routing table for the private subnets to use the firewall instance as the default gateway. Instances will always use the first IP address of the subnet as the default gateway. The AWS cloud fabric then internally reroutes the traffic to the configured network interface or instance.The route table attached to the public subnet does not need to be changed.

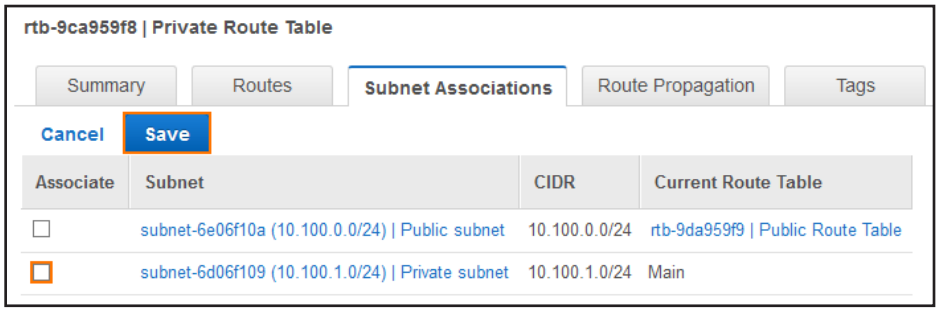
1. Log into the AWS console.
2. Click **Services** and select **VPC**
3. In the **Virtual Private Cloud** section of the left menu, click on **Route Tables**.
4. (optional) Filter the list using the VPC ID.
5. Select the route table that is not associated with the public subnet.



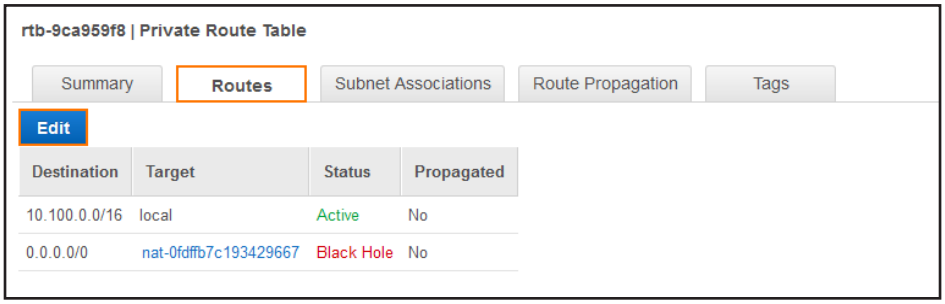
6. In the lower half of the page, click on the **Subnet Associations** tab.
7. Click **Edit**.

Select the private subnet and click **Save**.

If you are deploying with multiple network interfaces, you must create a route table for each private network. If you are using one network interface, associate all private subnets with this route table.



8. Click on the **Routes** tab.
9. Click **Edit**.



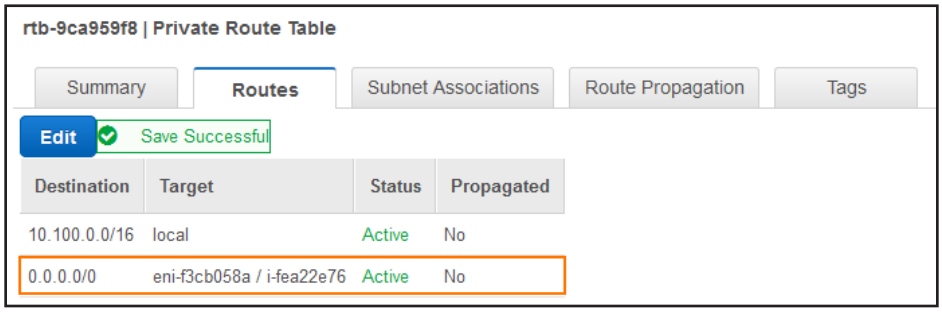
10. Depending on whether you are using single or multiple network interfaces:

    a. **Single NIC** – Enter the Instance ID of the firewall in the **Target** column of the route with the **Destination** 0.0.0.0/0.

    b. **Multiple NICs** – Enter the network interface ID of the network interface associated with this subnet in the **Target** column of the route with the **Destination** 0.0.0.0/0.

11. Click **Save**:

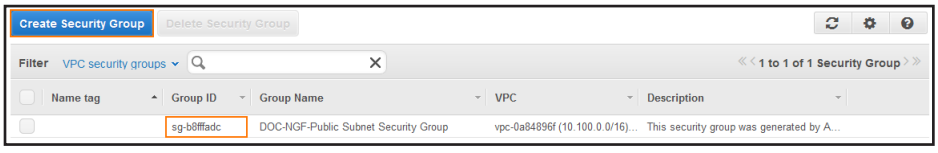
You now have a default route with the **Status** active and the target set to the correct firewall network interface.



Step 10. Create a Security Group

Create a security group for the private networks that allow all traffic from the security group assigned to the firewall.

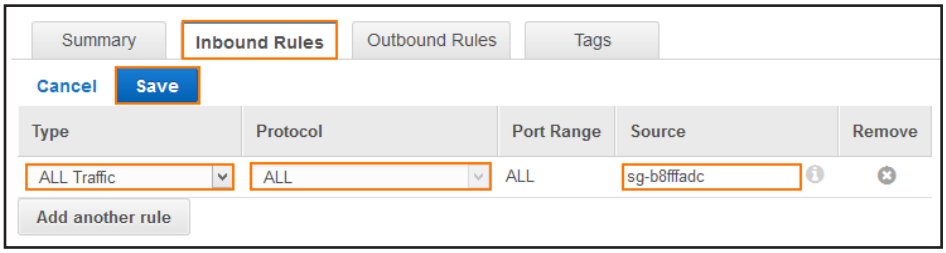
1. Log into the AWS console.
2. Click **Services** and select **VPC**
3. In the **Security** section of the left menu, click on **Security Groups**.
4. Locate the security group created during the firewall deployment, and copy the **Group ID**.



5. Click **Create Security Group**.

- **Group name** – Enter a name for the security group.
- **Description** – Enter a description for the security group.
- **VPC** – Select the VPC you created in step 3 from the list.

6. Click **Yes, Create**.
7. In the lower half of the page, click on the **Inbound Rules** tab.
8. Click **Edit**.
9. Create a rule to allow traffic from the firewall security group:
   - **Type** – Select **All Traffic**.
   - **Protocol** – Select **ALL**.
   - **Source** – Enter the group ID of the security group assigned to your firewall.
10. Click **Save**.



When deploying Instances to one of the private subnets, use this security group. This will allow traffic to and from the firewall.

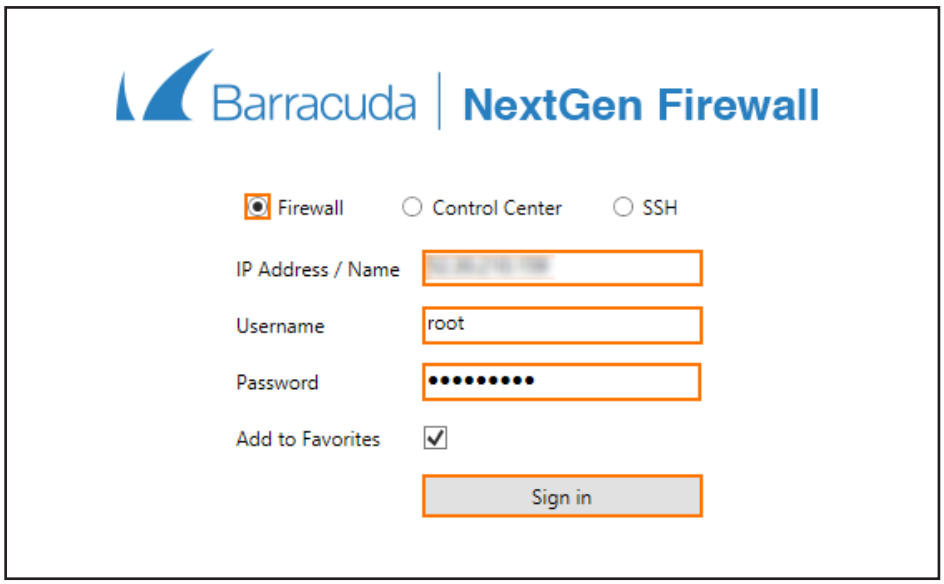
Step 11. (optional) Edit the Network ACLs

The Network ACLs created by the VPC wizard are configured by default to allow traffic through. If required, go **Network ACLs** to edit the network ACL assigned to your VPC.

Step 12. Log in via NextGen Admin

Use NextGen Admin to log into your firewall.

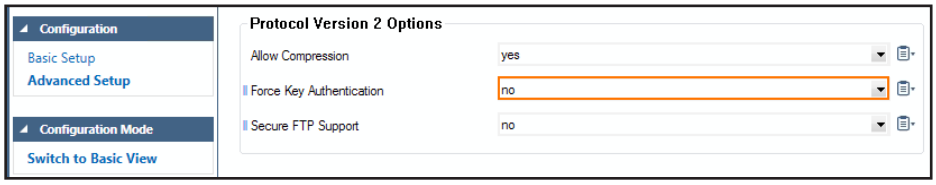
1. Launch NextGen Admin.
2. Log into the firewall:
   - Select **Firewall**.
   - **IP Address / Name** – Enter the elastic IP.
   - **Username** – Enter `root`.
   - **Password** – Enter the Instance ID of the firewall instance created in step 5.
3. Click **Sign in**.



3.11.1 Next Steps

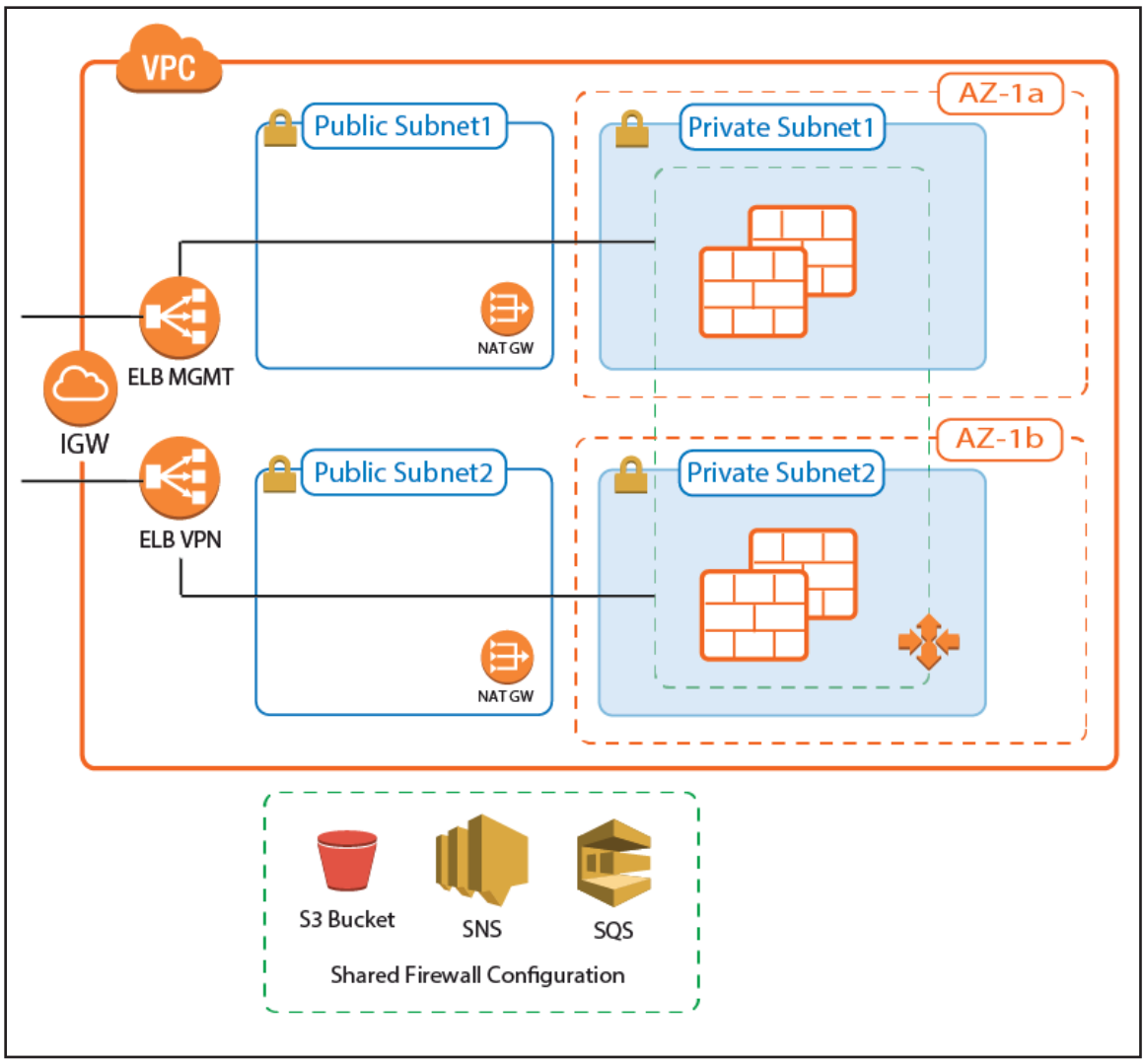
(BYOL only) License and activate the firewall. For more information, see How to Activate and License a Stand-alone Virtual or Public Cloud F-Series Firewall or Control Center.

