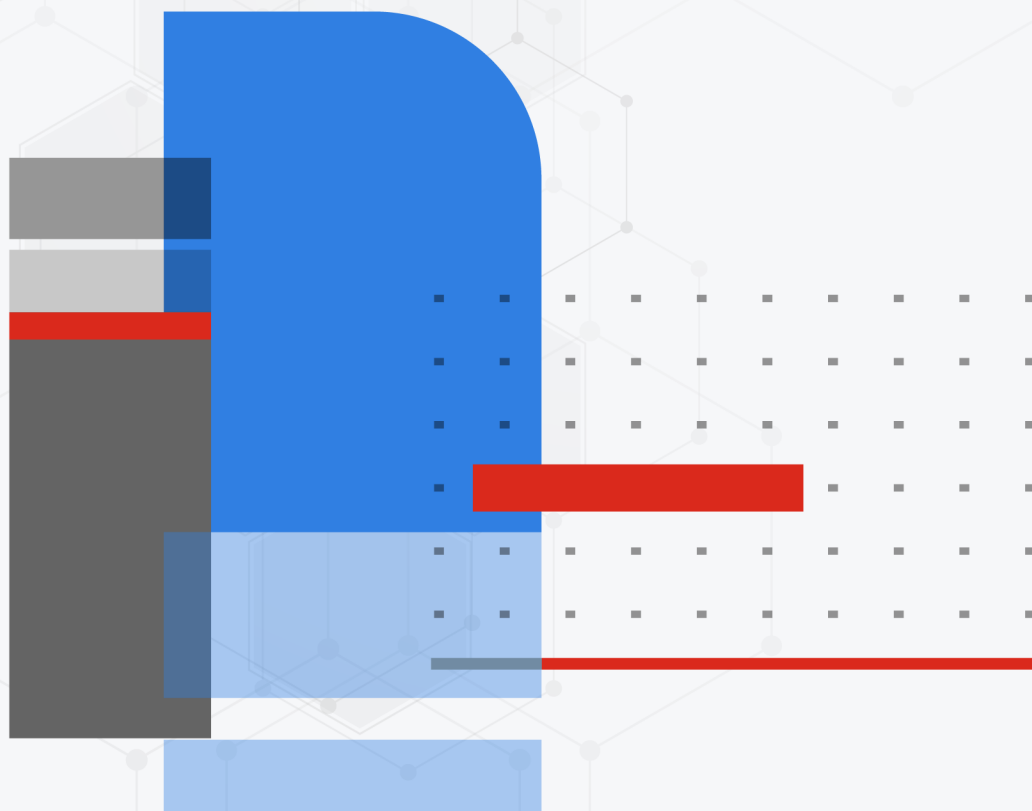




# AWS Administration Guide

FortiOS 7.4



**FORTINET DOCUMENT LIBRARY**

<https://docs.fortinet.com>

**FORTINET VIDEO LIBRARY**

<https://video.fortinet.com>

**FORTINET BLOG**

<https://blog.fortinet.com>

**CUSTOMER SERVICE & SUPPORT**

<https://support.fortinet.com>

**FORTINET TRAINING & CERTIFICATION PROGRAM**

<https://www.fortinet.com/training-certification>

**FORTINET TRAINING INSTITUTE**

<https://training.fortinet.com>

**FORTIGUARD LABS**

<https://www.fortiguard.com>

**END USER LICENSE AGREEMENT**

<https://www.fortinet.com/doc/legal/EULA.pdf>

**FEEDBACK**

Email: [techdoc@fortinet.com](mailto:techdoc@fortinet.com)



March 14, 2024

FortiOS 7.4 AWS Administration Guide

01-740-900312-20240314

# TABLE OF CONTENTS

|   |           |
|---|-----------|
| <b>About FortiGate-VM for AWS</b>   | <b>7</b>  |
| Instance type support   | 7         |
| Region support  | 13        |
| Models  | 15        |
| Licensing   | 17        |
| Order types   | 17        |
| Creating a support account  | 18        |
| FortiCloud-generated license adoption for AWS on-demand variant                                       | 20        |
| Migrating a FortiGate-VM instance between license types   | 21        |
| <b>Deploying FortiGate-VM on AWS</b>  | <b>22</b> |
| Launching FortiGate-VM on AWS   | 22        |
| Security best practices   | 23        |
| Opening ports in the security group   | 23        |
| Administrative access   | 23        |
| IAM roles   | 24        |
| Login credentials   | 24        |
| AWS services and components   | 25        |
| Ordinary FortiGate-VM single instance deployment or FortiGate-native active-passive high availability | 25        |
| Additional or alternative HA using AWS mechanisms   | 26        |
| Monitoring  | 26        |
| Related AWS services used as prerequisites for additional HA or extra features                        | 27        |
| Bootstrapping the FortiGate-VM at initial bootup using user data                                      | 27        |
| Setting up IAM roles  | 28        |
| Creating S3 buckets with license and firewall configurations  | 28        |
| Launching the instance using roles and user data  | 29        |
| Deploying from BYOL AMI   | 31        |
| Deploying on AWS China  | 35        |
| Creating a VPC and subnets  | 36        |
| Attaching the new VPC Internet gateway  | 36        |
| Launching the instance with shared FortiGate-VM AMI   | 37        |
| Connecting to the FortiGate-VM  | 38        |
| Upgrading the FortiGate-VM  | 42        |
| Backing up and restoring configuration  | 42        |
| <b>Deploying auto scaling on AWS</b>  | <b>43</b> |
| Planning  | 44        |
| Technical requirements  | 44        |
| Requirements when using an existing VPC   | 45        |
| Obtaining the deployment package  | 46        |
| BYOL license files  | 48        |
| Deploying the CloudFormation templates  | 48        |
| Deployment notes  | 48        |
| CFT parameters  | 52        |
| Optional settings   | 60        |

|  |            |
|--|------------|
| Completing the deployment .....  | 62         |
| Locating deployed resources .....  | 65         |
| Verifying the deployment .....   | 69         |
| Connecting to the primary FortiGate-VM .....   | 72         |
| Attaching a VPC to the TGW .....   | 75         |
| Troubleshooting .....  | 83         |
| CREATE_FAILED error in CloudFormation stack .....  | 83         |
| The election of the primary FortiGate-VM was not successful .....  | 83         |
| Resetting the elected primary FortiGate-VM .....   | 83         |
| Appendix .....   | 83         |
| FortiGate Autoscale for AWS features .....   | 83         |
| Deployment templates .....   | 86         |
| Cloud-init .....   | 87         |
| Architectural diagrams .....   | 87         |
| Upgrading the deployment .....   | 93         |
| FortiGate Autoscale for AWS document history .....   | 97         |
| <b>Single FortiGate-VM deployment .....</b>  | <b>98</b>  |
| Determining your licensing model .....   | 98         |
| Creating a VPC and subnets .....   | 99         |
| Attaching the new VPC Internet gateway .....   | 100        |
| Subscribing to the FortiGate .....   | 101        |
| Creating routing tables and associate subnets .....  | 103        |
| Connecting to the FortiGate-VM .....   | 109        |
| Setting up a Windows Server in the protected network .....   | 111        |
| <b>HA for FortiGate-VM on AWS .....</b>  | <b>113</b> |
| Deploying FortiGate-VM A-P HA on AWS within one zone .....   | 113        |
| Deploying FortiGate-VM active-passive HA AWS between multiple zones .....  | 121        |
| Deploying FortiGate-VM active-passive HA AWS between multiple zones manually with<br>Transit Gateway integration ..... | 129        |
| Creating VPCs and subnets .....  | 130        |
| Creating a Transit Gateway and related resources .....   | 131        |
| Creating an Internet gateway .....   | 136        |
| Creating VPC route tables .....  | 137        |
| Deploying FortiGate-VM from AWS marketplace .....  | 138        |
| Adding network interfaces and elastic IP addresses to the FortiGate-VMs .....  | 139        |
| Configuring the FortiGate-VMs .....  | 141        |
| Updating the route table and adding an IAM policy .....  | 142        |
| Testing FortiGate-VM HA failover .....   | 143        |
| <b>Deploying FortiGate-VM using Terraform .....</b>  | <b>145</b> |
| Support .....  | 145        |
| <b>Deploying FortiGate-VM on Snowball Edge .....</b>   | <b>146</b> |
| Ordering an Snowball Edge device .....   | 146        |
| Getting connected to the Snowball Edge device .....  | 146        |
| Configuring the Snowball Edge networking .....   | 147        |
| Configuring Snowball Edge via AWS OpsHub .....   | 148        |



|   |            |
|---|------------|
| Generating a raw image and registering it in OpsHub .....                     | 153        |
| Creating the FortiGate-VM Snowball Edge .....                                 | 155        |
| Creating a DNI .....  | 157        |
| Licensing the FortiGate-VM on Snowball Edge .....                             | 159        |
| Accessing and configuring the FortiGate-VM .....                              | 159        |
| FortiManager .....  | 161        |
| Creating a FortiManager on Snowball Edge .....                                | 161        |
| Configuring the FortiManager as a licensing server for the FortiGate-VM ..... | 163        |
| Use case: Traffic inspection with DNI and VLANs .....                         | 163        |
| Troubleshooting .....   | 165        |
| <b>SDN connector integration with AWS .....</b>                               | <b>167</b> |
| Access key-based SDN connector integration .....                              | 167        |
| Configuring an AWS SDN connector using IAM roles .....                        | 167        |
| SDN connector support for alternate resources .....                           | 169        |
| Filter keys .....   | 170        |
| Examples .....  | 171        |
| AWS EKS SDN connector .....   | 174        |
| Populating threat feeds with GuardDuty .....                                  | 175        |
| Security implications .....   | 176        |
| Parameters .....  | 176        |
| Installation .....  | 177        |
| Setting up CloudWatch .....   | 181        |
| Testing the setup .....   | 181        |
| (Optional) Generating sample findings in GuardDuty .....                      | 182        |
| Setting up the FortiGate(s) .....   | 182        |
| Cleanup .....   | 183        |
| Pipelined automation using AWS Lambda .....                                   | 183        |
| Creating an automation stitch .....   | 183        |
| Configuring an example automation stitch .....                                | 184        |
| Configuring FortiGate-VM load balancer using dynamic address objects .....    | 187        |
| Accessing a cloud server using an SDN connector via VPN .....                 | 188        |
| SDN connector support for AWS STS .....                                       | 191        |
| <b>VPN for FortiGate-VM on AWS .....</b>                                      | <b>195</b> |
| Connecting a local FortiGate to an AWS VPC VPN .....                          | 195        |
| Connecting a local FortiGate to an AWS FortiGate via site-to-site VPN .....   | 198        |
| <b>SD-WAN .....</b>   | <b>201</b> |
| SD-WAN cloud on-ramp .....  | 201        |
| SD-WAN Transit Gateway Connect .....  | 201        |
| Creating the TGW and related resources .....                                  | 202        |
| Configuring BGP .....   | 203        |
| Verifying the configuration .....   | 205        |
| Cloud WAN .....   | 206        |
| <b>Security inspection with GWLB integration .....</b>                        | <b>207</b> |
| North-south security inspection to customer VPC .....                         | 207        |
| Creating the GWLB and registering targets .....                               | 208        |

---

|   |            |
|---|------------|
| Creating the LB endpoint .....                                | 209        |
| VPC route tables .....  | 210        |
| Post-deployment configuration .....                           | 210        |
| Validating the configuration .....                            | 212        |
| East-west security inspection between two customer VPCs ..... | 214        |
| Creating the GWLB and registering targets .....               | 215        |
| Creating the LB endpoint .....                                | 216        |
| Creating the transit gateway .....                            | 216        |
| Route tables .....  | 217        |
| Post-deployment configuration .....                           | 218        |
| Validating the configuration .....                            | 220        |
| Multitenancy support with AWS GWLB .....                      | 221        |
| <b>Change log .....</b>                                       | <b>224</b> |

# About FortiGate-VM for AWS

By combining stateful inspection with a comprehensive suite of powerful security features, FortiGate next generation firewall technology delivers complete content and network protection. This solution is available for deployment on AWS.

In addition to advanced features such as an extreme threat database, vulnerability management, and flow-based inspection, features including application control, firewall, antivirus, IPS, web filter, and VPN work in concert to identify and mitigate the latest complex security threats.

The security-hardened FortiOS operating system is purpose-built for inspecting and identifying malware and supports direct single root I/O virtualization for higher and more consistent performance.

FortiGate-VM for AWS supports active/passive high availability (HA) configuration with FortiGate-native unicast HA synchronization between the primary and secondary nodes. When the FortiGate-VM detects a failure, the passive firewall instance becomes active and uses AWS API calls to configure its interfaces/ports.

FortiGate-VM also supports active/active HA using elastic load balancing, as well as autoscaling.

Highlights of FortiGate-VM for AWS include the following:

- Delivers complete content and network protection by combining stateful inspection with a comprehensive suite of powerful security features.
- IPS technology protects against current and emerging network-level threats. In addition to signature-based threat detection, IPS performs anomaly-based detection, which alerts users to any traffic that matches attack behavior profiles.
- Docker application control signatures protect your container environments from newly emerged security threats. See [FortiGate-VM on a Docker environment](#).

## Instance type support

FortiGate-VM supports the following instance types on AWS. Supported instances in the AWS marketplace listing may change without notice and vary between bring your own license (BYOL) and on-demand models. See [Order types on page 17](#). C3 and M-series instances do not appear as recommended instances.

When you run FortiGate-native active-passive high availability, each FortiGate-VM instance requires four network interfaces (port 1 to 4). See [Deploying FortiGate-VM A-P HA on AWS within one zone on page 113](#).

For up-to-date information on each instance type, see the following links:

- [Amazon EC2 Instance Types](#)
- [Elastic Network Interfaces](#)



Downgrading to a previous GA version that does not support UEFI boot mode when using a UEFI-enabled FortiGate instance is not supported. FortiOS 7.4 supports UEFI boot mode.

---



FortiGate-VM 7.4 AMIs are labeled with *UEFI-Preferred* boot mode, which allows for an instance with UEFI support to boot in UEFI boot mode without user interaction. The instance defaults to the legacy BIOS if the instance type does not support UEFI. See [Boot modes](#).

The following table summarizes instance type support for x64 instances. For a list of all supported instances for x64 FortiGate-VM deployments, see [Fortinet FortiGate \(BYOL\) Next-Generation Firewall](#).

| Instance category | Instance type | vCPU | Max NIC (AWS-enabled) | FortiGate minimum order (BYOL) to consume all instance CPU |
|-------------------|---------------|------|-----------------------|--|
| General purpose   | t2.small      | 1    | 2                     | FG-VM01 or FG-VM01v  |
|                   | t3.small      | 2    | 3                     | FG-VM02 or FG-VM02v  |
|                   | t3.medium     | 2    | 3                     |  |
|                   | t3.xlarge     | 4    | 4                     | FG-VM04 or FG-VM04v  |
| Compute optimized | c4.large      | 2    | 3                     | FG-VM02 or FG-VM02v  |
|                   | c4.xlarge     | 4    | 4                     | FG-VM04 or FG-VM04v  |
|                   | c4.2xlarge    | 8    | 4                     | FG-VM08 or FG-VM08v  |
|                   | c4.4xlarge    | 16   | 8                     | FG-VM16 or FG-VM16v  |
|                   | c4.8xlarge    | 36   | 8                     | FG-VMUL or FG-VMULv  |
|                   | c5.large      | 2    | 3                     | FG-VM02 or FG-VM02v  |
|                   | c5.xlarge     | 4    | 4                     | FG-VM04 or FG-VM04v  |
|                   | c5.2xlarge    | 8    | 4                     | FG-VM08 or FG-VM08v  |
|                   | c5.4xlarge    | 16   | 8                     | FG-VM16 or FG-VM16v  |
|                   | c5.9xlarge    | 36   | 8                     | FG-VMUL or FG-VMULv  |
|                   | c5.18xlarge   | 72   | 15                    |  |
|                   | c5d.large     | 2    | 3                     | FG-VM02 or FG-VM02v  |
|                   | c5d.xlarge    | 4    | 4                     | FG-VM04 or FG-VM04v  |
|                   | c5d.2xlarge   | 8    | 4                     | FG-VM08 or FG-VM08v  |
|                   | c5d.4xlarge   | 16   | 8                     | FG-VM16 or FG-VM16v  |
|                   | c5d.9xlarge   | 36   | 8                     | FG-VMUL or FG-VMULv  |

| Instance category | Instance type                          | vCPU | Max NIC (AWS-enabled) | FortiGate minimum order (BYOL) to consume all instance CPU |
|-------------------|--|------|-----------------------|--|
| Compute optimized | c5d.12xlarge                           | 48   | 8                     |  |
|                   | c5d.18xlarge                           | 72   | 15                    |  |
|                   | c5n.large                              | 2    | 3                     | FG-VM02 or FG-VM02v  |
|                   | c5n.xlarge                             | 4    | 4                     | FG-VM04 or FG-VM04v  |
|                   | c5n.2xlarge                            | 8    | 4                     | FG-VM08 or FG-VM08v  |
|                   | c5n.4xlarge                            | 16   | 8                     | FG-VM16 or FG-VM16v  |
|                   | c5n.9xlarge                            | 36   | 8                     | FG-VMUL or FG-VMULv  |
|                   | c5n.18xlarge                           | 72   | 15                    |  |
|                   | c6i.large                              | 2    | 3                     | FG-VM02 or FG-VM02v  |
|                   | c6i.xlarge<br>(recommended by default) | 4    | 4                     | FG-VM04 or FG-VM04v  |
|                   | c6i.2xlarge                            | 8    | 4                     | FG-VM08 or FG-VM08v  |
|                   | c6i.4xlarge                            | 16   | 8                     | FG-VM16 or FG-VM16v  |
|                   | c6i.8xlarge                            | 32   | 8                     | FG-VMUL or FG-VMULv  |
|                   | c6i.16xlarge                           | 64   | 15                    |  |
|                   | c6i.24xlarge                           | 96   | 15                    |  |
|                   | c6in.large                             | 2    | 3                     | FG-VM02 or FG-VM02v  |
|                   | c6in.xlarge                            | 4    | 4                     | FG-VM04 or FG-VM04v  |
|                   | c6in.2xlarge                           | 8    | 4                     | FG-VM08 or FG-VM08v  |
|                   | c6in.4xlarge                           | 16   | 8                     | FG-VM16 or FG-VM16v  |
|                   | c6in.8xlarge                           | 32   | 8                     | FG-VMUL or FG-VMULv  |
|                   | c6in.16xlarge                          | 64   | 15                    |  |
|                   | c6a.large                              | 2    | 3                     | FG-VM02 or FG-VM02v  |
|                   | c6a.xlarge                             | 4    | 4                     | FG-VM04 or FG-VM04v  |
|                   | c6a.2xlarge                            | 8    | 4                     | FG-VM08 or FG-VM08v  |
|                   | c6a.4xlarge                            | 16   | 8                     | FG-VM16 or FG-VM16v  |
|                   | c6a.8xlarge                            | 32   | 8                     | FG-VMUL or FG-VMULv  |
|                   | c6a.16xlarge                           | 64   | 15                    |  |

The following table summarizes instance type support for ARM-based CPU instances. For a list of all supported instances for ARM-based CPU FortiGate-VM deployments, see [Fortinet FortiGate \(BYOL\) Next-Generation Firewall \(ARM64/Graviton\)](#).

| Instance category | Instance type | vCPU | Max NIC (AWS-enabled) | FortiGate minimum order (BYOL) to consume all instance CPU |
|-------------------|---------------|------|-----------------------|--|
| General purpose   | t4g.small     | 2    | 3                     | FG-VM02 or FG-VM02v  |
|                   | t4g.medium    |      |                       |  |
|                   | t4g.large     |      |                       |  |
|                   | t4g.xlarge    | 4    | 4                     | FG-VM04 or FG-VM04v  |
|                   | t4g.2xlarge   | 8    |                       | FG-VM08 or FG-VM08v  |
|                   | m6g.large     | 2    | 3                     | FG-VM02 or FG-VM02v  |
|                   | m6g.xlarge    | 4    | 4                     | FG-VM04 or FG-VM04v  |
|                   | m6g.2xlarge   | 8    |                       | FG-VM08 or FG-VM08v  |
|                   | m6g.4xlarge   | 16   | 8                     | FG-VM16 or FG-VM16v  |
|                   | m6g.8xlarge   | 32   |                       | FG-VMUL or FG-VMULv  |
|                   | m6g.12xlarge  | 48   |                       |  |
|                   | m6g.16xlarge  | 64   | 15                    |  |

| Instance category | Instance type                       | vCPU | Max NIC (AWS-enabled) | FortiGate minimum order (BYOL) to consume all instance CPU |
|-------------------|-------------------------------------|------|-----------------------|--|
| Compute optimized | c6g.large                           | 2    | 3                     | FG-VM02 or FG-VM02v  |
|                   | c6g.xlarge                          | 4    | 4                     | FG-VM04 or FG-VM04v  |
|                   | c6g.2xlarge                         | 8    |                       | FG-VM08 or FG-VM08v  |
|                   | c6g.4xlarge                         | 16   | 8                     | FG-VM16 or FG-VM16v  |
|                   | c6g.8xlarge                         | 32   |                       | FG-VMUL or FG-VMULv  |
|                   | c6g.12xlarge                        | 48   |                       |  |
|                   | c6g.16xlarge                        | 64   | 15                    |  |
|                   | c6gn.medium                         | 1    | 2                     | FG-VM01 or FG-VM01v  |
|                   | c6gn.large                          | 2    | 3                     | FG-VM02 or FG-VM02v  |
|                   | c6gn.xlarge                         | 4    | 4                     | FG-VM04 or FG-VM04v  |
|                   | c6gn.2xlarge                        | 8    |                       | FG-VM08 or FG-VM08v  |
|                   | c6gn.4xlarge                        | 16   | 8                     | FG-VM16 or FG-VM16v  |
|                   | c6gn.8xlarge                        | 32   |                       | FG-VMUL or FG-VMULv  |
|                   | c6gn.16xlarge                       | 64   | 15                    |  |
|                   | c7g.medium                          | 1    | 2                     | FG-VM01 or FG-VM01v  |
|                   | c7g.large                           | 2    | 3                     | FG-VM02 or FG-VM02v  |
|                   | c7g.xlarge                          | 4    | 4                     | FG-VM04 or FG-VM04v  |
|                   | c7g.2xlarge                         | 8    |                       | FG-VM08 or FG-VM08v  |
|                   | c7g.4xlarge                         | 16   | 8                     | FG-VM16 or FG-VM16v  |
|                   | c7g.8xlarge                         | 32   |                       | FG-VMUL or FG-VMULv  |
|                   | c7g.16xlarge                        | 64   | 15                    |  |
|                   | c7gn.large (recommended by default) | 2    | 3                     | FG-VM02 or FG-VM02v  |
|                   | c7gn.xlarge                         | 4    | 4                     | FG-VM04 or FG-VM04v  |
|                   | c7gn.2xlarge                        | 8    | 4                     | FG-VM08 or FG-VM08v  |
|                   | c7gn.4xlarge                        | 16   | 8                     | FG-VM16 or FG-VM16v  |
|                   | c7gn.8xlarge                        | 32   | 8                     | FG-VMUL or FG-VMULv  |
|                   | c7gn.16xlarge                       | 64   | 15                    |  |

You can apply a smaller FortiGate-VM license if you are OK with consuming less CPU than is present on your instance. See [Models on page 15](#).

For more information about checking and enabling ENA support on AWS instances, see [Enable enhanced networking with the Elastic Network Adapter \(ENA\) on Linux instances](#).

FortiOS supports hot-adding vCPU and RAM. However, AWS may not support this. See [Requirements for changing the instance type](#).



## Region support

BYOL and on-demand deployments support the following regions. See [Order types on page 17](#).

Instance support may vary depending on the regions.

For details about regions, see [Regions and Availability Zones](#).

| Region name              | Region code    |
|--------------------------|----------------|
| US East (N. Virginia)    | us-east-1      |
| US East (Ohio)           | us-east-2      |
| US West (N. California)  | us-west-1      |
| US West (Oregon)         | us-west-2      |
| Africa (Cape Town)       | af-south-1     |
| Asia Pacific (Hong Kong) | ap-east-1      |
| Asia Pacific (Mumbai)    | ap-south-1     |
| Asia Pacific (Hyderabad) | ap-south-2     |
| Asia Pacific (Singapore) | ap-southeast-1 |
| Asia Pacific (Sydney)    | ap-southeast-2 |
| Asia Pacific (Jakarta)   | ap-southeast-3 |
| Asia Pacific (Melbourne) | ap-southeast-4 |
| Asia Pacific (Tokyo)     | ap-northeast-1 |
| Asia Pacific (Seoul)     | ap-northeast-2 |
| Asia Pacific (Osaka)     | ap-northeast-3 |
| Canada (Central)         | ca-central-1   |
| EU (Frankfurt)           | eu-central-1   |
| EU (Zurich)              | eu-central-2   |
| EU (Ireland)             | eu-west-1      |
| EU (London)              | eu-west-2      |
| EU (Paris)               | eu-west-3      |
| EU (Stockholm)           | eu-north-1     |
| EU (Milan)               | eu-south-1     |
| EU (Spain)               | eu-south-2     |

| Region name               | Region code   |
|---------------------------|---------------|
| Middle East (Bahrain)     | me-south-1    |
| Middle East (UAE)         | me-central-1  |
| South America (São Paulo) | sa-east-1     |
| AWS GovCloud (US-East)    | us-gov-east-1 |
| AWS GovCloud (US-West)    | us-gov-west-1 |

FortiOS 7.4.1 and later versions also support the following local zones with instance types c5d.2xlarge, c5d.4xlarge, and c5d.12xlarge:

| Local zone name       | Local zone code  |
|-----------------------|------------------|
| US East (Atlanta)     | us-east-1-atl-1a |
| US East (Boston)      | us-east-1-bos-1a |
| US East (Chicago)     | us-east-1-chi-1a |
| US West (Denver)      | us-west-2-den-1a |
| US West (Las Vegas)   | us-west-2-las-1a |
| US West (Los Angeles) | us-west-2-lax-1a |

AWS China is supported but does not appear with these regions when you log into the AWS portal. To use AWS resources on AWS China, you must have an AWS China account separate from your global AWS account.

FortiGate-VM for AWS China only supports the bring your own license model. To activate it, you must obtain a license. See [Deploying on AWS China on page 35](#).

## Models

FortiGate-VM is available with different CPU and RAM sizes. You can deploy FortiGate-VM on various private and public cloud platforms. The following table shows the models conventionally available to order, also known as bring your own license (BYOL) models. See [Order types on page 17](#).

| Model name      | vCPU    |           |
|-----------------|---------|-----------|
|                 | Minimum | Maximum   |
| FG-VM01/01v/01s | 1       | 1         |
| FG-VM02/02v/02s | 1       | 2         |
| FG-VM04/04v/04s | 1       | 4         |
| FG-VM08/08v/08s | 1       | 8         |
| FG-VM16/16v/16s | 1       | 16        |
| FG-VM32/32v/32s | 1       | 32        |
| FG-VMUL/ULv/ULs | 1       | Unlimited |



With the changes in the FortiGuard extended IPS database introduced in FortiOS 7.4.0, some workloads that depend on the extended IPS database must have the underlying VM resized to 8 vCPU or more to continue using the extended IPS database.

See [Support full extended IPS database for FortiGate VMs with eight cores or more](#).

For information about changing the instance type on an existing VM, see [Change the instance type](#).

For more information about Compute instances, see [Compute Optimized](#).



The v-series and s-series do not support virtual domains (VDMs) by default. To add VDMs, you must separately purchase perpetual VDM addition licenses. You can add and stack VDMs up to the maximum supported number after initial deployment.

Generally there are RAM size restrictions to FortiGate-VM BYOL licenses. However, these restrictions are not applicable to AWS deployments. Any RAM size with certain CPU models are allowed. Licenses are based on the number of CPUs only.

Previously, platform-specific models such as FortiGate-VM for AWS with an AWS-specific orderable menu existed. However, the common model is now applicable to all supported platforms.

For information about each model's order information, capacity limits, and adding VDMs, see the [FortiGate-VM datasheet](#).

The primary requirement for the provisioning of a FortiGate-VM may be the number of interfaces it can accommodate rather than its processing capabilities. In some cloud environments, the options with a high number of interfaces tend to have high numbers of vCPUs.

The licensing for FortiGate-VM does not restrict whether the FortiGate can work on a VM instance in a public cloud that uses more vCPUs than the license allows. The number of vCPUs indicated by the license does not restrict the FortiGate-VM from working, regardless of how many vCPUs are included in the virtual instance. However, only the licensed number of vCPUs process traffic and management. The rest of the vCPUs are unused.

The following shows an example for a FGT-VM08 license:

| License  | 1 vCPU | 2 vCPU | 4 vCPU | 8 vCPU | 16 vCPU  | 32 vCPU  |
|----------|--------|--------|--------|--------|--|--|
| FGT-VM08 | OK     | OK     | OK     | OK     | FortiOS uses eight vCPUs for traffic and management. It does not use the rest. | FortiOS uses eight vCPUs for traffic and management. It does not use the rest. |

You can provision a VM instance based on the number of interfaces you need and license the FortiGate-VM for only the processors you need.

# Licensing

You must have a license to deploy FortiGate-VM for AWS.

## Order types

On AWS, there are usually two order types: bring your own license (BYOL) and on-demand.

BYOL offers perpetual (normal series and v-series) and annual subscription (s-series) licensing as opposed to on-demand, which is an hourly subscription available with marketplace-listed products. BYOL licenses are available for purchase from resellers or your distributors, and the publicly available price list, which Fortinet updates quarterly, lists prices. BYOL licensing provides the same ordering practice across all private and public clouds, no matter what the platform is. You must activate a license for the first time you access the instance from the GUI or CLI before you can start using various features.

With an on-demand subscription, the FortiGate-VM becomes available for use immediately after you create the instance. The marketplace product page mentions term-based prices (hourly or annual).

For BYOL and on-demand deployments, cloud vendors charge separately for resource consumption on computing instances, storage, and so on, without use of software running on top of it (in this case the FortiGate-VM).

For BYOL, you typically order a combination of products and services including support entitlement. S-series SKUs contain the VM base and service bundle entitlements for easier ordering. On-demand includes support, for which you must contact [Fortinet Support](#) with your customer information. See [Support Information](#).

To purchase on-demand, all you need to do is subscribe to the product on the marketplace. However, you must contact Fortinet Support with your customer information to obtain support entitlement. See [Creating a support account on page 18](#). For the latest on-demand pricing and support details, see the [FortiGate-VM on-demand marketplace product page](#).



On-demand FortiGate-VM instances do not support the use of virtual domains (VDOMs). If you plan to use VDOMs, deploy BYOL instances instead.



On-demand and BYOL licensing and payment models are not interchangeable. For example, once you spin up a FortiGate-VM on-demand instance, you cannot inject a BYOL license on the same VM. Likewise, you cannot convert a FortiGate-VM BYOL instance to on-demand.

---

When using a FortiGate-VM on-demand instance prior to version 6.4.2, the FortiOS GUI may display expiry dates for FortiGuard services. However, these expiries are automatically extended for as long as the on-demand instance's lifespan. You do not need to be concerned about the expiry of FortiGuard services. For example, the following screenshot shows 2038/01/02.

| FortiGuard Distribution Network       |                                  |   |
|---------------------------------------|----------------------------------|---|
| License Information                   |                                  |   |
| Entitlement                           | Status                           |   |
| FortiCare Support                     | Not Supported                    |   |
| Firmware & General Updates            | Licensed - expires on 2038/01/02 |   |
| Application Control Signatures        | Version 16.00975                 | U |
| Device & OS Identification            | Version 1.00110                  |   |
| Internet Service Database Definitions | Version 7.01212                  |   |
| Intrusion Prevention                  | Licensed - expires on 2038/01/02 |   |
| IPS Definitions                       | Version 16.00975                 | U |
| IPS Engine                            | Version 5.00021                  |   |

FortiOS 6.4.2 and later versions do not display dates.

| FortiGuard Distribution Network |                |                      |
|---------------------------------|----------------|----------------------|
| License Information 1           |                |                      |
| Entitlement                     | Status         |                      |
| FortiCare Support               | Not Registered | Actions              |
| Virtual Machine                 | Valid          | FortiGate VM License |
| Firmware & General Updates      | Licensed       |                      |
| Intrusion Prevention            | Licensed       |                      |
| AntiVirus                       | Licensed       |                      |
| Web Filtering                   | Licensed       |                      |
| Outbreak Prevention             | Licensed       |                      |
| SD-WAN Network Monitor          | Not Licensed   | Purchase             |
| Security Rating                 | Licensed       |                      |

## Creating a support account

FortiGate-VM for AWS supports on-demand and bring your own license licensing models. See [Order types on page 17](#).

To use Fortinet technical support and ensure products function properly, you must complete certain steps to activate your entitlement. The Fortinet support team can identify your registration in the system thereafter.

First, if you do not have a Fortinet account, create one at [Customer Service & Support](#).

## BYOL

You must obtain a license to activate the FortiGate-VM. If you have not activated the license, you see the license upload screen when you log in to the FortiGate-VM and cannot proceed to configure the FortiGate-VM.

You can obtain licenses for the bring your own license (BYOL) licensing model through any Fortinet partner. If you do not have a partner, contact [awsales@fortinet.com](mailto:awsales@fortinet.com) for assistance in purchasing a license.

After you purchase a license or obtain an evaluation license, you receive a PDF with an activation code.

FortiOS has a permanent trial license, which requires a FortiCare account. This trial license has limited features and capacity. The trial license only applies to BYOL deployments for FortiGate-VM on AWS. See [Permanent trial mode for FortiGate-VM](#).

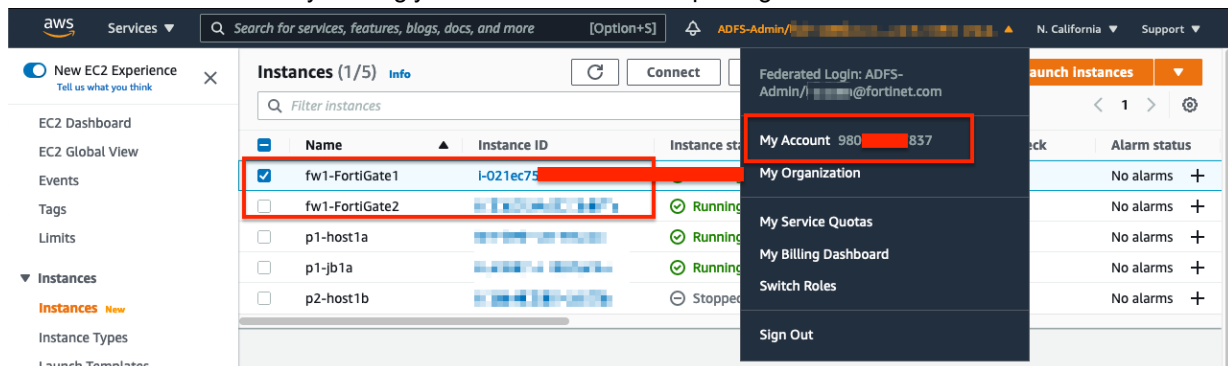
## To register a BYOL license:

1. Go to [Customer Service & Support](#) and create a new account or log in with an existing account.
2. Click *Register Now* to start the registration process.
3. In the *Registration Code* field, enter the registration code that was emailed to you, and select *Next* to access the registration form.
4. If you register the s-series subscription model, the site prompts you to select one of the following:
  - a. Click *Register* to newly register the code to acquire a new serial number with a new license file.
  - b. Click *Renew* to renew and extend the licensed period on top of the existing serial number, so that all features on the VM node continue working uninterrupted upon license renewal.
5. At the end of the registration process, download the license (.lic) file to your computer. You upload this license later to activate the FortiGate-VM. After registering a license, Fortinet servers may take up to 30 minutes to fully recognize the new license. When you upload the license (.lic) file to activate the FortiGate-VM, if you get an error that the license is invalid, wait 30 minutes and try again.

## On-demand

### To create a support account for on-demand deployments:

1. Deploy and boot the FortiGate-VM on-demand Elastic Compute Cloud (EC2) instance.
2. In the AWS management console, view the newly booted instance's instance ID. You can see the account that this instance was launched in by clicking your credentials on the top navigation bar.



3. Obtain the FortiGate-VM serial number:
  - a. Log into the FortiGate-VM GUI management console.
  - b. From the Dashboard, copy the FortiGate-VM serial number. You provide this serial number when registering your instance with Fortinet. Your instance requires Internet access for FortiCloud to assign this serial number to it. The instance also requires Internet access to report the AWS account ID that it detects to FortiCloud.
4. Go to [Customer Service & Support](#) and create a new account or log in with an existing account.
5. Go to *Asset Management > Register Now* to start the registration process.
6. In the *Registration Code* field, enter the serial number, and select *Next*.
7. In the AWS account ID field, enter the account ID that you gathered from AWS. In this example, the AWS account ID is 980...837.

If you provide an AWS account ID that does not match the one that the FortiGate reported to FortiCloud during its initial startup, FortiCloud rejects it.

8. Complete the registration.
9. After completing registration, contact [Fortinet Customer Support](#) and provide your FortiGate instance's serial number and the email address associated with your Fortinet account.