(optional) Re-enable SSH logins via password by setting **Force Key Authentication** to **No** in the **Advanced View** of the **CONFIGURATION > Configuration Tree > Box > Advanced Configuration > SSH > Advanced Setup** page.



3.12 How to Deploy a NextGen Firewall Auto Scaling Cluster in AWS

A NextGen Firewall Auto Scaling cluster automatically scales with demand, thereby creating a cost-effective, robust solution for securing and connecting to your cloud resources. The firewall cluster integrates tightly with AWS services and APIs. Configuration changes are synchronized securely over the AWS backend, with all instances sharing the same configuration. For the admin, the firewall cluster handles like a single NextGen Firewall. The firewall cluster uses the PAYG image of the Barracuda NextGen Firewall in the AWS Marketplace to allow you to quickly deploy without the need for long-term licensing commitments. NextGen Firewall clusters cannot be managed by a NextGen Control Center. The following custom metrics are collected from the firewall cluster:



3.12.1 AWS Reference Architectures

This article is used in the following AWS reference architectures:

2.2 NextGen Firewall Auto Scaling Cluster(page 29)

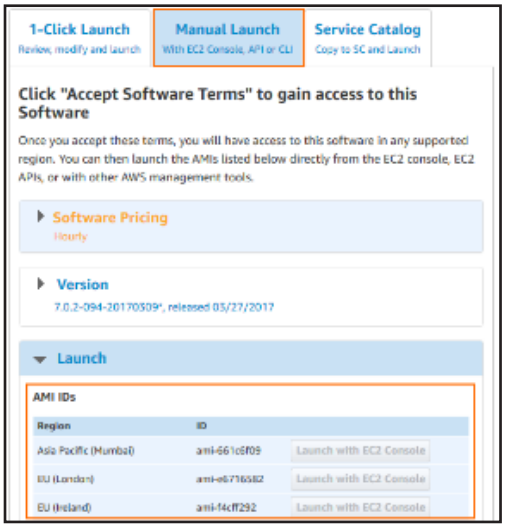
2.3 NextGen Firewall Cold Standby Cluster(page 49)

Before You Begin

Download the template from the Barracuda Network GitHub account:

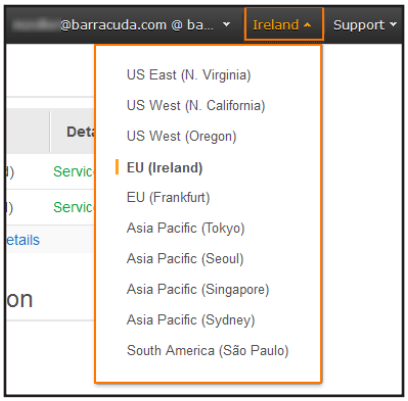
https://github.com/barracudanetworks/ngf-aws-templates.

- **NextGen Firewall Auto Scaling Cluster** – Download **autoscale.json**
- **NextGen Firewall Cold Standby Cluster** – Download **coldstandby.json**
- Verify that the AMI image IDs used in the CloudFormation template match the IDs for the NextGen Firewall image listed in the AWS Marketplace. The AMI disk images change for every released version and differ for each region.



Step 1. Select the AWS Datacenter

1. Log into the AWS console.
2. In the upper right, click the datacenter location, and select the datacenter you want to deploy to from the list.



The selected datacenter location is now displayed in the AWS console.

Step 2. Create an IAM Role for the Firewall

Create an IAM Role to allow the firewall instances to make the required API calls.

For more information, see 3.1 How to Create an IAM Role for an F-Series Firewall in AWS(page 79)

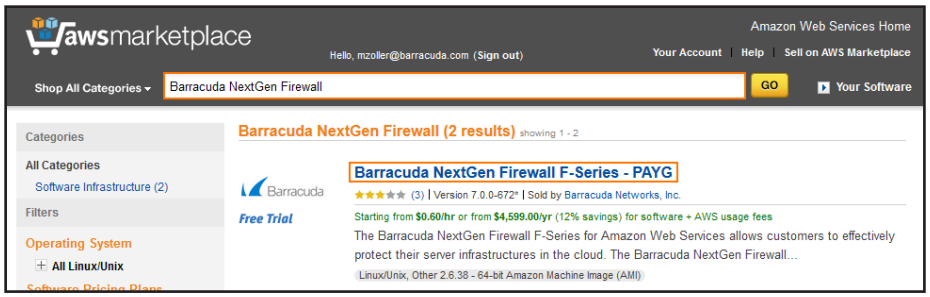
Step 3. Subscribe to Barracuda NextGen Firewall F-Series PAYG AMI in AWS Marketplace

To be able to deploy a NextGen Firewall PAYG image via the CloudFormation template, you must agree to the **Terms of Service** and subscribe to the image in the AWS Marketplace. You need to do this only once per account,

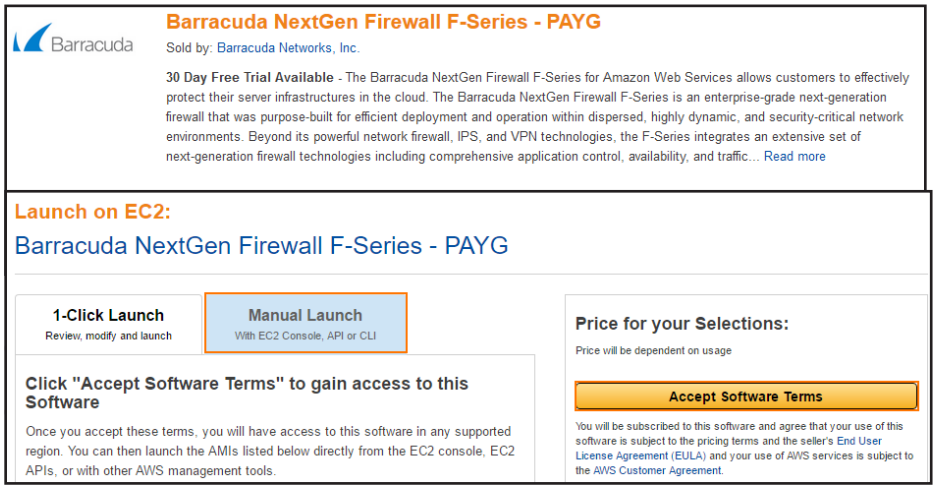
Go to the AWS Marketplace: https://aws.amazon.com/marketplace/

Search for `Barracuda NextGen Firewall.`

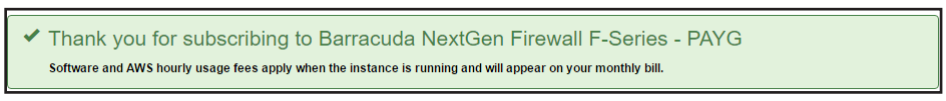
Click the **Barracuda NextGen Firewall F-Series PAYG** or **Barracuda NextGen Firewall F-Series BYOL** image.



Click **Continue**.



You will now receive an email from Amazon confirming your subscription. You can now use the provided AMI in your CloudFormation templates.

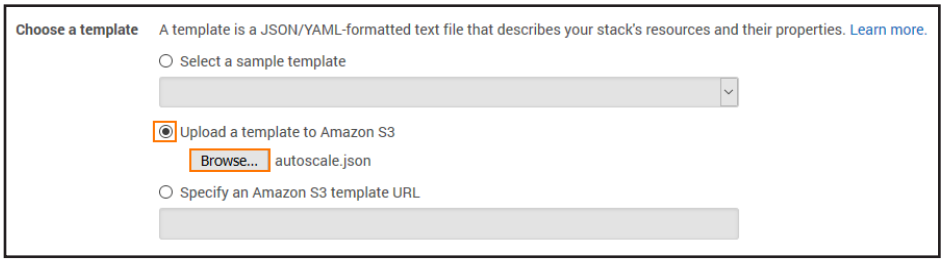


Step 4. Deploy the CloudFormation Template

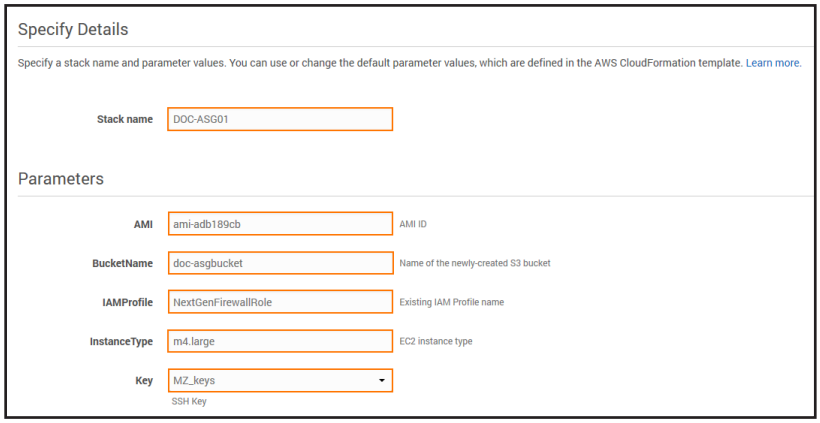
CloudFormation templates can be deployed via the AWS web console, CLI, REST, or PowerShell.

1. Log into the AWS console.
2. Click **Services** and select **CloudFormation**.
3. Click **Create Stack**

4. Select **Upload a template to Amazon S3**.
5. Click **Browse** and select the template file.



6. Click **Next**.
7. Enter the **Stack name**.
8. Fill in the template **Parameters**.
   - **Stack Name** – Enter a name.
   - **AMI** – Enter the ID for the Barracuda NextGen Firewall PAYG AMI for your AWS region.
   - **BucketName** – Enter the name for the S3 bucket used to store the firewall configuration.
   - **IAMProfile** – Enter the IAM role created for the NextGen Firewall.
   - **InstanceType** – Enter a supported instance type. Default `m4.large`.
   - **Key** – Select the key pair from the list. You must have access to the private key of the selected key pair to log in via SSH.

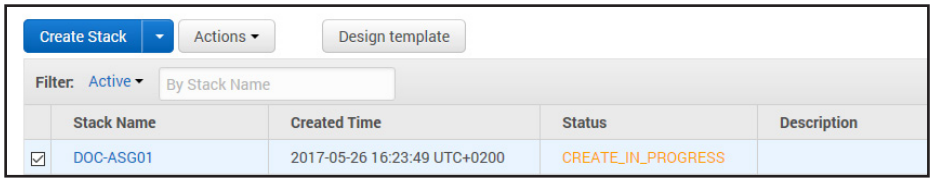


9. Click **Next**.
10. (optional) Enter **Tags** for your stack.
11. In the **Advanced** section, set additional options for your stack:
    - **Notification options**
    - **Timeout** – Set the timeout in minutes.

Rollback on failure – When set to **yes**, the deployment will be rolled back if any errors are encountered.

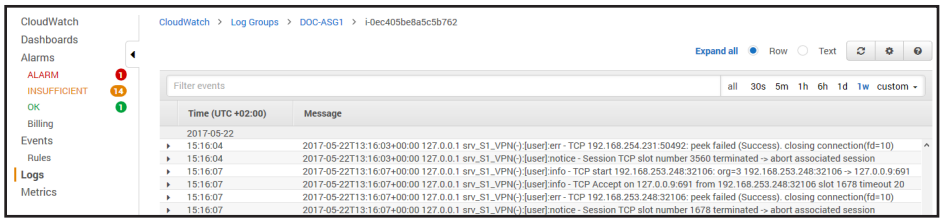
12. Click **Next**.
13. Review the settings and click **Create**.

The resources defined in the template are now deployed. This may take a couple of minutes. When the **Status** column shows **CREATE_COMPLETE**, the template has been deployed successfully. If the firewall fetches a PAR file from a Control Center, it may take a couple of minutes for the firewall to be available.



Step 5. Configure Log Streaming to AWS CloudWatch

Log files are generated and stored on each firewall instance in the Auto Scaling Group. To aggregate and store the log files generated on the firewall cluster, configure the NextGen Firewall cluster to stream all logs to AWS CloudWatch.



For more information, see 3.2 How to Configure Log Streaming to AWS CloudWatch(page 87)

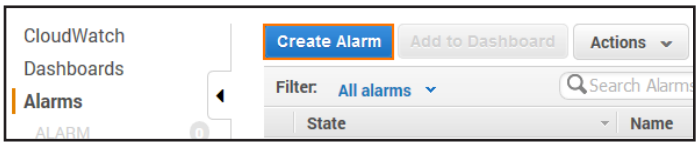
3.13 How to Configure Scaling Policies for a NextGen Firewall Auto Scaling Cluster

Scaling policies are required for the firewall cluster to adjust the capacity in response to changes in demand. Define CloudWatch alarms for the high and low thresholds. Use the custom metrics collected from the firewall cluster or the default EC2 system metrics. Add scaling policies to the Auto Scaling group that trigger a scaling action when the health check is in alarm state.

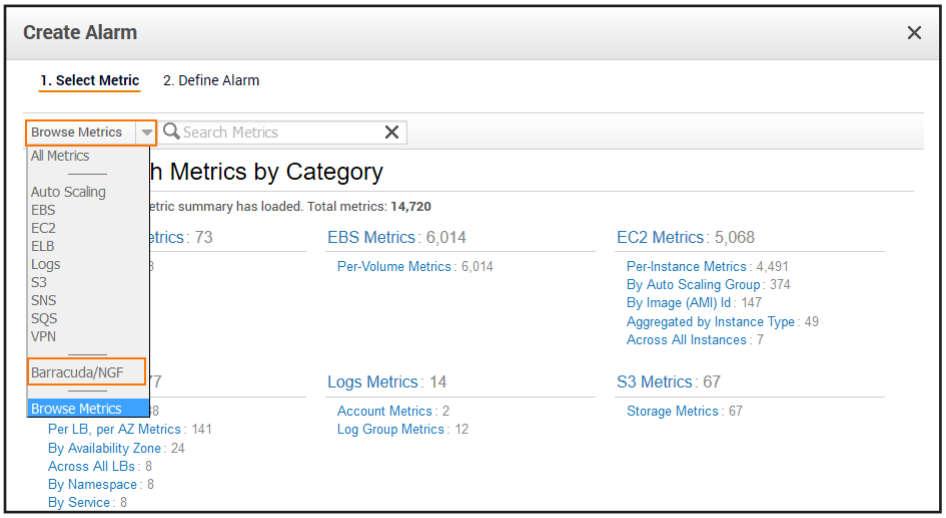
Step 1. Create CloudWatch Alarm

Create two CloudWatch alarms, one for the high and one for the low alarm threshold.

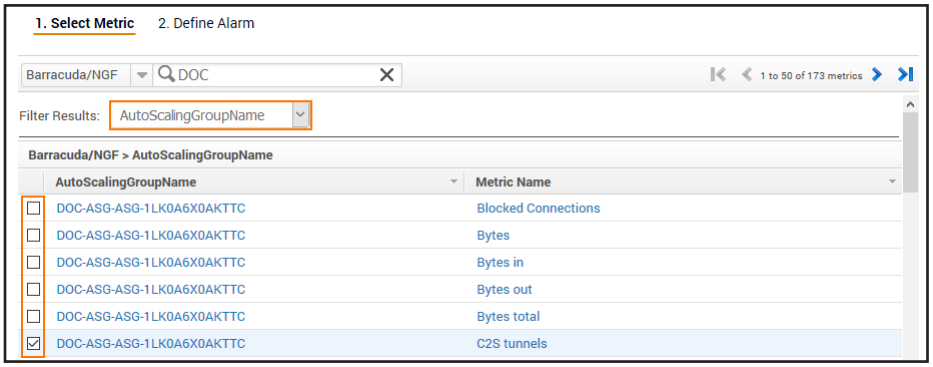
1. Log into the AWS console.
2. Click **Services** and select **CloudWatch**.
3. In the left menu, click **Alarms**.
4. Click **Create Alarm**.



5. From the **Browse Metrics** drop-down list, select **Barracuda/NGF**.



6. From the **Filter Results** drop-down list, select **AutoScalingGroupName**.
7. Select the check box for the metric.

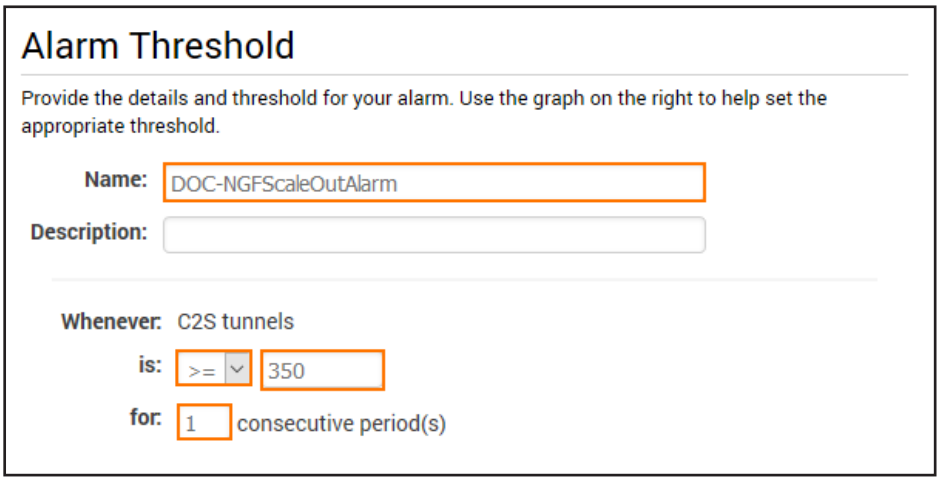


8. Click **Next**.

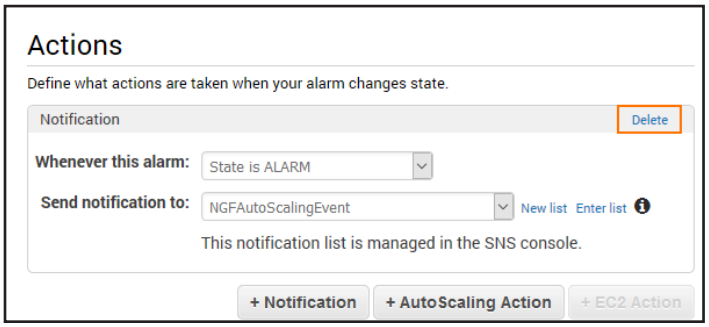
9. Enter a **Name**.

10. Configure the **Alarm Threshold**:

- **Logic operator** – Select **>=** when defining an alarm to scale out, **<=** when defining and alarm to scale in.
- **Alarm threshold** – Depending on the instance and metric type, enter the threshold. If unsure, use CloudWatch to monitor your cluster under load to determine the correct value to match your workload.
- **Period** – Enter the time period the threshold must be exceeded for alarm to be triggered.

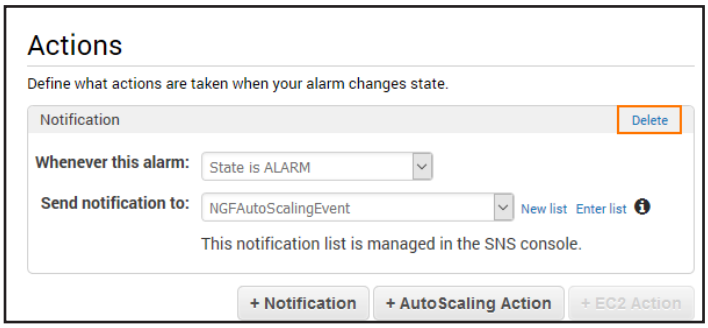


11. In the **Alarms section**, click **delete** to not receive a notification when the alarm is triggered. Alternatively, select an SNS topic that is configured to send notification emails when the alarm is triggered.



12. From the **Period** drop-down list, select the number of minutes.

13. From the Statistics drop-down list, select **Average** or **Sum** depending on the metric.

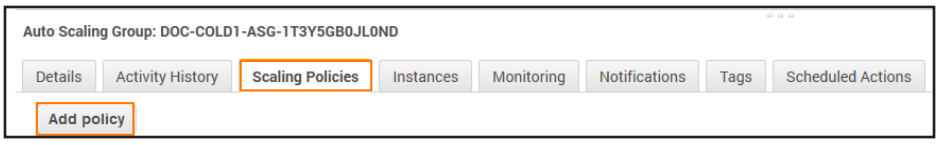


14. Click **Create Alarm**.

The alarm is in the **INSUFFICIENT** state until there is enough data for the alarm. As soon as enough data is available, the alarm state changes to **OK** or **Alarm**.

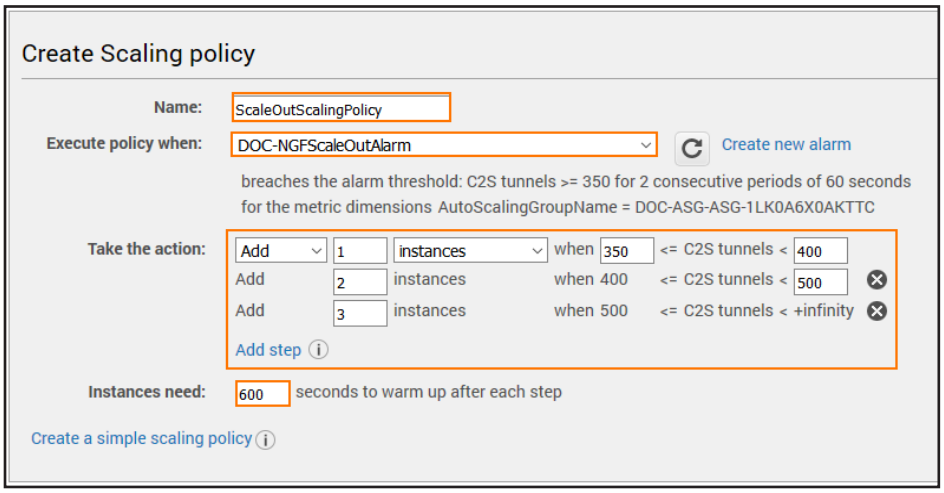
Step 2. Add Scaling Policy to Scale Out

1. Log into the AWS console.
2. Click **Services** and select **EC2**.
3. In the left menu, click **Auto Scaling Groups**.
4. Select the NextGen Firewall Auto Scaling group.
5. In the lower half, click the **Scaling Policies** tab.
6. Click **Add policy**.



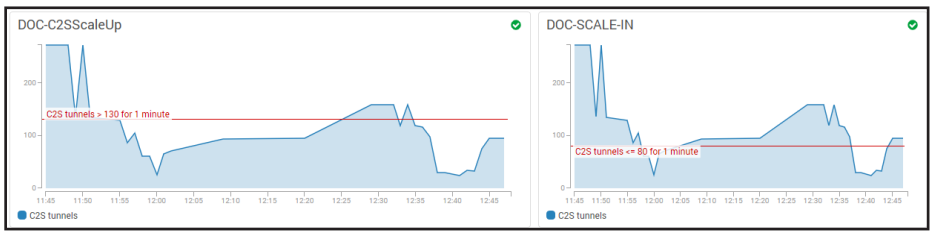
7. Enter a **Name**.
8. From the **Execute policy when** drop-down list, select the matching CloudWatch alarm created in Step 1.
9. Configure the action:
   - **Action** – Select **add** to scale out, or **Remove** to scale in. Click **set** to use an explicit number of instances.
   - **Number of instances** – Depending on the action, enter the number of instances to scale (add / remove) or the number of instances to scale to (set).

10. (optional) Click **add steps** to define a more granular scaling policy that takes into account by how much the threshold is exceeded.
11. In the **Instances need** text box, enter the number of seconds to wait before the next scaling action.



12. Click **Create**.

Repeat this for both Scale In and Scale Out policies. Use CloudWatch dashboard widgets to visualize the alarm thresholds.



3.14 How to Configure an AWS Elastic Load Balancer for F-Series Firewalls in AWS

The Elastic Load Balancer is a manged layer 4 load balancer by AWS. The ELB can be deployed as a public-facing load balancer or internally in your VPC. Instances are added either manually or, if associated with an Auto Scaling group, automatically. The load balancer continuously checks the health of the instances and takes unhealthy instances out of rotation. By enabling cross-zone loadbalancing, the load balancer spreads out the load evenly over multiple availability zones.

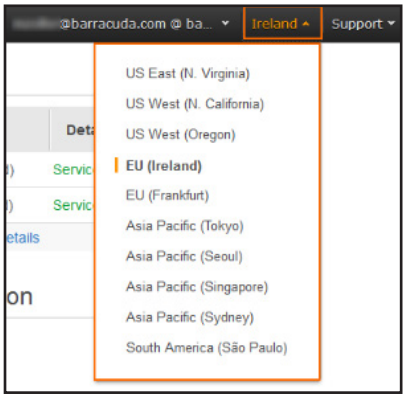
3.14.1 AWS Reference Architectures

This article is used in the following AWS reference architectures:

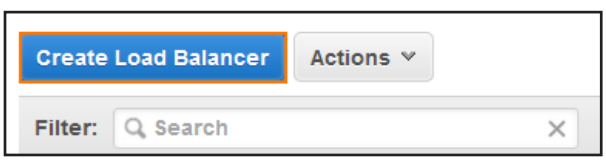
2.1 NextGen Firewall High Availability Cluster with Route Shifting(page 21)

3.14.2 Create an AWS Load Balancer

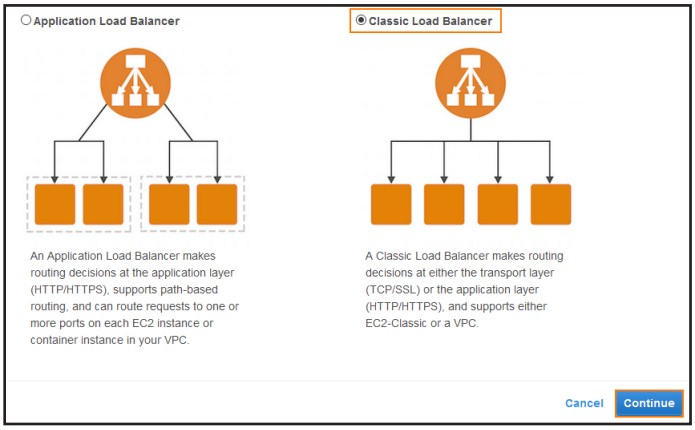
1. Log into the AWS console.
2. In the upper right, click on the datacenter location, and select the datacenter you want to deploy to from the list.