## FortiCloud-generated license adoption for AWS on-demand variant

FortiGate-VM AWS on-demand instances can obtain FortiCloud-generated licenses and register to FortiCloud.

The valid license allows you to register to FortiCloud to use features including FortiToken with the FortiGate-VM instance.

The FortiGate-VM must be able to reach FortiCloud to receive a valid on-demand license. Ensure connectivity to FortiCloud (<https://directregistration.fortinet.com/>) by checking all related setup on security groups, access control lists, Internet gateways, route tables, public IP addresses, and so on.

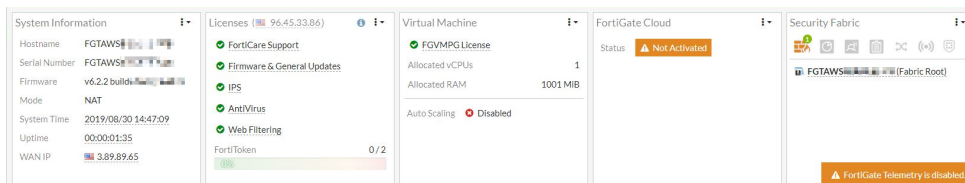
If you created the FortiGate-VM in a closed environment or it cannot reach FortiCloud, the FortiGate-VM self-generates a local license as in previous FortiOS versions. You can obtain a FortiCloud license, ensure that the FortiGate-VM can connect to FortiCloud, then run the `execute vm-license` command to obtain the license from FortiCloud.

### To deploy a FortiGate-VM AWS on-demand instance:

When deploying a FortiGate-VM on-demand instance for AWS, you use the `FGT_VM64_AWS-v7-buildXXXX-FORTINET.out` image. After deployment with this image, running `get system status` results in output that includes the following lines:

```
Version: FortiGate-VM64-AWS v7.4.x,buildXXXX,XXXXXX (GA)
Virus-DB: 71.00242(2019-08-30 08:19)
Extended DB: 1.00000(2018-04-09 18:07)
Extreme DB: 1.00000(2018-04-09 18:07)
IPS-DB: 6.00741(2015-12-01 02:30)
IPS-ETDB: 0.00000(2001-01-01 00:00)
APP-DB: 6.00741(2015-12-01 02:30)
INDUSTRIAL-DB: 6.00741(2015-12-01 02:30)
Serial-Number: FGTAWS12345678
```





## Migrating a FortiGate-VM instance between license types

When deploying a FortiGate-VM on public cloud, you determine the license type (on-demand or bring your own license (BYOL)) during deployment. The license type is fixed for the VM's lifetime. The image that you use to deploy the FortiGate-VM on the public cloud marketplace predetermines the license type.

Migrating a FortiGate-VM instance from one license type to another requires a new deployment. You cannot simply switch license types on the same VM instance. However, you can migrate the configuration between two VMs running as different license types. There are also FortiOS feature differences between on-demand and BYOL license types. For example, a FortiGate-VM on-demand instance is packaged with Unified Threat Management protection and does not support virtual domains, whereas a FortiGate-VM BYOL instance supports greater protection levels and features depending on its contract.

### To migrate FortiOS configuration to a FortiGate-VM of another license type:

1. Connect to the FortiOS GUI or CLI and back up the configuration. See [Configuration backups](#).
2. Deploy a new FortiGate-VM instance with the desired license type. If deploying a BYOL instance, you must purchase a new license from a Fortinet reseller. You can apply the license after deployment via the FortiOS GUI or bootstrap the license and configuration during initial bootup using custom data as [Bootstrapping the FortiGate-VM at initial bootup using user data on page 27](#) describes.
3. Restore the configuration on the FortiGate-VM instance that you deployed in step 2. As with the license, you can inject the configuration during initial bootup. Alternatively, you can restore the configuration in the FortiOS GUI as described in [Configuration backups](#).
4. If you deployed an on-demand instance in step 2, register the license. To receive support for an on-demand license, you must register the license as [Creating a support account on page 18](#) describes.

# Deploying FortiGate-VM on AWS

The following sections offer different options for FortiGate-VM single deployment on AWS:

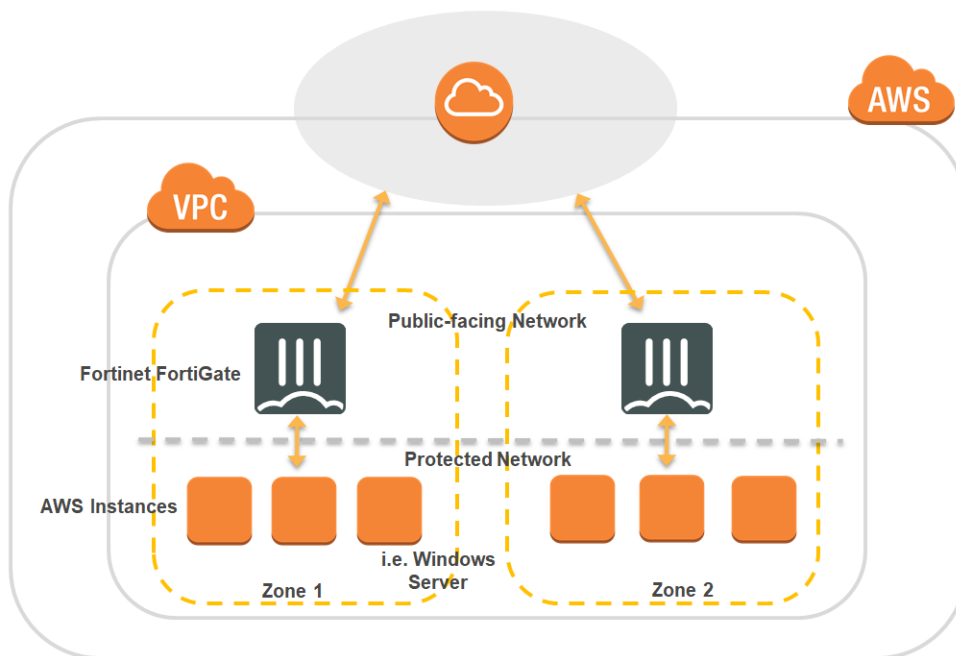
- [Launching FortiGate-VM on AWS on page 22](#)
- [Bootstrapping the FortiGate-VM at initial bootup using user data on page 27](#)
- [Deploying from BYOL AMI on page 31](#)
- [Deploying on AWS China on page 35](#)
- [Deploying a FortiGate CNF instance](#)

## Launching FortiGate-VM on AWS



Downgrading to a previous GA version that does not support UEFI boot mode when using a UEFI-enabled FortiGate instance is not supported. FortiOS 7.4 supports UEFI boot mode.

See [Single FortiGate-VM deployment on page 98](#).



The most basic deployment consists of one FortiGate-VM with two elastic network interfaces (ENIs) facing a public subnet and private subnet, with the FortiGate-VM deployed inline between the two subnets. A single FortiGate-VM protects a single virtual private cloud (VPC) with a single availability zone (AZ). The public subnet's default gateway is an AWS Internet gateway, and the FortiGate-VM's private subnet-facing ENI is the private subnet's default gateway. Protected EC2 instances such as web servers, database servers, or other endpoints are assumed to exist in the private

subnet. One elastic/public IP address or IPv4 DNS name must be allocated to the FortiGate-VM in the public subnet for you to access the FortiGate-VM remotely via HTTPS or SSH over the Internet for initial configuration.

## Security best practices

You can find general AWS security best practices at [Best Practices for Security, Identity, & Compliance](#).

In addition to following the general AWS guidelines, there are best practices to follow when deploying FortiGate-VM for AWS.

### Opening ports in the security group

By default, when you deploy FortiGate-VM, there is a predefined security group that you can select based on Fortinet's recommendation. The following ports are allowed in the predefined security group assuming immediate and near-future needs.

|          | Protocol/ports | Purpose  |
|----------|----------------|--|
| Incoming | TCP 22         | SSH  |
|          | TCP 80         | HTTP   |
|          | TCP 443        | HTTPS, management GUI access to the FortiGate-VM   |
|          | TCP 541        | Management by FortiManager located outside AWS   |
|          | TCP 3000       | Not immediately required, but typically used for incoming access to web servers, and so on |
|          | TCP 8080       |  |
| Outgoing | Any            |  |

FortiGate-specific open ports are explained in [Fortinet Communication Ports and Protocols](#).

To configure bare-minimum access that gives the most strict incoming access, allow only TCP 443 to access the FortiGate-VM GUI console as mentioned in [Connecting to the FortiGate-VM on page 109](#) and close all other ports. You may want to allow ICMP for ping, and so on, as needed.

### Administrative access

This is rather an ordinary consideration than AWS-specific to secure the FortiGate-VM and protect it by configuring allowed and restricted protocols and ports in corporate security scenes.

One example is to configure the local admin access to one of the FortiGate-VM's local network interfaces. Log into the GUI, go to *Network > Interfaces*, then chose the desired port to configure under *Administrative Access*.

To configure general firewall policies to protect VMs in the networks, see [Setting up a Windows Server in the protected network on page 111](#) or the [FortiOS documentation](#) for details.

## IAM roles

To deploy FortiGate-VM on the marketplace, you must log into the AWS portal as an AWS user. Your organization's administrator may have granted permissions via certain IAM roles. AWS security best practices explain when and in what use cases you need IAM roles. How you manage IAM users and roles is up to your organization.

When deploying FortiGate-VM on marketplace web or EC2 console, your AWS account must have appropriate permissions, including being able to subscribe to AWS resources through the marketplace, access EC2 resources, browse AWS resource groups, and so on.

## Login credentials

By default, you can log into the FortiGate-VM through HTTPS or SSH using the username "admin" and the FortiGate-VM's instance ID as the initial password. SSH also requires your AWS key.

The instance ID is relatively secure as it is visible only within the AWS portal or by running the AWS CLI. However, it may be viewable to those who have access to AWS resources but should not have access to the FortiGate-VM within the same organization. It is strongly recommended to change the initial password the first time you log in or activate the

license. You can also create other administrative users using more complex character strings than "admin" in a manner difficult to guess, or add two-factor authentication or other methods to secure login.

## AWS services and components

FortiGate-VM for AWS is an Elastic Compute Cloud (EC2) instance with an Elastic Block Store (EBS) volume attached. The following lists AWS services and components that you must understand when deploying FortiGate-VM for different purposes:



AWS EBS disk encryption is a host-based feature. The disk encryption is transparent to the FortiGate-VM OS. For EBS encryption, you can choose to enable the encryption at time of deployment. However, you must ensure the proper permissions and access to KMS or other keys when enabling encryption.

For information about EBS disk encryption, see [Amazon EBS encryption](#).



At this time, AWS does not support encrypting an existing FortiGate-VM's EBS disks. To take advantage of encrypted EBS volumes, encrypt the EBS volumes at the time of FortiGate-VM deployment.

## Ordinary FortiGate-VM single instance deployment or FortiGate-native active-passive high availability

| Service/component           | Description   |
|-----------------------------|---|
| Virtual private cloud (VPC) | This is where the FortiGate-VM and protected VMs are situated and users control the network. The public-facing interface is routed to the Internet gateway, which is created within the VPC.  |
| EC2                         | FortiGate-VM for AWS is an EC2 VM instance. Every instance has a unique instance ID.  |
| Subnets, route tables       | You must appropriately configure FortiGate-VM with subnets and route tables to handle traffic.  |
| Internet gateways           | AWS gateway as a VPC component that allows communication between instances in your VPC and the Internet.  |
| Elastic IP address          | You must allocate at least one public IP address to the FortiGate-VM to access and manage it over the Internet.   |
| Security groups             | AWS public-facing protection. Allow only necessary ports and protocols.   |
| AMI                         | Special deployable image type used on AWS. You can launch FortiGate-VM (bring your own license (BYOL)) directly from the publicly available FortiGate AMI instead of using the marketplace. See <a href="#">Deploying from BYOL AMI on page 31</a> .<br>The on-demand AMI is launchable but does not allow you to properly boot up as it is not intended to be deployed from AMI. |

| Service/component              | Description   |
|--------------------------------|---|
| CloudFormation Templates (CFT) | <p>FortiGate instances can be deployed using CFTs where tailor-made resource instantiation is defined. Fortinet provides CFTs for the following use cases:</p> <ul style="list-style-type: none"> <li>• Deploying FortiGate-native A-P HA</li> <li>• Customer-required scenarios with particular topologies</li> </ul> <p>CFTs are available on <a href="#">GitHub</a>.</p> <p>Fortinet-provided CFTs are not supported within the regular Fortinet technical support scope. Contact <a href="mailto:awssales@fortinet.com">awssales@fortinet.com</a> with questions.</p> |

## Additional or alternative HA using AWS mechanisms

| Service/component | Description   |
|-------------------|---|
| Auto Scaling      | <p>Auto scaling can automatically scale out by instantiating additional FortiGate-VM instances at times of high workloads. See <a href="#">Deploying auto scaling on AWS on page 43</a>.</p> <p>To run auto scaling, you must enable/subscribe to coexisting AWS services:</p> <ul style="list-style-type: none"> <li>• Route 53</li> <li>• API gateway</li> <li>• Load Balancer</li> <li>• CloudWatch</li> <li>• Lambda</li> <li>• SNS</li> <li>• DynamoDB</li> <li>• Simple Storage Services (S3) (BYOL only)</li> </ul> <p>These services are not always required for AWS auto scaling in general, but are predefined in Fortinet-provided Lambda scripts.</p> |
| Load Balancer     | <p>Also called Elastic Load Balancer (ELB). A network load balancer automatically distributes traffic across multiple FortiGate-VM instances when configured properly. Topologies differ depending on how you distribute incoming and outgoing traffic and cover AZs. Used in conjunction with auto scaling. See <a href="#">Deploying auto scaling on AWS on page 43</a>.</p>  |

## Monitoring

| Service/component | Description   |
|-------------------|---|
| CloudWatch        | <p>Monitoring service for various AWS resources. You can use CloudWatch in three scenarios with FortiGate-VM:</p> <ul style="list-style-type: none"> <li>• Monitor FortiGate-VM instance health and alert when needed.</li> <li>• Define auto scaling scale-out triggers to fire alarms</li> <li>• Monitor GuardDuty events</li> </ul> <p>You must subscribe to CloudWatch to use corresponding features.</p> |

## Related AWS services used as prerequisites for additional HA or extra features

| Service/component | Description   |
|-------------------|---|
| Lambda            | <p>AWS Lambda lets you run certain scripts and codes without provisioning servers. Fortinet provides Lambda scripts for:</p> <ul style="list-style-type: none"> <li>Running auto scaling</li> <li>GuardDuty integration</li> </ul> <p>To use the scripts, you must subscribe to Lambda. Fortinet-provided Lambda scripts are not supported within the regular Fortinet technical support scope. Contact <a href="mailto:awssales@fortinet.com">awssales@fortinet.com</a> with questions.</p>  |
| API Gateway       | <p>It acts as a front door by providing a callback URL for the FortiGate-VM to send its API calls and process FortiGate-VM config-sync tasks to synchronize OS configuration across multiple FortiGate-VM instances at the time of auto scaling scale-out. It is required if the config-sync feature needs to be incorporated into auto scaling.</p>  |
| DynamoDB          | <p>A handy flexible database. Fortinet-provided scripts use DynamoDB to store information about varying states of auto scaling conditions.</p>  |
| SNS               | <p>Managed message service used to communicate between AWS components. Fortinet-provided scripts use SNS to deliver subscription notifications from CFTs to Lambda for auto scaling.</p>  |
| GuardDuty         | <p>Managed threat detection service that monitors unwanted behaviors/activities related to AWS resources. Fortinet can leverage externally available lists of malicious IP addresses stored at certain locations. GuardDuty can be used to populate such a list. See <a href="#">Populating threat feeds with GuardDuty on page 175</a>. To use this feature, you must subscribe to GuardDuty.</p>  |
| S3                | <p>AWS storage. You can use S3 in four scenarios with FortiGate-VM:</p> <ul style="list-style-type: none"> <li>As the location where the list of blocklisted IP addresses is stored which is pointed by the FortiGate-VM in integrating with GuardDuty. See <a href="#">Populating threat feeds with GuardDuty on page 175</a>. You must allow the FortiGate-VM access to the S3 bucket/directory on S3 configuration.</li> <li>To store license keys which are parsed when provisioning additional FortiGate-VM instances in the event of auto scaling scale-out.</li> <li>To store a license key and the FortiGate-VM config file to bootstrap the FortiGate-VM at initial boot-up. See <a href="#">Bootstrapping the FortiGate-VM at initial bootup using user data on page 27</a>.</li> <li>To store license keys which are parsed when provisioning A-P HA.</li> </ul> |

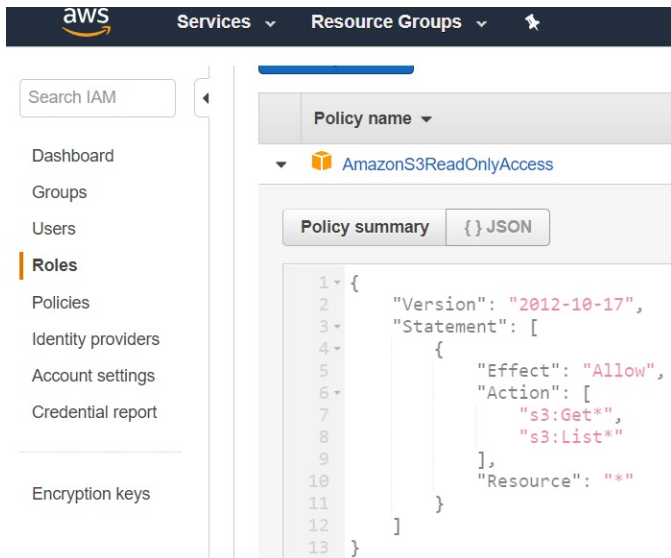
## Bootstrapping the FortiGate-VM at initial bootup using user data

If you are installing and configuring your applications on Amazon EC2 dynamically at instance launch time, you typically must pull and install packages, deploy files, and ensure services are started. The following bootstrapping instructions help simplify, automate, and centralize FortiGate-VM next generation firewall deployment directly from the configuration scripts stored in AWS S3. This is also called "cloud-init".

## Setting up IAM roles

Identity & Access Management roles need S3 bucket read access. This example applies the existing AmazonS3ReadOnlyAccess policy to the role by adding the following code or selecting S3ReadOnlyAccess from the policy list in adding to the role:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Resource": "*"
    }
  ]
}
```



See [IAM roles for Amazon EC2](#).

## Creating S3 buckets with license and firewall configurations

**To create S3 buckets with license and firewall configurations:**

1. On the AWS console, create an Amazon S3 bucket at the root level for the bootstrap files.
2. Upload the license file and configuration file(s) to the S3 bucket. This example uploads one license file and two configuration files. The example has the following FortiOS CLI command statement in the config file:

```
config sys global
  set hostname cloudinit
end
```

This is to set a hostname as part of initial configuration at first-time launch.

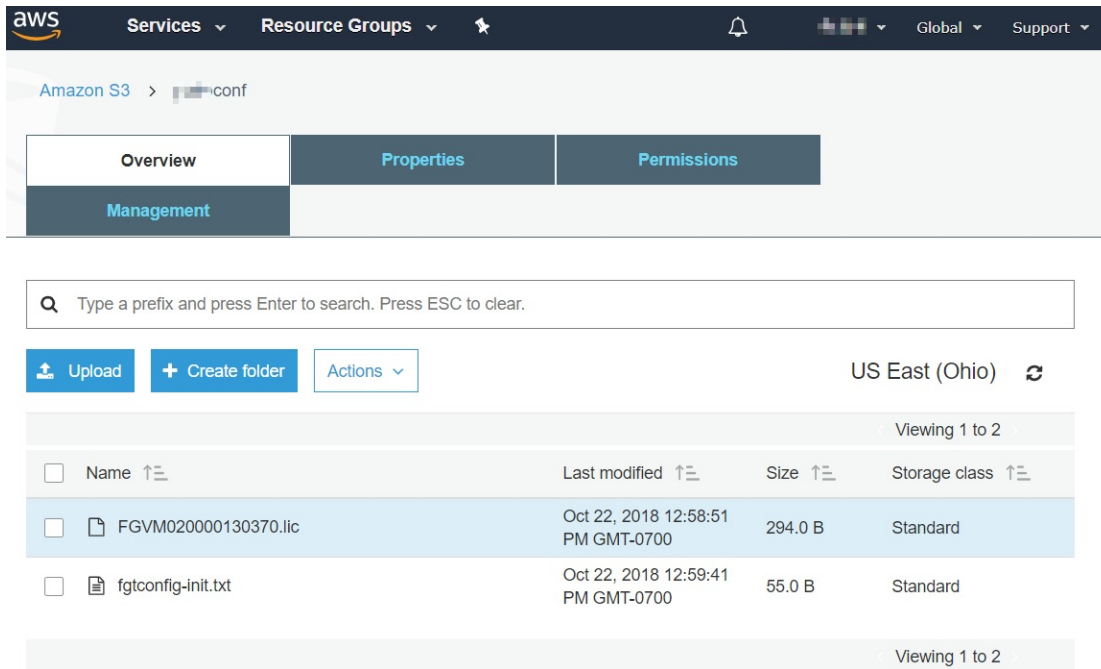
3. Amazon S3 creates the bucket in a region you specify. You can choose any AWS region that is geographically close to you to optimize latency, minimize costs, or address regulatory requirements. To choose a region, use the



following code:

```
{
  "bucket" : "conf",
  "region" : "us-east-2",
  "license" : "/FGVM020000130370.lic",
  "config" : "/fgtconfig-init.txt"
}
```

Although the S3 bucket and the firewall can be in different regions, keeping them in the same region is recommended to speed up the bootstrapping process.



## Launching the instance using roles and user data

Follow the normal procedure to launch the instance from the AWS marketplace.

When selecting the VPC subnet, the instance must with the role that was created and specify the information about the license file and configuration file from the AWS S3 bucket previously configured under *Advanced Settings*. In this example, the role name is ec2-s3.

**Step 3: Configure Instance Details**

Auto-assign Public IP Use subnet setting (Enable)

Placement group ☐ Add instance to placement group.

IAM role ec2-s3 [Create new IAM role](#)

Shutdown behavior Stop

Enable termination protection ☒ Protect against accidental termination

Monitoring ☒ Enable CloudWatch detailed monitoring  
[Additional charges apply.](#)

Tenancy Shared - Run a shared hardware instance   
[Additional charges will apply for dedicated tenancy.](#)

T2/T3 Unlimited ☒ Enable  
[Additional charges may apply](#)

▼ Advanced Details

User data ☒ As text ☐ As file ☐ Input is already base64 encoded

```
{
  "bucket": "s3://.conf",
  "region": "us-east-2",
  "license": "FGVM020000130370.lic",
  "config": "fgtconfig-init.txt"
}
```

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

After you launch the FortiGate-VM for the first time and log into the management GUI, FortiOS validates the license instead of displaying the license upload prompt.

**FortiGate VM License**

License is being validated by FortiGuard.

VM license has been validated. You need to re-login.

Username

Password

[Login](#)

After logging in, you can see that the license was activated and that the specified hostname was configured.

The screenshot displays the FortiGate-VM web interface with the following sections:

- System Information:**
  - Hostname: cloudinit
  - Serial Number: FGVM02
  - Firmware: v6.0.3 build0200 (GA)
  - Mode: NAT (Flow-based)
  - System Time: 2018/10/22 13:48:36
  - Uptime: 00:00:09:58
  - WAN IP: Unknown
- Licenses (209.222.136.7):**
  - FortiCare Support: ✓
  - Firmware & General Updates: ✓
  - IPS: ✓
  - AntiVirus: ✓
  - Web Filtering: ✓
  - FortiClient: 0 / 10 (0%)
  - FortiToken: 0 / 2 (0%)
- Virtual Machine:**
  - FGVM02 License: ✓
- FortiCloud:**
  - Status: ⚠ Not Activated

Check the serial number that you have with the license.