3. Log into the AWS console.
4. Click **Services** and select **EC2**.
5. In the **Load Balancing** section of the left menu, click **Load Balancer**.
6. Click **Create Load Balancer**.

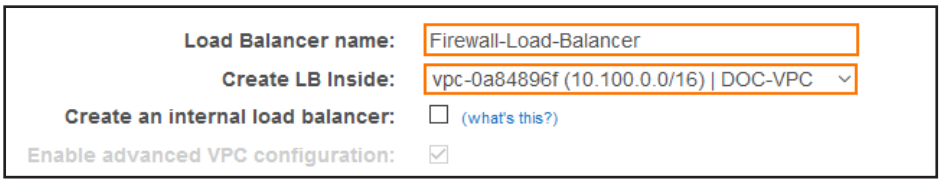


7. Select **Classic Load Balancer** and click **Continue**.



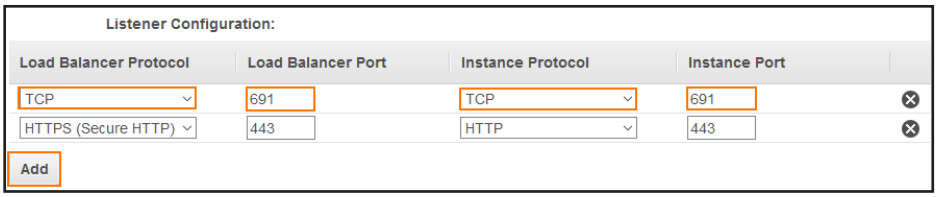
8. Enter the **Basic Configuration Settings**:

   - **Load Balancer name** – Enter name for the load balancer.
   - **Create LB inside** – Select the VPC the firewalls are deployed to from the list.
   - **Create an internal load balancer** – Select the check box to create an internal load balancer. Internal load balancers are reachable from within the VPC and do not have a public IP address.

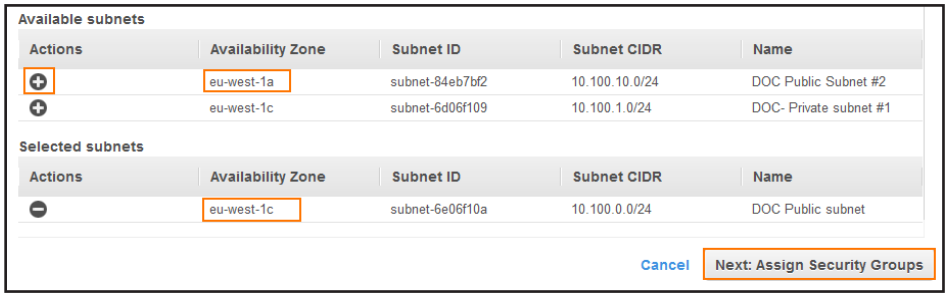


9. For each **Listener**, click **Add** and enter:

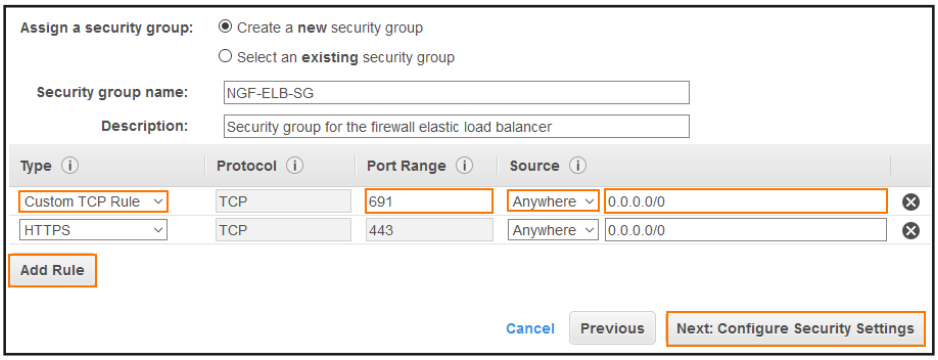
   - **Load Balancer Protocol** – Select the protocol from the list. Supported protocols: **TCP**, **HTTP**, **HTTPS**, **SSL (Secure TCP)**.
   - **Load Balancer Port** – Enter the external port.
   - **Instance Protocol** – Enter the protocol. In most cases, this is the same protocol as the **Load Balancer Protocol**. To offload SSL encryption to the ELB, different protocols can be selected (e.g, HTTPS to HTTP).
   - **Instance Port** – Enter the port number of the service on the instance.



10. Click **+** in the **Actions** column to add subnets to the load balancer. Add the subnets containing the firewall instances. Each subnet should be in a different Availability Zone.

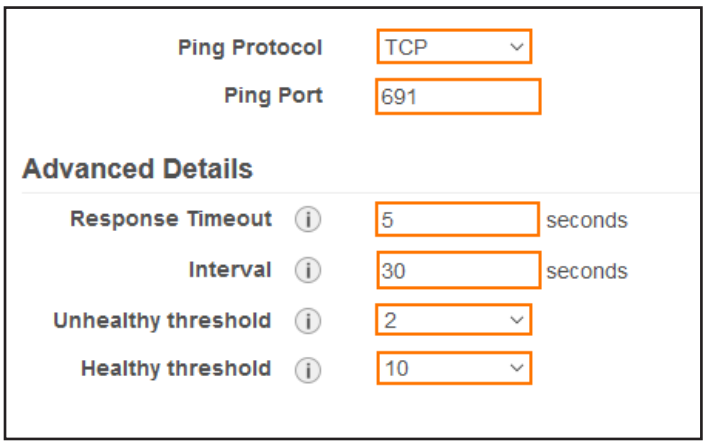


11. Click **Next: Assign Security Groups**.
12. Click **Create new security group**.
13. For each load balancer listener, create a **Rule**. Click **Add Rule** for each additional security group rule required.
    - **Type** – Select the protocol or type of traffic. e.g., **Custom TCP Rule** for TCP, or **HTTPS** for SSL-encrypted web traffic.
    - **Port Range** – Enter the port. e.g., 691 for TINA VPN
    - **Source** – Select the source of the traffic. For Internet traffic, select **Anywhere** and enter `0.0.0.0/0.`



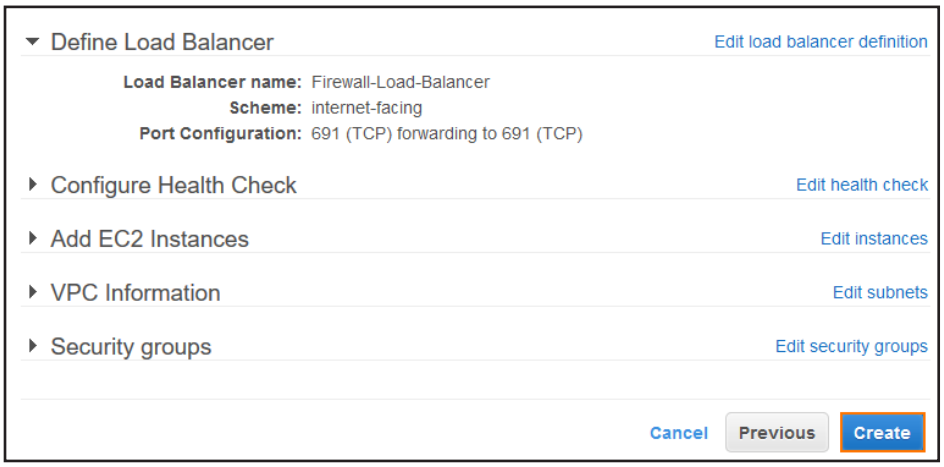
14. Configure the **Health Check**.
    - **Ping Protocol** – Select the protocol from the list.
    - **Ping Port** – Enter the port. e.g, 691 for TINA VPN, or 443 for HTTPS
    - **Response Timeout** – Enter the number of seconds the probe waits for an answer.
    - **Interval** – Enter the number of seconds between two probes.
    - **Unhealthy threshold** – Enter the number of failed heath checks for the instance to be considered unhealthy.

      Unhealthy health checks are taken out of rotation until healthy again.
    - **Healthy threshold** – Enter the the number of successful heath checks for the instance to be considered healthy.



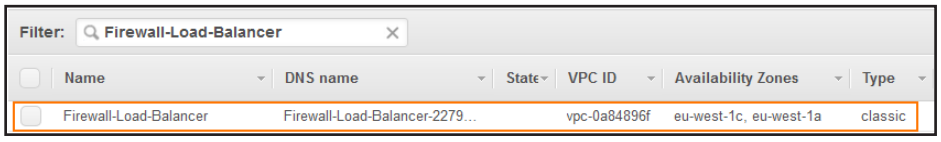
15. Click **Next: Add EC2 Instances**.
16. (optional) If the firewall EC2 instances are already deployed, select the EC2 instances.
17. Select **Enable Cross-Zone Load Balancing**.

18. Click **Next: Add Tags**.
19. (optional) Add **Key** / **Value** tags to the resource. Click **Create Tag** to add additional tags.
20. Click **Review and Create**.



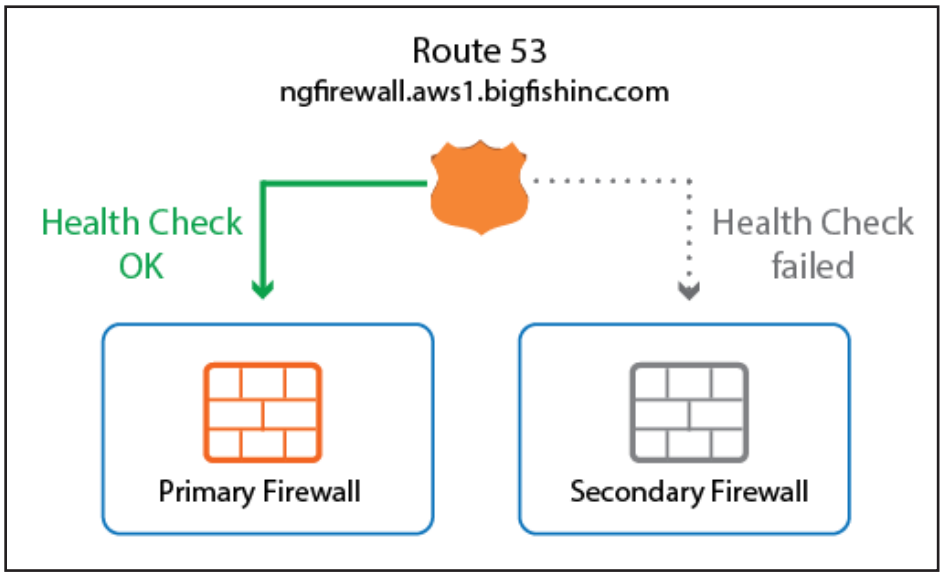
21. Review the settings and click **Create**.

The Elastic Load Balancer is now deployed and ready for use.



3.15 How to Configure Route 53 for F-Series Firewalls in AWS

If you are running multiple stacks in different AWS regions, or multiple deployments in a single region, you must configure AWS Route 53 to access your services behind the NextGen Firewalls. Also use Route 53 if you are using UDP-based services since the Elastic Load Balancer supports only TCP connections. To always route traffic to the active firewall in the HA cluster, define two record sets with a failover policy. The record set for the first firewall is combined with a health check. As long as the health check is valid, the DNS name for the firewall is resolved to the primary firewall. When the virtual server fails over to the secondary firewall, the health check for the primary firewall fails, and after the TTL of the DNS record has expired, the DNS name for the firewall cluster resolves to the IP address in the secondary record set. When the primary firewall is active again, the health check will again show a healthy state and the DNS record will point to the IP address of the primary firewall.



3.15.1 Alternative

If you are not using Elastic IP addresses for your firewalls, you can also use the DNS name of the firewall for the health check and create a CNAME DNS record.

3.15.2 Before You Begin

- Set up a domain or subdomain in Route 53 and create a public hosted zone.

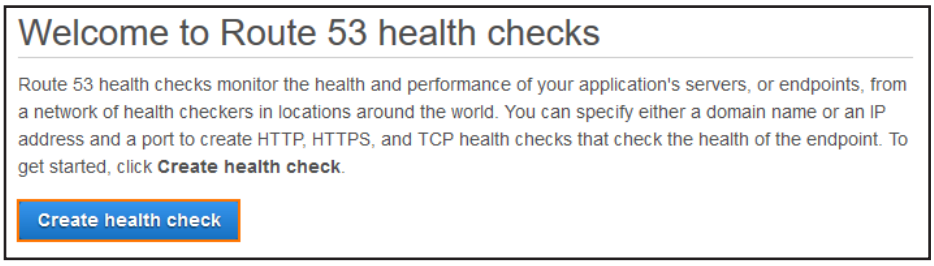
Deploy a multi-AZ high availability cluster. For more information, see How to Configure a Multi-AZ High Availability Cluster in AWS using the Web Portal.

Look up the DNS names, and public or Elastic IP address for the primary and secondary firewalls.

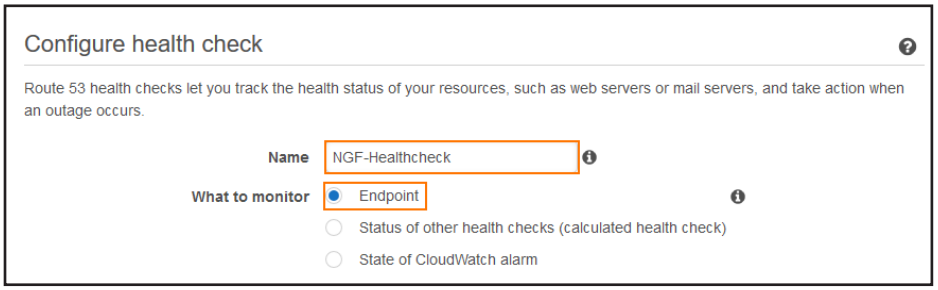
Step 1. Create a Route 53 Heath Check for the Primary Firewall

Configure the health check for a service running on the virtual server, such as the VPN service. Do not create a check for box-level services because these services will not fail over to the secondary firewall.

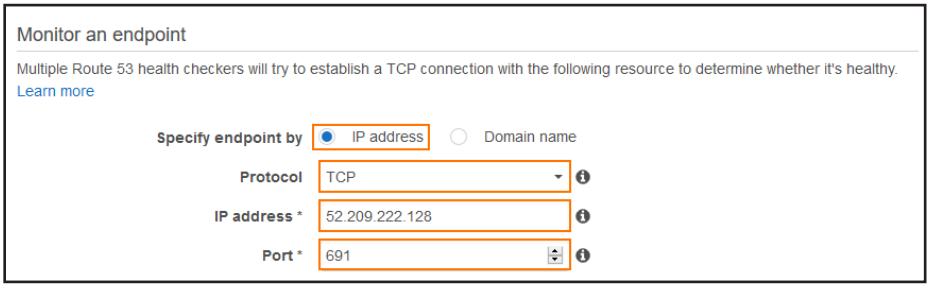
1. Log into the AWS console.
2. Click **Services** and select **Route 53**.
3. In the left menu, click **Health checks**.
4. Click **Create health check**.



5. Enter a **Name**.
6. From the **What to monitor** list select **Endpoint**.

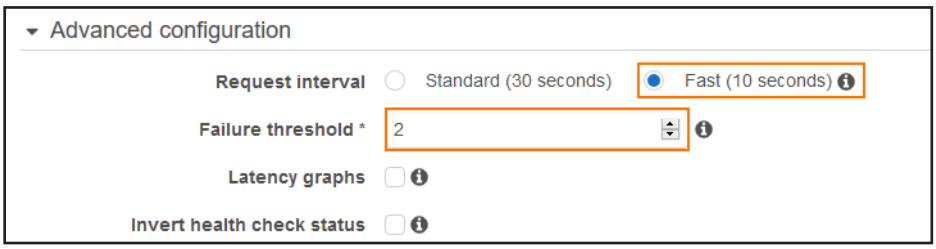


7. Configure the service to be monitored:
   - **Specify and endpoint by** – Select **IP address**.
   - **Protocol** – Select **TCP**.
   - **IP address** – Enter the public IP address for the primary firewall.
   - **Port** – Enter `691` to monitor the VPN service. The VPN service must be running on your virtual server. Alternatively, you can also select another port on your firewall.

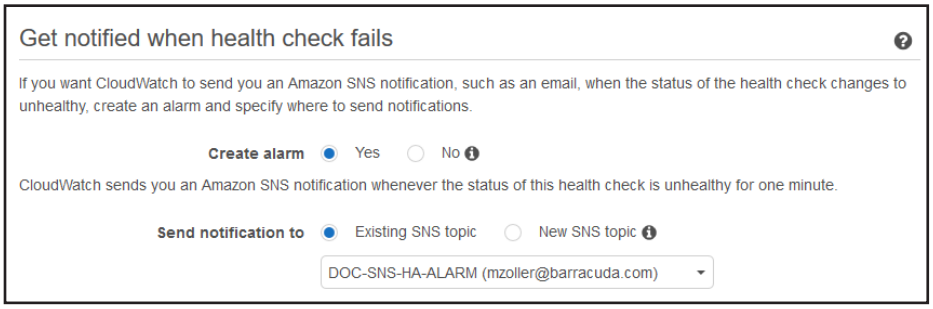


8. (optional) Expand the **Advanced configuration** section and adjust the following settings to improve failover times:

- **Request interval** – Select **Fast (10 seconds)**.
- **Failure threshold** – Select **2**.

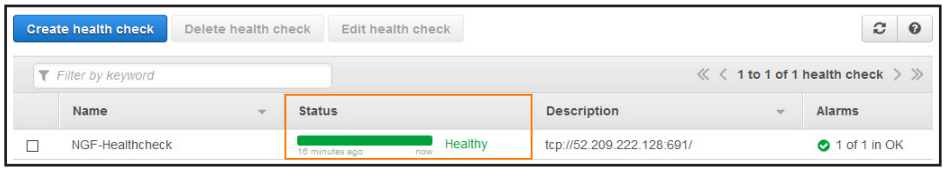


9. Click **Next**.
10. (optional) Set **Create alarm** to **yes** and select an **Existing SNS topic** or create a **New SNS topic** to receive a notification.



11. Click **Create health check**.

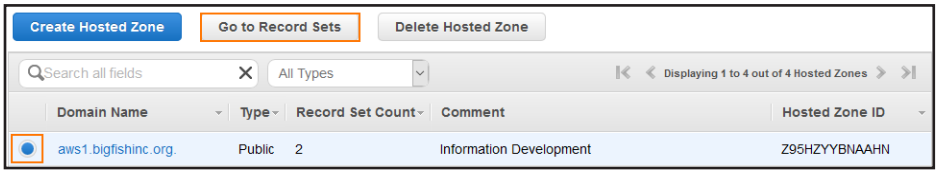
The health check is now active. Depending on the request interval and failover threshold, the **Status** of the health check changes from **Unknown** to **Healthy**.



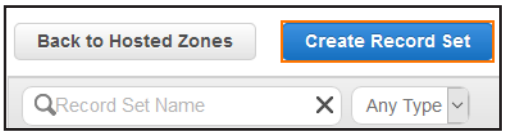
Step 2. Create a Failover Record Set for the Primary Firewall

Create the DNS record for the primary firewall. Use a **failover** routing policy and add the health check you just created as a condition.

1. Log into the AWS console.
2. Click **Services** and select **Route 53**.
3. In the left menu, click **Hosted zones**.
4. Select your **Domain Name** and click **Go to Record Sets**.

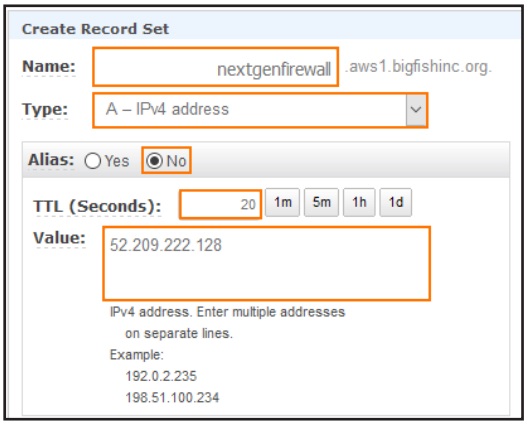


5. Click **Create Record Set**.



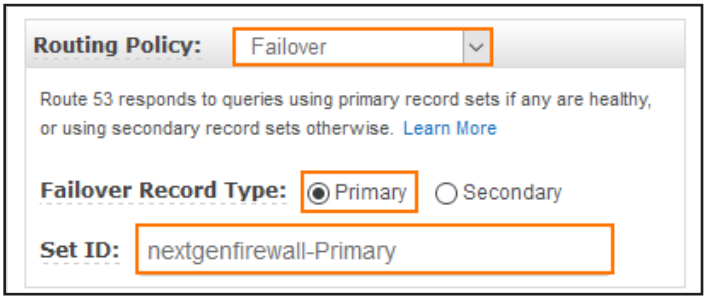
6. In the right column, create the record set:

   - **Name** – Enter the DNS name.
   - **Type** – Select **A - IPv4 address**.
   - **Alias** – Select **No**.
   - **TTL (Seconds)** – Set the number of seconds the DNS records can be cached by non-authoritative DNS servers.
   - **Value** – Enter the EIP or public IP address for the primary firewall.



7. In the right column, configure the **Routing Policy**:

   - **Routing Policy** – Select **Failover**.
   - **Failover Record Type** – Select **Primary**.
   - **Set ID** – Enter a unique ID to differentiate from other failover record sets using the same name and type.



8. In the right column, configure the **Health Check**:
   - **Associate with Heath Check** – Select **yes**.
   - **Health Check to Associate** – Select the health check created in step 1.

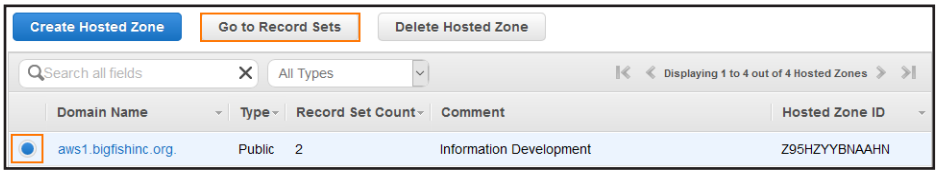
9. Click **Create**.

The record set for the primary firewall is now listed with the other DNS records of this hosted zone.

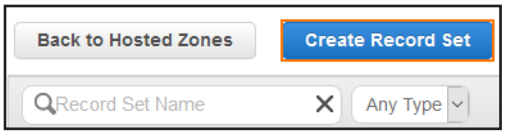
Step 3. Create a Failover Record Set for the Secondary Firewall

Create the DNS record for the secondary firewall. Use a **failover** routing policy.

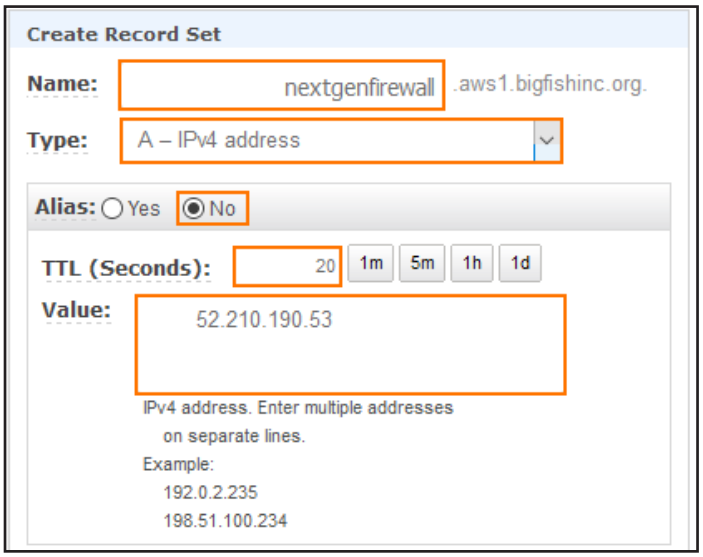
1. Log into the AWS console.
2. Click **Services** and select **Route 53**.
3. In the left menu, click **Hosted zones**.
4. Select your **Domain Name** and click **Go to Record Sets**.



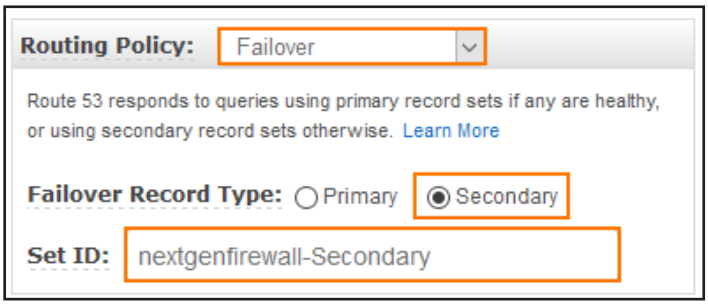
5. Click **Create Record Set**.