```
cloudinit # get system status
Version: FortiGate-VM64-AWS v6.0.3,build0200,181009 (GA)
Virus-DB: 1.00000(2018-04-09 18:07)
Extended DB: 1.00000(2018-04-09 18:07)
Extreme DB: 1.00000(2018-04-09 18:07)
IPS-DB: 6.00741(2015-12-01 02:30)
IPS-ETDB: 0.00000(2001-01-01 00:00)
APP-DB: 6.00741(2015-12-01 02:30)
INDUSTRIAL-DB: 6.00741(2015-12-01 02:30)
Serial-Number: FGVM020000130370
```

You can view the cloud-init log in *Log & Report > System Events*.

The screenshot shows the 'Log & Report > System Events' page. The main table lists events with columns for #, Date/Time, Level, User, and Message. A specific event is highlighted in yellow:

| #  | Date/Time      | Level | User  | Message  |
|----|----------------|-------|-------|--|
| 18 | 12 minutes ago | Info  | admin | User changed hostname global setting to jkatocloudinit from cloudinitd |

The 'Log Details' panel on the right shows the following information:

- General:**
  - Date: 2018/10/22
  - Time: 13:39:05
  - Virtual Domain: root
  - Log Description: Global setting changed
- Security:**
  - Level: Info
- Event:**
  - User Interface: cloudinitd
  - Message: User changed hostname global setting to jkatocloudinit from cloudinitd
- Other:**
  - Field: hostname
  - Old Value: FGVM020000130370
  - Log event original timestamp: 1540240745
  - Log ID: 32222
  - Sub Type: system
  - New Value: jkatocloudinit

## Deploying from BYOL AMI

You can deploy FortiGate-VM outside the marketplace launcher if you want to install it manually from the AMI for some reason, such as if your organization does not allow access to the AWS marketplace website. There are AMI images publicly available in various regions for the versions already listed in the marketplace. This deployment works only with AMI for bring your own license (BYOL) licensing. Deploying from AMI designed for on-demand is not supported.

If you want to install the latest FortiGate-VM versions immediately after release from Fortinet but you do not see them published in the marketplace or publicly available in the AWS portal, you can deploy older FortiGate-VM versions

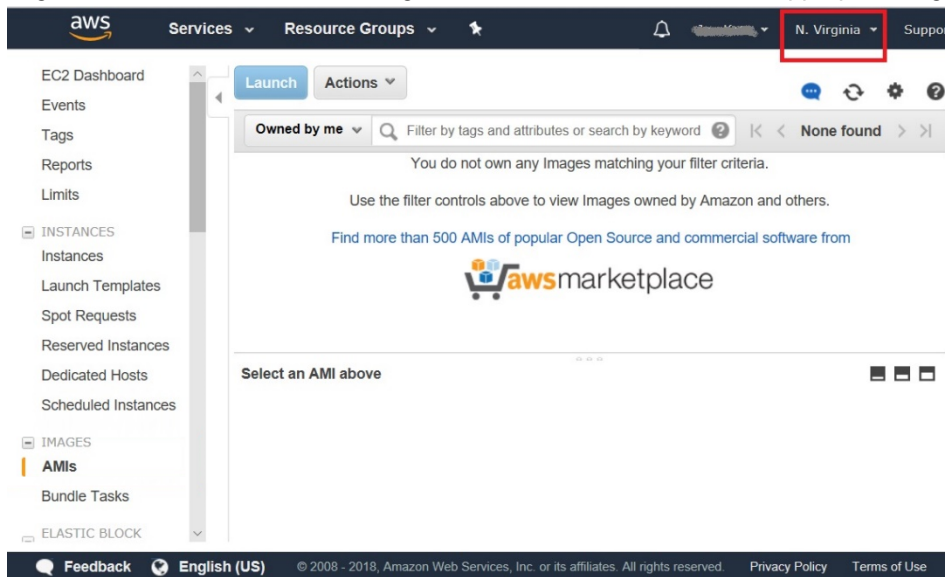
available on the marketplace or the AWS portal as publicly available AMIs, then upgrade using the .out upgrade files, which are available at [Customer Service & Support](#).



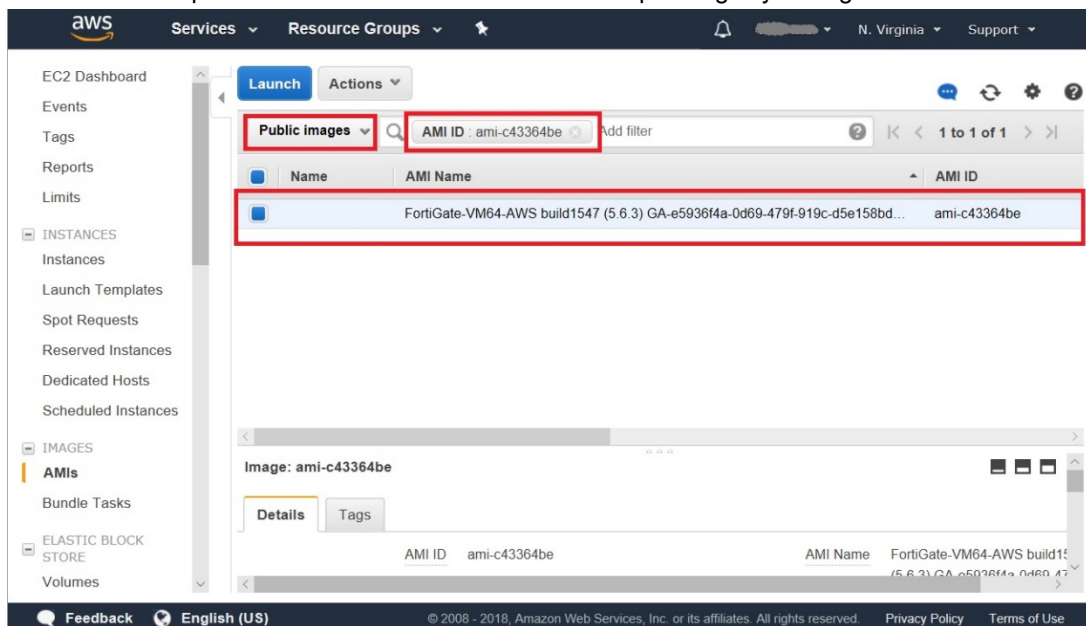
Downgrading to a previous GA version that does not support UEFI boot mode when using a UEFI-enabled FortiGate instance is not supported. FortiOS 7.4 supports UEFI boot mode.

### To deploy from BYOL AMI:

1. Select the desired AMI:
  - a. Log into the AWS EC2 console and go to **IMAGES > AMIs**. Select the appropriate region.



- b. Find the desired public AMI from the list of AMI IDs corresponding to your region.



- c. Select the AMI and click **Launch**.

2. Choose a supported instance. See [Instance type support on page 7](#).

| Instance type     | vCPUs       | Memory (GiB) | Storage | EBS only | Yes/No | Network bandwidth |
|-------------------|-------------|--------------|---------|----------|--------|-------------------|
| General purpose   | m4.16xlarge | 64           | 256     | EBS only | Yes    | 25 Gigabit        |
| Compute optimized | c5.large    | 2            | 4       | EBS only | Yes    | Up to 10 Gigabit  |
| Compute optimized | c5.xlarge   | 4            | 8       | EBS only | Yes    | Up to 10 Gigabit  |
| Compute optimized | c5.2xlarge  | 8            | 16      | EBS only | Yes    | Up to 10 Gigabit  |
| Compute optimized | c5.4xlarge  | 16           | 32      | EBS only | Yes    | Up to 10 Gigabit  |
| Compute optimized | c5.9xlarge  | 36           | 72      | EBS only | Yes    | 10 Gigabit        |
| Compute optimized | c5.18xlarge | 72           | 144     | EBS only | Yes    | 25 Gigabit        |
| Compute optimized | c4.large    | 2            | 3.75    | EBS only | Yes    | Moderate          |
| Compute optimized | c4.xlarge   | 4            | 7.5     | EBS only | Yes    | High              |
| Compute optimized | c4.2xlarge  | 8            | 15      | EBS only | Yes    | High              |

3. Click *Next: Configure Instance Details*.

4. Configure the instance details:

- In the *Network* field, select the VPC that you created.
- In the *Subnet* field, select the public subnet.
- In the *Network interfaces* section, you see the entry for eth0 that was created for the public subnet. Select *Add Device* to add another network interface (in this example, eth1), and select the private subnet. Assigning static IP addresses is recommended.
- When you have two network interfaces, a global IP address is not assigned automatically. You must manually assign a global IP address later. Select *Review and Launch*, then select *Launch*.

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 Launch into Auto Scaling Group

Purchasing option: ☐ Request Spot instances

Network:  Create new VPC

Subnet:  Create new subnet

Auto-assign Public IP:

IAM role:  Create new IAM role

Shutdown behavior:

Enable termination protection: ☐ Protect against accidental termination

Monitoring: ☐ Enable CloudWatch detailed monitoring

Tenancy:  Additional charges will apply for dedicated tenancy

T2 Unlimited: ☐ Enable

Network interfaces:

| Device | Network interface     | Subnet          | Primary IP  | Secondary IP addresses | IPv6 IPs |
|--------|-----------------------|-----------------|-------------|------------------------|----------|
| eth0   | New network interface | subnet-07a6517c | Auto-assign | Add IP                 |          |

Add Device

Advanced Details

5. Click *Next: Add Storage*. In *Step 4: Add Storage*, you can leave the fields as-is, or change the size of /dev/sdb as desired. The second volume is used for logging.

**Step 4: Add Storage**

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

| Volume Type | Device    | Snapshot               | Size (GiB) | Volume Type               | IOPS       | Throughput (MB/s) |
|-------------|-----------|------------------------|------------|---------------------------|------------|-------------------|
| Root        | /dev/sda1 | snap-0a2577b6234d25348 | 2          | General Purpose SSD (GP2) | 100 / 3000 | N/A               |
| EBS         | /dev/sdb  | Search (case-insensit) | 30         | Magnetic                  | N/A        | N/A               |

[Add New Volume](#)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Tags](#)

6. Click **Next: Add Tags**. You can add tags for convenient management.

**Step 5: Add Tags**

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

| Key (127 characters maximum) | Value (255 characters maximum)  | Instances                           | Volumes                             |
|------------------------------|---------------------------------|-------------------------------------|-------------------------------------|
| Name                         | jkato-FGT-5.6.3-Install-AMI-004 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

[Add another tag](#) (Up to 50 tags maximum)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Security Group](#)

7. Click **Next: Configure Security Groups**. Allowing some incoming ports is important. Allow TCP port 8443 for management from the GUI. You can also allow TCP port 22 for SSH login. Allow other ports where necessary as noted. The [FortiOS documentation](#) explains port use.

| Incoming TCP ports allowed | Purpose  |
|----------------------------|--|
| 22                         | SSH  |
| 443                        | Management using the GUI                       |
| 541                        | Management by FortiManager located outside AWS |
| 8000                       | Fortinet Single Sign On                        |
| 10443                      | SSLVPN   |



You can change the source address later.

**Step 6: Configure Security Group**

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name:

Description:

| Type  | Protocol | Port Range | Source                | Description |
|-------|----------|------------|-----------------------|-------------|
| SSH   | TCP      | 22         | Custom 0.0.0.0/0      | SSH         |
| HTTPS | TCP      | 443        | Custom 0.0.0.0/0, ::0 | GUI access  |

[Add Rule](#)

**Warning**  
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses.

[Cancel](#) [Previous](#) [Review and Launch](#)

8. Click **Review and Launch**. If everything looks good, go to next by clicking **Launch**.
9. Then select the appropriate keypair, then click **Launch Instance**. Deploying the instance may take 15 to 30 minutes. To access the FortiGate and complete post-install setup, see [Connecting to the FortiGate-VM on page 109](#).

**Step 7: Review Instance Launch**

**Select an existing key pair or create a new key pair**

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Choose an existing key pair

Select a key pair

☒ I acknowledge that I have access to the selected private key file (jtkato-Ohio.pem), and that without this file, I won't be able to log into my instance.

[Cancel](#) [Launch Instances](#)

## Deploying on AWS China

Deploying FortiGate-VM for AWS China has separate requirements than deploying FortiGate-VM for global AWS. To use AWS resources on AWS China, you must have an AWS China account separate from your global AWS account.

FortiGate-VM for AWS China only supports the bring your own license licensing model. To activate it, you must obtain a license. Complete the following steps to deploy FortiGate-VM on AWS China:

1. [Creating a support account on page 18](#)
2. [Creating a VPC and subnets on page 36](#)
3. [Attaching the new VPC Internet gateway on page 36](#)
4. [Launching the instance with shared FortiGate-VM AMI on page 37](#)
5. [Connecting to the FortiGate-VM on page 38](#)

## Creating a VPC and subnets

This section shows you how to create an AWS VPC and create two subnets in it. For many steps, you have a choice to make that can be specific to your own environment.

### To create a VPC and subnets:

1. Change your language to English and log into the [AWS Management Console](#).
2. Go to *Services > Networking > VPC*.
3. Go to *Virtual Private Cloud > Your VPCs*, then select *Create VPC*.
4. In the *Name tag* field, set the VPC name.
5. In the *CIDR block* field, specify an IPv4 address range for your VPC.
6. In the *Tenancy* field, select *Default*.
7. Select *Yes, Create*.
8. Go to *Virtual Private Cloud > Subnets*, then select *Create Subnet*. Create a public subnet (in this example, *Subnet1*) and a private subnet (*Subnet2*), as shown in this example. Both subnets belong to the VPC that you created.

The screenshot shows the 'Create Subnet' dialog box in the AWS Management Console. It includes a title bar with a close button. Below the title bar is a note: 'Use the CIDR format to specify your subnet's IP address block (e.g., 10.0.0.0/24). Note that block sizes must be between a /16 netmask and /28 netmask. Also, note that a subnet can be the same size as your VPC.' The form contains the following fields: 'Name tag' with the value 'Public-FortiGate', 'VPC' with a dropdown showing 'vpc-69522e0d | FortiGateVPC', 'Availability Zone' with a dropdown showing 'No Preference', and 'CIDR block' with the value '10.0.0.0/24'. At the bottom right, there are two buttons: 'Cancel' and 'Yes, Create' (which is highlighted with a red rectangle).

The screenshot shows the 'Create Subnet' dialog box in the AWS Management Console. It includes a title bar with a close button. Below the title bar is a note: 'Use the CIDR format to specify your subnet's IP address block (e.g., 10.0.0.0/24). Note that block sizes must be between a /16 netmask and /28 netmask. Also, note that a subnet can be the same size as your VPC.' The form contains the following fields: 'Name tag' with the value 'Private', 'VPC' with a dropdown showing 'vpc-69522e0d | FortiGateVPC', 'Availability Zone' with a dropdown showing 'No Preference', and 'CIDR block' with the value '10.0.1.0/24'. At the bottom right, there are two buttons: 'Cancel' and 'Yes, Create' (which is highlighted with a red rectangle).

## Attaching the new VPC Internet gateway

This section shows how to connect the new VPC to the Internet gateway. If you use the default VPC, the Internet gateway should already exist.

### To attach the new VPC Internet gateway:

1. Go to *Virtual Private Cloud > Internet Gateways*, then select *Create internet Gateway*.
2. In the *Name tag* field, set the Internet gateway name, then select *Create*.

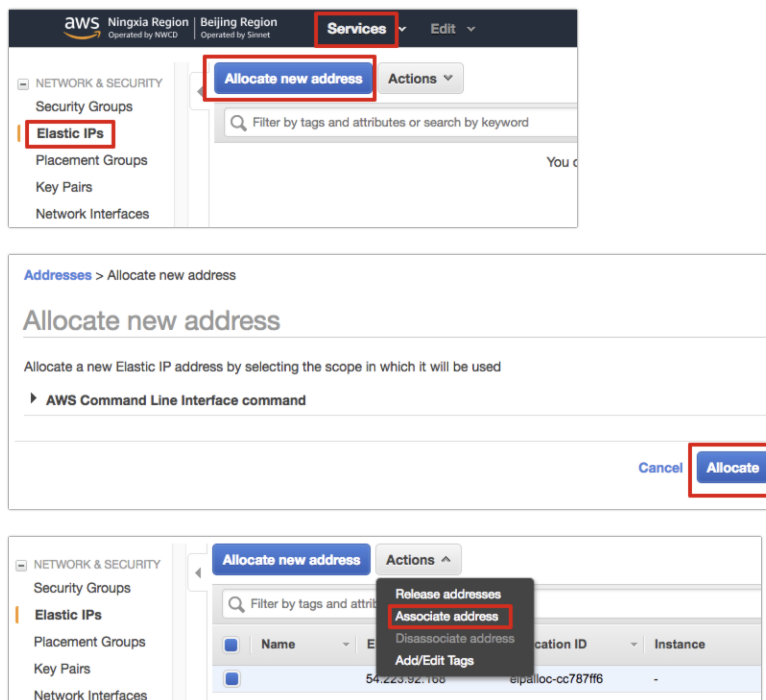


3. Select the Internet gateway, then select *Attach to VPC*.
4. Select the created VPC and select *Attach*. The Internet gateway state changes from *detached* to *attached*.

## Launching the instance with shared FortiGate-VM AMI

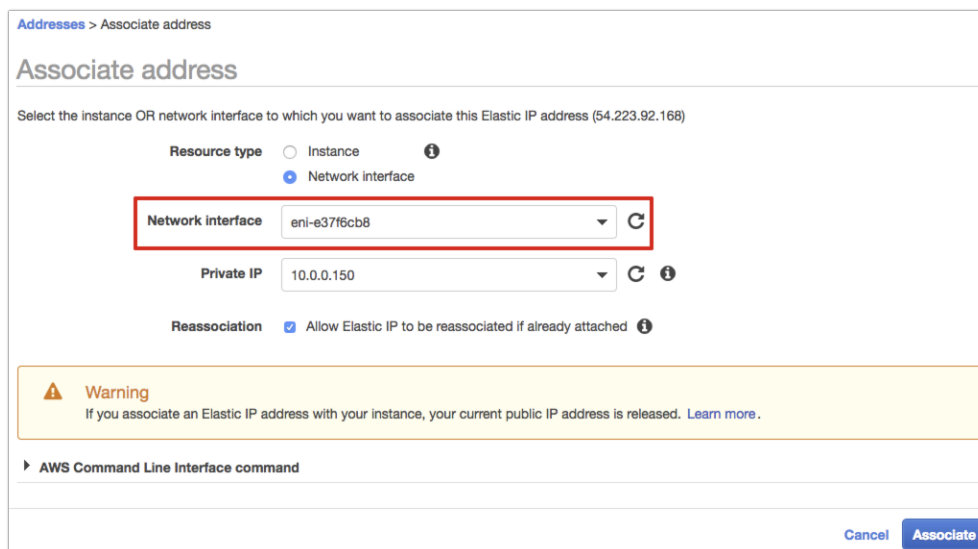
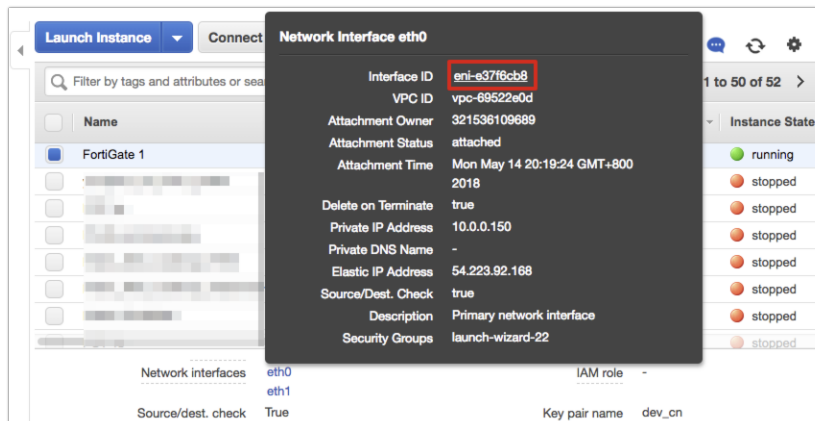
### To launch the instance with FortiGate-VM AMI:

1. In the *Services-EC2* Dashboard, go to *INSTANCES > Instances*, then select *Launch Instance*.
2. Select *AWS Marketplace*. Search for *FortiGate*. Click *Select*.
3. Select an instance type, then select *Next: Configure Instance Details*.
4. Configure the instance details:
  - a. In the *Network* field, select the VPC you created.
  - b. In the *Subnet* field, select the public subnet.
  - c. In the *Network interfaces* section, you see the entry for *eth0* that was created for the public subnet. Select *Add Device* to add another network interface (in this example, *eth1*), and select the private subnet.
  - d. When you have two network interfaces, a global IP address is not assigned automatically. You must manually assign a global IP address later. Select *Review and Launch*, then select *Launch*.
  - e. Select an existing key pair or create a new key pair. Select the acknowledgment checkbox. Select *Launch Instances*.
  - f. To easily identify the instance, set a name for it in the *Name* field.
  - g. Go to *NETWORK & SECURITY > Elastic IPs*, select a global IP address that is available for use. Select *Actions > Allocate new address*. If you do not have a global IP address available to use, create one.



- h. In the *Resource type* section, select *Network Interface*.
- i. In the *Network interface* field, select the Interface ID of the network interface that you created for the public subnet (in this example, *eth0*). In the *Private IP* field, select the IP address that belongs to the public subnet. To find these values, go to the EC2 Management Console, select *Instances*, and select the interface in the *Network interfaces* section in the lower pane of the page (*Interface ID* and *Private IP Address* fields). Select

**Associate.** A message displays indicating the address association succeeded. If you do not associate the Internet gateway with a VPC, the elastic IP address assignment fails.



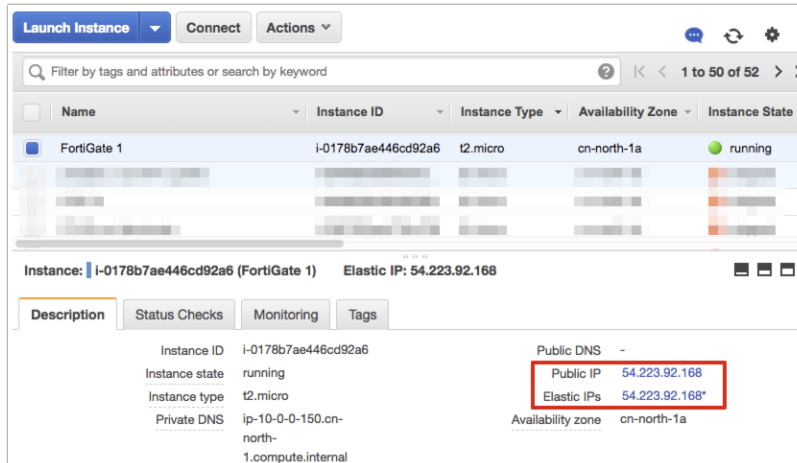
## Connecting to the FortiGate-VM

To connect to the FortiGate-VM, you need your login credentials, the FortiGate-VM's elastic IP address (EIP), SSH client, and an FTP server.

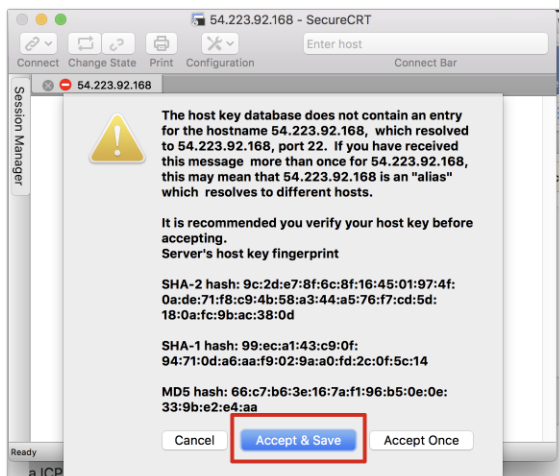
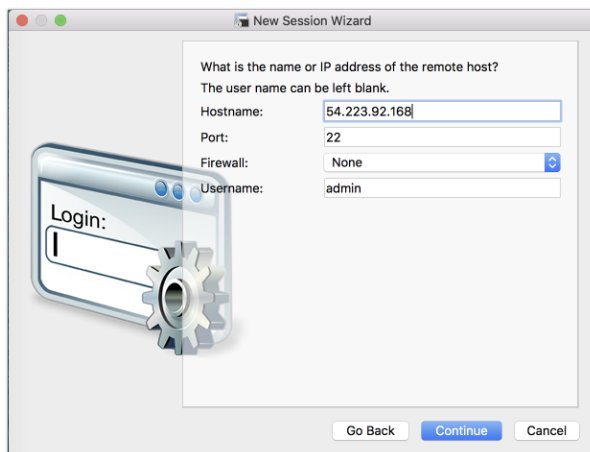
The default username is admin and the default password is the instance ID.

### To connect to the FortiGate-VM:

1. Find the public IP address in the EC2 management console. Select *Instances* and look at the *Public IP* field in the lower pane.



- Each public IP address in China should obtain an ICP license. Otherwise ports 80, 443, and 8080 cannot visit it. You cannot initially access the FortiGate-VM web GUI via the default HTTPS port. You can access the FortiGate-VM via SSH, then upload a BYOL license to the FortiGate-VM via FTP or TFTP. After activating the FortiGate-VM, you can modify the default admin HTTPS port to any port, such as 8443. Then you can go to the FortiGate-VM via `https://<FortiGate-VM EIP>:8443`.



The default password is the instance ID as shown.

| Name        | Instance ID         | Instance Type | Availability Zone | Instance State |
|-------------|---------------------|---------------|-------------------|----------------|
| FortiGate 1 | i-0178b7ae446cd92a6 | t2.micro      | cn-north-1a       | running        |

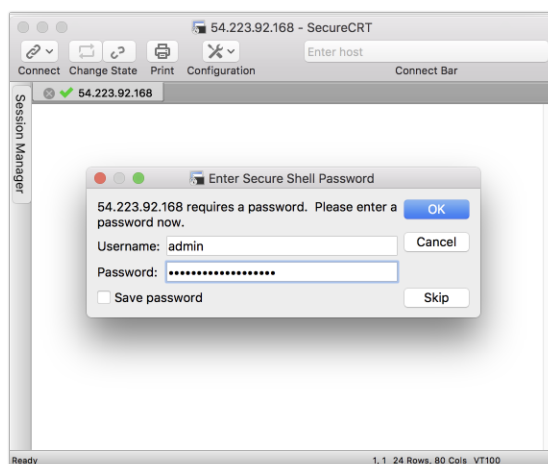
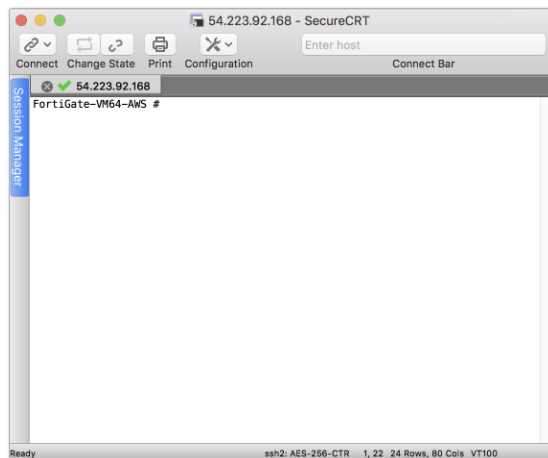
|   |  |                           |
|---|--|---------------------------|
| Instance: i-0178b7ae446cd92a6 (FortiGate 1) |  | Elastic IP: 54.223.92.168 |
|---|--|---------------------------|

|             |               |            |      |
|-------------|---------------|------------|------|
| Description | Status Checks | Monitoring | Tags |
|-------------|---------------|------------|------|

|                |                     |             |                |
|----------------|---------------------|-------------|----------------|
| Instance ID    | i-0178b7ae446cd92a6 | Public DNS  | -              |
| Instance state | running             | Public IP   | 54.223.92.168  |
| Instance type  | t2.micro            | Elastic IPs | 54.223.92.168* |



- Set up an FTP/TFTP server and ensure the FortiGate can log onto and download a BYOL license from it.
- On the FortiGate, use one of the following CLI commands to restore the VM license.

```
exec restore vmlicense tftp <license file name> <IP address>
exec restore vmlicense ftp <license name (path) on the remote server> <ftp server address>[:ftp port]
```

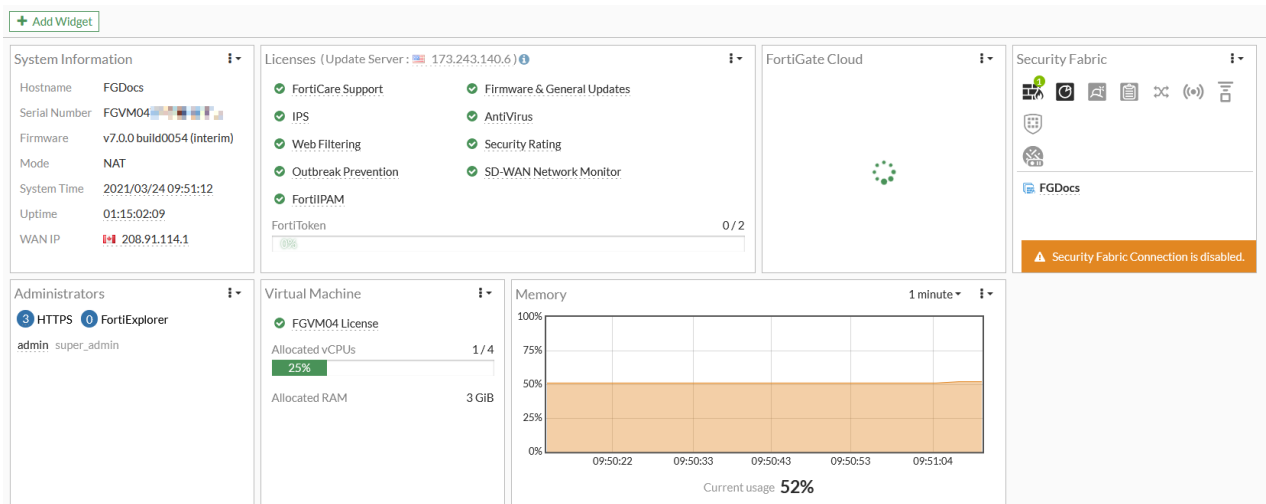
```
FortiGate-VM64-AWS # exec restore vmlicense tftp FGVM64-VM64-1.lic 168.168.1.3
This operation will overwrite the current VM license!
Do you want to continue? (y/n)
Please wait...
Connect to tftp server 168.168.1.3 ...
Get VM license from tftp server OK.
VM license install succeeded. Rebooting firewall.
```

If the license installation succeeds, the FortiGate-VM reboots automatically. After it restarts, log in.

5. Change the default port to any port, such as 8443. Do not use ports 443, 8080, or 80.