6. In the right column, create the record set:
   - **Name** – Enter the DNS name you used for the primary firewall.
   - **Type** – Select **A - IPv4 address**.
   - **Alias** – Select **No**.
   - **TTL (Seconds)** – Set the number of seconds the DNS records can be cached by non-authoritative DNS servers.
   - **Value** – Enter the EIP or public IP address for the secondary firewall.



7. In the right column, configure the **Routing Policy**:
   - **Routing Policy** – Select **Failover**.
   - **Failover Record Type** – Select **Secondary**.
   - **Set ID** – Enter a unique ID to differentiate from other failover record sets using the same name and type.

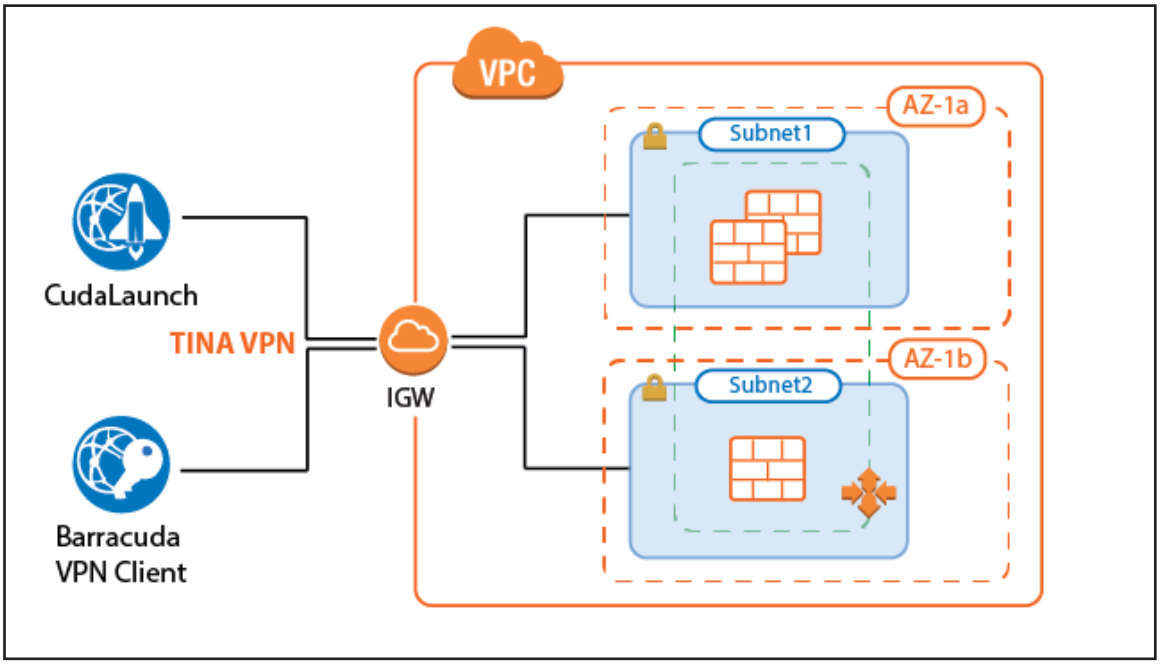


8. In the right column, configure the **Health Check**:
   - **Associate with Health Check** – Select **No**.
9. Click **Create**.

Both record sets for the primary and secondary firewalls are now listed in the hosted zone.

3.16 How to Configure a Client-to-Site VPN Group Policy for a NextGen Firewall Auto Scaling Cluster in AWS

Create a client-to-site group policy for remote users connecting to your network in a NextGen Firewall Auto Scaling Cluster in AWS. Configure a VPN client network, create the policy, and configure the network settings for the client-to-site connections. Then, create a Source NAT access rule to allow the clients to connect to your network. VPN clients can be authenticated either through external authentication schemes, client certificates, or a combination thereof.



3.16.1 Supported Clients

- Barracuda VPN Client for Windows, macOS, Linux, and OpenBSD
- CudaLaunch for Windows, macOS, and Android. A CudaLaunch version for iOS with support for NextGen Firewall clusters is coming soon.

3.16.2 Before You Begin

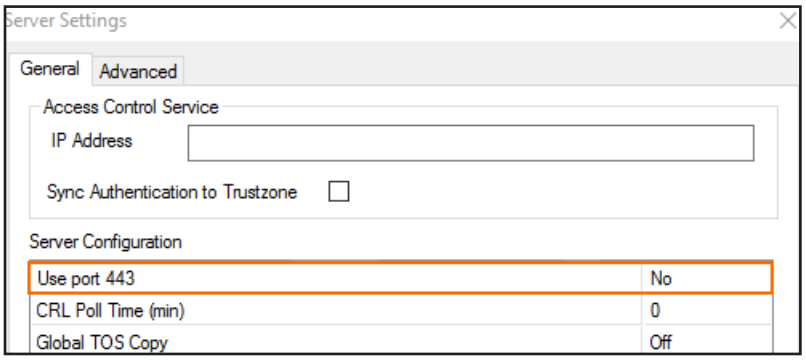
Set up the VPN certificates for External CA or Barracuda VPN CA. For more information, see How to Set Up External CA VPN Certificates, or How to Set Up Barracuda VPN CA VPN Certificates.

Configure an authentication scheme. For more information, see Authentication.

Step 1. Disable Port 443 for Client-to-Site VPN

To use SSL VPN and client-to-site VPN simultaneously, the listener on port 443 for the VPN service must be disabled.

1. Go to **CONFIGURATION > Configuration Tree > Box > Virtual Servers >** `your virtual server` **> Assigned Services > VPN > VPN Settings**.
2. Click **Lock**.
3. Select **Click here for Server Settings**. The **Server Settings** window opens.
4. In the **Server Configuration** section, set **Use port 443** to **No**.



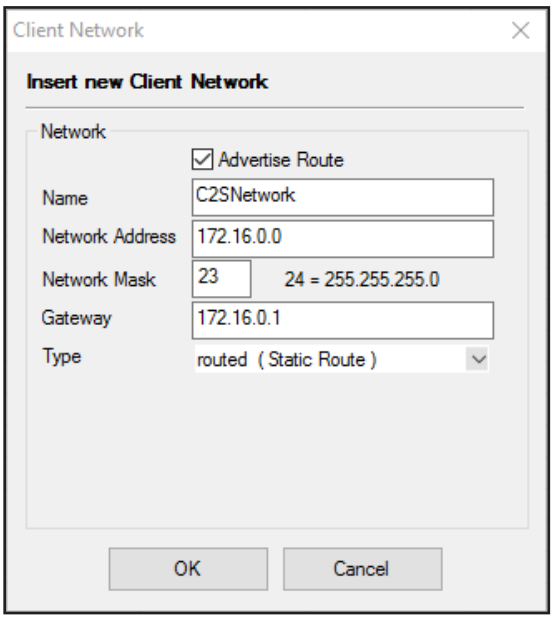
5. Click **OK**.
6. Click **Send Changes** and **Activate**.

Step 2. Configure the VPN Client Network

Configure the client network. When the VPN clients connect, they are assigned an IP address out of this network. Make sure to size the client-to-site network according to the number of client-to-site connections you are expecting to use on one instance of your Auto Scaling cluster. The source IP address for all connections from the VPN client network are rewritten to use the firewall's IP address.

1. Go to **CONFIGURATION > Configuration Tree > Box > Virtual Servers >** `your virtual server` **> Assigned Services > VPN > VPN Settings** .
2. Click **Lock**.
3. Click the **Client Networks** tab.
4. Right-click the table, and select **New Client Network**.
5. In the **Client Network** window, configure the following settings:
   - **Name** – Enter a descriptive name for the network.
   - **Network Address** – Enter the default network address, e.g.: `172.16.0.0`
   - **Network Mask** – Specify the appropriate subnet mask, e.g.: `23`
   - **Gateway** – Enter the gateway network address, e.g.: `172.16.0.1`

- **Type** – Select **routed (Static Route)**. A static route on the NextGen Firewall routes traffic between the VPN client subnet and the local network.

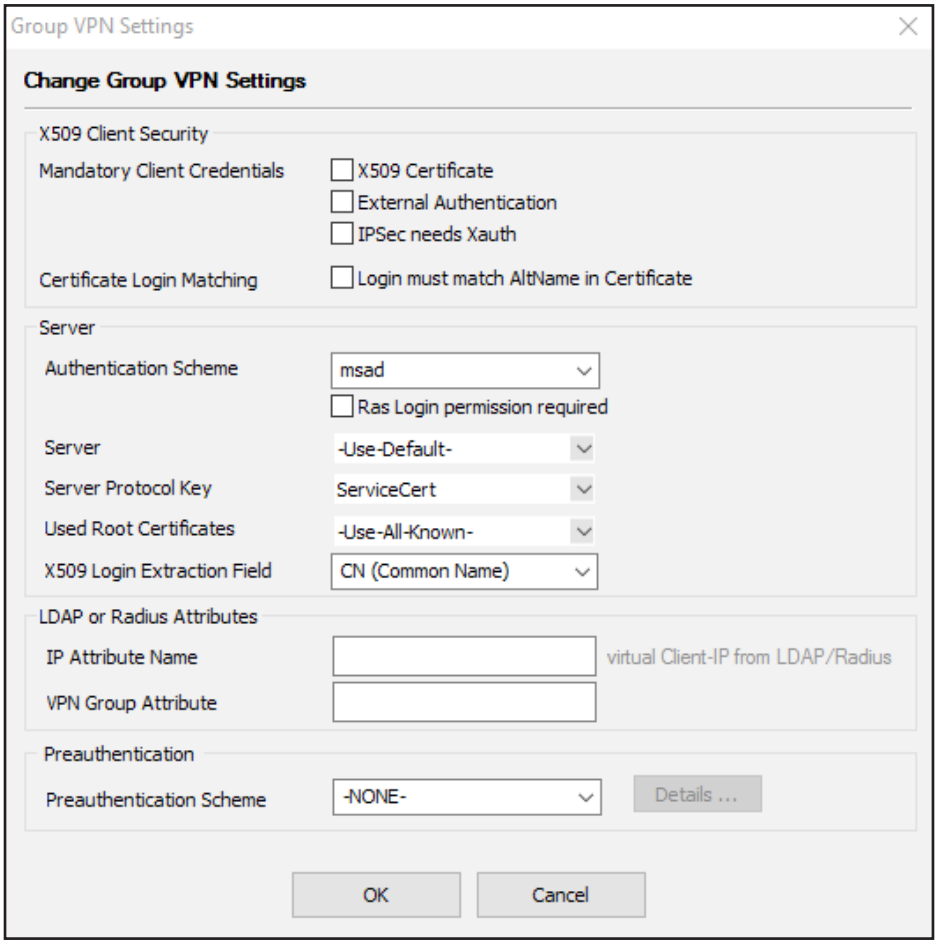


6. Click **OK**.
7. Click **Send Changes** and **Activate**.

Step 3. Configure Group Policy Settings

Configure the authentication setting for the client-to-site VPN. The firewall must have access to the authentication service.

1. Go to **CONFIGURATION > Configuration Tree > Box > Virtual Servers >** `your virtual server` **> Assigned Services > VPN > Client-to-Site**.
2. Click **Lock**.
3. Click the **External CA** tab and then the **Group Policy** tab.
4. Click the **Click here for options** link.
5. In the **Server** section of the **Group VPN Settings** window, select the **Authentication Scheme**. e.g., **msad**
6. Configure which certificates are used. By selecting a specific certificate, all VPN group policies must use this certificate:
   - **(optional) Server** – Select a server certificate, or use the default server certificate configured in the VPN settings.
   - **Server Protocol Key** – Select the service certificate.
   - **(optional) Used Root Certificates** – Select a root certificate, or use the default server certificate configured in the VPN settings.
   - **(optional) X509 Login Extraction Field** – Select the X509 field containing the user name.
7. (optional) If needed, select the **Preauthentication Scheme**.



8. Click **OK**.

Only X.509 certificate conditions can be assigned because IPsec XAUTH authentication will not work if group patterns are defined in the **External Group Condition** section.

Step 4. Create a VPN Group Policy

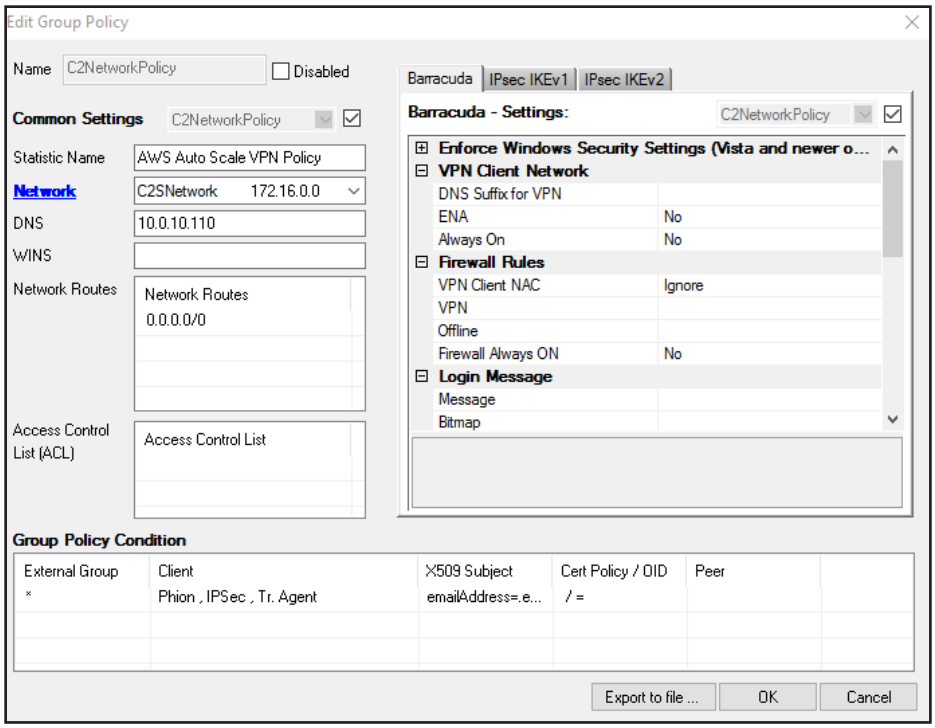
Create a group policy and configure the network settings for the client-to-site connections. If you want the client to send all traffic through the VPN tunnel, enter `0.0.0.0/0` as the network.

1. Go to **CONFIGURATION > Configuration Tree > Box > Virtual Servers >** `your virtual server` **> Assigned Services > VPN > Client-to-Site**.
2. Click the **External CA** tab and then click the **Group Policy** tab.
3. Right-click the table and select **New Group Policy**.
4. In the **Edit Group Policy** window, edit the following settings:
   - **Name** – Enter a name for this policy.
   - **Common Settings** – Select the check box.
   - **Statistics Name** – To better allocate statistics entries, enter a name.
   - **Network** – Select the required client network.

- **DNS** – Enter a DNS server for the clients.
- **Network Routes** – Add all networks that should be reachable by the VPN clients. Enter `0.0.0.0/0` for all traffic to be sent through the client-to-site VPN.

5. Right-click the **Group Policy Condition** field and select **New Rule**.
6. In the **X509 Certificate Conditions** section of the **Group Policy Condition** window, set filters for the certificate. For example, to let everyone with a valid certificate log on, click **Edit/Show** to add the following condition to the **Subject** field: `CN=*`

Certificate condition entries are case insensitive and can contain the quantification patterns **?** (zero or one) and ***** (zero or more).



7. Click **OK**.
8. Click **OK**.
9. Click **Send Changes** and **Activate**.

Step 5. (optional) Adjust Barracuda (TINA) Settings

1. Go to **CONFIGURATION > Configuration Tree > Box > Virtual Servers >** `your virtual server` **> Assigned Services > VPN-Service > Client-to-Site**.
2. Click **Lock**.
3. Click the **External CA** tab and then click the **Group Policy** tab.
4. Double-click the VPN group policy created in step 3.
5. In the **Barracuda** tab configure:

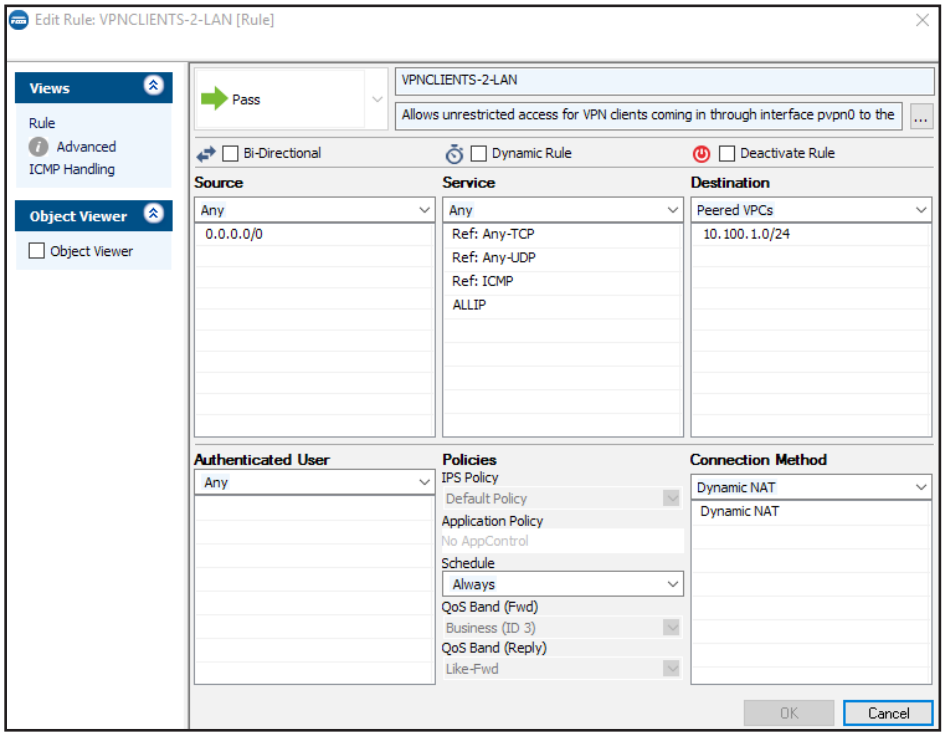
- **Windows Security Settings**
- **VPN Client Network**
- **Firewall Rules**
- **Login Message**
- **Ciphers**

6. Click **OK**.
7. Click **Send Changes** and **Activate**.

Step 6. Add Access Rules

For each service and/or destination network, create an access rule to allow traffic from the client VPN network to your AWS resources. The access rules must always use a **Dynamic NAT** or **Translated IP from DHCP** connection method.

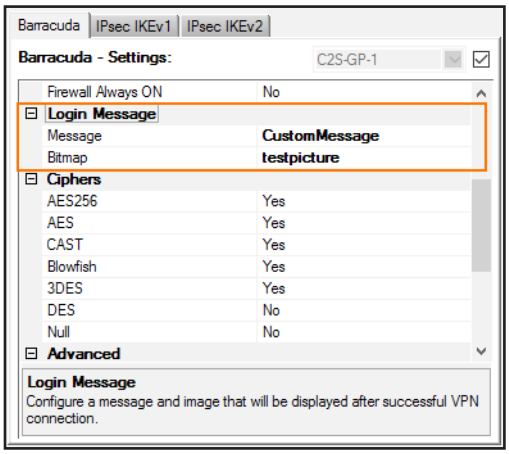
- **Action** – Select **Pass**.
- **Source** – Select **Any**.
- **Service** – Select the allowed services, or **Any** to allow all services.
- **Destination** – Select the network object containing the networks the VPN clients can access in AWS.
- **Connection Method** – Select **Dynamic NAT**.



3.16.3 Configure a Custom Login Message

When using a Barracuda VPN client, you can define a custom welcome messages as well as upload your company logo as a custom **Picture**. Custom message and pictures can be selected in the **Barracuda - Settings** of the VPN group policy.

- **Messages** – Create a custom message in the **Message** tab of the **Client-to-Site** page and then select the customized welcome message in the **Barracuda Settings** tab of the VPN group policies.
- **Bitmap/Pictures** – Upload a 150x80 pixel, 256 color BMP bitmap in the **Pictures** tab of the **Client-to-Site** page and then select the custom bitmap in the **Barracuda Settings** tab of the VPN group policies.



3.16.4 Troubleshooting

NextGen Admin only displays the logs on one firewall instance. To troubleshoot multiple client-to-site connections in an AWS Auto Scaling cluster, use CloudWatch.

For more information, see 3.2 How to Configure Log Streaming to AWS CloudWatch(page 87)

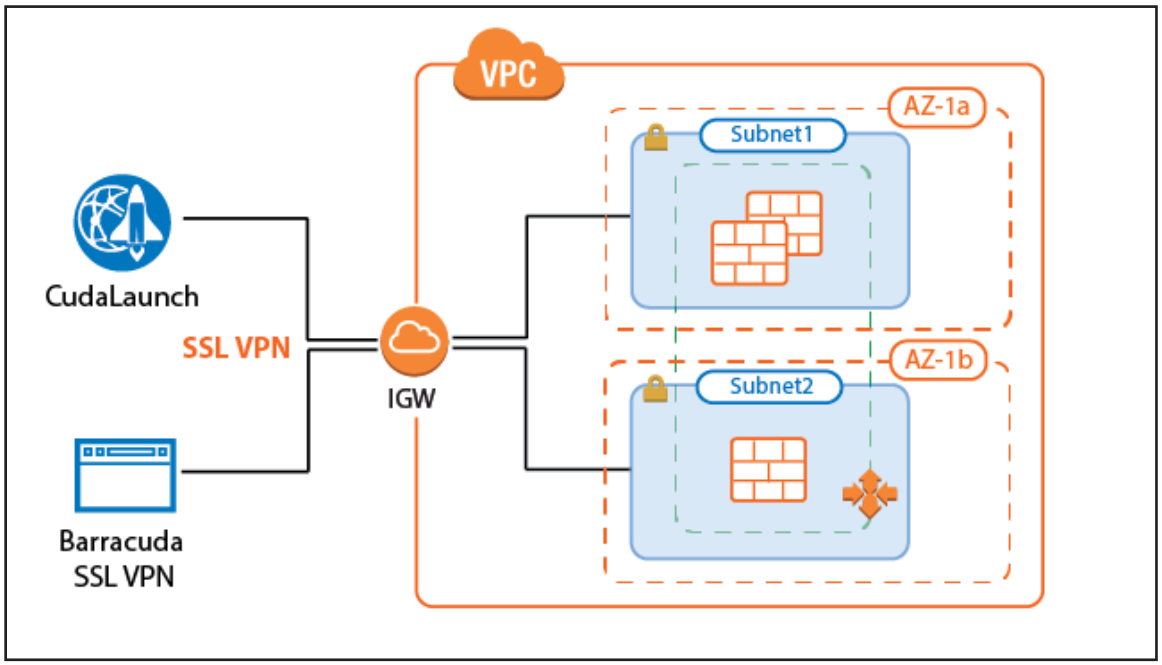
3.16.5 Next Steps

Configure the remote access clients to connect to the client-to-site VPN. For more information, see Remote Access Clients.

Configure SSL VPN and CudaLaunch. For more information, see SSL VPN and F-Series Firewall Configuration for CudaLaunch.

3.17 How to Configure the SSL VPN Services for AWS Auto Scaling Clusters

Let your users connect to a network in an AWS Auto Scaling cluster using SSL VPN. Enable the SSL VPN service and CudaLaunch, create a group access policy, and configure the login and authentication settings for the SSL VPN connections. To use SSL VPN, you must upload a certificate to the AWS certificate manager. For CudaLaunch on iOS, NextGen Firewall Auto Scaling Clusters are supported for CudaLaunch 2.3.0 or higher.

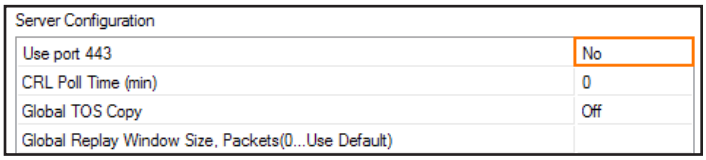


3.17.1 Before You Begin

Configure an external authentication server or NGF local authentication. For more information, see Authentication.

Step 1. Disable Port 443 for Site-to-Site and Client-to-Site VPN

1. Go to **CONFIGURATION > Configuration Tree > Box > Virtual Servers >** `your virtual server` **> Assigned Services > VPN > VPN Settings**.
2. Click **Lock**.
3. Click the **Click here for Server Settings** link. The **Server Settings** window opens.
4. Set **Use Port 443** to **No**.