```
FGVM04@FGVM04:~$ # config system global
FGVM04@FGVM04:~$ (global) # set admin-sport 8443
FGVM04@FGVM04:~$ (global) # end
```

You now see the FortiGate-VM dashboard. Depending on your license type, the information in the license widget on the dashboard may vary.



6. Select **Network > Interfaces**, and edit the interfaces, if required. If the IP address or subnet mask is missing for port 1 or port 2, configure these values.

The screenshot shows the 'Edit Interface' configuration window for 'port2'. The interface is a physical interface with VRF ID 0 and an undefined role. Under the 'Address' section, the addressing mode is set to 'Manual' with an IP/Netmask of '10.0.1.5/24'. IPv6 addressing is also set to 'Manual' with an address/prefix of '::/0'. Administrative access is configured for both IPv4 and IPv6, with HTTPS, HTTP, SSH, and PING enabled. LLDP settings are set to 'Use VDOM Setting'. DHCP and SLAAC services are disabled.

| Section  | Field  | Value  |
|----------|--|--|
| General  | Name   | port2  |
|          | Alias  |  |
|          | Type   | Physical Interface   |
|          | VRF ID                                       | 0  |
| Role     | Role   | Undefined  |
|          | Role   | Undefined  |
| Address  | Addressing mode                              | Manual   |
|          | IP/Netmask                                   | 10.0.1.5/24  |
|          | IPv6 addressing mode                         | Manual   |
|          | IPv6 Address/Prefix                          | ::/0   |
|          | Auto configure IPv6 address                  | Off  |
|          | DHCPv6 prefix delegation                     | Off  |
|          | Secondary IP address                         | Off  |
|          | Administrative Access                        |  |
|          | IPv4   | <input checked="" type="checkbox"/> HTTPS<br><input type="checkbox"/> FMG-Access<br><input type="checkbox"/> FTM |
|          | IPv6   | <input type="checkbox"/> HTTPS<br><input type="checkbox"/> SSH   |
| LLDP     | Receive LLDP                                 | Use VDOM Setting   |
|          | Transmit LLDP                                | Use VDOM Setting   |
| Services | DHCP Server                                  | Off  |
|          | Stateless Address Auto-configuration (SLAAC) | Off  |
|          | DHCPv6 Server                                | Off  |

## Upgrading the FortiGate-VM

For the recommended upgrade path, see the [Upgrade Path Tool Table](#). Select *FortiGate-VM-AWS* or *FortiGate-VM-AWSONDEMAND*, and the current and target upgrade versions.

For upgrade instructions, see [Upgrading the individual device firmware](#).

## Backing up and restoring configuration

See [Configuration backups](#).

# Deploying auto scaling on AWS

You can deploy FortiGate virtual machines (VMs) to support Auto Scaling on AWS. Optionally, AWS Transit Gateway can be used to connect Amazon Virtual Private Clouds (Amazon VPCs) and their on-premises networks to a single gateway. This integration extends the FortiGate protection to all networks connected to the Transit Gateway. Consolidate logging and reporting for your FortiGate cluster by integrating FortiAnalyzer. Fortinet provides FortiGate Autoscale for AWS deployment packages to facilitate the deployment.

Multiple FortiGate-VM instances form an Auto Scaling group to provide highly efficient clustering at times of high workloads. FortiGate-VM instances can be scaled out automatically according to predefined workload levels. When a spike in traffic occurs, FortiGate-VM instances are automatically added to the Auto Scaling group. Auto Scaling is achieved by using FortiGate-native High Availability (HA) features that synchronize operating system (OS) configurations across multiple FortiGate-VM instances at the time of scale-out events.

FortiGate Autoscale for AWS is available with FortiOS 6.2.5, FortiOS 6.4.6, FortiOS 7.0.0, and FortiOS 7.0.1 and supports any combination of On-demand and Bring Your Own License (BYOL) instances. FortiAnalyzer 6.4.6 can be incorporated into Fortinet FortiGate Autoscale to use extended features that include storing logs into FortiAnalyzer.



Fees are incurred based on the Amazon Elastic Compute Cloud (Amazon EC2) instance type. Additionally, a license is required for each FortiGate Bring Own License (BYOL) instance you might use.

---

FortiGate Autoscale for AWS uses AWS CloudFormation Templates (CFTs) to deploy components.

Deployments without Transit Gateway integration have:

- A highly available architecture that spans two Availability Zones.\*
- An Amazon VPC configured with public and private subnets according to AWS best practices, to provide you with your own virtual network on AWS.\*
- An Internet gateway to allow access to the Internet.\*
- In the public subnets:
  - (Optional) A FortiAnalyzer instance, which consolidates logging and reporting for your FortiGate cluster.
  - Two or more FortiGate-VM instances, which complement AWS security groups. FortiGates provide intrusion protection, web filtering, and threat detection to help protect your services from cyberattacks. Each instance also provides VPN access for authorized users. VPN connections use the Diffie-Hellman Group 14 and SHA256 (Secure Hash Algorithm 2).
  - A cluster of FortiGate-VM instances in the Auto Scaling groups, where one FortiGate-VM acts as the primary while the others act as the secondary. The primary FortiGate-VM also acts as the NAT gateway by default, allowing egress Internet access for resources in the private subnets.
- A public-facing network load balancer that distributes inbound traffic across the FortiGate-VM instances. An internal-facing network load balancer is optional.
- AWS Lambda, which provides the core Auto Scaling functionality between FortiGates-VM instances.
- Amazon Simple Storage Service (Amazon S3) to host artifacts for Lambda functions and logs.
- Amazon DynamoDB to store information about Auto Scaling condition states.

\* Deployment into an existing VPC does not create the marked components in the list. You are prompted for your existing VPC configuration.

Deployments with Transit Gateway integration have:

- A highly available architecture that spans two Availability Zones.
- An Amazon VPC configured with public and private subnets according to AWS best practices, to provide you with your own virtual network on AWS.
- An Internet gateway to allow access to the Internet.
- In the public subnets:
  - (Optional) A FortiAnalyzer instance, which consolidates logging and reporting for your FortiGate cluster.
  - Two or more FortiGate-VM instances, which complement AWS security groups. FortiGates provide intrusion protection, web filtering, and threat detection to help protect your services from cyberattacks. Each instance also provides VPN access for authorized users. VPN connections use the Diffie-Hellman Group 14 and SHA256 (Secure Hash Algorithm 2).
  - A primary FortiGate-VM instance in the Auto Scaling group(s).
- AWS Lambda, which provides the core Auto Scaling functionality between FortiGate-VM instances.
- Amazon Simple Storage Service (Amazon S3) to host artifacts for Lambda functions and logs.
- Amazon DynamoDB to store information about Auto Scaling condition states.
- Site-to-Site VPN connections.

## Planning

This deployment requires familiarity with the configuration of a FortiGate using the CLI as well as with the following AWS services:

- [Amazon Elastic Cloud Compute \(Amazon EC2\)](#)
- [Amazon EC2 Auto Scaling](#)
- [Amazon VPC](#)
- [AWS CloudFormation](#)
- [AWS Lambda](#)
- [Amazon DynamoDB](#)
- [Amazon API Gateway](#)
- [Amazon CloudWatch](#)
- [Amazon S3](#)

Deployments with Transit Gateway integration require knowledge of the following:

- [AWS Transit Gateway](#)
- Border Gateway Protocol (BGP)
- Equal-cost multi-path (ECMP)

If you are new to AWS, go to the [Getting Started Resource Center](#) and the [AWS Training and Certification website](#).

It is expected that DevOps engineers or advanced system administrators who are familiar with the listed items will deploy FortiGate Autoscale for AWS.

## Technical requirements

To start the deployment, you must have an AWS account. If you do not already have one, create one at <https://aws.amazon.com/> by following the on-screen instructions. Part of the sign-up process involves receiving a phone



call and entering a PIN. Your AWS account is automatically signed up for all AWS services. You are charged only for the services you use.

Log into your AWS account and verify the following:

- **IAM permissions.** Ensure that the AWS user deploying the template has sufficient permissions to perform the required service actions on resources. At a minimum, the following are required: *Service:* IAM; *Actions:* CreateRole; *Resource:* \*. The FortiGate Autoscale for AWS template increases the security level of the deployment stack by narrowing down the scope of access to external resources belonging to the same user account as well as restricting access to resources within the deployment.
- **Region.** Use the region selector in the navigation bar to choose the AWS region where you want to deploy FortiGate Autoscale for AWS.



This deployment includes *AWS Auto Scaling*, which currently not all AWS regions support. For a current list of supported regions, see the AWS documentation [Service Endpoints and Quotas](#).

---

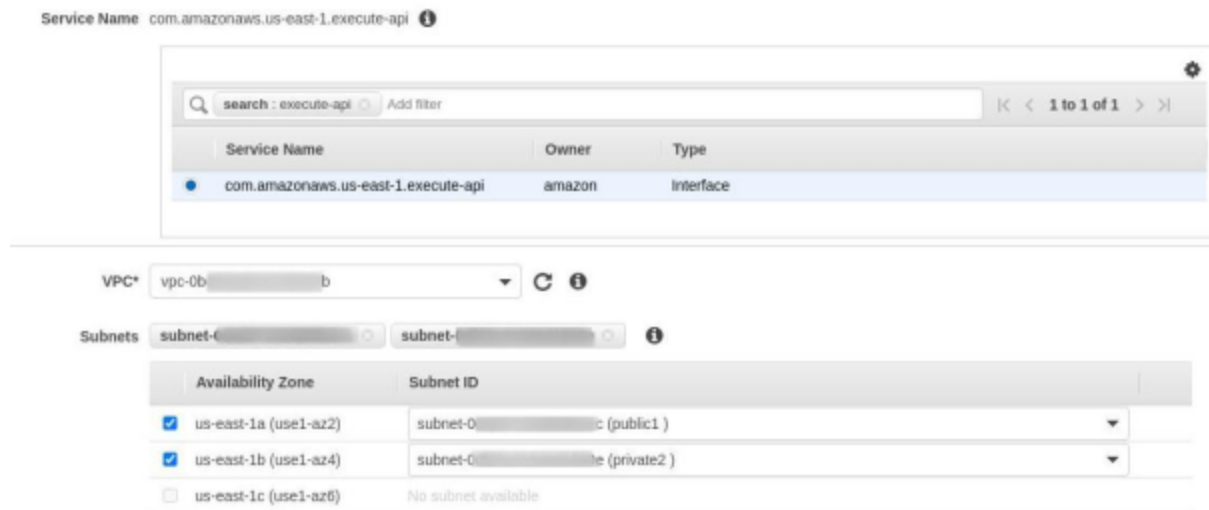
- **Instance Type.** This deployment offers a range of instance types, some of which not all AWS regions support. Ensure that your desired instance type is available in your region by checking the [Instance types](#) page for your region.
- **FortiGate subscription(s).** Confirm that you have a valid subscription to the [On-demand FortiGate](#) and/or [BYOL FortiGate](#) marketplace listings, as required for your deployment.
  - If you are not subscribed, open the subscription page and click *Continue to Subscribe*.
  - Review the terms and conditions for software usage, and then choose *Accept Terms*. A confirmation page loads, and an email confirmation is sent to the account owner.
  - Exit out of AWS Marketplace without further action. Do not provision the software from AWS Marketplace.
- **Key pair.** Ensure at least one Amazon EC2 [key pair](#) exists in your AWS account in the region where you plan to deploy FortiGate Autoscale for AWS. Make note of the key pair name.
- **Resources.** If necessary, request [service quota increases](#). This is necessary when you might exceed the default quotas with this deployment. The [Service Quotas console](#) displays your usage and quotas for some aspects of some services. For more information, see the [AWS documentation](#). The default instance type is *c5.large*.
- **FortiGate licenses.** Ensure you have a license for each FortiGate BYOL instance you might use. Licenses can be purchased from FortiCare. In the section [BYOL license files on page 48](#), you place the license files in an S3 bucket for use by the deployment.

## Requirements when using an existing VPC

When using an existing VPC, there are additional requirements:

- The VPC must have the option *DNS hostnames* enabled.
- Each of the two Availability Zones in the VPC must have at least 1 public subnet and at least 1 private subnet.

- A VPC Endpoint for the `execute-api` service under the *AWS services* category is required. This VPC Endpoint must have the *Private DNS Name* option enabled and must be associated with the VPC:



After deployment, you must associate the created security group with the VPC endpoint. For details, see [Post-deployment activities on page 64](#).

## Obtaining the deployment package





The FortiGate Autoscale for AWS deployment package is located in the Fortinet [GitHub project](#).

To obtain the deployment package, use one of the following:

- Download the package *aws-cloudformation.zip* directly from the [GitHub project release page](#).
- Manually generate the deployment package in your local workspace:
  - a. From the [GitHub project release page](#), download the source code (.zip or .tar.gz) for the latest version.
  - b. Extract the source code into the project directory in your local workspace.
  - c. Run `npm install` to initialize the project at the project root directory.
  - d. Run `npm run build-artifacts` to generate the local deployment package.  
The deployment package *aws-cloudformation.zip* is available in the *dist/artifacts* directory.

Once you have the deployment package *aws-cloudformation.zip*:

1. Unzip the file on your local PC. The following files and folders are extracted:

| Name   | Size      | Modified |
|--|-----------|----------|
|  <b>assets</b>    | 1 item    | 20:09    |
|  <b>functions</b> | 1 item    | 20:09    |
|  <b>README.md</b> | 609 bytes | 20:09    |
|  <b>templates</b> | 14 items  | 20:09    |

2. Log into your AWS account.
3. In the Amazon S3 service, create an S3 bucket as the root folder for your deployment. In the example below, the folder is named *fortigate-autoscale*.
4. Inside this folder, create another folder to store the deployment resources. In the example below, this folder is named *deployment-package*.
5. Navigate to this second folder and upload the files and folders you extracted in step 2 to this location. In the example below, we navigate to *Amazon S3 > fortigate-autoscale > deployment-package*.

Amazon S3 > fortigate-autoscale > deployment-package

fortigate-autoscale

Overview

 Type a prefix and press Enter to search. Press ESC to clear.



Upload



Create folder





Download

Actions 

US West (Oregon)



Viewing 1 to 4

| <input type="checkbox"/> | Name ▼  | Last modified ▼                  | Size ▼  | Storage class ▼ |
|--------------------------|---|----------------------------------|---------|-----------------|
| <input type="checkbox"/> |  assets    | --                               | --      | --              |
| <input type="checkbox"/> |  functions | --                               | --      | --              |
| <input type="checkbox"/> |  templates | --                               | --      | --              |
| <input type="checkbox"/> |  README.md | Aug 26, 2020 2:22:46 PM GMT-0700 | 609.0 B | Standard        |



This *assets* folder contains configuration files that you can modify to meet your network requirements. See [Major components on page 83](#) > *The "assets" folder in the S3 bucket.*

## BYOL license files

If you will be using BYOL instances, the deployment package will look for FortiGate license files in a location that ends with *license-files* > *fortigate*. This location can be created within the *assets* folder of the deployment package location or within a custom asset location.



If a custom asset location is used, you must specify the location in the parameters described in the table [Custom asset location configuration on page 60](#).

Examples:

- If the deployment package is located at *Amazon S3* > *fortigate-autoscale* > *deployment-package*, license files would be uploaded to *Amazon S3* > *fortigate-autoscale* > *deployment-package* > *assets* > *license-files* > *fortigate*.
- If you will be storing license files in a custom S3 location and you have created the S3 bucket *custom-s3-bucket-name* with the directory *custom-asset-directory*, you would upload the license files to *Amazon S3* > *custom-s3-bucket-name* > *custom-asset-directory* > *license-files* > *fortigate*.


## Deploying the CloudFormation templates

FortiGate Autoscale for AWS can be deployed:

- *with Transit Gateway integration* (using a new Transit Gateway or integrating with your existing Transit Gateway). This option builds a new AWS environment consisting of the VPC, subnets, security groups, and other infrastructure components. It then deploys FortiGate Autoscale into this new VPC and attaches this new VPC to the Transit Gateway.
- *without Transit Gateway integration*. This option allows for deployment into a new VPC or into an existing VPC.

## Deployment notes

| Deployment option                               | Notes  |
|---|--|
| with Transit Gateway integration (new VPC only) | One inbound route domain and one outbound route domain will be created for the new or existing Transit Gateway. FortiGate Autoscale for AWS will be attached to the Transit Gateway. |

| Deployment option    | Notes   |
|----------------------|---|
| into an existing VPC | <ul style="list-style-type: none"> <li>Incoming requests go through a connection that flows through the internet gateway, Network Load Balancer, and FortiGate Auto Scaling group before reaching the protected instances in the private subnets in your existing VPC. The protected instances return a response using the same connection.</li> <li>Outgoing requests from the protected instances go through one FortiGate-VM instance in an Auto Scaling group and the internet gateway to the public network. The public network returns the response using the same path.</li> </ul> |
|                      | <ul style="list-style-type: none"> <li>  FortiGate Autoscale manages the 0.0.0.0/0 route for overall egress traffic. For details on using other NAT gateways, see <a href="#">How to partially route egress traffic on page 85</a>. </li> </ul>  |

### To deploy the CloudFormation templates:

- Navigate to the S3 folder you uploaded files to in the previous section. In the example below, we navigate to *Amazon S3 > fortigate-autoscale > deployment-package*.
- Click *templates* and select the appropriate entry template to start the deployment. To deploy:
  - with Transit Gateway integration, click `autoscale-tgw-new-vpc.template.yaml`
  - without Transit Gateway integration, click `autoscale-new-vpc.template.yaml` to deploy into a new VPC
  - without Transit Gateway integration, click `autoscale-existing-vpc.template.yaml` to deploy into an existing VPC

Amazon S3 > fortigate-autoscale > deployment-package > templates

fortigate-autoscale

Overview

Q Type a prefix and press Enter to search. Press ESC to clear.

prefix autoscale X

Upload Create folder Download Actions

US West (Oregon)

Viewing 1 to 4

| <input type="checkbox"/> | Name                                 | Last modified                     | Size    | Storage class |
|--------------------------|--------------------------------------|-----------------------------------|---------|---------------|
| <input type="checkbox"/> | autoscale-existing-vpc.template.yaml | Aug 17, 2020 10:46:28 PM GMT-0700 | 41.4 KB | Standard      |
| <input type="checkbox"/> | autoscale-tgw-new-vpc.template.yaml  | Aug 17, 2020 10:46:27 PM GMT-0700 | 44.0 KB | Standard      |
| <input type="checkbox"/> | autoscale-new-vpc.template.yaml      | Aug 17, 2020 10:46:27 PM GMT-0700 | 47.5 KB | Standard      |
| <input type="checkbox"/> | autoscale-main.template.yaml         | Aug 17, 2020 10:46:27 PM GMT-0700 | 78.3 KB | Standard      |

3. Copy the *Object URL* of the template you picked in the previous step. In our example, the template chosen is for deploying into a new VPC.

[Amazon S3](#) > [fortigate-autoscale](#) > [deployment-package](#) > [templates](#) > [autoscale-new-vpc.template](#)

autoscale-new-vpc.template

Latest version ▼

Overview

Properties

Permissions

Select from

Open

Download

Download as

Make public

Copy path

Owner

[7f85a2f1-4a8a2a71-c38538a038f77e4d8528a0379a](#)

Last modified

Aug 17, 2020 10:46:27 PM GMT-0700

Etag

[7f85a2f1-4a8a2a71-c38538a038f77e4d8528a0379a](#)

Storage class

Standard

Server-side encryption

None

Size

47.5 KB

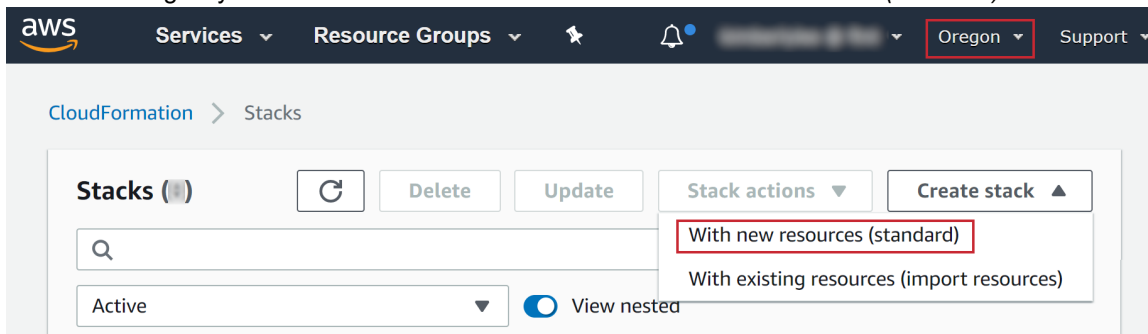
Key

deployment-package/templates/autoscale-new-vpc.template.yaml

Object URL

<https://fortigate-autoscale.s3-us-west-2.amazonaws.com/deployment-package/templates/autoscale-new-vpc.template.yaml>

4. Click *Services*, and then *Management & Governance* > *CloudFormation*.
5. Confirm the region you are in and then click *Create Stack* > *With new resources (standard)*.



6. Paste the *Object URL* from step 3 into the *Amazon S3 URL* field as shown below.

CloudFormation > Stacks > Create stack

## Create stack

### Prerequisite - Prepare template

**Prepare template**  
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

☒ Template is ready ☐ Use a sample template ☐ Create template in Designer

### Specify template

A template is a JSON or YAML file that describes your stack's resources and properties.

**Template source**  
Selecting a template generates an Amazon S3 URL where it will be stored.

☒ Amazon S3 URL ☐ Upload a template file

Amazon S3 URL

`https://fortigate-autoscale.s3-us-west-2.amazonaws.com/deployment-package/templates/autoscale-new-vpc.templ`

Amazon S3 template URL

S3 URL: `https://fortigate-autoscale.s3-us-west-2.amazonaws.com/deployment-package/templates/autoscale-new-vpc.template.yaml` [View in Designer](#)

Cancel [Next](#)

7. Click *Next*.
8. On the *Specify stack details* page, enter a stack name and review parameters for the template, providing values for parameters that require input.

CloudFormation &gt; Stacks &gt; Create stack

## Specify stack details

## Stack name

Stack name

*Enter a stack name*

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

## Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

## Resource tagging configuration

## Resource tag prefix

The ResourceGroup Tag Key used on all resources and as the name prefix of all applicable resources. Can only contain numbers, lowercase letters, uppercase letters, ampersat(@), hyphens (-), period (.), and hash (#). Max length is 50.

## Resource name prefix

An alternative name prefix to be used on a resource that the 'Resource tag prefix' cannot apply to. Can only contain numbers, lowercase letters, and uppercase letters. Max length is 10.

fgtASG

## Network configuration

## CFT parameters

The following sections provide descriptions of the available parameters. Some parameters are specific to certain templates, and are only displayed when that template is selected.

After entering required parameters, click *Next*.

## Resource tagging configuration

| Parameter label<br>(name)                  | Default               | Description  |
|--|-----------------------|--|
| Resource tag prefix<br>(ResourceTagPrefix) | <i>Requires input</i> | <i>ResourceGroup</i> Tag Key used on all resources and as the name prefix of all applicable resources. Can only contain uppercase letters, lowercase letters, numbers, ampersand (@), hyphens (-), period (.), and hash (#). Maximum length is 50. |
| Resource name prefix<br>(CustomIdentifier) | fgtASG                | Alternative name prefix to be used on a resource that the <i>Resource tag prefix</i> cannot apply to. Can only contain uppercase letters, lowercase letters, and numbers. Maximum length is 10.  |



## Network configuration (New VPC, no Transit Gateway)

| Parameter label (name)                          | Default               | Description  |
|---|-----------------------|--|
| Availability Zones<br>(AvailabilityZones)       | <i>Requires input</i> | List of Availability Zones to use for the subnets in the VPC. The FortiGate Autoscale solution uses two Availability Zones from your list and preserves the logical order you specify. |
| VPC CIDR (VpcCidr)                              | 192.168.0.0/16        | Classless Inter-Domain Routing (CIDR) block for the FortiGate Autoscale VPC.   |
| Autoscale subnet 1 CIDR<br>(PublicSubnet1Cidr)  | 192.168.0.0/24        | CIDR block for the subnet located in Availability Zone 1 where FortiGate Autoscale instances will be deployed to.  |
| Autoscale subnet 2 CIDR<br>(PublicSubnet2Cidr)  | 192.168.1.0/24        | CIDR block for the subnet located in Availability Zone 2 where FortiGate Autoscale instances will be deployed to.  |
| Protected subnet 1 CIDR<br>(PrivateSubnet1Cidr) | 192.168.2.0/24        | CIDR block for the private subnet located in Availability Zone 1 where it is protected by the FortiGate-VMs in the public subnet of the same Availability Zone.                        |
| Protected subnet 2 CIDR<br>(PrivateSubnet2Cidr) | 192.168.3.0/24        | CIDR block for the private subnet located in Availability Zone 2 where it is protected by the FortiGate-VMs in the public subnet of the same Availability Zone.                        |

## Network configuration (Existing VPC, no Transit Gateway)


| Parameter label (name)                   | Default               | Description  |
|--|-----------------------|--|
| VPC ID (VpcId)                           | <i>Requires input</i> | ID of the existing VPC where FortiGate Autoscale will be deployed. The VPC must have the option <i>DNS hostnames</i> enabled and each of the two Availability Zones in the VPC must have at least 1 public subnet and at least 1 private subnet. |
| VPC CIDR (VpcCidr)                       | <i>Requires input</i> | CIDR block of the selected existing VPC into which FortiGate Autoscale will be deployed. This can be found in parentheses in the VPC ID parameter selection.   |
| Private VPC Endpoint<br>(VpcEndpointId)  | <i>Requires input</i> | ID of the Private VPC Endpoint associated with the existing VPC. A Private VPC Endpoint is required for FortiGate Autoscale and is a VPC Endpoint that has enabled <i>Private DNS names</i> .  |
| Autoscale subnet 1 ID<br>(PublicSubnet1) | <i>Requires input</i> | ID of the public subnet 1 located in Availability Zone 1 of the selected existing VPC. The FortiGate Autoscale instances will be deployed here.  |
| Autoscale subnet 2 ID<br>(PublicSubnet2) | <i>Requires input</i> | ID of the public subnet 2 located in Availability Zone 2 of the selected existing VPC. The FortiGate Autoscale instances will be deployed here.  |




| Parameter label (name)                                  | Default               | Description  |
|---|-----------------------|--|
| Private subnet 1 ID<br>(PrivateSubnet1)                 | <i>Requires input</i> | ID of the private subnet 1 located in Availability Zone 1 of the selected existing VPC. This subnet will be protected by the FortiGate-VMs in the public subnet of the same Availability Zone. |
| Private subnet 2 ID<br>(PrivateSubnet2)                 | <i>Requires input</i> | ID of the private subnet 2 located in Availability Zone 2 of the selected existing VPC. This subnet will be protected by the FortiGate-VMs in the public subnet of the same Availability Zone. |
| Private subnet route table<br>(PrivateSubnetRouteTable) | <i>Requires input</i> | ID of the route table associated with the two private subnets.   |

## Network configuration (Transit Gateway integration)

| Parameter label (name)                         | Default               | Description  |
|--|-----------------------|--|
| Availability Zones<br>(AvailabilityZones)      | <i>Requires input</i> | The list of Availability Zones to use for the subnets in the VPC. The FortiGate Autoscale solution uses two Availability Zones from your list and preserves the logical order you specify. |
| VPC CIDR (VpcCidr)                             | 192.168.0.0/16        | The Classless Inter-Domain Routing (CIDR) block for the FortiGate Autoscale VPC.   |
| Autoscale subnet 1 CIDR<br>(PublicSubnet1Cidr) | 192.168.0.0/24        | The CIDR block for the subnet located in Availability Zone 1 where FortiGate Autoscale instances will be deployed to.  |
| Autoscale subnet 2 CIDR<br>(PublicSubnet2Cidr) | 192.168.1.0/24        | The CIDR block for the subnet located in Availability Zone 2 where FortiGate Autoscale instances will be deployed to.  |



## FortiGate configuration

| Parameter label (name)   | Default               | Description   |
|--|-----------------------|---|
| Instance type<br>(FortiGateInstanceType)   | c5.large              | Instance type for the FortiGate-VMs in the Auto Scaling group. There are t2.small and compute-optimized instances such as c4 and c5 available with different vCPU sizes and bandwidths. For more information about instance types, see <a href="#">Instance Types</a> . |
| FortiOS version<br>(FortiOSVersion)  | 7.0.1                 | FortiOS version supported by FortiGate Autoscale for AWS.   |
|  Requires one or more subscriptions to Fortinet FortiGate on-demand or BYOL AMIs. |                       |   |
| FortiGate PSK secret<br>(FortiGatePskSecret)   | <i>Requires input</i> | Secret preshared key used by the FortiGate-VM instances to securely communicate with each other. Must contain numbers and letters and may contain special characters. Maximum length is 128.  |

| Parameter label (name)                   | Default               | Description   |
|--|-----------------------|---|
|  |                       |  <p>Changes to the PSK secret after FortiGate Autoscale for AWS has been deployed are not reflected here. For new instances to be spawned with the changed PSK secret, this environment variable will need to be manually updated.</p>                           |
| Admin port<br>(FortiGateAdminPort)       | 8443                  | A port number for FortiGate administration. Minimum is 1. Maximum is 65535. Do not use the FortiGate reserved ports 443, 541, 514, or 703.  |
| Admin CIDR block<br>(FortiGateAdminCidr) | <i>Requires input</i> | <p>CIDR block for external administrator management access.</p>  <p>0.0.0.0/0 accepts connections from any IP address. Use a constrained CIDR range to reduce the potential of inbound attacks from unknown IP addresses.</p>                                    |
| Key pair name<br>(KeyPairName)           | <i>Requires input</i> | Amazon EC2 Key Pair for admin access.   |
| BGP ASN (BgpAsn)                         | 65000                 | <p>The Border Gateway Protocol (BGP) Autonomous System Number (ASN) of the Customer Gateway of each FortiGate-VM instance in the Auto Scaling group. This value ranges from 64512 to 65534.</p>  <p>Only for deployments with Transit Gateway integration.</p> |

## FortiGate Auto Scaling group configuration

| Parameter label (name)                                 | Default | Description   |
|--|---------|---|
| Desired capacity (BYOL)<br>(FgtAsgDesiredCapacityByol) | 2       | Number of FortiGate-VM instances the BYOL Auto Scaling group should have at any time. For High Availability in BYOL-only and Hybrid use cases, ensure at least 2 FortiGate-VMs are in the group. For specific use cases, set to 0 for On-demand-only, and $\geq 2$ for BYOL-only or hybrid licensing. |
| Minimum group size (BYOL)<br>(FgtAsgMinSizeByol)       | 2       | Minimum number of FortiGate-VM instances in the BYOL Auto Scaling group. For specific use cases, set to 0 for On-demand-only, and $\geq 2$ for BYOL-only or hybrid licensing.   |

| Parameter label (name)  | Default | Description   |
|---|---------|---|
|   |         |  <p>For BYOL-only and hybrid licensing deployments, this parameter must be at least 2. If it is set to 1 and the instance fails to work, the current FortiGate-VM configuration will be lost.</p>                                |
| Maximum group size (BYOL)<br>(FgtAsgMaxSizeByol)                      | 2       | Maximum number of FortiGate-VM instances in the BYOL Auto Scaling group. For specific use cases, set to 0 for On-demand-only, and $\geq 2$ for BYOL-only or hybrid licensing. This number must be greater than or equal to the <i>Minimum group size (BYOL)</i> .   |
| Desired capacity (On-demand instances)<br>(FgtAsgDesiredCapacityPayg) | 0       | Number of FortiGate-VM instances the On-demand Auto Scaling group should have at any time. For High Availability in an On-demand-only use case, ensure at least 2 FortiGate-VMs are in the group. For specific use cases, set to 0 for BYOL-only, $\geq 2$ for On-demand-only, and $\geq 0$ for hybrid licensing. |
| Minimum group size (On-demand instances) (FgtAsgMinSizePayg)          | 0       | Minimum number of FortiGate-VM instances in the On-demand Auto Scaling group. For specific use cases, set to 0 for BYOL-only, $\geq 2$ for On-demand-only, and $\geq 0$ for hybrid licensing.   |
|   |         |  <p>For On-demand-only deployments, this parameter must be at least 2. If it is set to 1 and the instance fails to work, the current FortiGate-VM configuration will be lost.</p>  |
| Maximum group size (On-demand instances) (FgtAsgMaxSizePayg)          | 0       | Maximum number of FortiGate-VM instances in the On-demand Auto Scaling group. For specific use cases, set to 0 for BYOL-only, $\geq 2$ for On-demand-only, and $\geq 0$ for hybrid licensing. This number must be greater than or equal to the <i>Minimum group size (On-demand instances)</i> .                  |
| Scale-out threshold<br>(FgtAsgScaleOutThreshold)                      | 80      | Threshold (in percentage) for the FortiGate Auto Scaling group to scale out (add) 1 instance. Minimum is 1. Maximum is 100.   |
| Scale-in threshold<br>(FgtAsgScaleInThreshold)                        | 25      | Threshold (in percentage) for the FortiGate Auto Scaling group to scale in (remove) 1 instance. Minimum is 1. Maximum is 100.   |
| Primary election timeout<br>(PrimaryElectionTimeout)                  | 300     | Maximum time (in seconds) to wait for the election of the primary instance to complete. Minimum is 30. Maximum is 3600.   |

| Parameter label (name)                                      | Default | Description  |
|---|---------|--|
| Get license grace period<br>(GetLicenseGracePeriod)         | 600     | Minimum time (in seconds) permitted before a distributed license can be revoked from a non-responsive FortiGate-VM and re-distributed. Minimum is 300. |
| Health check grace period<br>(FgtAsgHealthCheckGracePeriod) | 300     | Length of time (in seconds) that Auto Scaling waits before checking an instance's health status.<br>Minimum is 60.                                     |
| Scaling cooldown period<br>(FgtAsgCooldown)                 | 300     | Auto Scaling group waits for the cooldown period (in seconds) to complete before resuming scaling activities.<br>Minimum is 60. Maximum is 3600.       |
| Instance lifecycle timeout<br>(LifecycleHookTimeout)        | 480     | Amount of time (in seconds) that can elapse before the FortiGate Autoscale lifecycle hook times out. Minimum is 60. Maximum is 3600.                   |

### Transit Gateway configuration (Transit Gateway integration)

| Parameter label (name)                                    | Default                                 | Description   |
|---|---|---|
| Transit Gateway support<br>(TransitGatewaySupportOptions) | create one                              | Create a Transit Gateway for the FortiGate Autoscale VPC to attach to, or specify to use an existing one.   |
| Transit Gateway ID<br>(TransitGatewayId)                  | <i>Conditionally<br/>requires input</i> | ID of the Transit Gateway that the FortiGate Autoscale VPC will be attached to. Required when <i>Transit Gateway support</i> is set to "use an existing one". |

### Load balancing configuration (no Transit Gateway integration)


| Parameter label (name)  | Default                                   | Description  |
|---|---|--|
| Traffic protocol<br>(LoadBalancingTrafficProtocol)            | HTTPS                                     | Protocol used to load balance traffic.   |
| Traffic port (LoadBalancingTrafficPort)                       | 443                                       | Port number used to balance web service traffic if the internal web service load balancer is enabled.<br>Minimum is 1. Maximum is 65535.                                 |
| Health check threshold<br>(LoadBalancingHealthCheckThreshold) | 3   | Number of consecutive health check failures required before considering a FortiGate-VM instance unhealthy.<br>Minimum 3.   |
| Internal ELB options<br>(InternalLoadBalancingOptions)        | add a new<br>internal<br>load<br>balancer | (Optional) Predefined Elastic Load Balancer (ELB) to route traffic to web service in the private subnets. You can optionally use your own one or decide to not need one. |

| Parameter label (name)                                    | Default               | Description  |
|---|-----------------------|--|
| Health check path<br>(InternalTargetGroupHealthCheckPath) | /                     | (Optional) Destination path for health checks. This path must begin with a forward slash (/) and can be at most 1024 characters in length.   |
| Internal ELB DNS name<br>(InternalLoadBalancerDnsName)    | <i>Requires input</i> | (Optional) DNS name of an existing internal load balancer used to route traffic from a FortiGate-VM to targets in a specified target group. Leave it blank if you don't use an existing load balancer. |

## Failover management configuration

| Parameter label (name)   | Default | Description  |
|--|---------|--|
| Heart beat interval (HeartBeatInterval)  | 30      | Length of time (in seconds) that a FortiGate-VM instance waits between sending heartbeat requests to the Autoscale handler. Minimum is 30. Maximum is 90.  |
| Heart beat loss count (HeartBeatLossCount)   | 10      | Number of consecutively lost heartbeats. When the Heartbeat loss count has been reached, the FortiGate-VM is deemed unhealthy and failover activities will commence.   |
| Heart beat delay allowance<br>(HeartBeatDelayAllowance)                            | 2       | Maximum amount of time (in seconds) allowed for network latency of the FortiGate-VM heartbeat arriving at the FortiGate Autoscale handler. Minimum is 0.   |
| Autoscale notifications subscriber email<br>(AutoscaleNotificationSubscriberEmail) | -       | The email address (AWS SNS Topic subscriber) to receive Autoscale notifications. If provided, the template can only accept one email address. An email will be sent to the address to confirm the subscription.                  |
| Terminate unhealthy VM<br>(TerminateUnhealthyVm)                                   | no      | Set to <b>yes</b> to terminate any VM that is deemed unhealthy by FortiGate Autoscale.   |
| Autoscale sync recovery count<br>(SyncRecoveryCount)                               | 3       | Number of consecutive on-time heartbeats required for a VM to become healthy again. This parameter is only used when <i>Terminate unhealthy VM</i> is set to <b>no</b> and allows for the VM to recover from an unhealthy state. |

## FortiAnalyzer integration

| Parameter label (name)  | Default               | Description   |
|---|-----------------------|---|
| FortiAnalyzer integration<br>(FortiAnalyzerIntegrationOptions)            | yes                   | Set to <i>no</i> if you do not want to incorporate FortiAnalyzer into FortiGate Autoscale to use extended features that include storing logs into FortiAnalyzer.  |
| FortiAnalyzer version<br>(FortiAnalyzerVersion)                           | 6.4.6                 | FortiAnalyzer version supported by FortiGate Autoscale.<br><br> Requires a subscription to the "Fortinet FortiAnalyzer Centralized Logging/Reporting (10 managed devices)" AMI.   |
| FortiAnalyzer instance type<br>(FortiAnalyzerInstanceType)                | m5.large              | Instance type to launch as FortiAnalyzer on-demand instances. There are compute-optimized instances, such as m4 and c4, available with different vCPU sizes and bandwidths. For more information about instance types, see <a href="#">Instance Types</a> .   |
| Autoscale admin user name<br>(FortiAnalyzerAutoscaleAdminUsername)        | <i>Requires input</i> | Name of the secondary administrator-level account in the FortiAnalyzer, which FortiGate Autoscale uses to connect to the FortiAnalyzer to authorize any FortiGate device in the Auto Scaling group. To conform to the FortiAnalyzer naming policy, the user name can only contain numbers, lowercase letters, uppercase letters, and hyphens. It cannot start or end with a hyphen (-). |
| Autoscale admin password<br>(FortiAnalyzerAutoscaleAdminPassword)         | <i>Requires input</i> | Password for the "Autoscale admin user name." The password must conform to the FortiAnalyzer password policy and have a minimum length of 8 and a maximum length of 128. To enable KMS encryption, see the documentation.   |
| FortiAnalyzer private IP address<br>(FortiAnalyzerCustomPrivateIpAddress) | <i>Requires input</i> | Custom private IP address to be used by the FortiAnalyzer. Must be within the public subnet 1 CIDR range. Required if "FortiAnalyzer integration" is set to <i>yes</i> . If "FortiAnalyzer integration" is set to <i>no</i> , any input will be ignored.  |

## Custom asset location configuration

| Parameter label (name)                                | Default               | Description   |
|---|-----------------------|---|
| Use custom asset location<br>(UseCustomAssetLocation) | no                    | Set to <b>yes</b> to use a custom S3 location for custom assets such as licenses and customized configsets.   |
| Custom asset S3 bucket<br>(CustomAssetContainer)      | <i>Requires input</i> | Name of the S3 bucket that contains your custom assets. Required if 'Use custom asset location' is set to <b>yes</b> . Can only contain numbers, lowercase letters, uppercase letters, and hyphens (-). It cannot start or end with a hyphen (-).   |
| Custom asset folder<br>(CustomAssetDirectory)         | <i>Requires input</i> | The sub path within the 'custom asset container' that serves as the top level directory of all your custom assets. If 'Use custom asset location' is set to <b>yes</b> , and this value is left empty, the 'custom asset container' will serve as the top level directory. Can only contain numbers, lowercase letters, uppercase letters, hyphens (-), and forward slashes (/). If provided, it must end with a forward slash (/). |

## Deployment resources configuration

| Parameter label (name)              | Default               | Description  |
|-------------------------------------|-----------------------|--|
| S3 bucket name<br>(S3BucketName)    | <i>Requires input</i> | Name of the S3 bucket (created in step 4 of <a href="#">Obtaining the deployment package on page 46</a> ) that contains the FortiGate Autoscale deployment package. Can only contain numbers, lowercase letters, uppercase letters, and hyphens (-). It cannot start or end with a hyphen (-).                               |
| S3 resource folder<br>(S3KeyPrefix) | <i>Requires input</i> | Name of the S3 folder (created in step 5 of <a href="#">Obtaining the deployment package on page 46</a> ) that stores the FortiGate Autoscale deployment resources. Can only contain numbers, lowercase letters, uppercase letters, hyphens (-), and forward slashes (/). If provided, it must end with a forward slash (/). |

## Optional settings

You can configure optional settings on the *Configure stack options* page:



## Configure stack options

### Tags

You can specify tags (key-value pairs) to apply to resources in your stack. You can add up to 50 unique tags for each stack. [Learn more.](#)

### Permissions


Choose an IAM role to explicitly define how CloudFormation can create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses permissions based on your user credentials. [Learn more.](#)

#### IAM role - optional

Choose the IAM role for CloudFormation to use for all operations performed on the stack.