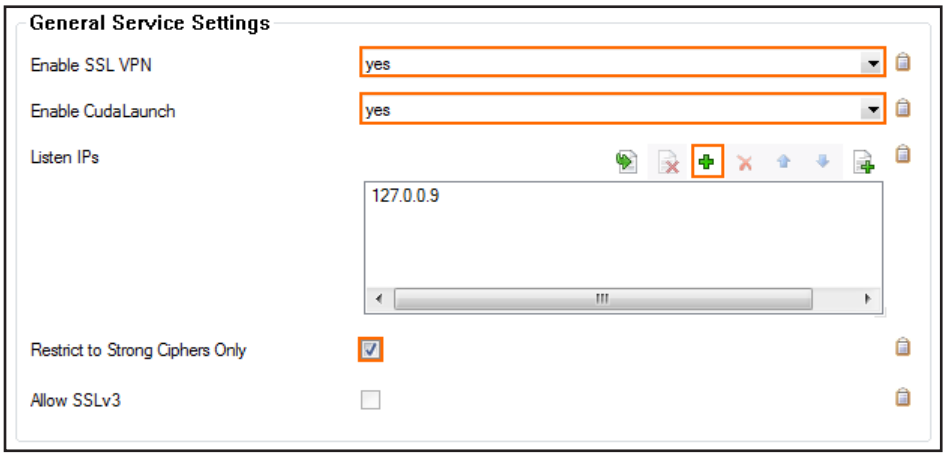


5. Click **OK**.
6. Click **Send Changes** and **Activate**.

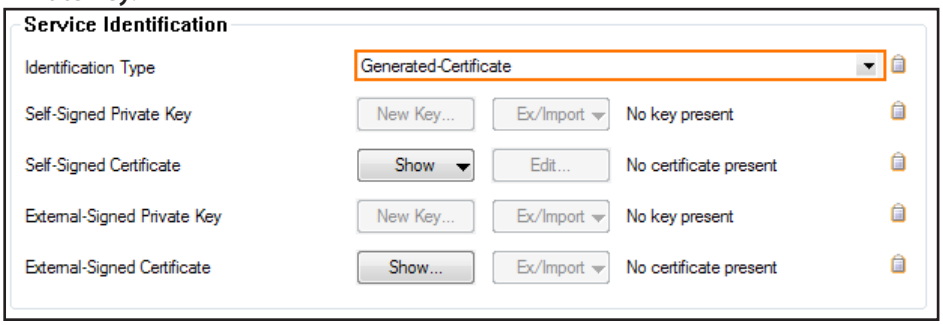
Step 2. Configure SSL VPN General Service Settings

Enable the SSL VPN service and add the listening IP addresses.

1. Go to **CONFIGURATION > Configuration Tree > Box > Virtual Servers >** `your virtual server` **> Assigned Services > VPN-Service > SSL-VPN**.
2. Click **Lock**.
3. Set **Enable SSL VPN** to **Yes**.
4. (optional) Set **Enable CudaLaunch** to **yes**.
5. Click **+** to add a **Listen IP**.
6. Enter the IP address of the VPN service. e.g., `127.0.0.9`



7. (recommended) Enable **Restrict to Strong Ciphers Only**.
8. Select the **Identification Type**:
   - **Generated-Certificate** – The certificate and the private key is automatically created by the firewall.
   - **Self-Signed-Certificate** – Click **New** to create a **Self-Signed Private Key** and then **Edit** to create the **Self-Signed Certificate**.
   - **External-Certificate** – Click **Ex/Import** to import the CA-signed **External Certificate** and the **External-Signed Private Key**.

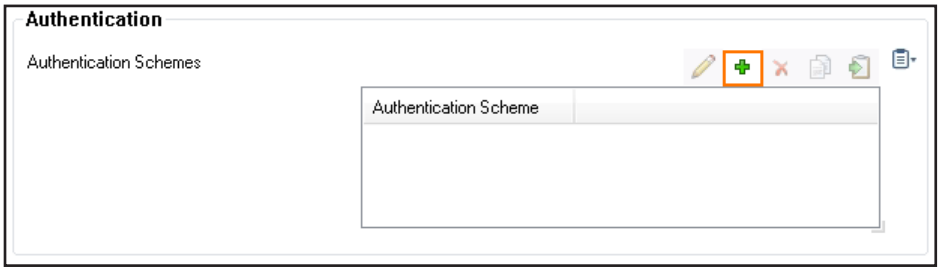


9. Click **Send Changes** and **Activate**.

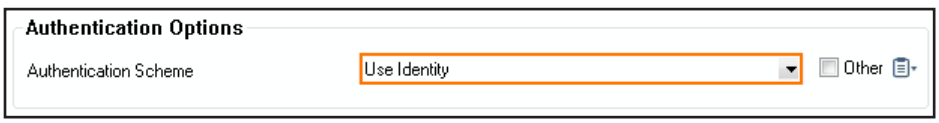
Step 3. Configure User Identity Access Control Policy

1. Go to **CONFIGURATION > Configuration Tree > Box > Virtual Servers >** `your virtual server` **> Assigned Service > VPN-Service > SSL-VPN**.
2. In the left menu, click **Access Control Policies**.
3. Click **Lock**.
4. Click **+** to add an **Access Control Policy**.
5. Enter the **Name** for the access control policy.
6. Click **OK**.
7. In the **Access Control Policy** section, select the **Active** check box.

8. In the **Group Access** section, click **+** to add **Allowed Groups** and **Blocked Groups**. Click **x** to remove the entry from the table.
9. In **Allowed Groups**, either add an asterisk (*) to allow all groups, or enter one or more group names. Leaving the **Allowed Groups** empty causes the **Access Control Policy** to block all authentication attempts.
10. In the **Authentication** section, click **+** to add an **Authentication Scheme**.



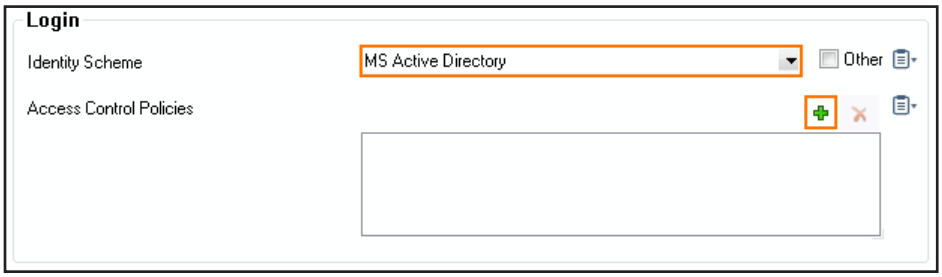
11. Select **Use Identity** from the **Authentication Scheme** drop-down list.



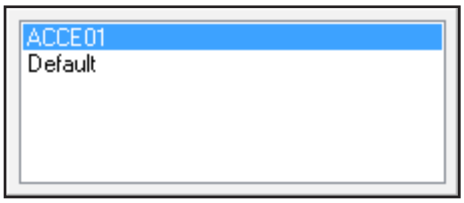
12. Click **OK**.
13. Click **Send Changes** and **Activate**.

Step 4. Configure Login to Log In with User Identity

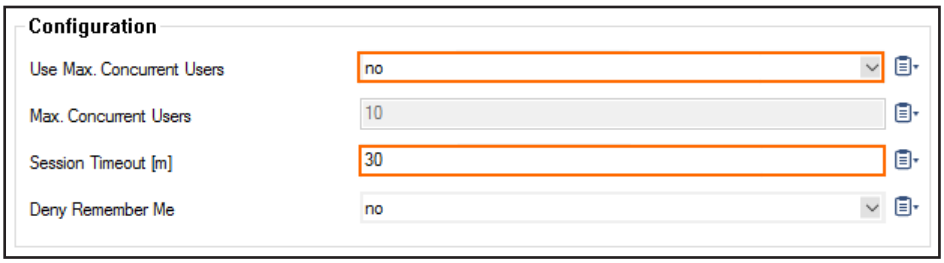
1. Go to **CONFIGURATION > Configuration Tree > Box > Virtual Servers >** `your virtual server` **> Assigned Services > VPN-Service > SSL-VPN**.
2. In the left menu, click **Login**.
3. Click **Lock**.
4. In the **Login** section, set the **Identity Scheme** to your preferred authentication method, e.g., **MS-Active Directory**.
5. Click **+** to add your access control policy to the list of **Access Control Policies**.



6. From the pop-up menu, select the access control policy that you configured in Step 3 for **Use Identity**, i.e., ACCE01.



7. Configure the following settings:
   - **Use Max Concurrent Users** – Set to **no**.
   - **Session Timeout (m)** – Set to 30. This setting must match with the timeout on the ELB.



8. (optional) Customize the login messages and logos:

Import a 200 x 66-pixel PNG or JPG image to customize the **Logo**.

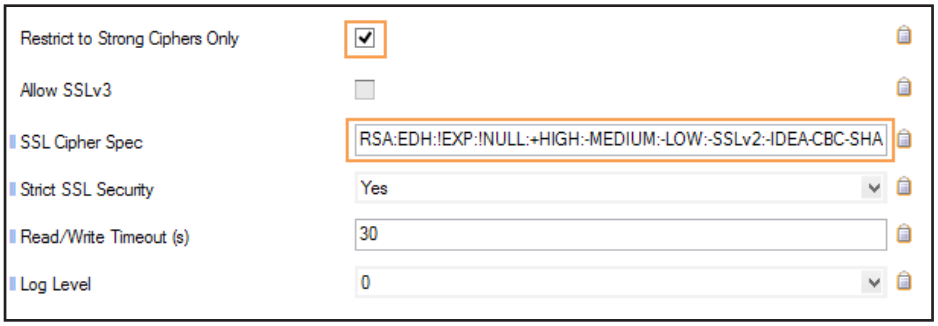
- Enter a plain text **Login Message**. e.g., `Welcome to the Barracuda NextGen Firewall SSL VPN`.
- Enter an HTML **Help Text**.

9. Click **Send Changes** and **Activate**.

Step 5. (optional) Use Custom Cipher String

Configure a custom cipher string to be used by the SSL VPN service.

1. Go to **CONFIGURATION > Configuration Tree > Box > Virtual Servers >** `your virtual server` **> Assigned Services > VPN-Service > SSL-VPN**.
2. In the left menu, click **Basic Setup**.
3. Click **Lock**.
4. In the left menu, expand **Configuration Mode** and click on **Switch to Advanced View**.
5. Disable **Allow SSLv3**.
6. Enable **Restrict to Strong Ciphers Only**.
7. Enter your custom **SSL Cipher Spec** string.



8. Set **Strict SSL Security** to **yes**.

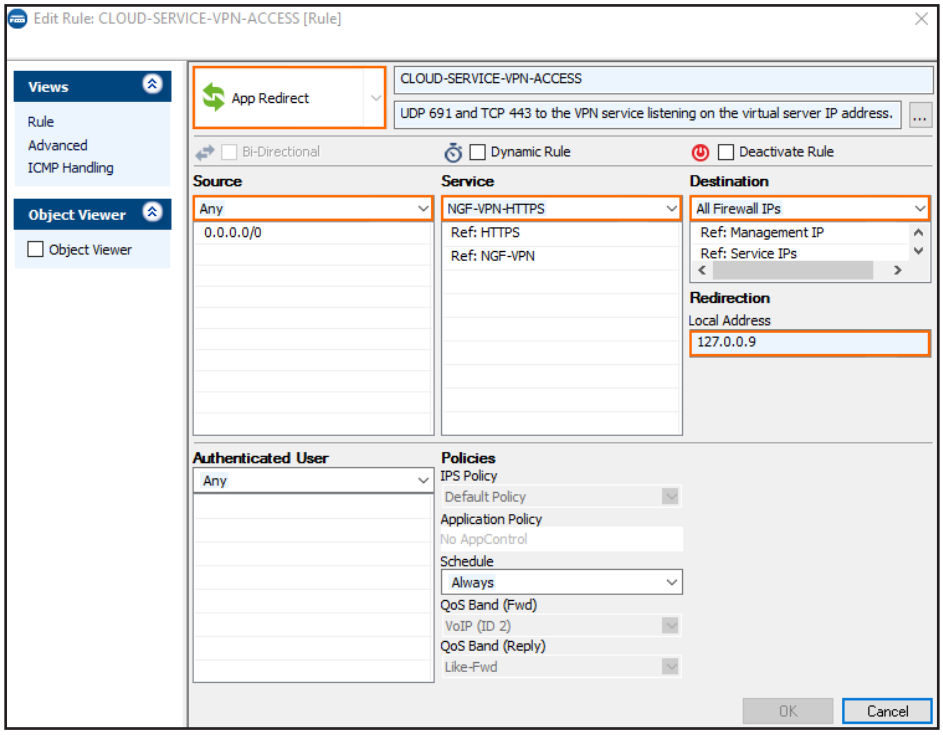
> This setting might break access for some older client SSL implementation. Disable if you experience problems when using older browsers.

9. Click **Send Changes** and **Activate**.

Step 6. Create Access Rules

Verify the the access rule CLOUD-SERVICE-VPN-ACCESS is present in the forwarding ruleset. If not, create the rule. Use the following settings:

- **Action** – Select **App Redirest**.
- **Source** – Select **Any**.
- **Service** – Select **NGF-VPN-HTTPS**.
- **Destination** – Select the network object containing all firewall IPs.
- **Redirection** – Enter the IP address of the VPN service. e.g., `127.0.0.9`.



3.17.2 Troubleshooting

- If the **sslvpn** log contains the following line: `http_listener: failed to listen on <IP address>@443` verify that no other service on the firewall is running on that port and that no Dst NAT access rules are forwarding TCP port 443 (HTTPS) traffic.
- Updating certificates requires the SSL VPN service to be restarted. To do this in an ASG, scale the ASG to a size of one. Then restart the VPN (SSL VPN) service. Then scale out, or wait for the scaling policies to scale your ASG out to the desired size.

Barracuda®