1. Specify *Tags* and *Permissions* as desired:
  - a. *Tags*: Key-Value pairs for resources in your stack.
  - b. *Permissions*: An [IAM role](#) that AWS CloudFormation uses to create, modify, or delete resources in your stack.
2. Under *Advanced options*, disabling the [Stack creation option Rollback on failure](#) is recommended to allow for a better troubleshooting experience.


## Advanced options

You can set additional options for your stack, like notification options and a stack policy. [Learn more](#) 

### ► Stack policy

Defines the resources that you want to protect from unintentional updates during a stack update.

### ► Rollback configuration

Specify alarms for CloudFormation to monitor when creating and updating the stack. If the operation breaches an alarm threshold, CloudFormation rolls it back. [Learn more](#) 

### ► Notification options

### ▼ Stack creation options

#### Rollback on failure

Specifies whether the stack should be rolled back if stack creation fails.

- ☐ Enabled
- ☒ Disabled

#### Timeout

The number of minutes before a stack creation times out.

#### Termination protection

Prevents the stack from being accidentally deleted. Once created, you can update this through stack actions.

- ☒ Disabled
- ☐ Enabled

3. Other advanced options can be specified as desired.
4. When done, click *Next*.

## Completing the deployment

On the *Review* page, review and confirm the template, the stack details, and the stack options. Under *Capabilities*, select both check boxes to acknowledge that the template creates IAM resources and might require the ability to automatically expand macros.

## Capabilities

**The following resource(s) require capabilities: [AWS::CloudFormation::Stack]**

This template contains Identity and Access Management (IAM) resources. Check that you want to create each of these resources and that they have the minimum required permissions. In addition, they have custom names. Check that the custom names are unique within your AWS account. [Learn more](#)

For this template, AWS CloudFormation might require an unrecognized capability: CAPABILITY\_AUTO\_EXPAND. Check the capabilities of these resources.

- ☒ I acknowledge that AWS CloudFormation might create IAM resources with custom names.
- ☒ I acknowledge that AWS CloudFormation might require the following capability: CAPABILITY\_AUTO\_EXPAND

Cancel

Previous

Create change set

**Create stack**

Click **Create stack** to deploy the stack.

Creation status is shown in the *Status* column. To see the latest status, refresh the view. It takes about 10 minutes to create the stack. Deployment has completed when each stack (including the main stack and all nested stacks) has a status of *CREATE\_COMPLETE*.

CloudFormation > Stacks > FortiGate-Autoscale-CI-1597944688632

**Stacks (26)**

FortiGate-Autoscale-CI-

Active View nested

1

NESTED

FortiGate-Autoscale-CI-1597944688632-StackMainWorkload-19V11L-StackCreateDyna

moDBTable-19G8K07BFHR6Y

2020-08-20 10:33:07 UTC-0700

**CREATE\_COMPLETE**

NESTED

FortiGate-Autoscale-CI-1597944688632-StackMainWorkload-19V11LXJZNW4

2020-08-20 10:32:59 UTC-0700

**CREATE\_COMPLETE**

NESTED

FortiGate-Autoscale-CI-1597944688632-StackCreateNewVPC-1YOMPAOW4D2FE

2020-08-20 10:31:36 UTC-0700

**CREATE\_COMPLETE**

FortiGate-Autoscale-CI-1597944688632

2020-08-20 10:31:29 UTC-0700

**CREATE\_COMPLETE**

### FortiGate-Autoscale-CI-1597944688632

Delete Update Stack actions Create stack

Stack info **Events** Resources Outputs Parameters Template Change sets

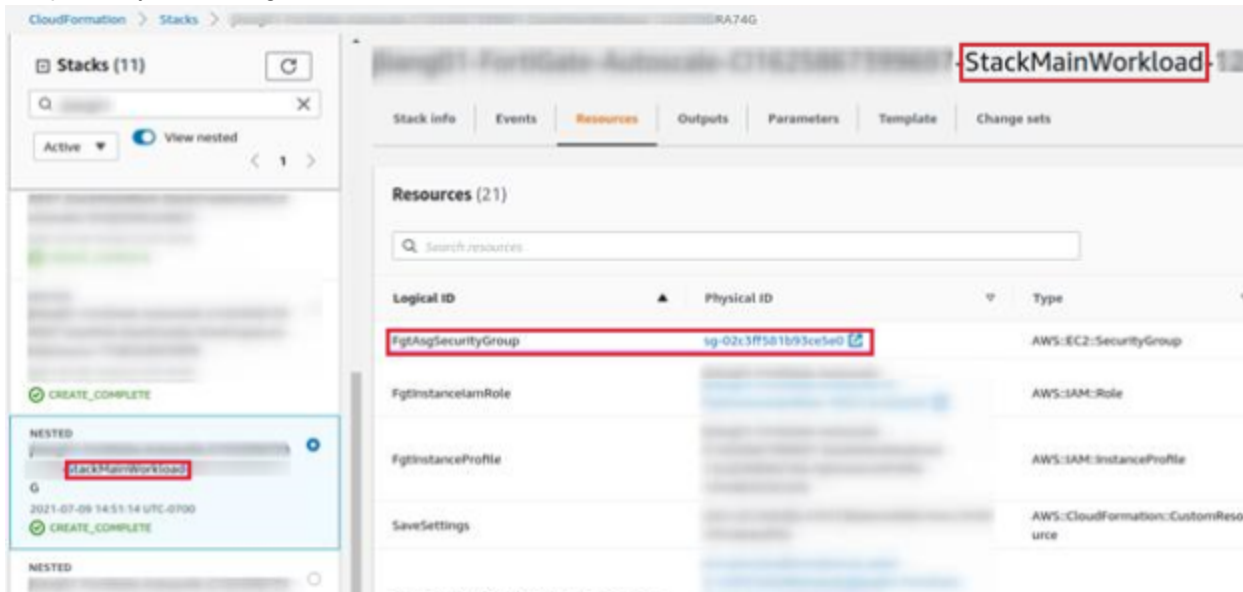
**Events (8)**

Search events

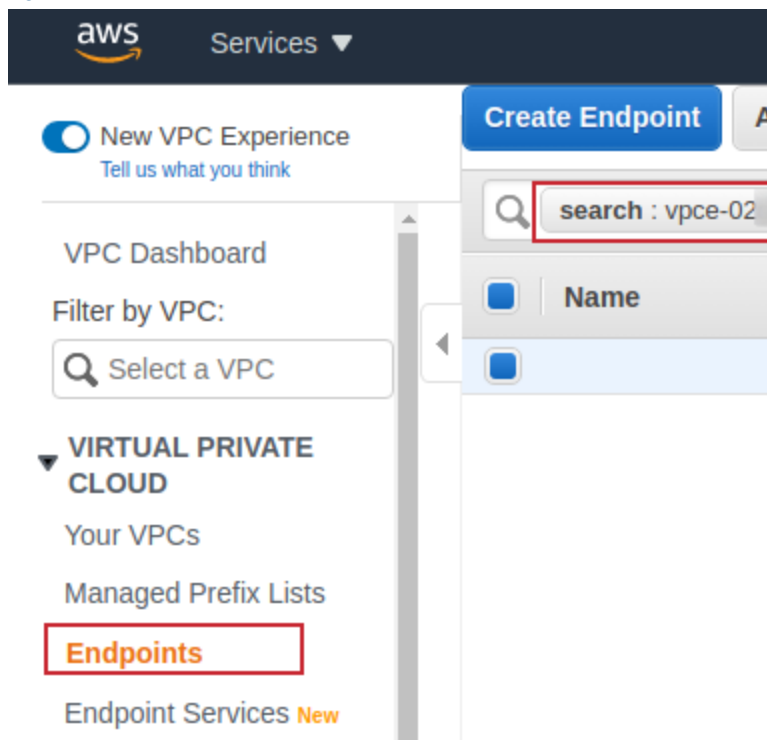
| Timestamp                    | Logical ID                           | Status                    | Status reason               |
|------------------------------|--------------------------------------|---------------------------|-----------------------------|
| 2020-08-20 10:39:43 UTC-0700 | FortiGate-Autoscale-CI-1597944688632 | <b>CREATE_COMPLETE</b>    | -                           |
| 2020-08-20 10:39:41 UTC-0700 | StackMainWorkload                    | <b>CREATE_COMPLETE</b>    | -                           |
| 2020-08-20 10:33:00 UTC-0700 | StackMainWorkload                    | <b>CREATE_IN_PROGRESS</b> | Resource creation Initiated |
| 2020-08-20 10:32:59 UTC-0700 | StackMainWorkload                    | <b>CREATE_IN_PROGRESS</b> | -                           |

## Post-deployment activities

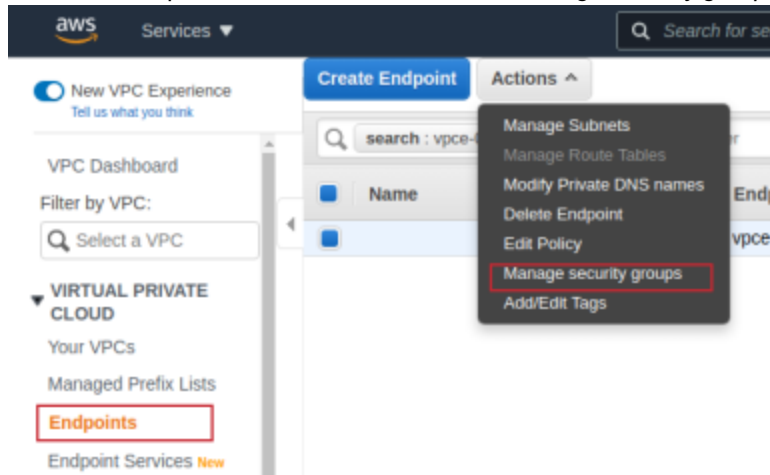
If you deployed into an existing VPC, locate and select *StackMainWorkload* from the left column. Make note of *Physical ID* for the *Logical ID* *FgtAsgSecurityGroup*. You will need to associate this security group with the Private VPC Endpoint of your existing VPC.



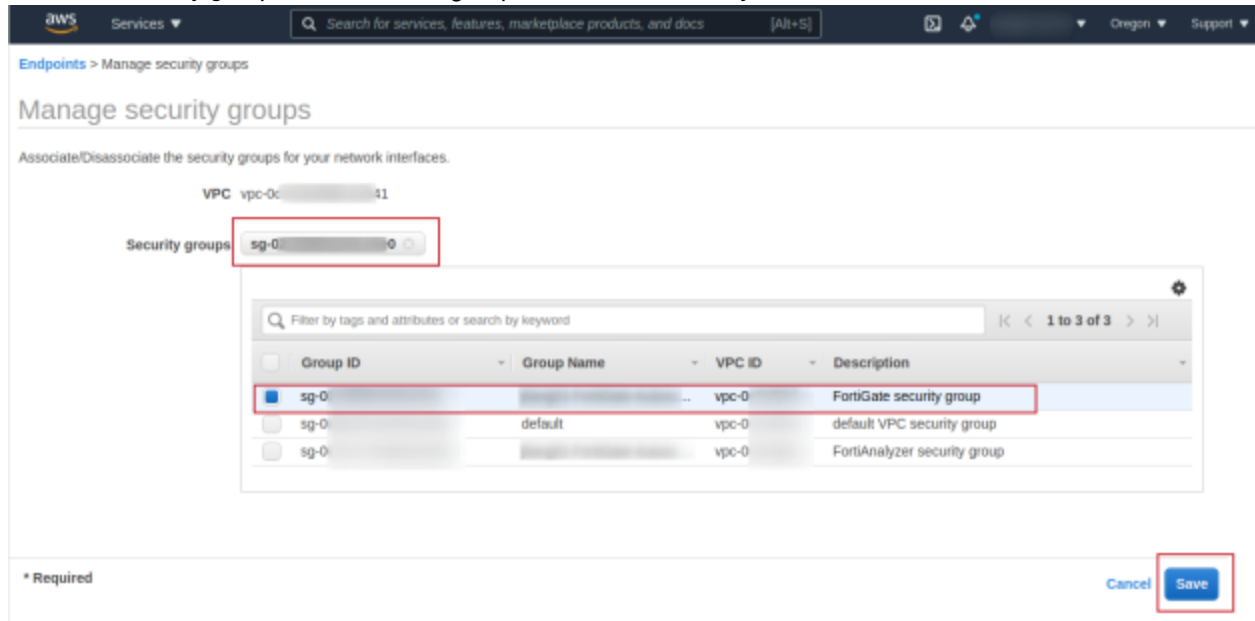
1. In the AWS console, select *Services > Network & Content Delivery > VPC*.
2. In the left navigation tree, click *Endpoints*.
3. Click the filter box and search for the VPC Endpoint created in [Requirements when using an existing VPC on page 45](#).



4. Select the endpoint and under *Actions*, select *Manage security groups*.



5. From the *Security groups* list, select the group that matches the *Physical ID*.



6. Click *Save*.

## Locating deployed resources

To locate a newly deployed resource, searching for it using the *ResourceTagPrefix*, also referred to as the *ResourceGroup Tag Key*, is recommended. Alternatively, the *UniqueID* can be used. For items that need a shorter prefix, the *CustomIdentifier* can be used. These keys are found on the *Outputs* tab as shown below. Note that the *UniqueID* is at the end of the *ResourceTagPrefix*.

## FortiGate-Autoscale-CI-1597944702684

Delete

Update

Stack actions ▼

Create stack ▼

Stack info

Events

Resources

**Outputs**

Parameters

Template

Change sets

## Outputs (5)



Q Search outputs

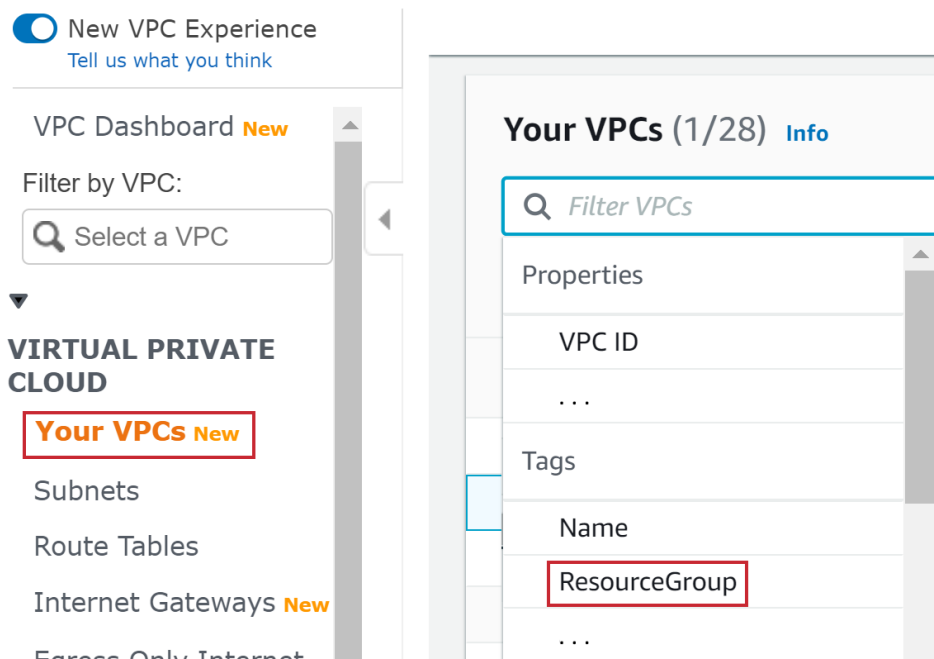


| Key ▲             | Value ▼   | Description ▼   | Export name ▼ |
|-------------------|---|---|---------------|
| CustomIdentifier  | qsY46W65a2  | The custom identifier specified for this stack. This is used as a resource name prefix on those resources that have a strict naming requirement.  | -             |
| FgtLicensingModel | Hybrid  | The FortiGate licensing model in the Auto Scaling group(s) for the initial deployment of this stack. (Options: On-Demand-Only, BYOL-Only, Hybrid) | -             |
| FortiOSVersion    | 6.2.3   | The selected FortiOS version.   | -             |
| ResourceTagPrefix | fortigate-autoscale-has-a-very-long-res-tag-prefix-038d9fe0 | The value for the Tag Key 'ResourceGroup' on all resources deployed in this stack.  | -             |
| Uniqueld          | 038d9fe0  | An automatically generated random string as a unique ID for all resources in the deployment stack and nested stacks.                              | -             |

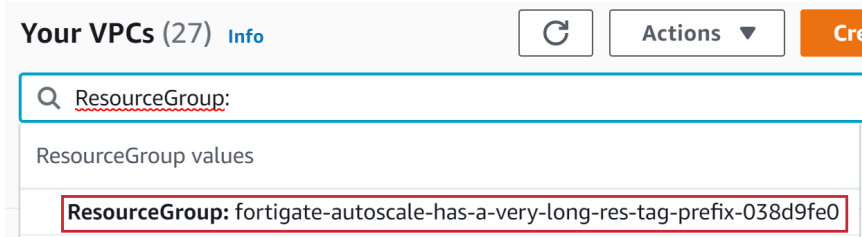
**To look up the newly deployed VPC using the ResourceGroup Tag Key:**

1. In the AWS console, select *Services > Network & Content Delivery > VPC*.
2. In the left navigation tree, click *Your VPCs*.

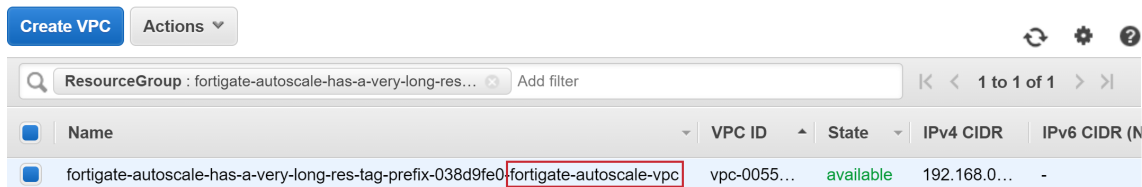
- Click the filter box and under *Tags*, select *ResourceGroup*.



- Select your *ResourceTagPrefix* from the list of Tags.



Your VPC will be displayed. The *Name* of VPC is of the format *<ResourceTagPrefix>-fortigate-autoscale-vpc*.



### To look up the newly deployed VPC subnets using the ResourceGroup Tag Key:

- In the AWS console, select *Services > Network & Content Delivery > VPC*.
- In the left navigation tree, click *VIRTUAL PRIVATE CLOUD > Subnets*.
- Click the filter box and select *Tag Keys > ResourceGroup*.
- Select your *ResourceTagPrefix* from the list of Tag Keys.

Your VPC subnets will be displayed. The *Name* of each subnets will be of the format *<ResourceTagPrefix>-fortigate-autoscale-vpc-subnet#<#>*.

Create subnet Actions

ResourceGroup : fortigate-autoscale-has-a-very-long-res...

| Name  | Subnet ID   |
|---|-------------|
| fortigate-autoscale-has-a-very-long-res-tag-prefix-038d9fe0-fortigate-autoscale-vpc-public-subnet#2 | subnet-04fd |
| fortigate-autoscale-has-a-very-long-res-tag-prefix-038d9fe0-fortigate-autoscale-vpc-public-subnet#1 | subnet-0b8t |

### To look up the newly deployed DynamoDB tables using the UniqueID

1. In the AWS console, select *Services > Database > DynamoDB*.
2. In the left navigation tree, click *Tables*.
3. Click the filter box and enter the *UniqueID*.

The DynamoDB tables will be displayed. The *Name* of each DynamoDB table will be of the format *<ResourceTagPrefix>-<table-name>*.

Create table Delete table

038d9fe0 Choose a table group Actions

1 to 8 of 8 Tables

| Name  | Status | Partition key             |
|---|--------|---------------------------|
| fortigate-autoscale-has-a-very-long-res-tag-prefix-038d9fe0-Autoscale       | Active | vmId (String)             |
| fortigate-autoscale-has-a-very-long-res-tag-prefix-038d9fe0-CustomLog       | Active | id (String)               |
| fortigate-autoscale-has-a-very-long-res-tag-prefix-038d9fe0-LicenseStock    | Active | checksum (String)         |
| fortigate-autoscale-has-a-very-long-res-tag-prefix-038d9fe0-LicenseUsage    | Active | checksum (String)         |
| fortigate-autoscale-has-a-very-long-res-tag-prefix-038d9fe0-LifecycleItem   | Active | vmId (String)             |
| fortigate-autoscale-has-a-very-long-res-tag-prefix-038d9fe0-PrimaryElection | Active | scalingGroupName (String) |
| fortigate-autoscale-has-a-very-long-res-tag-prefix-038d9fe0-Settings        | Active | settingKey (String)       |
| fortigate-autoscale-has-a-very-long-res-tag-prefix-038d9fe0-VpnAttachment   | Active | vmId (String)             |

### To look up the newly deployed Lambda Functions using the CustomIdentifier or the UniqueID:

1. In the AWS console, select *Services > Compute > Lambda*.
2. In the left navigation tree, click *Functions*.
3. Click the filter box and enter the *CustomIdentifier* or the *UniqueID*.

The Lambda Functions will be displayed. Each *Function name* will be of the format *<CustomIdentifier>-<UniqueID>-<LambdaFunctionName>*.



Lambda > Functions

**Functions (52)** Actions ▾ Create function

Q Add filter ? < 1 > ⚙

Keyword : qsY46w65a2 ⓧ

|                       | Function name ▾   | Description  | Runtime ▾    | Code size ▾ | Last modified ▾ |
|-----------------------|---|--|--------------|-------------|-----------------|
| <input type="radio"/> | <a href="#">qsY46W65a2-038d9fe0-fortigate-autoscale-handler</a>                 | FortiGate Autoscale handler function.                                    | Node.js 12.x | 2.6 MB      | yesterday       |
| <input type="radio"/> | <a href="#">qsY46W65a2-038d9fe0-fortigate-autoscale-byol-license</a>            | FortiGate Autoscale BYOL license handler function.                       | Node.js 12.x | 2.6 MB      | yesterday       |
| <input type="radio"/> | <a href="#">qsY46W65a2-038d9fe0-fortigate-transit-gateway-vpn-handler</a>       | A service for Transit Gateway VPN management.                            | Node.js 12.x | 2.6 MB      | yesterday       |
| <input type="radio"/> | <a href="#">qsY46W65a2-038d9fe0-fortigate-autoscale-auto-scaling-event</a>      | FortiGate Autoscale Auto Scaling event handler.                          | Node.js 12.x | 2.6 MB      | yesterday       |
| <input type="radio"/> | <a href="#">qsY46W65a2-038d9fe0-fortigate-autoscale-cloud-formation-service</a> | FortiGate Autoscale service provider function for Cloud Formation stack. | Node.js 12.x | 2.6 MB      | yesterday       |

Click the *Function name* to go directly to the function.

## Verifying the deployment

FortiGate Autoscale for AWS creates two Auto Scaling groups with instances as specified in the CFT parameters. One of these instances is the elected primary instance. Verify the following:

- [the Auto Scaling groups](#)
- [the primary election](#)

If deploying with Transit Gateway integration, you will also need to verify:

- [the Transit Gateway](#)

### To verify the Auto Scaling groups:

1. In the AWS console, select the *Services > Compute > EC2*.
2. In the left navigation tree, click *AUTO SCALING > Auto Scaling Groups*.
3. Click the filter box and look up the Auto Scaling groups using the *Unique ID*.
4. The name of each group starts with the prefix you specified in *Resource tag prefix*. Confirm that the number in the *Instances* column is equal to or greater than the *Desired capacity* you specified.

EC2 > Auto Scaling groups

**Auto Scaling groups (10)** Refresh Edit Delete Create an Auto Scaling group

Search: 038d9fe0 X 2 matches < 1 > ⚙️

| <input type="checkbox"/> | Name   | Launch templ...         | Instances | Desired capacity |
|--------------------------|--|-------------------------|-----------|------------------|
| <input type="checkbox"/> | fortigate-.....-038d9fe0-fortigate-byol-auto-scaling-group | fortigate-autoscale-... | 2         | 2                |
| <input type="checkbox"/> | fortigate-.....-038d9fe0-fortigate-payg-auto-scaling-group | fortigate-autoscale-... | 0         | 0                |

5. In the left navigation tree, click **INSTANCES** > **Instances**.
6. Click the filter box and look up instances using the *ResourceTagPrefix*.
7. Instances will be listed with their current state.

**Instances (4)** Info Refresh Actions Launch instances

Filter instances:

ResourceGroup: fortigate-autoscale-has-a-very-long-res-tag-prefix-038d9fe0 X Clear filters

| <input type="checkbox"/> | Name   | AutoscaleRole | Instance ID   | Instance state | Instance type |
|--------------------------|--|---------------|---------------|----------------|---------------|
| <input type="checkbox"/> | fortigate-autoscale-has-a-very-long-res-tag-prefix-... | -             | i-033fb972... | Running        | c5.xlarge     |
| <input type="checkbox"/> | fortigate-autoscale-has-a-very-long-res-tag-prefix-... | primary       | i-0c74632d... | Running        | c5.xlarge     |

### To verify the primary election:

The primary instance is noted in the *AutoscaleRole* column:

**Instances (4)** Info Refresh Actions Launch instances

Filter instances:

ResourceGroup: fortigate-autoscale-has-a-very-long-res-tag-prefix-038d9fe0 X Clear filters

| <input type="checkbox"/> | Name   | AutoscaleRole | Instance ID   | Instance state | Instance type |
|--------------------------|--|---------------|---------------|----------------|---------------|
| <input type="checkbox"/> | fortigate-autoscale-has-a-very-long-res-tag-prefix-... | -             | i-033fb972... | Running        | c5.xlarge     |
| <input type="checkbox"/> | fortigate-autoscale-has-a-very-long-res-tag-prefix-... | primary       | i-0c74632d... | Running        | c5.xlarge     |

If the *AutoscaleRole* column is not displayed, click the *Preferences* cog and locate the *Tag columns* dropdown. Select *AutoscaleRole* and then click *Confirm*.

Tag columns

Select visible tag columns

Search for tags keys

Name X

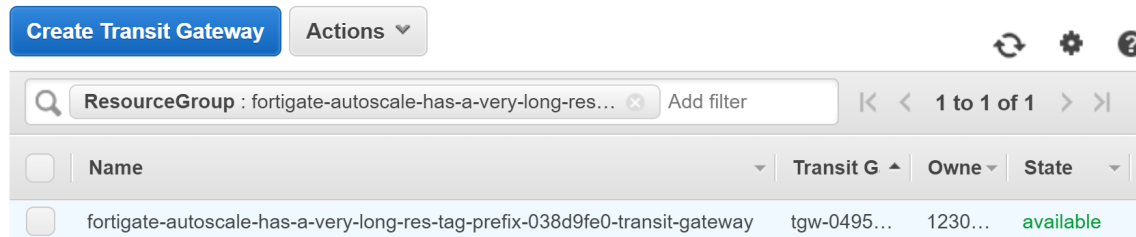
AutoscaleRole X

Cancel

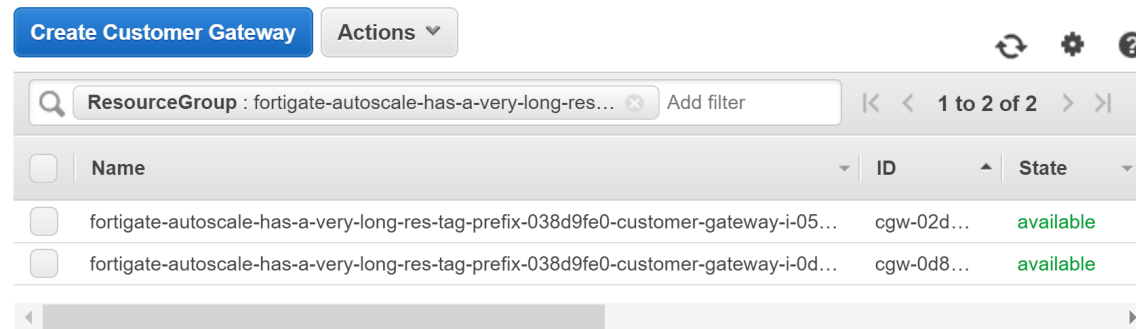
Confirm

**To verify the Transit Gateway:**

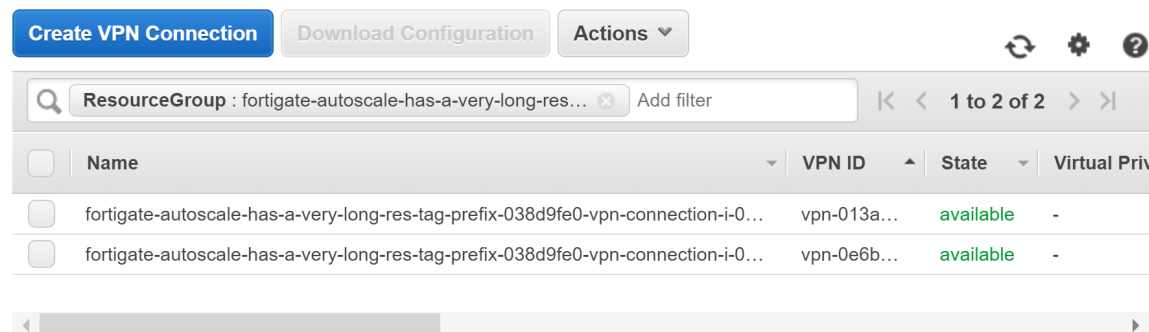
1. In the AWS console, select the *Services > Network & Content Delivery > VPC*.
2. In the left navigation tree, click *TRANSIT GATEWAYS > Transit Gateways*.
3. Filter by the Tag Key *ResourceGroup*. There should be one result.



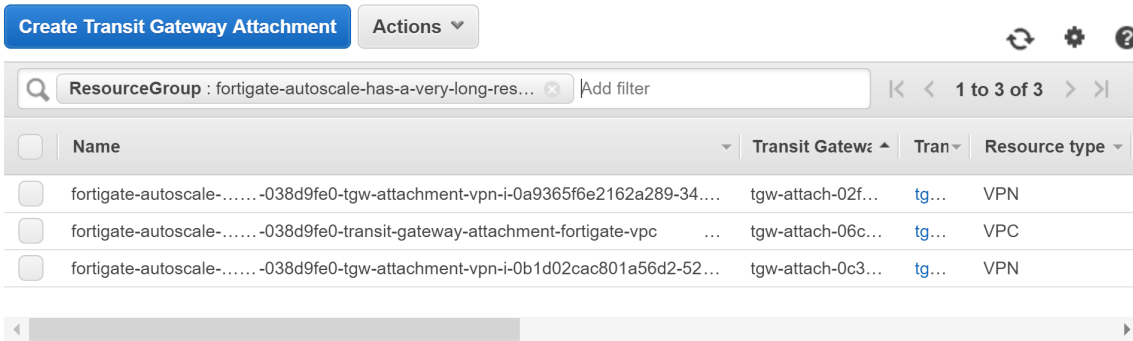
4. In the left navigation tree, click *VIRTUAL PRIVATE NETWORK (VPN) > Customer Gateways*.
5. Filter by the Tag Key *ResourceGroup*. There should be one customer gateway per running FortiGate-VM instance (2 at the start).



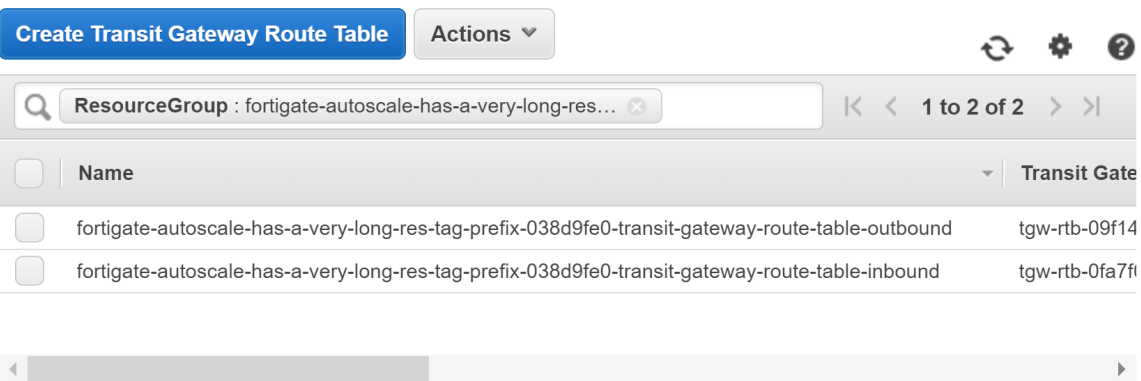
6. In the left navigation tree, click *VIRTUAL PRIVATE NETWORK (VPN) > Site-to-Site VPN Connections*.
7. Filter by the Tag Key *ResourceGroup*. There should be two items, 1 per FortiGate-VM instance, each with a corresponding Transit Gateway attachment.



8. In the left navigation tree, click *TRANSIT GATEWAYS > Transit Gateway Attachments*.
9. Filter by the Tag Key *ResourceGroup*. There should be one VPC, and one VPN per running FortiGate-VM instance in the Auto Scaling group. (2 at the start, one primary and one secondary). The VPN name will contain the public IP address of the VPN.



10. In the left navigation tree, click **TRANSIT GATEWAYS** > **Transit Gateway Route Tables**.
11. Filter by the Tag Key **ResourceGroup**. There should be two items, one for inbound and one for outbound. For diagrams, see [Appendix on page 83](#).



## Connecting to the primary FortiGate-VM

To connect to the primary FortiGate-VM instance, you need a login URL, a username, and a password.

### To connect to the primary FortiGate-VM:

1. Construct a login URL in this way: `https://<IPAddress>:<Port>/`, where:
  - *Port* refers to the *Admin port* specified in the section [FortiGate configuration on page 54](#).
  - *IPAddress* refers to the *Public IPv4 address* of the FortiGate-VM and is listed on the *Details* tab for the instance. In the EC2 Management console, locate the primary instance as described in the section [To verify the](#)

primary election: on page 70. Click the *Instance ID* for the primary instance.

The screenshot displays the AWS Management Console interface for an EC2 instance. At the top, there's a search bar with the text 'search: i-0b1d02cac801a56d2'. Below this, a table lists instances. The first instance is 'fortigate-autoscale-has-a-very-long-res-tag-prefix-03...' with Instance ID 'i-0b1d02cac801a56d2', which is highlighted with a red box. The instance state is 'Running' and the type is 'c5.xlarge'. Below the table, the 'Details' tab is selected, showing the 'Instance summary' with the Instance ID, Public IPv4 address '52.25.66.115' (highlighted with a red box), and Private IPv4 address '192.168.0.172'.

Make note of the *InstanceID* as you will need it to log in.

2. Open an HTTPS session in your browser and go to the login URL. Your browser displays a certificate error message. This is normal because the default FortiGate certificate is self-signed and browsers do not recognize it. Proceed past this error. At a later time, you can upload a publicly signed certificate to avoid this error.
3. Log in with the username *admin* and the *Instance ID* of the primary FortiGate-VM instance.




As the primary FortiGate-VM propagates the password to all secondary FortiGate instances, this is the initial password for all FortiGate-VM instances.

You need this initial password if failover occurs prior to changing the password, as the newly elected primary FortiGate-VM still has the previous primary's initial password.

4. FortiOS prompts to change the password at initial login. Doing so at this time is recommended.

### Change Password

 You are required to change the default password.

New password must include:

8 Minimum length

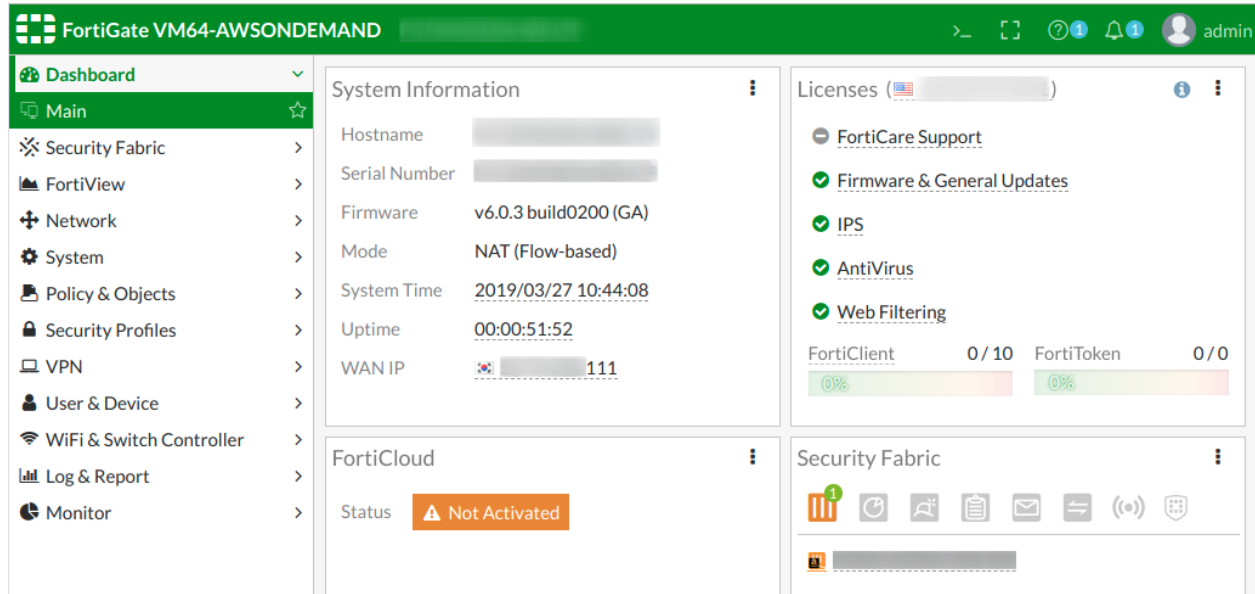
OK

Logout



You should only change the password on the primary FortiGate-VM. The primary FortiGate-VM will propagate the password to all secondary FortiGate-VMs. Any password changed on a secondary FortiGate-VM will be overwritten.

5. You will now see the FortiGate-VM dashboard. The information displayed in the license widget of the dashboard depends on your license type.



## Attaching a VPC to the TGW

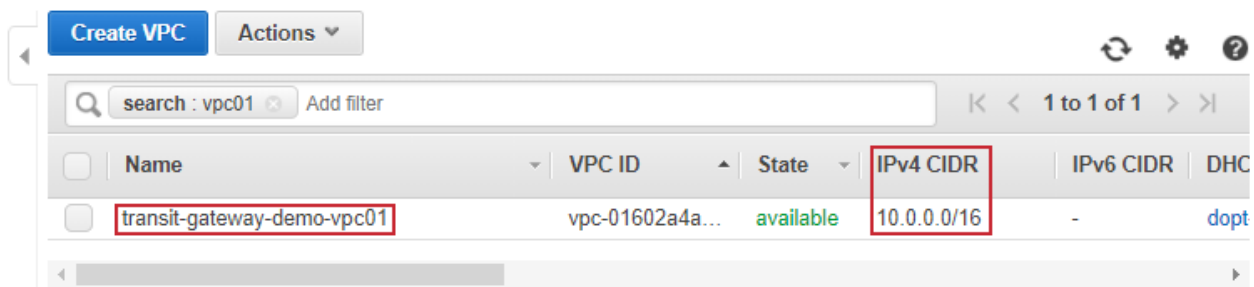
You can attach an existing VPC to the FortiGate Autoscale with Transit Gateway (TGW) environment by manually creating a TGW attachment and adding the necessary routes, propagations, and associations:

1. [Create a TGW attachment.](#)
2. [Create a route to the TGW.](#)
3. [Create a propagation in the inbound route table.](#)
4. [Create an association in the outbound route table.](#)



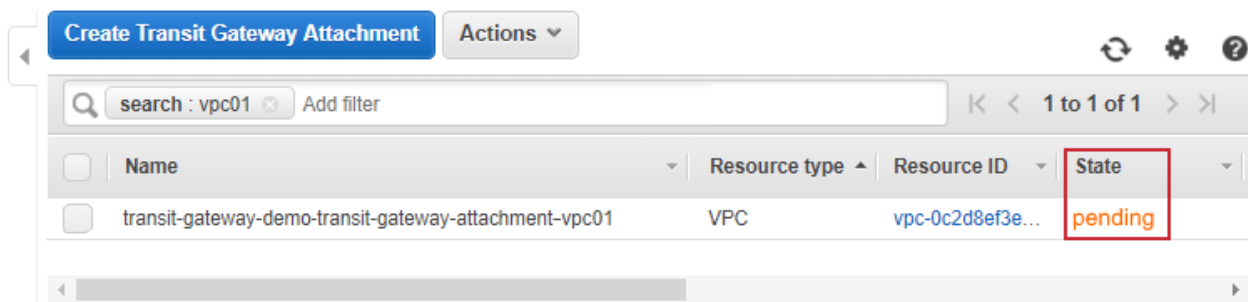
The CIDR block for the VPC you are attaching must differ from that of the FortiGate Autoscale VPC.

The following instructions attach the VPC *transit-gateway-demo-vpc01* with CIDR *10.0.0.0/16* to the FortiGate Autoscale with Transit Gateway environment.



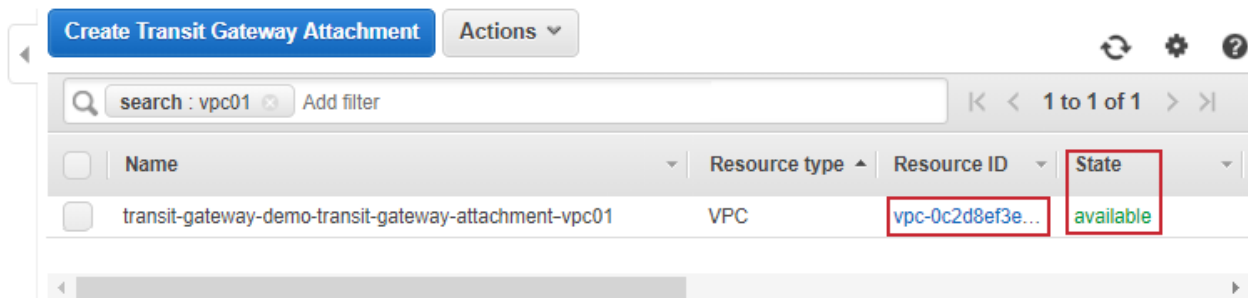
**To create a TGW attachment:**

1. In the left navigation tree, click *TRANSIT GATEWAYS > Transit Gateway Attachment*.
2. Click *Create Transit Gateway Attachment*.
3. Specify information as follows:
  - a. *Transit Gateway ID*: select the desired TGW ID from the dropdown list.
  - b. *Attachment type*: select *VPC*.
  - c. *Attachment name tag*: enter a desired tag.
  - d. *VPC ID*: select from the dropdown menu
  - e. *Subnet IDs*: This option appears once the *VPC ID* has been selected. Check the Availability Zone check box(es) and choose one subnet per Availability Zone.For everything else, use the default settings.
4. Click *Create attachment*.
5. Wait for the *State* to change from *pending* to *available*.



The *Name* is what you specified for the *Attachment name tag*.

6. When the *State* is *available*, click the *Resource ID* to go to the VPC.





**To create a route to the TGW:**

1. In the VPC, click the *Route table*.

The screenshot shows the AWS Management Console interface for a VPC. At the top, there's a search bar with 'vpc-01602a4a72064e48a' and a table with columns: Name, VPC ID, State, and IPv4 CIDR. The table lists 'transit-gateway-demo-vpc01' with VPC ID 'vpc-01602a4a72064e48a', State 'available', and IPv4 CIDR '10.0.0.0/16'. Below this, the VPC details for 'vpc-01602a4a72064e48a' are shown. The 'Description' tab is active, displaying various attributes. The 'Route table' attribute is highlighted with a red box, showing the value 'rtb-055f16b87c4302f4c'.

| Name                       | VPC ID                | State     | IPv4 CIDR   |
|----------------------------|-----------------------|-----------|-------------|
| transit-gateway-demo-vpc01 | vpc-01602a4a72064e48a | available | 10.0.0.0/16 |

VPC: vpc-01602a4a72064e48a

Attributes:

|                  |                       |                         |          |
|------------------|-----------------------|-------------------------|----------|
| VPC ID           | vpc-01602a4a72064e48a | Tenancy                 | default  |
| State            | available             | Default VPC             | No       |
| IPv4 CIDR        | 10.0.0.0/16           | Classic link            | Disabled |
| IPv6 CIDR        | -                     | DNS resolution          | Enabled  |
| Network ACL      | -                     | DNS hostnames           | Enabled  |
| DHCP options set | -                     | ClassicLink DNS Support | Disabled |
| Route table      | rtb-055f16b87c4302f4c | Owner                   | -        |

2. Click the *Routes* tab and then click *Edit routes*.

The screenshot shows the AWS Management Console interface for a Route Table. At the top, there's a search bar with 'Route Table ID : rtb-037eaf3e7e5d0f165' and a table with columns: Name, Route Table ID, Explicitly Associated with, and Main. The table lists 'rtb-037eaf3e7e5d0f165' with Route Table ID 'rtb-037eaf3e7e5d0f165', Explicitly Associated with '-', and Main 'Yes'. Below this, the Route Table details for 'rtb-037eaf3e7e5d0f165' are shown. The 'Routes' tab is active and highlighted with a red box. Below the tabs, the 'Edit routes' button is highlighted with a red box. The 'View' dropdown is set to 'All routes'. The table below shows the routes for this table.

| Name                  | Route Table ID        | Explicitly Associated with | Main |
|-----------------------|-----------------------|----------------------------|------|
| rtb-037eaf3e7e5d0f165 | rtb-037eaf3e7e5d0f165 | -                          | Yes  |

Route Table: rtb-037eaf3e7e5d0f165

Tabs: Summary, Routes, Subnet Associations, Route Propagation, Tags

Buttons: Edit routes

View: All routes

| Destination | Target                | Status | Propagated |
|-------------|-----------------------|--------|------------|
| 10.0.0.0/16 | local                 | active | No         |
| 0.0.0.0/0   | nat-017c74b8c872dff70 | active | No         |

3. Click **Add route** and specify the *Destination*, for example, 10.1.0.0/16. Under *Target*, select *Transit Gateway*.

[Route Tables](#) > [Edit routes](#)

## Edit routes

| Destination | Target                | Status | Propagated |
|-------------|-----------------------|--------|------------|
| 10.0.0.0/16 | local                 | active | No         |
| 0.0.0.0/0   | nat-017c74b8c872dff70 | active | No         |
| 10.1.0.0/16 |                       |        | No         |

**Add route**

\* Required

Cancel **Save routes**

Egress Only Internet Gateway  
 Instance  
 Internet Gateway  
 NAT Gateway  
 Network Interface  
 Peering Connection  
**Transit Gateway**  
 Virtual Private Gateway

4. The dropdown displays available TGWs. Select the one that the deployment stack created and click *Save routes*.

[Route Tables](#) > [Edit routes](#)

## Edit routes

| Destination | Target                | Status | Propagated |
|-------------|-----------------------|--------|------------|
| 10.0.0.0/16 | local                 | active | No         |
| 0.0.0.0/0   | nat-017c74b8c872dff70 | active | No         |
| 10.1.0.0/16 | tgw-                  |        | No         |

**Add route**

\* Required

Cancel **Save routes**

tgw-092e9c685c54d1172 transit-gateway-demo-000001-has-a-very-long-prefix-512db680-transit-gateway



If you want to route all traffic to the TGW, add a new route for destination 0.0.0.0/0. If this route already exists, remove the route and add a new one for the same destination with the target set to the TGW that the deployment stack.

**To create a propagation in the inbound route table:**

1. In the left navigation tree, click *Transit Gateways > Transit Gateway Route Tables*.
2. Select the *<ResourceTagPrefix>-transit-gateway-route-table-inbound* route table.

The screenshot shows the AWS Management Console interface for Transit Gateway Route Tables. At the top, there's a search bar and a list of route tables. The selected route table is 'transit-gateway-demo-000001-has-a-very-long-prefix-512db680-transit-gateway-route-table-inbound'. Below the list, the 'Propagations' tab is selected, and the 'Create propagation' button is highlighted. The message 'This route table does not have any propagated attachments' is displayed at the bottom.

3. Click the *Propagations* tab and then click *Create propagation*.
4. From *Choose attachment to propagate*, select the attachment created in the section [To create a TGW attachment: on page 76](#).

[Transit Gateway Route Tables](#) > Create propagation

## Create propagation

Adding a propagation will allow routes to be propagated from an attachment to the target Transit Gateway route table. An attachment can be propagated to multiple route tables.

Transit Gateway ID tgw-09844e6562e187959

Transit Gateway route table ID tgw-rtb-0e2cd1bf0d609d7b9

Choose attachment to propagate\*

Filter by attributes

| Attachment ID                | Name tag  | Resource ID           | Resource owner ID | Association route table |
|------------------------------|---|-----------------------|-------------------|-------------------------|
| tgw-attach-0adeba36ce982a638 | transit-gateway-demo-transit-gateway-attachment-vpc01 | vpc-022728efe8f41cb7f | 254414331203      |                         |

\* Required

- Click *Create propagation* and then click *Close*.
- The new propagation with *Resource type* VPC is now listed on the *Propagations* tab.

Create Transit Gateway Route Table

Actions ▾

Filter by tags and attributes or search by keyword
 1 to 3 of 3

| Name   | Transit Gateway route table |
|--|-----------------------------|
| transit-gateway-demo-000001-has-a-very-long-prefix-512db680-transit-gateway-route-table-inbound  | tgw-rtb-0e2cd1bf0d609d7b9   |
| transit-gateway-demo-000001-has-a-very-long-prefix-512db680-transit-gateway-route-table-outbound | tgw-rtb-04821b397ed         |

Transit Gateway Route Table: tgw-rtb-0e2cd1bf0d609d7b9

Details

Associations

Propagations

Routes

Tags

Create propagation

Delete propagation

Filter by attributes or search by keyword
 1 to 1 of 1

| Attachment ID                | Resource type | Resource ID           |
|------------------------------|---------------|-----------------------|
| tgw-attach-0adeba36ce982a638 | VPC           | vpc-022728efe8f41cb7f |

- Click the *Routes* tab to see that the route for your VPC has been automatically propagated.

**Transit Gateway Route Table:** tgw-rtb-0e2cd1bf0d609d7b9

Details Associations Propagations **Routes** Tags

The table below will return a maximum of 1000 routes. Narrow the filter or use export routes to view more routes.

Create route Replace route Delete route

Filter by attributes or search by keyword

| CIDR        | Attachment   | Resource type | Route type |
|-------------|--|---------------|------------|
| 10.0.0.0/16 | tgw-attach-0adeba36ce982a638   vpc-022728efe8f41cb7f | VPC           | propagated |

### To create an association in the outbound route table:

- In the left navigation tree, click *Transit Gateways > Transit Gateway Route Tables*.
- Select the *<ResourceTagPrefix>-transit-gateway-route-table-outbound* route table.

Create Transit Gateway Route Table Actions

Filter by tags and attributes or search by keyword

| Name   | Transit Gateway route     |
|--|---------------------------|
| transit-gateway-demo-000001-has-a-very-long-prefix-512db680-transit-gateway-route-table-inbound  | tgw-rtb-0e2cd1bf0d609d    |
| transit-gateway-demo-000001-has-a-very-long-prefix-512db680-transit-gateway-route-table-outbound | tgw-rtb-04821b397ed85652a |

**Transit Gateway Route Table:** tgw-rtb-04821b397ed85652a

Details **Associations** Propagations Routes Tags

Create association Delete association

Filter by attributes or search by keyword

| Attachment ID                | Resource type | Resource ID           |
|------------------------------|---------------|-----------------------|
| tgw-attach-0d55b7a5da4e3595a | VPC           | vpc-0b540c0a075009c1c |

- Click the *Associations* tab and then click *Create association*.

4. From *Choose attachment to associate*, select the attachment created in the section [To create a TGW attachment: on page 76](#).

[Transit Gateway Route Tables](#) > Create association

## Create association

Associating an attachment to a route table allows traffic to be sent from the attachment to the target route table. An attachment can only be associated to one route table.

Transit Gateway ID `tgw-09844e6562e187959`

Transit Gateway route table ID `tgw-rtb-04821b397ed85652a`

Choose attachment to associate\*

\* Required

| Attachment ID                             | Name tag  | Resource ID                        | Resource owner ID | Association route table |
|---|---|------------------------------------|-------------------|-------------------------|
| <code>tgw-attach-0adeba36ce982a638</code> | transit-gateway-demo-transit-gateway-attachment-vpc01 | <code>vpc-022728efe8f41cb7f</code> | 254414331203      |                         |

5. Click *Create association* and then click *Close*.
6. The new association with *Resource type* VPC is now listed on the *Associations* tab.

Transit Gateway Route Table: `tgw-rtb-04821b397ed85652a`

Details **Associations** Propagations Routes Tags

Create association Delete association

Filter by attributes or search by keyword

|                          | Attachment ID                             | Resource type | Resource ID                        |
|--------------------------|---|---------------|------------------------------------|
| <input type="checkbox"/> | <code>tgw-attach-0d55b7a5da4e3595a</code> | VPC           | <code>vpc-0b540c0a075009c1c</code> |
| <input type="checkbox"/> | <code>tgw-attach-0adeba36ce982a638</code> | VPC           | <code>vpc-022728efe8f41cb7f</code> |

The VPC is now connected to the FortiGate Autoscale TGW. For a technical view of attaching VPCs to the FortiGate Autoscale TGW, see the architectural diagram.

## Troubleshooting

### CREATE\_FAILED error in CloudFormation stack

If you encounter a CREATE\_FAILED error when you launch the Quick Start, relaunching the template with *Rollback on failure* set to *Disabled* is recommended. (This setting is under *Advanced options* in the AWS CloudFormation console, *Configuring option settings* page.) With this setting, the stack's state is retained and the instance is left running, so you can troubleshoot the issue.



When you set *Rollback on failure* to *Disabled*, you continue to incur AWS charges for this stack. Ensure that you delete the stack when you finish troubleshooting.

---

See [Troubleshooting AWS CloudFormation](#) on the AWS website.

The deployment also fails if you select an instance type that is not supported in the region that was selected. Your desired instance type is available in your region if it is listed on the [Instance types](#) page for your region.

### The election of the primary FortiGate-VM was not successful

If the election of the primary FortiGate-VM is not successful, reset the elected primary FortiGate-VM. If the reset does not solve the problem, contact support.

### Resetting the elected primary FortiGate-VM

To reset the elected primary FortiGate-VM, navigate to the DynamoDB table `<ResourceTagPrefix>-FortiGatePrimaryElection`. Click the *Items* tab and delete the only item in the table.

A new primary FortiGate-VM is elected and a new record is created as a result.

For details on locating the DynamoDB table `<ResourceTagPrefix>-FortiGatePrimaryElection`, see [Locating deployed resources on page 65](#).

## Appendix

### FortiGate Autoscale for AWS features

#### Major components

- *The BYOL Auto Scaling group.* This Auto Scaling group contains zero to many FortiGate-VMs of the BYOL licensing model and will dynamically scale-out or scale-in based on the scaling metrics specified by the parameters *Scale-out threshold* and *Scale-in threshold*. For each instance you must provide a valid license purchased from FortiCare.



For BYOL-only and hybrid licensing deployments, the Minimum group size (`FgtAsgMinSizeByol`) must be at least 2. These FortiGate-VMs are the main instances and are fixed and running 7x24. If it is set to 1 and the instance fails to work, the current FortiGate-VM configuration will be lost.

- *The On-demand Auto Scaling group.* This Auto Scaling group contains 0 to many FortiGate-VMs of the On-demand licensing model and will dynamically scale-out or scale-in based on the scaling metrics specified by the parameters *Scale-out threshold* and *Scale-in threshold*.



For on-demand-only deployments, the minimum group size (`FgtAsgMinSizePayg`) must be at least 2. These FortiGate-VMs are the main instances and are fixed and running 7x24. If it is set to 1 and the instance fails to work, the current FortiGate-VM configuration will be lost.

- *The “assets” folder in the S3 Bucket.*
  - The *configset* folder contains files that are loaded as the initial configuration for a new FortiGate-VM instance.
    - *baseconfig* is the base configuration. This file can be modified as needed to meet your network requirements. Placeholders such as `{SYNC_INTERFACE}` are explained in the [Configset placeholders on page 84](#) table below
    - *httproutingpolicy* and *httpsroutingpolicy* are provided as part of the base configset - for a common use case - and specify the FortiGate firewall policy for VIPs for *http* routing and *https* routing respectively. This common use case includes a VIP on port 80 and a VIP on port 443 with a policy that points to an internal load balancer. The port numbers are configurable and can be changed during CFT deployment. Additional VIPs can be added here as needed.



In FortiOS 6.2.3, any VIPs created on the primary instance will not sync to the secondary instances. Any VIP you wish to add must be added as part of the base configuration.

If you set the *Internal ELB options* parameter to `do not need one`, then you must include your VIP configuration in the base configuration.

- The `... >license-files > fortigate` folder contains BYOL license files.
- *Tables in DynamoDB.* These tables are required to store information such as health check monitoring, primary election, state transitions, etc. These records should not be modified unless required for troubleshooting purposes.
- *Networking Components* These are the network load balancers, the target group, and the VPC and subnets. You are expected to create your own client and server instances that you want protected by the FortiGate-VM.

## Configset placeholders

When the FortiGate-VM requests the configuration from the Auto Scaling Handler function, the placeholders in the table below will be replaced with actual values about the Auto Scaling group.

| Placeholder                   | Type | Description  |
|-------------------------------|------|--|
| <code>{SYNC_INTERFACE}</code> | Text | The interface for FortiGate-VMs to synchronize information.<br>Specify as port1, port2, port3, etc.<br>All characters must be lowercase. |
| <code>{CALLBACK_URL}</code>   | URL  | The endpoint URL to interact with the auto scaling handler script.   |



| Placeholder           | Type   | Description  |
|-----------------------|--------|--|
|                       |        | Automatically generated during CloudFormation deployment.  |
| {PSK_SECRET}          | Text   | The Pre-Shared Key used in FortiOS.<br>Specified during CloudFormation deployment.   |
| {ADMIN_PORT}          | Number | A port number specified for admin login.<br>A positive integer such as 443 etc.<br>Specified during CloudFormation deployment.   |
| {HEART_BEAT_INTERVAL} | Number | The time interval (in seconds) that the FortiGate-VM waits between sending heartbeat requests to the Autoscale handler function. |

## Auto Scaling Handler environment variables

| Variable name       | Description   |
|---------------------|---|
| UNIQUE_ID           | Reserved, empty string.   |
| CUSTOM_ID           | Reserved, empty string.   |
| RESOURCE_TAG_PREFIX | The value of the CFT parameter <i>Resource tag prefix</i> which is described in the section <a href="#">Resource tagging configuration on page 52</a> . |

## AWS GovCloud (US) support

The AWS GovCloud (US) regions `us-gov-east-1` and `us-gov-west-1` are supported.

AWS may have service limitations, restrictions, or different implementations for these regions. Review [AWS documentation](#) for more information.

As service is provided differently than it is for commercial regions, if you encounter errors when deploying to these regions, report them on the [Issues](#) tab of the FortiGate Autoscale for AWS GitHub project.

## How to partially route egress traffic

By default, FortiGate Autoscale manages the route `0.0.0.0/0` in the route table associated with the FortiGate-VM cluster. As such, all egress traffic will be routed to the primary FortiGate-VM. If desired, you can add firewall policies to the FortiGate-VM with more customized egress rules.

In addition to the `0.0.0.0/0` route via FortiGate Autoscale, egress traffic can be also routed via other NAT gateways. This is done by creating a route with a specific destination with the NAT device as the target. This route must be next to the route `0.0.0.0/0` in the Autoscale route table and the route destination must be a valid CIDR. For example, for egress traffic to the IP address range `10.0.0.0/16` to use a different NAT device, create a route with destination `10.0.0.0/16` and the NAT device as the target. Egress traffic to `10.0.0.0/16` will now flow through the NAT device while the rest will still flow through FortiGate.

However, you cannot use the route with destination `0.0.0.0/0` because FortiGate Autoscale is managing it and will overwrite it whenever the FortiGate primary role has been switched.

## Deployment templates

Deploying FortiGate Autoscale for AWS requires the use of deployment templates. There are two types of templates:

- *Entry template*. This template could run as the entry point of a deployment.
- *Dependency template*. This template is automatically run by the deployment process as a Nested Stack. It cannot be run as an entry template. A dependency template is run based on user selected options.

Following are descriptions of the templates included in the FortiGate Autoscale for AWS deployment package.

| Template  | Type                | Description  |
|---|---------------------|--|
| autoscale-new-vpc.template.yaml                 | Entry template      | Deploys the Auto Scaling solution to a new VPC.  |
| autoscale-existing-vpc.template.yaml            | Entry template      | Deploys the Auto Scaling solution to an existing VPC.  |
| autoscale-tgw-new-vpc.template.yaml             | Entry template      | Deploys the Auto Scaling solution with Transit Gateway Integration to a new VPC.   |
| autoscale-main.template.yaml                    | Dependency template | Does the majority of the work for deploying FortiGate Autoscale.   |
| configure-fortianalyzer-service.template.yaml   | Dependency template | Configure the FortiAnalyzer integration additional services.   |
| copy-objects.template.yaml                      | Dependency template | Creates an S3 bucket in the same region where the stack is launched and copies deployment related objects to this S3 bucket. |
| create-autoscale-handler.template.yaml          | Dependency template | Creates a FortiGate Autoscale Handler Lambda function and an API Gateway.  |
| create-db-table.template.yaml                   | Dependency template | Creates all necessary DynamoDB tables for the FortiGate Autoscale solution.  |
| create-fortianalyzer-components.template.yaml   | Dependency template | Deploys a FortiAnalyzer to a selected subnet and configures all FortiGates to connect to it.                                 |
| create-fortigate.template.yaml                  | Dependency template | Deploys a FortiGate EC2 instance to a subnet using a given FortiGate AML, security group, and instance profile.              |
| create-hybrid-auto-scaling-group.template.yaml  | Dependency template | Deploys the hybrid licensing FortiGate Auto Scaling groups.  |
| create-load-balancer.template.yaml              | Dependency template | Deploys network traffic Load Balancers and components for FortiGate Autoscale.   |
| create-new-vpc.template.yaml                    | Dependency template | Creates a new VPC in which to deploy the FortiGate Autoscale solution.   |
| create-transit-gateway-components.template.yaml | Dependency template | Creates a Transit Gateway for FortiGate Autoscale for AWS.   |
| create-tgw-vpn-handler.template.yaml            | Dependency template | Creates a service for Transit Gateway VPN management.  |

## Cloud-init

In autoscaling, a FortiGate-VM uses the `cloud-init` feature to preconfigure the instances when they first come up. During template deployment, an internal API gateway endpoint is created.

A FortiGate-VM sends requests to the endpoint to retrieve necessary configuration after initialization.

Use this FortiOS CLI command to display information for your devices:

```
diagnose debug cloudinit show
```

You can retrieve VPN output with this FortiOS CLI command:

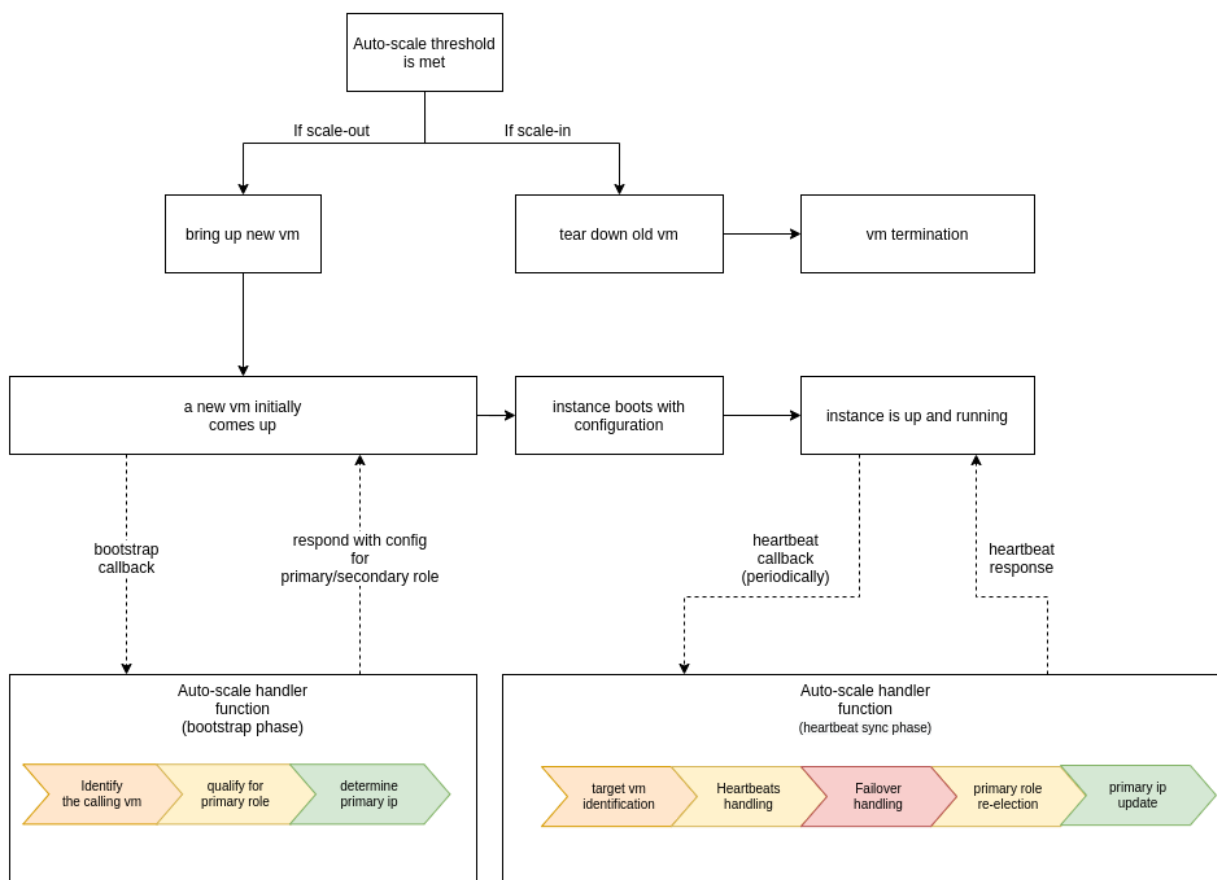
```
# diagnose vpn tun list
```

## Architectural diagrams

The following diagrams illustrate the different aspects of the architecture of FortiGate Autoscale for AWS.

### Autoscale handler flowchart

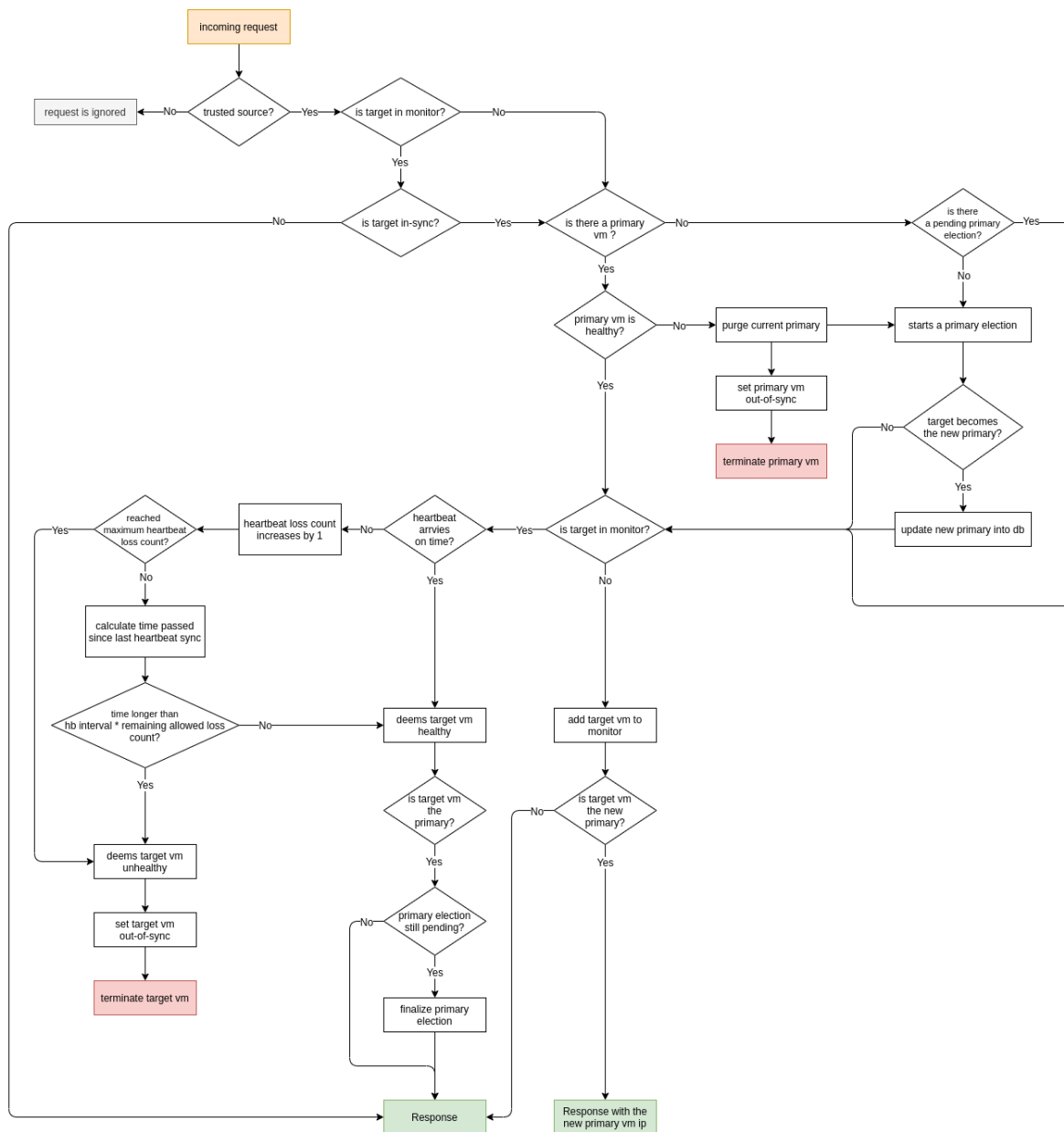
#### Autoscale handler flowchart



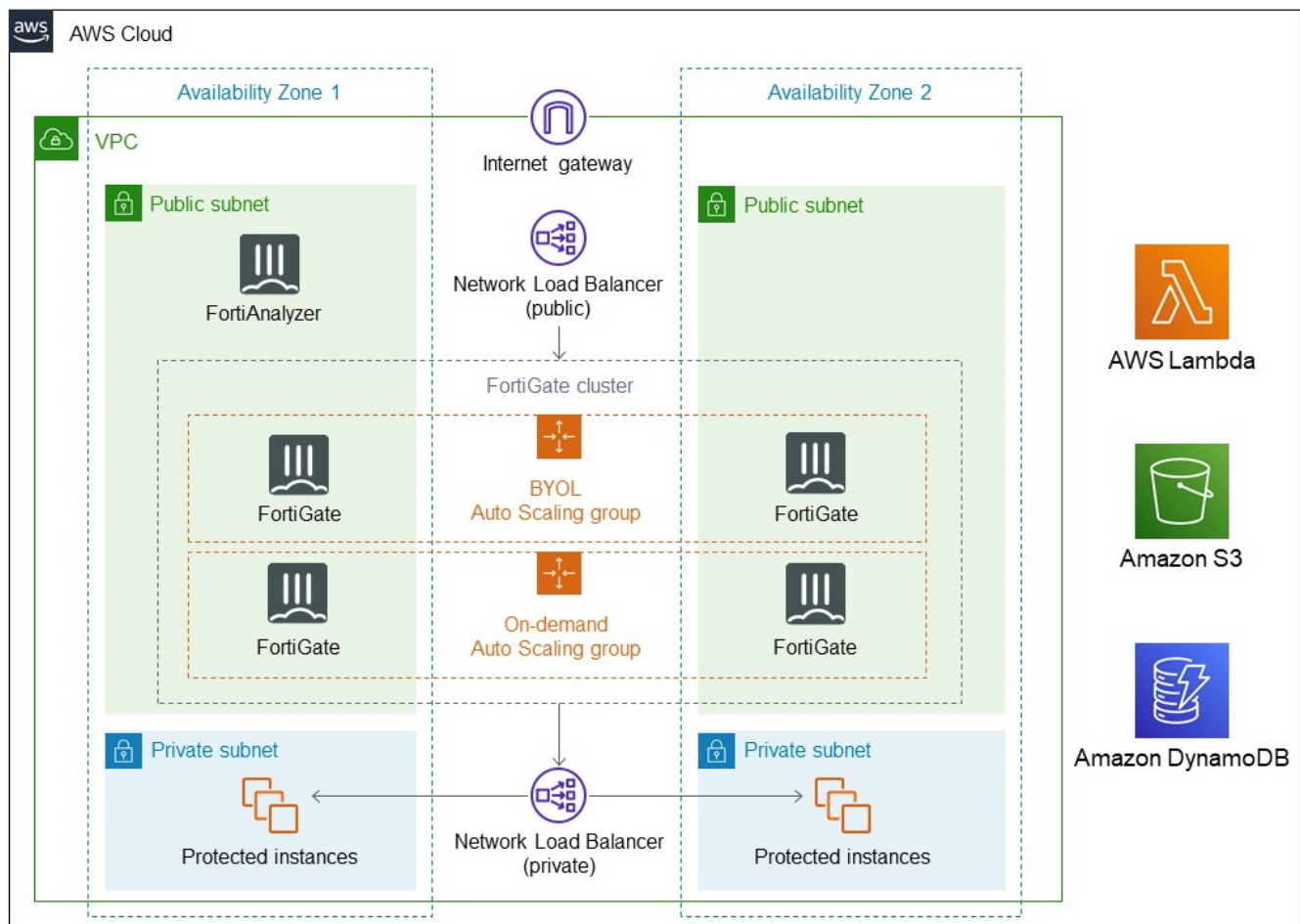
## Primary election

### FortiGate Autoscale

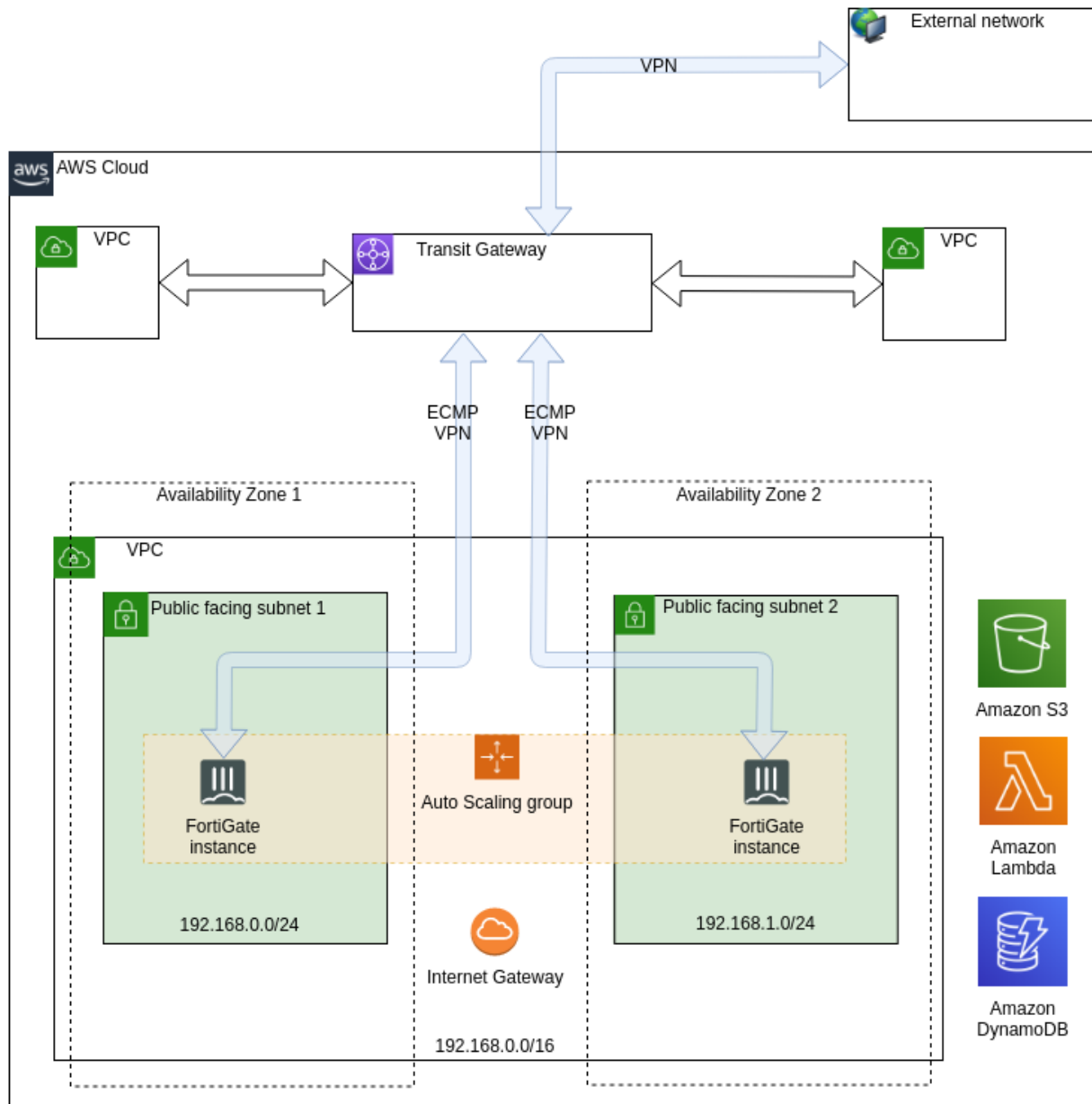
with heartbeat response & failover management



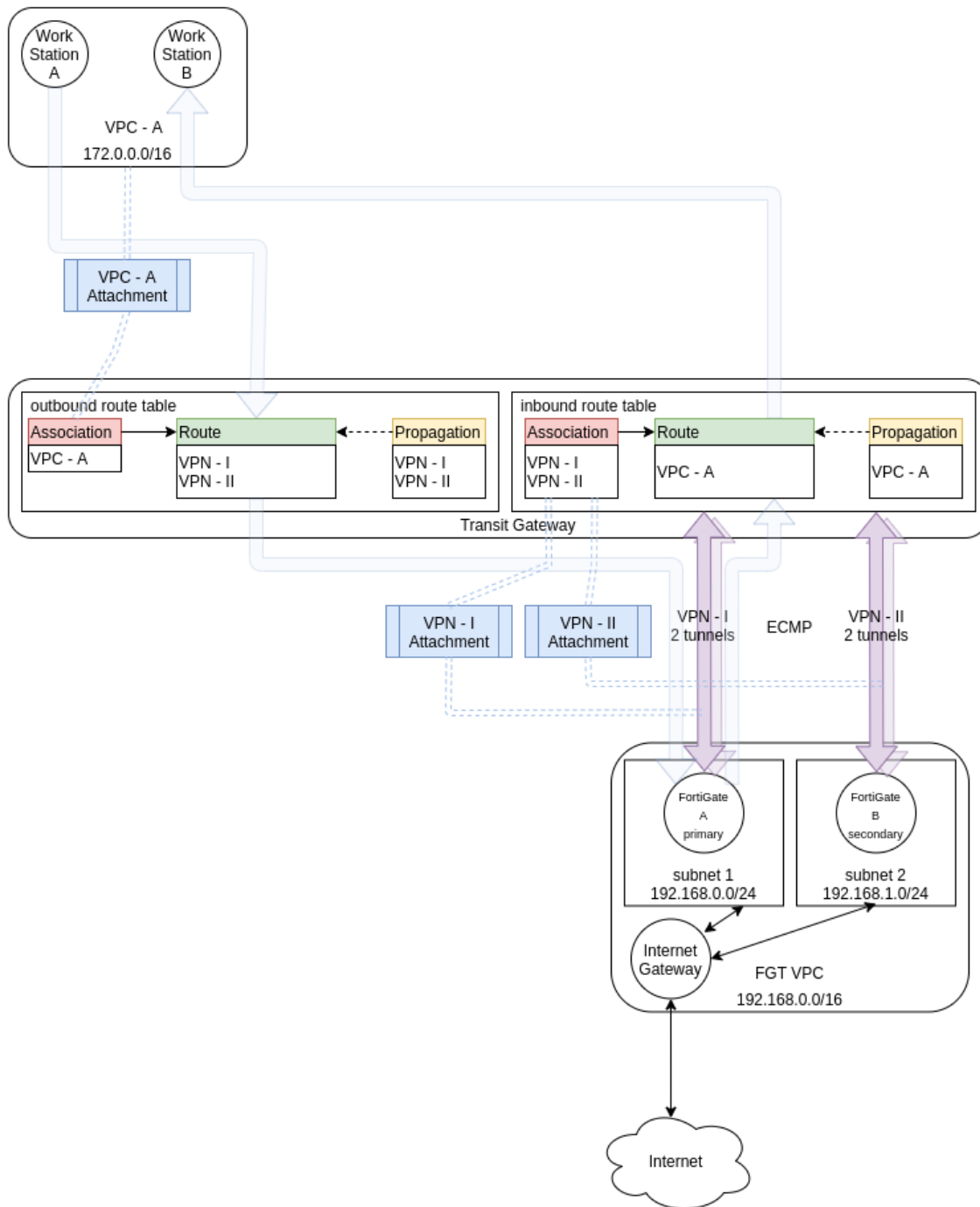
## FortiGate Autoscale VPC



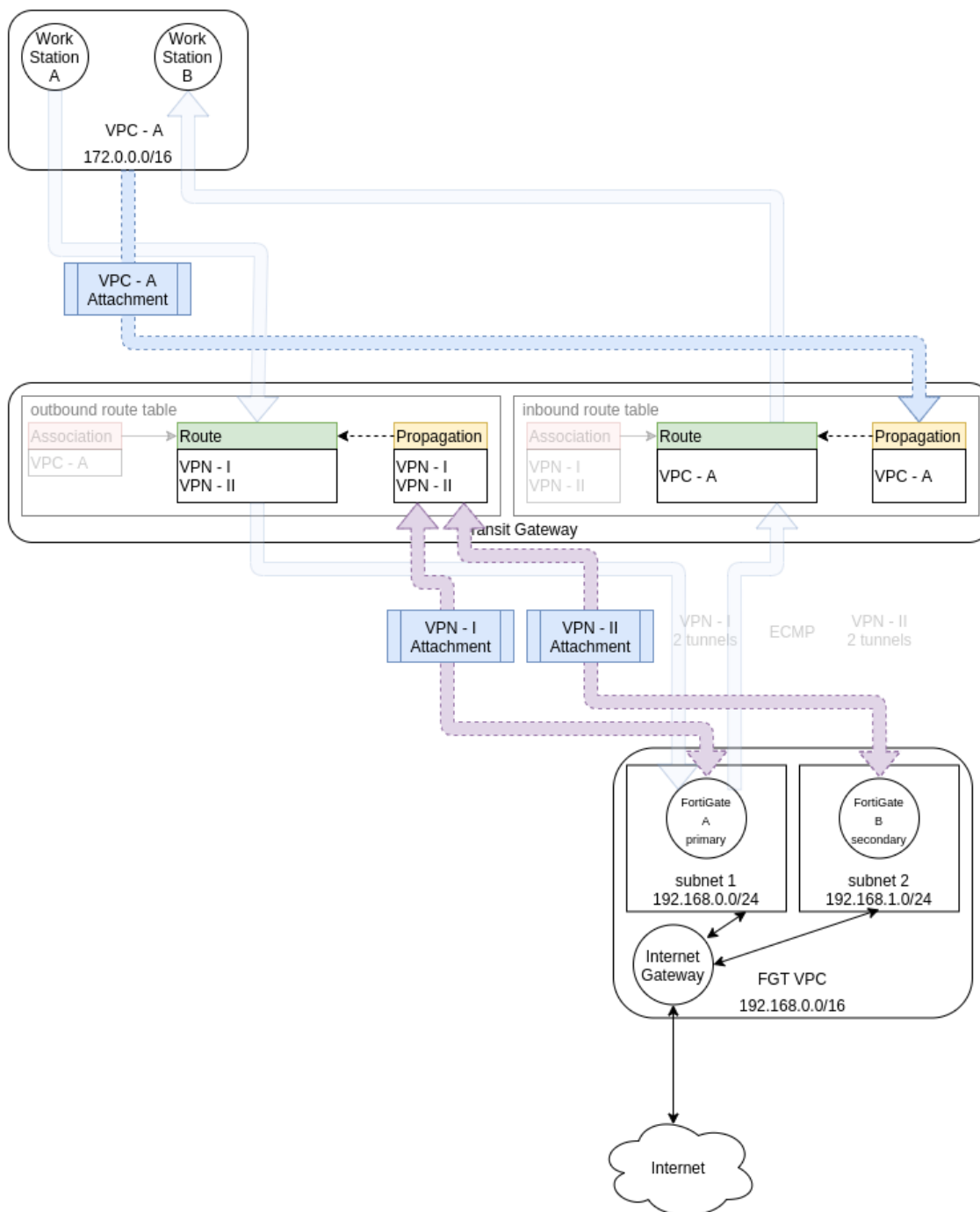
## FortiGate Autoscale VPC attached to a Transit Gateway



## FortiGate Autoscale VPC integration with Transit Gateway

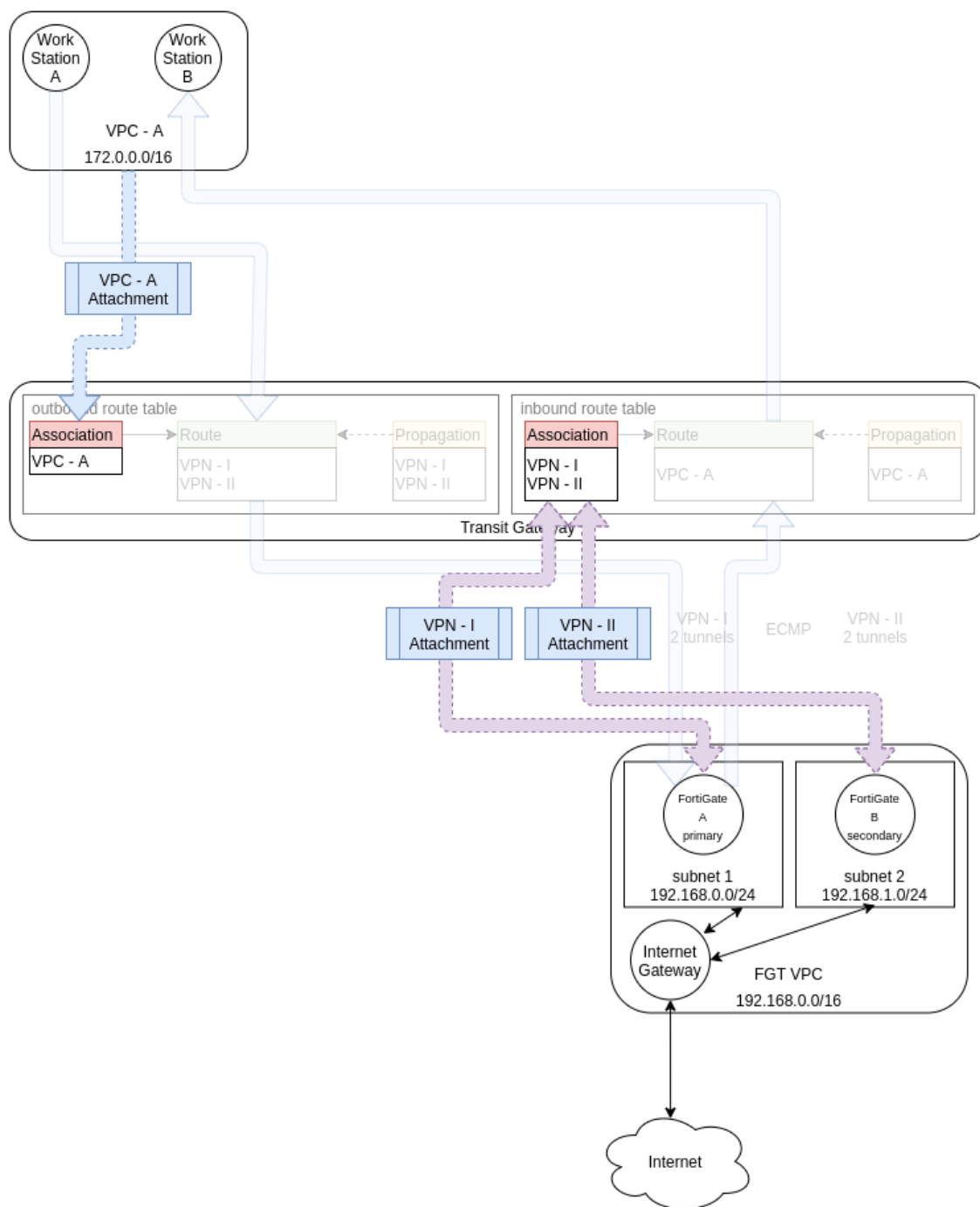


## Route propagation





## Route associations



## Upgrading the deployment

The following provides steps to apply firmware updates to the FortiGate instances that the AWS Autoscaling deployment deployed.



Back up all FortiGate configurations prior to upgrading the FortiGate instances.

---

**To upgrade the deployment:**

1. Edit the autoscaling group to suspend the health check, launch, and terminate processes:
    - a. In the AWS management console, go to *EC2 > Auto Scaling > Auto Scaling Groups*.
    - b. Edit the desired pay-as-you-go (PAYG) and/or bring your own license (BYOL) autoscaling group.
    - c. On the *Details* tab, go to *Advanced Configurations*, then click *Edit*.
    - d. From the *Suspended Processes* dropdown list, select *Health Check*, *Launch*, and *Terminate*.
    - e. Click *Update* to save the changes.
- 



Using the *Instance Refresh* option is not recommended, as this is designed for truly ephemeral instances, which the FortiGate instances may not be.

---

2. Confirm the new AMI ID for PAYG or BYOL as desired for your region.
- 



You can find the specific FortiGate AMI ID by going to the marketplace listing for FortiGate PAYG or BYOL, selecting *Subscribe*, continuing to configuration and confirming the desired region, then copying the AMI ID.

---

3. Edit the launch template or create a new one. You will need to create a new template version that references the new FortiGate version's AMI ID, so that autoscaling uses the new version for new instances:
  - a. Go to *EC2 > Instances > Launch Templates*.
  - b. Select the desired launch template for FortiGate BYOL and PAYG.
  - c. From the *Actions* menu, select *Modify Template (Create new version)*.
  - d. Under *Application and OS Images*, paste the AMI ID that you confirmed in step 2 in the searchbar.
  - e. Select the desired FortiGate marketplace offering.
  - f. Click *Continue*. EC2 may display a warning that your security group rules may be overridden if you proceed. Under *Network settings > Firewall (security groups)*, click *Select existing security group*, and select the

previously selected security group before saving or creating a new version of the launch template.

▼
**Network settings**

Subnet
[Info](#)

Don't include in launch template ▼

↺
Create new subnet

When you specify a subnet, a network interface is automatically added to your template.

**Firewall (security groups)**

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Select existing security group
☐ Create security group

Common security groups
[Info](#)

Select security groups ▼

↺
Compare security group rules

asg-stack-fgt-update-StackMainWorkload-12PRQIE5J48GG-FgtAsgSecurityGroup-AWGX01ISDF6Z

×

sg-0488a66c972fdf6be

VPC: vpc-0885c45da7317926d

Security groups that you add or remove here will be added to or removed from all your network interfaces.

▶
**Advanced network configuration**


4. Edit the BYOL or PAYG autoscaling group and update the launch template version to the new version:
  - a. Go to *EC2 > Auto Scaling > Auto Scaling Groups*.
  - b. Select the desired scaling group.
  - c. In *LAUNCH TEMPLATE*, select *Edit*.


- d. From the *Version* dropdown list, select the new version.

## Edit asg-stack-fgt-update-fd2163a0-fortigate-byol-auto-scaling-group [Info](#)


**Launch template** [Info](#)
[Switch to launch configuration](#)


Launch template  
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

asg-stack-fgt-update-fd2163a0-fortigate-byol-autoscale-launch-template ▼ 

[Create a launch template](#) 

Version

▲ 

[Version](#) 

|             |
|-------------|
| Latest (2)  |
| Default (2) |
| 2           |
| 1           |

|  |   |                                     |
|--|---|-------------------------------------|
| <b>AMI ID</b><br>ami-0c186535bfe65e6a9     | <b>Launch template</b><br>asg-stack-fgt-update-fd2163a0-fortigate-byol-autoscale-launch-template <a href="#">Info</a><br>lt-0feabc34b9919fb96 | <b>Instance type</b><br>c5.xlarge   |
| <b>Key pair name</b><br>[redacted]-keypair | <b>Security groups</b><br>-   | <b>Request Spot Instances</b><br>No |
| <b>Additional details</b>                  |   |                                     |
| <b>Storage (volumes)</b><br>-              | <b>Date created</b><br>Mon Mar 07 2022 08:58:50 GMT-0800 (Pacific Standard Time)  |                                     |

- e. Click *Update*.
- Manually apply the update to existing instances. Starting the upgrade process on the secondary autoscale FortiGate, then the primary FortiGate, is recommended. The firmware upgrade option is only available when logged in with administrator read-write privileges. Do one of the following:
    - In FortiOS, go to *System > Firmware*. Select *FortiGuard Firmware*, then *Backup Config*. Upgrade to the latest available firmware.
    - Log in to FortiOS as the admin user. Go to *System Firmware*. Under *Upload Firmware*, browse to and locate the previously downloaded firmware image file. Click *Backup config and upgrade*. The FortiGate backs up the current configuration to the management computer, uploads the firmware image file, upgrades to the new firmware version, and restarts. This process takes a few minutes.
  - Resume health check, launch, and terminate processes:
    - Go to *EC2 > Auto Scaling Groups*.
    - Edit the desired autoscaling group.

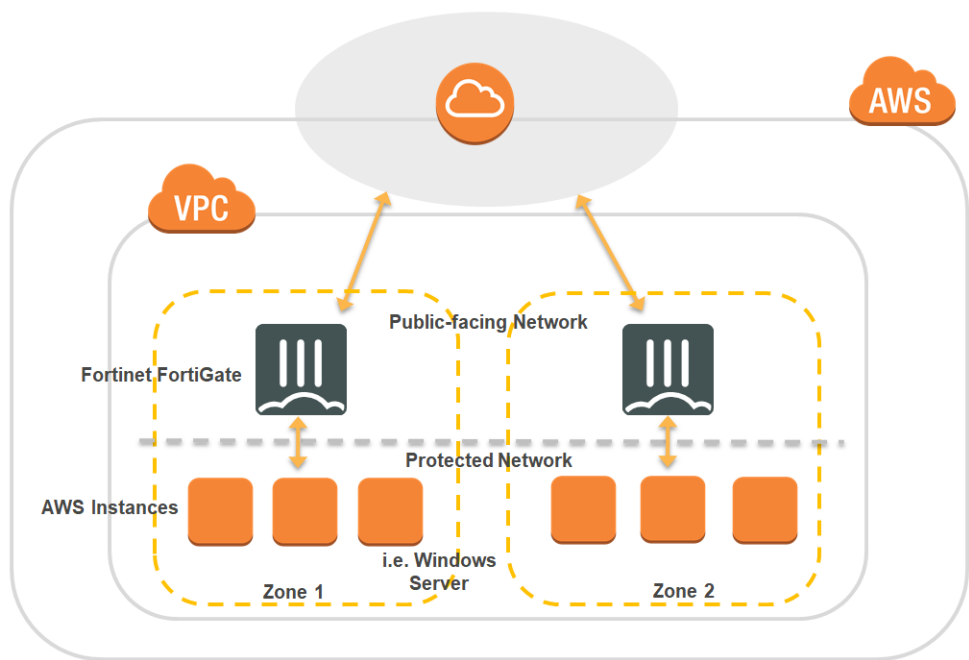
- c. Go to *Advanced Configuration > Edit*.
- d. Deselect *Health Check*, *Launch Instances*, and *Terminate Instances*.
- e. Click *Update*.

## FortiGate Autoscale for AWS document history

| Template       | Details  |
|----------------|--|
| 3.4.0          | Updated the <a href="#">CFT parameters on page 52</a>  |
| 3.3.2 (latest) | Added support for FortiOS 6.2.5, FortiOS 6.4.6, FortiOS 7.0.0, and FortiOS 7.0.1. Removed support for FortiOS 6.2.3 and FortiOS 6.4.4. Added support for FortiAnalyzer 6.4.6. Removed support for FortiAnalyzer 6.2.5 and FortiAnalyzer 6.4.4.   |
| 3.3.1          | Added <a href="#">Requirements when using an existing VPC on page 45</a>   |
| 3.3            | Added support for AWS GovCloud (US); VPN connections now use Diffie-Hellman Group 14 and SHA256 (Secure Hash Algorithm 2); increased stack security.   |
| 3.2            | Added support for FortiOS 6.4.4. FortiAnalyzer can now be integrated into the deployment.  |
| 3.0            | Supports any combination of BYOL and On-demand instances as well as the option for Transit Gateway integration. Requires FortiOS 6.2.3.  |
| 2.0            | <p>Added support for Hybrid Licensing (any combination of BYOL and/or On-demand instances) with no Transit Gateway integration. Transit Gateway support is only for On-demand instances. Documentation is no longer maintained and is only available as a PDF:</p> <ul style="list-style-type: none"> <li>• <a href="#">Deploying auto scaling on AWS without Transit Gateway integration 2.0</a> <ul style="list-style-type: none"> <li>• Requires FortiOS 6.2.3.</li> </ul> </li> <li>• <a href="#">Deploying auto scaling on AWS with Transit Gateway integration 1.0</a> <ul style="list-style-type: none"> <li>• Requires FortiOS 6.2.1.</li> </ul> </li> </ul> |
| 1.0            | <p>Supports auto scaling for On-demand instances; does not support Transit Gateway integration. Requires FortiOS 6.0.6 or FortiOS 6.2.1. Documentation is no longer maintained and is only available as a PDF:</p> <ul style="list-style-type: none"> <li>• <a href="#">Deploying auto scaling on AWS 1.0</a></li> </ul>   |

# Single FortiGate-VM deployment

You can deploy the FortiGate-VM enterprise firewall for AWS as a virtual appliance in AWS (infrastructure as a service (IaaS)). This section shows you how to install and configure a single instance FortiGate-VM in AWS to provide a full next generation firewall/unified threat management security solution to protect your workloads in the AWS IaaS.



Networking is a core component in using AWS services, and using virtual private clouds, subnets, and virtual gateways help you to secure your resources at the networking level.

This section covers the deployment of simple web servers, but you can use this deployment type for any type of public resource protection with only slight modifications. With this architecture as a starting point, you can implement more advanced solutions, including multitiered solutions.

The example creates two subnets:

| Subnet  | Connects the FortiGate-VM to...                |
|---------|--|
| Subnet1 | AWS virtual gateway on the public-facing side. |
| Subnet2 | Windows server on the private side.            |

## Determining your licensing model

On-demand users do not need to register from the FortiGate-VM GUI console. If you are using an on-demand licensing model, once you create the FortiGate-VM instance in AWS, contact [Fortinet Customer Support](#) with the following information:

- Your FortiGate-VM instance serial number
- Your Fortinet account email ID. If you do not have a Fortinet account, you can create one at [Customer Service & Support](#).

If you are deploying a FortiGate-VM in the AWS marketplace with bring your own license, you must obtain a license to activate it.

See [Creating a support account on page 18](#).

## Creating a VPC and subnets

This section shows you how to create an AWS VPC and create two subnets in it. For many steps, you have a choice to make that can be specific to your own environment.

If deploying to outposts, create the subnets on the outpost, per AWS documentation.

### To create a VPC and subnets:

1. Log in to the [AWS Management Console](#).
2. Go to *Networking & Content Delivery > VPC*.
3. Go to *Virtual Private Cloud > Your VPCs*, then select *Create VPC*.
4. In the *Name tag* field, set the VPC name.
5. In the *CIDR block* field, specify an IPv4 address range for your VPC.
6. In the *Tenancy* field, select *Default*.
7. Select *Yes, Create*.
8. In the *Virtual Private Cloud* menu, select *Subnets*, then select *Create Subnet*. Create a public subnet (in this example, *Subnet1*) and a private subnet (*Subnet2*), as shown in this example. Both subnets belong to the VPC that you created.

### Create Subnet ✕

Use the CIDR format to specify your subnet's IP address block (e.g., 10.0.0.0/24). Note that block sizes must be between a /16 netmask and /28 netmask. Also, note that a subnet can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag

?

VPC

vpc-76581310 | FortiGateVPC

?

VPC CIDRs

| CIDR        | Status     | Status Reason |
|-------------|------------|---------------|
| 10.0.0.0/16 | associated |               |

Availability Zone

No Preference

?

IPv4 CIDR block

✕
?

Cancel

Yes, Create

## Create Subnet

Use the CIDR format to specify your subnet's IP address block (e.g., 10.0.0.0/24). Note that block sizes must be between a /16 netmask and /28 netmask. Also, note that a subnet can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag  ⓘ

VPC  ⓘ

VPC CIDRs

| CIDR        | Status     | Status Reason |
|-------------|------------|---------------|
| 10.0.0.0/16 | associated |               |

Availability Zone  ⓘ

IPv4 CIDR block  ⓘ

Cancel

Yes, Create

VPC Dashboard

Filter by VPC:

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Endpoints

NAT Gateways

Peering Connections

Create Subnet

Subnet Actions

<< 1 to 2 of 2 Subnets >>

| <input type="checkbox"/>            | Name              | Subnet ID       | State     | VPC                         | IPv4 CIDR   | Available IPv4 | IPv6 CIDR |
|-------------------------------------|-------------------|-----------------|-----------|-----------------------------|-------------|----------------|-----------|
| <input checked="" type="checkbox"/> | Private-FortiGate | subnet-0d900e6b | available | vpc-76581310   FortiGateVPC | 10.0.1.0/24 | 251            |           |
| <input type="checkbox"/>            | Public-FortiGate  | subnet-00960866 | available | vpc-76581310   FortiGateVPC | 10.0.0.0/24 | 251            |           |

subnet-0d900e6b | Private-FortiGate

Summary

Route Table

Network ACL

Flow Logs

Tags

Subnet ID: subnet-0d900e6b | Private-FortiGate

Availability Zone: us-west-2a

IPv4 CIDR: 10.0.1.0/24

Route table: rtb-145ce76d

IPv6 CIDR:

Network ACL: acl-7952ba00

State: available

Default subnet: no

## Attaching the new VPC Internet gateway

If you are using the default VPC, the Internet gateway should already exist.

**To attach the new VPC Internet gateway:**

1. In the *Virtual Private Cloud* menu, select *Internet Gateways*, then select *Create Internet Gateway*.
2. In the *Name tag* field, set the Internet gateway name, then select *Yes, Create*.
3. Select the Internet gateway, then select *Attach to VPC*.



4. Select the VPC that you created and select *Yes, Attach*. The Internet gateway state changes from *detached* to *attached*.

## Subscribing to the FortiGate



Downgrading to a previous GA version that does not support UEFI boot mode when using a UEFI-enabled FortiGate instance is not supported. FortiOS 7.4 supports UEFI boot mode.

---

### To subscribe to the FortiGate:

1. Go to the AWS Marketplace's page for Fortinet [FortiGate-VM \(BYOL\)](#) or [FortiGate-VM \(on-demand\)](#). Select *Continue*.
2. Select *Manual Launch*.
3. Select *Launch with EC2 Console* beside the region you want to launch.
4. Select an instance type, then select *Next: Configure Instance Details*.
5. Configure instance details:
  - a. In the *Network* field, select the VPC that you created.
  - b. In the *Subnet* field, select the public subnet.
  - c. In the *Network interfaces* section, you see the entry for *eth0* that was created for the public subnet. Select *Add Device* to add another network interface (in this example, *eth1*), and select the private subnet. Assigning static IP addresses is recommended.
  - d. When you have two network interfaces, an elastic IP address (EIP) is not assigned automatically. You must manually assign one later. Select *Review and Launch*, then select *Launch*.
6. Select an existing key pair or create a new key pair. Select the acknowledgment checkbox. Select *Launch Instances*.
7. To easily identify the instance, set a name for it in the *Name* field.
8. On-demand FortiGate-VMs require connectivity to [FortiCare](#) to obtain a valid license. Without connectivity to FortiCare, the FortiGate-VM shuts down for self-protection. Ensure the following:
  - a. Outgoing connectivity to <https://directregistration.fortinet.com:443> is allowed in security groups and ACLs.
  - b. You have assigned a public IP address (default or EIP). If you have not enabled a public address during instance creation, follow the remaining steps to assign an EIP and bring up the FortiGate-VM again.
9. Configure an EIP:
  - a. In the *Network & Security* menu, select *Elastic IPs*, then select one that is available for you to use or create one. Select *Actions > Associate Address*. If you do not have one available to use, create one.

The screenshot displays the AWS Elastic IP management console. On the left, the navigation pane shows 'STORE' (Volumes, Snapshots), 'NETWORK & SECURITY' (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), and 'LOAD BALANCING'. The 'Elastic IPs' section is active. The main area shows a table of Elastic IPs with columns: Elastic IP, Allocation ID, Instance, and Private IP address. There are 4 items in total. The bottom item, with IP 34.215.95.19 and Allocation ID eipalloc-b735bd8a, is selected. The 'Actions' dropdown menu is open, showing options: Release addresses, Associate address (highlighted in orange), Disassociate address, Move to VPC scope, and Restore to EC2 scope.

| <input type="checkbox"/>            | Elastic IP    | Allocation ID     | Instance | Private IP address |
|-------------------------------------|---------------|-------------------|----------|--------------------|
| <input type="checkbox"/>            | 34.210.111.61 | eipalloc-e0e736dd |          | 10.2.3.10          |
| <input type="checkbox"/>            | 34.213.132.1  | eipalloc-d6f120eb |          | 10.2.1.10          |
| <input type="checkbox"/>            | 34.213.249.84 | eipalloc-6819c855 |          | 10.2.1.67          |
| <input checked="" type="checkbox"/> | 34.215.95.19  | eipalloc-b735bd8a | -        | -                  |

- b. In the *Resource type* section, select *Network Interface*.
- c. In the *Network interface* field, select the interface ID of the network interface that you created for the public subnet (in this example, *eth0*). In the *Private IP* field, select the IP address that belongs to the public subnet. To find these values, go to the EC2 Management Console, select *Instances*, and select the interface in the *Network interfaces* section in the lower pane of the page (*Interface ID* and *Private IP Address* fields). Select *Associate*. A message displays indicating the address association succeeded. If the Internet gateway is not associated with a VPC, the EIP assignment fails.

[Addresses](#) > Associate address

## Associate address

Select the instance OR network interface to which you want to associate this Elastic IP address (34.215.95.19)

Resource type ☐ Instance ?  
☒ Network interface

Network interface eni-bd2aa69c ↻

Private IP 10.0.0.220 ↻ ?

Reassociation ☒ Allow Elastic IP to be reassociated if already attached ?



### Warning

If you associate an Elastic IP address with your instance, your current public IP address is released. [Learn more](#).

\* Required

[Cancel](#)[Associate](#)

**Network Interface eth0**

|                     |  |
|---------------------|--|
| Interface ID        | eni-bd2aa69c   |
| VPC ID              | vpc-76581310   |
| Attachment Owner    | 137016737462   |
| Attachment Status   | attached   |
| Attachment Time     | Mon Oct 16 21:27:48 GMT-700 2017                     |
| Delete on Terminate | true   |
| Private IP Address  | 10.0.0.220   |
| Private DNS Name    | -  |
| Elastic IP Address  | -  |
| Source/Dest. Check  | true   |
| Description         | Primary network interface                            |
| Security Groups     | Fortinet FortiGate-VM -BYOL--v5-6-2-AutogenByAWSMP-1 |

Platform - Network interfaces **eth0** eth1  
 IAM role - Source/dest. check True

## Creating routing tables and associate subnets

Configure the routing tables. Since the FortiGate-VM has two interfaces, one for the public subnet and one for the private subnet, you must configure two routing tables.

**To create routing tables and associate subnets:**

1. To configure the public subnet's routing table, go to *Networking & Content Delivery > VPC* in the AWS management console. In the VPC Dashboard, select *Your VPCs*, and select the VPC you created. In the *Summary* tab in the lower pane, select the route table ID located in the *Route table* field. To easily identify the route table, set a name for it in the *Name* field.

The screenshot shows the AWS VPC console. At the top, there's a 'Create VPC' button and an 'Actions' dropdown. Below is a search bar and a table of VPCs. The table has columns: Name, VPC ID, State, IPv4 CIDR, and IPv6 CIDR. Three VPCs are listed: '-vpc1' (vpc-2806934e), 'FortiGateVPC' (vpc-76581310), and '-default' (vpc-5bf17a3d). The 'FortiGateVPC' is selected. Below the table, the details for 'vpc-76581310 | FortiGateVPC' are shown in the 'Summary' tab. The details include VPC ID, Network ACL, State, Tenancy, IPv4 CIDR, DNS resolution, IPv6 CIDR, DNS hostnames, DHCP options set, ClassicLink DNS Support, and a highlighted 'Route table: rtb-fb12a982'.

| Name         | VPC ID       | State     | IPv4 CIDR     | IPv6 CIDR |
|--------------|--------------|-----------|---------------|-----------|
| -vpc1        | vpc-2806934e | available | 10.2.0.0/16   |           |
| FortiGateVPC | vpc-76581310 | available | 10.0.0.0/16   |           |
| -default     | vpc-5bf17a3d | available | 172.31.0.0/16 |           |

**vpc-76581310 | FortiGateVPC**

**Summary** | CIDR Blocks | Flow Logs | Tags

VPC ID: vpc-76581310 | FortiGateVPC  
 State: available  
 IPv4 CIDR: 10.0.0.0/16  
 IPv6 CIDR:  
 DHCP options set: dopt-80bd75e6  
 Route table: **rtb-fb12a982**

Network ACL: acl-7952ba00  
 Tenancy: Default  
 DNS resolution: yes  
 DNS hostnames: yes  
 ClassicLink DNS Support: no

2. In the *Routes* tab, select *Edit*, then select *Add another route*. In the *Destination* field, type *0.0.0.0/0*. In the *Target* field, type *igw* and select the Internet gateway from the auto-complete suggestions. Select *Save*. The default route on the public interface in this VPC is now the Internet gateway.

[Create Route Table](#)
[Delete Route Table](#)
[Set As Main Table](#)

» 1 to 1 of 1 Route Table »

| <input type="checkbox"/>            | Name                      | Route Table ID | Explicitly Associat | Main | VPC                    |
|-------------------------------------|---------------------------|----------------|---------------------|------|------------------------|
| <input checked="" type="checkbox"/> | FortiGate-Pub-route-table | rtb-fb12a982   | 1 Subnet            | Yes  | vpc-76581310   FortiGa |

rtb-fb12a982

[Summary](#)
[Routes](#)
[Subnet Associations](#)
[Route Propagation](#)
[Tags](#)

[Cancel](#)
[Save](#)

View: All rules

| Destination                            | Target   | Status | Propagated | Remove         |
|--|--|--------|------------|----------------|
| 10.0.0.0/16                            | local  | Active | No         |                |
| <input type="text" value="0.0.0.0/0"/> | <input type="text" value="igw-f4a78093"/> <span>×</span> | Active | No         | <span>×</span> |

[Add another route](#)
igw-f4a78093 | Forti Internet Gateway

3. In the *Subnet Associations* tab, select *Edit*, and select the public subnet to associate it with this routing table. Select *Save*.

rtb-fb12a982 | FortiGate-Pub-route-table

[Summary](#)
[Routes](#)
[Subnet Associations](#)

[Cancel](#)
[Save](#)

| Associate                           | Subnet                             | IPv4 CIDR   | IPv6 CIDR |
|-------------------------------------|------------------------------------|-------------|-----------|
| <input checked="" type="checkbox"/> | subnet-00960866   Public-FortiGate | 10.0.0.0/24 | -         |

4. To configure the routing table for the private subnet, select *Create Route Table*. To easily identify the route table, set a name for it in the *Name* field. Select the VPC you created. Select *Yes, Create*.

## Create Route Table

✕

A route table specifies how packets are forwarded between the subnets within your VPC, the Internet, and your VPN connection.

Name tag

i

VPC

vpc-76581310
|
FortiGateVPC
i

Cancel
Yes, Create

5. In the *Routes* tab, select *Edit*, then select *Add another route*. In the *Destination* field, type *0.0.0.0/0*. In the *Target* field, enter the interface ID of the private network interface. To find the interface ID, go to the EC2 Management Console, select *Instances*, and select the interface in the *Network interfaces* section in the lower pane of the page (*Interface ID* field). Select *Save*. The default route on the private subnet in this VPC is now the FortiGate's private network interface.

Create Route Table
Delete Route Table
Set As Main Table

✕
<< 1

|                                     | Name                       | Route Table ID | Explicitly Associat | Main |
|-------------------------------------|----------------------------|----------------|---------------------|------|
| <input checked="" type="checkbox"/> | Fortigate-Priv-route-table | rtb-2f48c956   | 0 Subnets           | No   |
| <input type="checkbox"/>            | FortiGate-Pub-route-table  | rtb-fb12a982   | 1 Subnet            | Yes  |

<
rtb-2f48c956 | Fortigate-Priv-route-table

Summary
Routes
Associations
Route Propagation
Tags

Cancel
Save

View: A

| Destination   | Status |
|---|--------|
| 10.0.0.0/16   |        |
| <input style="width: 100%;" type="text" value="0.0.0.0/0"/> |        |

eni-

eni-032da122

eni-13c54b32

eni-3f07881e

eni-54da4475

eni-55ea6474

eni-7b45ca5a

eni-960e81b7

eni-bd2aa69c

eni-

✕

Add another route

6. In the *Subnet Associations* tab, select *Edit*, select the private subnet to associate it with this routing table. Select *Save*. You have now created two routing tables, one for the public segment and one for the private segment, with default routes.



## rtb-145ce76d | FortiGate 1 routing table

| Summary Routes Subnet Associations Route Propagation Tags |                                     |             |           |  |
|---|-------------------------------------|-------------|-----------|--|
| Cancel Save   |                                     |             |           |  |
| Associate   | Subnet                              | IPv4 CIDR   | IPv6 CIDR | Current Route Table                      |
| <input type="checkbox"/>                                  | subnet-00960866   Public-FortiGate  | 10.0.0.0/24 | -         | rtb-145ce76d   FortiGate 1 routing table |
| <input checked="" type="checkbox"/>                       | subnet-0d900e6b   Private-FortiGate | 10.0.1.0/24 | -         | rtb-145ce76d   FortiGate 1 routing table |

7. In the EC2 Management Console, select *Instances*, and select the network interface that you created for the private subnet (in this example, *eth1*) in the *Network interfaces* section in the lower pane. Select the interface ID.

The screenshot shows the AWS EC2 Management Console. On the left, the 'INSTANCES' section is expanded, and 'Instances' is selected. The main pane shows a list of instances. A modal window titled 'Network Interface eth1' is open, displaying details for the interface 'eni-960e81b7'. The details include:

- Interface ID: eni-960e81b7
- VPC ID: vpc-76581310
- Attachment Owner: 137016737462
- Attachment Status: attached
- Attachment Time: Tue Oct 17 00:00:18 GMT-700 2017
- Delete on Terminate: true
- Private IP Address: 10.0.1.11
- Private DNS Name: ip-10-0-1-11.us-west-2.compute.internal
- Elastic IP Address: -
- Source/Dest. Check: false
- Description: -
- Security Groups: Fortinet FortiGate-VM -BYOL--v5-6-2-AutogenByAWSMP-2

The interface is attached to the instance 'Private-FortiGate-VM'.

8. Select the network interface, select the *Actions* dropdown list, select *Change Source/Dest. Check*. Select *Disabled*. Select *Save*.

The screenshot shows the AWS Management Console interface for a network interface. At the top, there are buttons for 'Create Network Interface', 'Attach', 'Detach', 'Delete', and 'Actions'. A search bar contains 'eni-960e81b7'. Below the search bar is a table with columns: Name, Network interface, Subnet ID, VPC ID, Zone, and Security groups. The table lists one interface: 'eni-960e81b7' associated with 'vpc-76581310' in 'us-west-2a' zone, attached to 'Fortinet FortiGate-VM'. An 'Actions' dropdown menu is open, showing options like 'Attach', 'Detach', 'Delete', 'Manage IP Addresses', 'Associate Address', 'Disassociate Address', 'Change Termination Behavior', 'Change Security Groups', 'Change Source/Dest. Check' (highlighted), 'Add/Edit Tags', 'Change Description', and 'Create Flow Log'. Below the table, the 'Details' tab is selected for the network interface 'eni-960e81b7'. It shows fields for Network interface ID, VPC ID, MAC address, Security groups, Status, Private DNS (IPv4), Subnet ID, Availability Zone, Description, Owner ID, Primary private IPv4 IP, and IPv4 Public IP. At the bottom, the 'Change Source/Dest. Check' dialog is open, showing the network interface 'eni-960e81b7' and the 'Source/dest. check' option set to 'Disabled' (radio button selected). There are 'Cancel' and 'Save' buttons at the bottom of the dialog.

**Change Source/Dest. Check** X

Network Interface eni-960e81b7

Source/dest. check ☐ Enabled ☒ Disabled

Cancel Save

If you have multiple network interfaces, you must disable **Source/Dest. Check** in each interface. You can confirm by looking at the interface information shown as *false*.



| Network Interface eth3 |   |
|------------------------|---|
| Interface ID           | eni-765ebc72                                      |
| VPC ID                 | vpc-e1e4b587                                      |
| Attachment Owner       | 123073262904                                      |
| Attachment Status      | attached  |
| Attachment Time        | Wed Dec 20 11:53:39 GMT-800 2017                  |
| Delete on Terminate    | false   |
| Private IP Address     | 10.0.4.211  |
| Private DNS Name       | ip-10-0-4-211.us-west-2.compute.internal          |
| Elastic IP Address     | -   |
| Source/Dest. Check     | false   |
| Description            | -   |
| Security Groups        | Fortinet FortiGate-VM -BYOL-5-4-6-AutogenByAWSMP- |

## Connecting to the FortiGate-VM

To connect to the FortiGate-VM, you need your login credentials and its public DNS address.

The default username is admin and the default password is the instance ID.

### To connect to the FortiGate-VM:

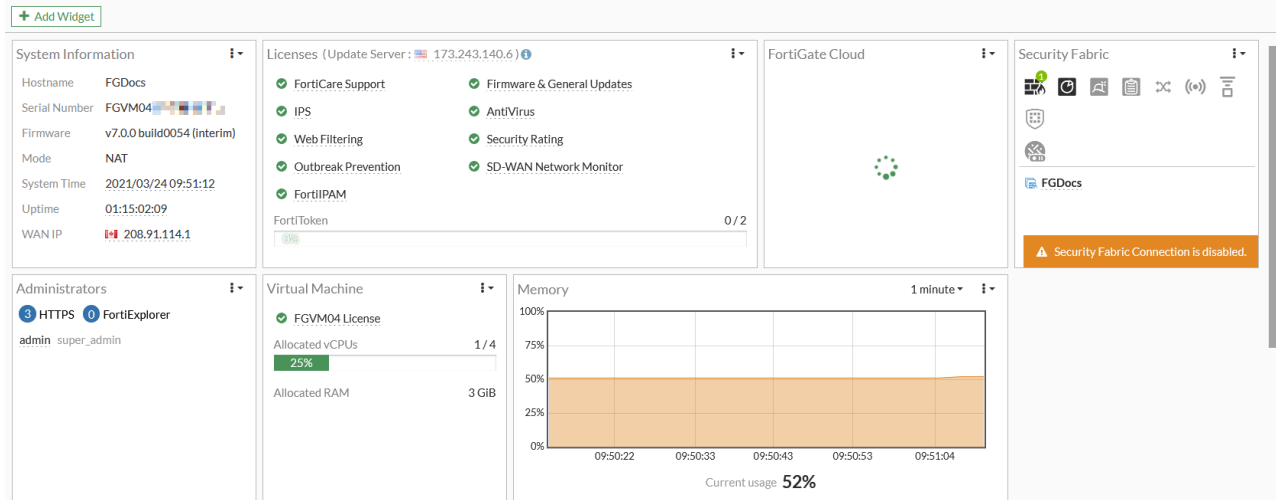
1. You can find the public DNS address in the EC2 management console. Select *Instances* and look at the *Public DNS (IPv4)* field in the lower pane. If you do not see the DNS address, you may need to enable DNS host assignment on your VPC. In this case, go back to the VPC management console, select *Your VPCs*, and select your VPC. From the *Action* dropdown list, and select *Edit DNS Hostnames*. Select *Yes*. Select *Save*.

The screenshot shows the AWS Management Console interface for an EC2 instance named 'Primary FortiGate 1'. The instance is in the 'running' state, located in the 'us-west-2a' availability zone. The public DNS (IPv4) address is highlighted in a red box: `ec2-34-215-95-19.us-west-2.compute.amazonaws.com`. Other details visible include the instance type 'm4.large', the private IP address '10.0.0.220', and the private DNS name 'ip-10-0-0-220.us-west-2.compute.internal'.

2. Open an HTTPS session using the public DNS address of the FortiGate-VM in your browser (`https://<public DNS>`). You see a certificate error message from your browser, which is normal because the default FortiGate certificate is

self-signed and browsers do not recognize it. Proceed past this error. At a later time, you can upload a publicly-signed certificate to avoid this error. Log in to the FortiGate-VM with your username and password (the login credentials mentioned above).

- If you are using a BYOL license, upload your license (.lic) file to activate the FortiGate-VM. The FortiGate-VM automatically restarts. After it restarts, log in again. You now see the FortiGate-VM dashboard. Depending on your license type, the information in the license widget on the dashboard may vary.



- Select **Network > Interfaces**, and edit the interfaces, if required. If the IP address or subnet mask is missing for port 1 or port 2, configure these values.

The 'Edit Interface' window for 'port2' shows the following configuration:

- Name:** port2
- Alias:** (empty)
- Type:** Physical Interface
- VRF ID:** 0
- Role:** Undefined
- Addressing mode:** Manual (selected), DHCP, Auto-managed by FortiIPAM
- IP/Netmask:** 10.0.15/24
- IPv6 addressing mode:** Manual (selected), DHCP, Delegated
- IPv6 Address/Prefix:** ::/0
- Auto configure IPv6 address:** (disabled)
- DHCPv6 prefix delegation:** (disabled)
- Secondary IP address:** (disabled)
- Administrative Access:**
  - IPv4:**
    - HTTPS (checked), HTTP (checked), PING (checked)
    - FMG-Access (unchecked), SSH (checked), SNMP (unchecked)
    - FTM (unchecked), RADIUS Accounting (unchecked)
  - IPv6:**
    - HTTPS (unchecked), PING (unchecked), Security Fabric Connection (unchecked)
    - SSH (unchecked), SNMP (unchecked), FMG-Access (unchecked)
    - Security Fabric Connection (unchecked)
- Receive LLDP:** Use VDOM Setting, Enable, Disable
- Transmit LLDP:** Use VDOM Setting, Enable, Disable
- DHCP Server:** (disabled)
- Stateless Address Auto-configuration (SLAAC):** (disabled)
- DHCPv6 Server:** (disabled)

Buttons: OK, Cancel

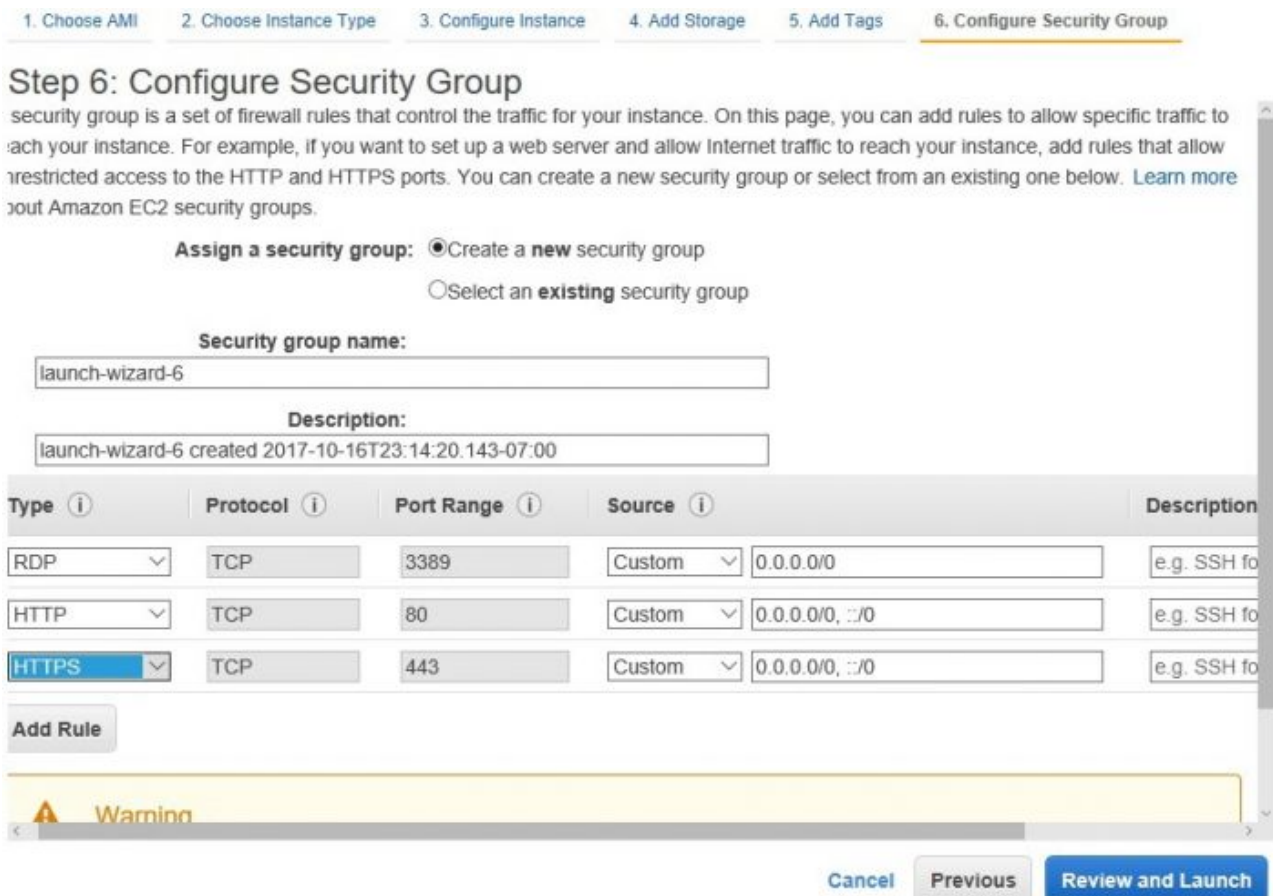
## Setting up a Windows Server in the protected network

To set up a Windows Server in the protected network:

1. In the AWS management console, select EC2. Select *Launch Instance*, then select the *Microsoft Windows Server 2012 R2* that applies to your environment. You use this to test connectivity with remote desktop access.



2. In the *Configure Instance Details* step, in the *Network* field, select the FortiGate-VM's VPC. In the *Subnet* field, select the private subnet.
3. In the *Configure Security Group* step, configure a security group for the Windows server so that it allows Internet access. In this example, we use Remote Desktop TCP port 3389, and other ports are optional. Select *Review and Launch*.



4. Select a key pair, select the acknowledgment checkbox, and select *Launch Instances*.

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Choose an existing key pair

Select a key pair

fctems-keypair

☒ I acknowledge that I have access to the selected private key file (fctems-keypair.pem), and that without this file, I won't be able to log into my instance.

Cancel

Launch Instances

FortiOS 7.4 AWS Administration Guide  
Fortinet Inc.

112

# HA for FortiGate-VM on AWS

## Deploying FortiGate-VM A-P HA on AWS within one zone

This guide provides sample configuration of active-passive (A-P) FortiGate-VM high availability (HA) on AWS within one zone.

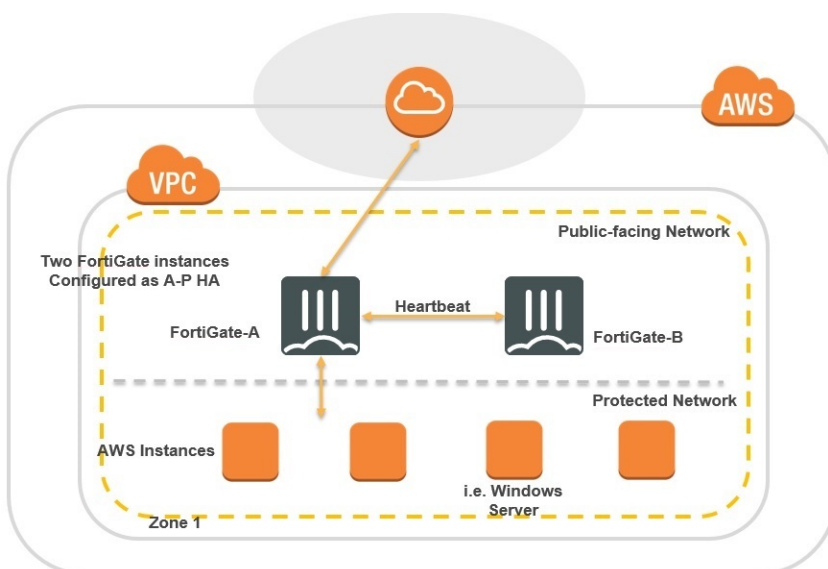
You can configure FortiGate's native HA feature without using an AWS supplementary mechanism with two FortiGate instances: one acting as the primary node and the other as the secondary node, located in two availability zones (AZs) within a single VPC.

This is called "unicast HA" specific to the AWS environment in comparison to an equivalent feature that physical FortiGate units provide. The FortiGates run heartbeats between dedicated ports and synchronize OS configurations. When the primary node fails, the secondary node takes over as the primary node so endpoints continue to communicate with external resources over the FortiGate.

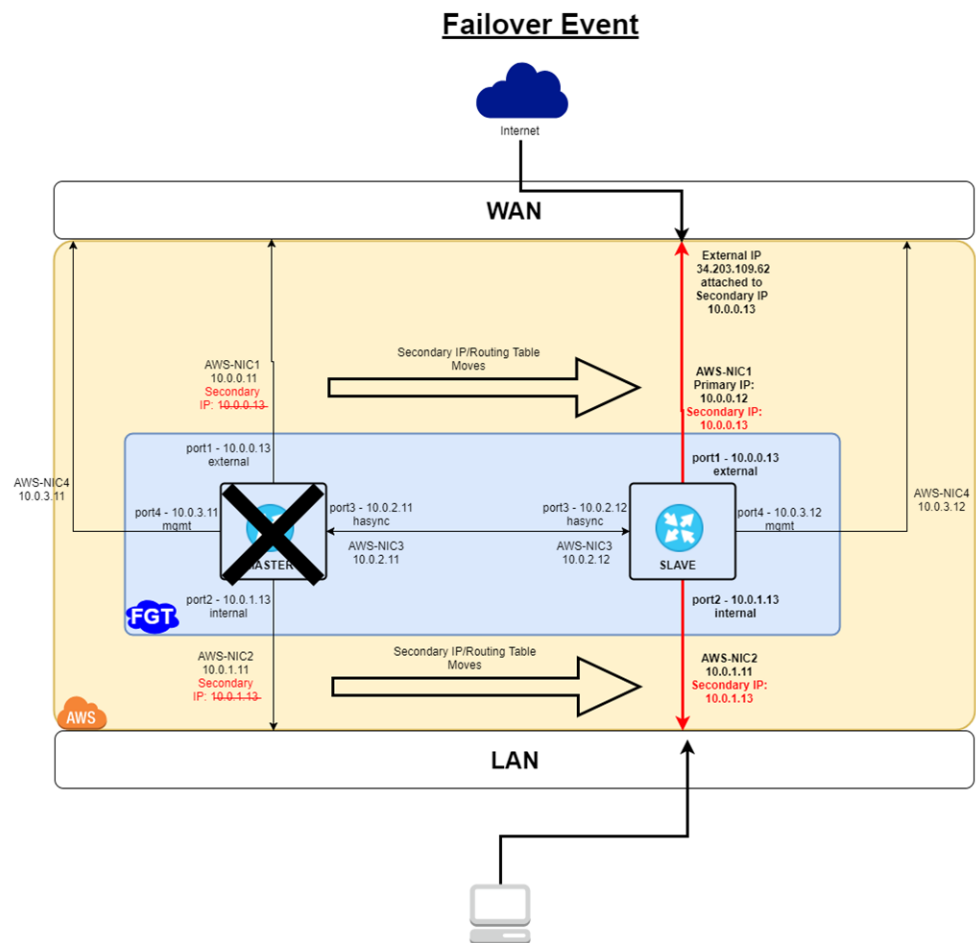
These paired FortiGate instances act as a single logical instance and share interface IP addressing. The main benefits of this solution are:

- Fast failover of FortiOS and AWS SDN without external automation/services
- Automatic AWS SDN updates to elastic IP addresses (EIP) and route targets
- Native FortiOS configuration synchronization
- Ease of use as the cluster is treated as single logical FortiGate

The following depicts the network topology for this sample deployment:



The following depicts a failover event for this sample deployment:



The following lists the IP address assignments for this sample deployment for FortiGate A:

| Port  | AWS primary address | AWS secondary address |
|-------|---------------------|-----------------------|
| port1 | 10.0.0.11           | 10.0.0.13             |
| port2 | 10.0.1.11           | 10.0.1.13             |
| port3 | 10.0.2.11           | N/A                   |
| port4 | 10.0.3.11           | N/A                   |

The following lists the IP address assignments for this sample deployment for FortiGate B:

| Port  | AWS primary address |
|-------|---------------------|
| port1 | 10.0.0.12           |
| port2 | 10.0.1.12           |
| port3 | 10.0.2.12           |
| port4 | 10.0.3.12           |

**To check the prerequisites:**

- Ensure that two FortiGates exist in the same VPC and AZ. The two FortiGates must also have the same build of FortiOS (FGT\_VM64\_AWS or FGT\_VM64\_AWSONDEMAND) installed.
- If using FGT\_VM64\_AWS, ensure that both FortiGates have valid licenses.

**To configure FortiGate-VM HA in AWS:**

1. In the AWS management console, create a VPC. The VPC in this example has been created with 10.0.0.0/16 CIDR.
2. Create four subnets. In this example, the four subnets are as follows:
  - a. Public WAN: 10.0.0.0/24
  - b. Internal network: 10.0.1.0/24
  - c. Heartbeat network: 10.0.2.0/24
  - d. Management network: 10.0.3.0/24

If the FortiGates are to be deployed on the outpost, create the subnets all on outposts.
3. Create a single open security group as shown:

Security Group: sg-0e53c6af6badd964e

Description Inbound Rules **Outbound Rules** Tags

Edit rules

| Type ⓘ      | Protocol ⓘ | Port Range ⓘ | Destination ⓘ |
|-------------|------------|--------------|---------------|
| All traffic | All        | All          | 0.0.0.0/0     |

Security Group: sg-0e53c6af6badd964e

Description **Inbound Rules** Outbound Rules Tags

Edit rules

| Type ⓘ      | Protocol ⓘ | Port Range ⓘ | Source ⓘ  |
|-------------|------------|--------------|-----------|
| All traffic | All        | All          | 0.0.0.0/0 |

4. Create an IAM role. The IAM role is necessary for HA failover. Ensure that the IAM role can read and write EC2 information to read, detach, and reattach network interfaces and edit routing tables.
5. Create five EIPs. Setting up the environment requires five EIPs, but you are left with three IP addresses at the end:
  - One public WAN IP address. This is attached to the instance NIC1's secondary IP address.
  - One FortiGate A management IP address
  - One FortiGate B management IP address
  - Two temporary IP addresses
6. Create two FortiGate instances. You can use any instance type with at least four vCPUs, since the configuration requires four NICs:
  - a. Configure FortiGate A:
    - i. Attach the IAM role created earlier.
    - ii. Create the instance in the VPC created earlier and in the public WAN subnet, with no ephemeral public IP address.
    - iii. Configure an internal IP address of 10.0.0.11, and a secondary IP address of 10.0.0.13.

▼ Network interfaces ⓘ

| Device | Network Interface       | Subnet            | Primary IP | Secondary IP addresses           | IPv6 IPs               |
|--------|-------------------------|-------------------|------------|----------------------------------|------------------------|
| eth0   | New network interface ▼ | subnet-0c6b210e ▼ | 10.0.0.11  | 10.0.0.13 <a href="#">Add IP</a> | <a href="#">Add IP</a> |

- iv. Attach a security group.

- b. Configure FortiGate B by repeating the steps for FortiGate A above. For FortiGate B, configure an internal IP address of 10.0.0.12, and no internal IP address.
- c. Attach three NICs to each FortiGate according to the IP assignment in the appropriate subnet:
  - FortiGate A:
    - port2 (AWS primary 10.0.1.11/AWS secondary 10.0.1.13) (internal network)
    - port3 (AWS primary 10.0.2.11) (Heartbeat network)
    - port4 (AWS primary 10.0.3.11) (management network)
  - FortiGate B:
    - port2 (AWS primary 10.0.1.12) (internal network)
    - port3 (AWS primary 10.0.2.12) (Heartbeat network)
    - port4 (AWS primary 10.0.3.12) (management network)
7. Attach the two temporary EIPs to the port1 primary IP addresses of FortiGate A and FortiGate B. This allows access to the FortiGates via SSH for configuration purposes. The default password for the FortiGates is their instance IDs. The following shows the temporary EIP assigned to FortiGate A:

▼ eth0: eni-016c35b5d998a3995 - port1 - 10.0.0.0/24

#### IPv4 Addresses

| Private IP                    | Public IP                            |
|-------------------------------|--------------------------------------|
| 10.0.0.11                     | 18.233.110.8                         |
| 10.0.0.13                     | 3.81.255.75 <a href="#">Unassign</a> |
| <a href="#">Assign new IP</a> |                                      |

The following shows the temporary EIP assigned to FortiGate B:

▼ eth0: eni-0d318aeb7cdf72fb - port1 - 10.0.0.0/24

#### IPv4 Addresses

| Private IP                    | Public IP   |
|-------------------------------|-------------|
| 10.0.0.12                     | 3.88.75.174 |
| <a href="#">Assign new IP</a> |             |

## To configure FortiGate A using the CLI:

Run the following commands in the FortiOS CLI on FortiGate A:

```
config system global
    set hostname master
end
config system interface
    edit port1
        set mode static
        set ip 10.0.0.13 255.255.255.0
        set allowaccess https ping ssh fgfm
        set alias external
    next
    edit port2
        set mode static
        set ip 10.0.1.13 255.255.255.0
        set allowaccess https ping ssh fgfm
        set alias internal
    next
    edit port3
        set mode static
        set ip 10.0.2.11 255.255.255.0
        set allowaccess https ping ssh fgfm
```



```
        set alias hasync
    next
    edit port4
        set mode static
        set ip 10.0.3.11 255.255.255.0
        set allowaccess https ping ssh fgfm
        set alias hamgmt
    next
    end
config router static
    edit 1
        set device port1
        set gateway 10.0.0.1
    next
    end
config system dns
    set primary 8.8.8.8
end
config firewall policy
    edit 0
        set name "outgoing"
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic disable
        set nat enable
    next
end
config system ha
    set group-name "test"
    set mode a-p
    set hbdev "port3" 50
    set session-pickup enable
    set ha-mgmt-status enable
    config ha-mgmt-interfaces
        edit 1
            set interface "port4"
            set gateway 10.0.3.1
        next
    end
    set override disable
    set priority 1
    set unicast-hb enable
    set unicast-hb-peerip 10.0.2.12
end
```

### **To configure FortiGate B using the CLI:**

Run the following commands in the FortiOS CLI on FortiGate B:

```
config system global
    set hostname slave
end
```

```
config system interface
  edit port1
    set mode static
    set ip 10.0.0.12 255.255.255.0
    set allowaccess https ping ssh fgfm
    set alias external
  next
  edit port2
    set mode static
    set ip 10.0.1.12 255.255.255.0
    set allowaccess https ping ssh fgfm
    set alias internal
  next
  edit port3
    set mode static
    set ip 10.0.2.12 255.255.255.0
    set allowaccess https ping ssh fgfm
    set alias hasync
  next
  edit port4
    set mode static
    set ip 10.0.3.12 255.255.255.0
    set allowaccess https ping ssh fgfm
    set alias hamgmt
  next
end
config router static
  edit 1
    set device port1
    set gateway 10.0.0.1
  next
end
config system dns
  set primary 8.8.8.8
end
config firewall policy
  edit 0
    set name "outgoing"
    set srcintf "port2"
    set dstintf "port1"
    set srcaddr "all"
    set dstaddr "all"
    set action accept
    set schedule "always"
    set service "ALL"
    set logtraffic disable
    set nat enable
  next
end
config system ha
  set group-name "test"
  set mode a-p
  set hbdev "port3" 50
  set session-pickup enable
  set ha-mgmt-status enable
  config ha-mgmt-interfaces
    edit 1
```

```

        set interface "port4"
        set gateway 10.0.3.1
    next
end
set override disable
set priority 1
set unicast-hb enable
set unicast-hb-peerip 10.0.2.11
end

```

After completing configuration of FortiGate B, remove the two temporary IP addresses. You can connect to the FortiGates via the management ports instead.

### To configure the routing tables in AWS:

You must configure three routing tables.

The following shows the public WAN routing table:

| Destination | Target               | Status | Propagated |
|-------------|----------------------|--------|------------|
| 10.0.0.0/16 | local                | active | No         |
| 0.0.0.0/0   | igw-011fd018b1e7227e | active | No         |

The following shows the internal network routing table. Ensure to point the 0.0.0.0/0 CIDR to FortiGate A's port2 NIC.

| Destination | Target                | Status | Propagated |
|-------------|-----------------------|--------|------------|
| 10.0.0.0/16 | local                 | active | No         |
| 0.0.0.0/0   | eni-0f19c02934d82c086 | active | No         |

The following shows the heartbeat and management networks' routing table:

| Destination | Target                | Status | Propagated |
|-------------|-----------------------|--------|------------|
| 10.0.0.0/16 | local                 | active | No         |
| 0.0.0.0/0   | igw-05af2c26b0e7ff1fa | active | No         |

### To test FortiGate-VM HA:

#### 1. Run `get system ha status` to check that the FortiGates are in sync:

```

master # get system ha stat
HA Health Status: OK
Model: FortiGate-VM64-AWSONDEMAND
Mode: HA A-P
Group: 0
Debug: 0
Cluster Uptime: 0 days 0:42:46
Cluster state change time: 2019-01-15 17:23:02
Master selected using:
  <2019/01/15 17:23:02> FGTAWS000F19C1A0 is selected as the master because it has the
    largest value of uptime.
  <2019/01/15 17:09:47> FGTAWS000F19C1A0 is selected as the master because it's the only
    member in the cluster.
ses_pickup: enable, ses_pickup_delay=disable
override: disable
unicast_hb: peerip=10.0.2.12, myip=10.0.2.11, hasync_port='port3'
Configuration Status:
  FGTAWS000F19C1A0(updated 4 seconds ago): in-sync

```

```

FGTAWS000ECBF4EF(updated 4 seconds ago): in-sync
System Usage stats:
FGTAWS000F19C1A0(updated 4 seconds ago):
  sessions=2, average-cpu-user/nice/system/idle=0%/0%/0%/100%, memory=5%
FGTAWS000ECBF4EF(updated 4 seconds ago):
  sessions=0, average-cpu-user/nice/system/idle=0%/0%/0%/100%, memory=5%
HBDEV stats:
FGTAWS000F19C1A0(updated 4 seconds ago):
  port3: physical/1000full, up, rx-bytes/packets/dropped/errors=3135309/12092/0/0,
  tx=9539178/17438/0/0
FGTAWS000ECBF4EF(updated 4 seconds ago):
  port3: physical/1000full, up, rx-bytes/packets/dropped/errors=9300105/17602/0/0,
  tx=3293016/11828/0/0
Master: master , FGTAWS000F19C1A0, HA cluster index = 0
Slave : slave , FGTAWS000ECBF4EF, HA cluster index = 1
number of vcluster: 1
vcluster 1: work 10.0.2.11
Master: FGTAWS000F19C1A0, HA operating index = 0
Slave : FGTAWS000ECBF4EF, HA operating index = 1

```

## 2. Ensure that failover functions as configured:

### a. Turn on debug mode on FortiGate B:

```

slave # di de en
slave # di de application awsd -1
Debug messages will be on for unlimited time.

```

### b. Shut down the primary FortiGate A. In the event of a successful failover, FortiGate B's CLI shows the following:

```

slave # Become HA master
send_vip_arp: vd root master 1 intf port1 ip 10.0.0.13
send_vip_arp: vd root master 1 intf port2 ip 10.0.1.13
awsd get instance id i-0ecbf4ef4c14ba1bb
awsd get iam role WikiDemoHARole
awsd get region us-east-1
awsd doing ha failover for vdom root
awsd moving secondary ip for port1
awsd moving secip 10.0.0.13 from eni-016c35b5d998a3995 to eni-0d318aeb7cdfe72fb
awsd move secondary ip successfully
awsd associate elastic ip allocation eipalloc-0e5ff7daabd5f46dc to 10.0.0.13 of eni
eni-0d318aeb7cdfe72fb
awsd associate elastic ip successfully
awsd moving secondary ip for port2
awsd moving secip 10.0.1.13 from eni-0f19c02934d82c086 to eni-004d87ffb05329b28
awsd move secondary ip successfully
awsd update route table rtb-0bc0aaaaa8fe56192, replace route of dst 0.0.0.0/0 to eni-
004d87ffb05329b28
awsd update route successfully

```

### c. Verify on AWS that the public and internal networks' secondary IP addresses moved, and that the routing table changes to point to FortiGate B's internal network ENI.

## 3. Initiate an SSH session (or another protocol with similar long keep-alive session characteristics) to an external IP address on Ubuntu or an internal VM used for testing purposes. Retest failover and check that the session continues to function without needing to reconnect, and that the session list on the primary and failed over secondary FortiGates are synced.

## 4. You must configure a VDOM exception to prevent interface synchronization between the two FortiGates. Run the following commands in the FortiOS CLI:

```

config system vdom-exception
edit 1
set object system.interface

```

```
next
edit 2
    set object router.static
next
edit 3
    set object firewall.vip
next
end
```

## Deploying FortiGate-VM active-passive HA AWS between multiple zones

This guide provides sample configuration of active-passive FortiGate-VM high availability (HA) on AWS between multiple zones.

You can configure FortiGate's native HA feature without using an AWS supplementary mechanism with two FortiGate instances: one acting as the master/primary node and the other as the slave/secondary node, located in two different availability zones (AZs) within a single VPC.

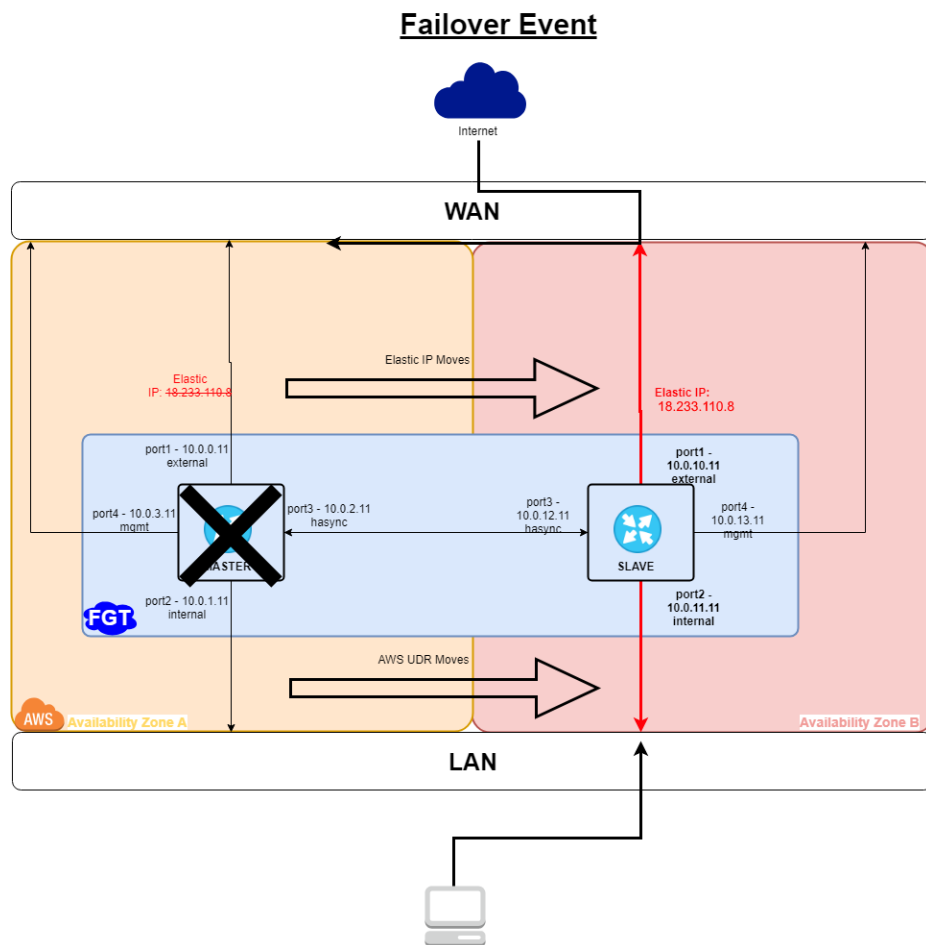
This is called "unicast HA" specific to the AWS environment in comparison to an equivalent feature that physical FortiGate units provide. The FortiGates run heartbeats between dedicated ports and synchronize OS configurations. When the primary node fails, the secondary node takes over as the primary node so endpoints continue to communicate with external resources over the FortiGate.

This feature is important because it solves a critical issue of HA, which is the ability to recover in the event of a catastrophic failure. In the case that both FortiGates are located in the same Availability Zone and that AZ happens to fail, then both FortiGates would go down and HA would be useless. Thus, there is a need to support HA configuration where both FortiGates are in separate AZs.

These paired FortiGate instances act as a single logical instance and share interface IP addressing. The main benefits of this solution are:

- Fast failover of FortiOS and AWS SDN without external automation/services
- Automatic AWS SDN updates to elastic IP addresses (EIP) and route targets
- Native FortiOS configuration synchronization
- Ease of use as the cluster is treated as single logical FortiGate

The following depicts the network topology for this sample deployment:



The following lists the IP address assignments for this sample deployment for FortiGate A:

| Port  | AWS primary address | Subnet          |
|-------|---------------------|-----------------|
| port1 | 10.0.0.11           | 10.0.0.0/24 EIP |
| port2 | 10.0.1.11           | 10.0.1.0/24     |
| port3 | 10.0.2.11           | 10.0.2.0/24     |
| port4 | 10.0.3.11           | 10.0.3.0/24 EIP |

The following lists the IP address assignments for this sample deployment for FortiGate B:

| Port  | AWS primary address | Subnet           |
|-------|---------------------|------------------|
| port1 | 10.0.10.11          | 10.0.10.0/24 EIP |
| port2 | 10.0.11.11          | 10.0.11.0/24     |
| port3 | 10.0.12.11          | 10.0.12.0/24     |
| port4 | 10.0.13.11          | 10.0.13.0/24 EIP |



IPsec VPN phase 1 configuration does not synchronize between primary and secondary FortiGates across AZs. Phase 2 configuration does synchronize.

### To check the prerequisites:

- Ensure that two FortiGates exist in the same VPC but different AZs. The two FortiGates must also have the same FortiOS build (FGT\_VM64\_AWS or FGT\_VM64\_AWSONDEMAND) installed.
- If using FGT\_VM64\_AWS, ensure that both FortiGates have valid licenses.
- The following summarizes minimum sufficient IAM roles for this deployment:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:Describe*",
        "ec2:AssociateAddress",
        "ec2:AssignPrivateIpAddresses",
        "ec2:UnassignPrivateIpAddresses",
        "ec2:ReplaceRoute"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

### To configure FortiGate-VM HA in AWS:

1. In the AWS management console, create a VPC. The VPC in this example has been created with 10.0.0.0/16 CIDR.
2. Create eight subnets. In this example, the eight subnets are as follows:
  - a. Four in AZ A:
    - i. Public WAN: 10.0.0.0/24
    - ii. Internal network: 10.0.1.0/24
    - iii. Heartbeat network: 10.0.2.0/24
    - iv. Management network: 10.0.3.0/24
  - b. Four in AZ B:
    - i. Public WAN: 10.0.10.0/24
    - ii. Internal network: 10.0.11.0/24
    - iii. Heartbeat network: 10.0.12.0/24
    - iv. Management: 10.0.13.0/24
3. Create a single, open security group as shown:

Security Group: sg-0e53c6af6badd964e

| Description | Inbound Rules | Outbound Rules | Tags      |
|-------------|---------------|----------------|-----------|
| Edit rules  |               |                |           |
| Type ⓘ      | Protocol ⓘ    | Port Range ⓘ   | Source ⓘ  |
| All traffic | All           | All            | 0.0.0.0/0 |

4. Create an IAM role. The IAM role is necessary for HA failover. Ensure that the IAM role can read and write EC2 information to read, detach, and reattach network interfaces and edit routing tables.
5. Create three elastic IP addresses:
  - a. One public WAN IP address. This will be attached to the instance NIC1's secondary IP address.
  - b. One FortiGate A management IP address
  - c. One FortiGate B management IP address
6. Create two FortiGate instances. You can use any instance type with at least four vCPUs, since the configuration requires four NICs:
  - a. Configure FortiGate A:
    - i. Attach the IAM role created earlier.
    - ii. Create the instance in the VPC created earlier and in the public WAN subnet, with no ephemeral public IP address.
    - iii. Configure an internal IP address of 10.0.0.11.



- iv. Attach a security group.
- b. Configure FortiGate B by repeating the steps for FortiGate A above. For FortiGate B, configure the instance in the public WAN subnet in AZ B, and configure an internal IP address of 10.0.10.11.
- c. Attach three NICs to each FortiGate according to the IP assignment in the appropriate subnet:
  - i. FortiGate A:
    - i. port2 (AWS primary 10.0.1.11) (internal network)
    - ii. port3 (AWS primary 10.0.2.11) (Heartbeat network)
    - iii. port4 (AWS primary 10.0.3.11) (management network)
  - ii. FortiGate B:
    - i. port2 (AWS primary 10.0.11.11) (internal network)
    - ii. port3 (AWS primary 10.0.12.11) (Heartbeat network)
    - iii. port4 (AWS primary 10.0.13.11) (management network)
7. Attach the two elastic IP addresses to the port1 primary IP addresses of FortiGate A and FortiGate B. This allows access to the FortiGates via SSH for configuration purposes. The default password for the FortiGates is their instance IDs. The following shows the elastic IP address assigned to FortiGate A:

▼ eth0: eni-02888b42018697ca2 - Primary network interface - 10.0.0.0/24

IPv4 Addresses

| Private IP | Public IP    |
|------------|--------------|
| 10.0.0.11  | 18.233.110.8 |

[Assign new IP](#)

The following shows the elastic IP address assigned to FortiGate B:

▼ eth0: eni-0ab045a4d6dce664a - Primary network interface - 10.0.10.0/24

IPv4 Addresses

| Private IP | Public IP   |
|------------|-------------|
| 10.0.10.11 | 3.88.75.174 |

[Assign new IP](#)

## To configure FortiGate A using the CLI:

Run the following commands in the FortiOS CLI on FortiGate A:



```
config sys glo
  set hostname master
end
config system interface
  edit port1
    set mode static
    set ip 10.0.0.11 255.255.255.0
    set allowaccess https ping ssh fgfm
    set alias external
  next
  edit port2
    set mode static
    set ip 10.0.1.11 255.255.255.0
    set allowaccess https ping ssh fgfm
    set alias internal
  next
  edit port3
    set mode static
    set ip 10.0.2.11 255.255.255.0
    set allowaccess https ping ssh fgfm
    set alias hasync
  next
  edit port4
    set mode static
    set ip 10.0.3.11 255.255.255.0
    set allowaccess https ping ssh fgfm
    set alias hamgmt
  next
end
config router static
  edit 1
    set device port1
    set gateway 10.0.0.1
  next
  edit 2
    set device port2
    set gateway 10.0.1.1
    set dst 10.0.11.0/24
  next
end
config firewall policy
  edit 0
    set name "outgoing"
    set srcintf "port2"
    set dstintf "port1"
    set srcaddr "all"
    set dstaddr "all"
    set action accept
    set schedule "always"
    set service "ALL"
    set logtraffic disable
    set nat enable
  next
end
config system ha
  set group-name "test"
  set mode a-p
```

```
set hbdev "port3" 50
set session-pickup enable
set ha-mgmt-status enable
config ha-mgmt-interfaces
  edit 1
    set interface "port4"
    set gateway 10.0.3.1
  next
end
set override disable
set priority 255
set unicast-hb enable
set unicast-hb-peerip 10.0.12.11
end
```

### **To configure FortiGate B using the CLI:**

Run the following commands in the FortiOS CLI on FortiGate B:

```
config sys glo
  set hostname slave
end
config system interface
  edit port1
    set mode static
    set ip 10.0.10.11 255.255.255.0
    set allowaccess https ping ssh fgfm
    set alias external
  next
  edit port2
    set mode static
    set ip 10.0.11.11 255.255.255.0
    set allowaccess https ping ssh fgfm
    set alias internal
  next
  edit port3
    set mode static
    set ip 10.0.12.11 255.255.255.0
    set allowaccess https ping ssh fgfm
    set alias hasync
  next
  edit port4
    set mode static
    set ip 10.0.13.11 255.255.255.0
    set allowaccess https ping ssh fgfm
    set alias hamgmt
  next
end
config router static
  edit 1
    set device port1
    set gateway 10.0.10.1
  next
  edit 2
    set device port2
    set gateway 10.0.11.1
    set dst 10.0.1.0/24
```

```

    next
end
config firewall policy
    edit 0
        set name "outgoing"
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic disable
        set nat enable
    next
end
config system ha
    set group-name "test"
    set mode a-p
    set hbdev "port3" 50
    set session-pickup enable
    set ha-mgmt-status enable
    config ha-mgmt-interfaces
        edit 1
            set interface "port4"
            set gateway 10.0.13.1
        next
    end
    set override disable
    set priority 1
    set unicast-hb enable
    set unicast-hb-peerip 10.0.2.11
end

```

After completing configuration of FortiGate B, remove the EIP to the FortiGate B public IP address. You can connect to the FortiGates via the management ports instead.

## To configure the routing tables in AWS:

You must configure three routing tables.

The following shows the public WAN routing table. Ensure to point the 0.0.0.0/0 CIDR to the Internet gateway:

| Destination | Target                | Status | Propagated |
|-------------|-----------------------|--------|------------|
| 10.0.0.0/16 | local                 | active | No         |
| 0.0.0.0/0   | igw-05af2c26b0e7ff1fa | active | No         |

The following shows the internal network routing table. Ensure to point the 0.0.0.0/0 CIDR to FortiGate A's port2 NIC.

| Destination | Target                | Status | Propagated |
|-------------|-----------------------|--------|------------|
| 10.0.0.0/16 | local                 | active | No         |
| 0.0.0.0/0   | eni-0c4c085477aaff8c5 | active | No         |

The following shows the Heartbeat and management networks' routing table:

| Destination | Target                | Status | Propagated |
|-------------|-----------------------|--------|------------|
| 10.0.0.0/16 | local                 | active | No         |
| 0.0.0.0/0   | igw-05af2c26b0e7ff1fa | active | No         |

**To configure a VDOM exception:**

You must configure a VDOM exception to prevent interface synchronization between the two FortiGates.

```
config system vdom-exception
  edit 1
    set object system.interface
  next
  edit 2
    set object router.static
  next
  edit 3
    set object firewall.vip
  next
end
```

**To test FortiGate-VM HA:****1. Run `get system ha status` to check that the FortiGates are in sync:**

```
master # get sys ha stat
HA Health Status: OK
Model: FortiGate-VM64-AWSONDEMAND
Mode: HA A-P
Group: 0
Debug: 0
Cluster Uptime: 3 days 1:50:18
Cluster state change time: 2019-01-31 18:20:47
Master selected using:
  <2019/01/31 18:20:47> FGTAWS0006AB1961 is selected as the master because it has the
    largest value of override priority.
  <2019/01/31 18:20:47> FGTAWS0006AB1961 is selected as the master because it's the only
    member in the cluster.
ses_pickup: enable, ses_pickup_delay=disable
override: disable
unicast_hb: peerip=10.0.12.11, myip=10.0.2.11, hasync_port='port3'
Configuration Status:
  FGTAWS0006AB1961(updated 3 seconds ago): in-sync
  FGTAWS000B29804F(updated 4 seconds ago): in-sync
System Usage stats:
  FGTAWS0006AB1961(updated 3 seconds ago):
    sessions=18, average-cpu-user/nice/system/idle=0%/0%/0%/100%, memory=10%
  FGTAWS000B29804F(updated 4 seconds ago):
    sessions=2, average-cpu-user/nice/system/idle=0%/0%/0%/100%, memory=10%
HBDEV stats:
  FGTAWS0006AB1961(updated 3 seconds ago):
    port3: physical/00, up, rx-bytes/packets/dropped/errors=430368/1319/0/0,
    tx=560457/1280/0/0
  FGTAWS000B29804F(updated 4 seconds ago):
    port3: physical/00, up, rx-bytes/packets/dropped/errors=870505/2061/0/0,
    tx=731630/2171/0/0
Master: master , FGTAWS0006AB1961, HA cluster index = 1
Slave : slave , FGTAWS000B29804F, HA cluster index = 0
number of vcluster: 1
vcluster 1: work 10.0.2.11
Master: FGTAWS0006AB1961, HA operating index = 0
Slave : FGTAWS000B29804F, HA operating index = 1
```

**2. Ensure that failover functions as configured:****a. Turn on debug mode on FortiGate B:**

```
slave # di de en
slave # di de application awsd -1
Debug messages will be on for unlimited time.
```

**b. Shut down the primary FortiGate A. In the event of a successful failover, FortiGate B's CLI shows the following:**

```
slave # Become HA master
send_vip_arp: vd root master 1 intf port1 ip 10.0.10.11
send_vip_arp: vd root master 1 intf port2 ip 10.0.11.11
awsd get instance id i-0b29804fd38976af4
awsd get iam role WikiDemoHARole
awsd get region us-east-1
awsd get vpc id vpc-0ade7ea6e64befbfc
awsd doing ha failover for vdom root
awsd associate elastic ip for port1
awsd associate elastic ip allocation eipalloc-06b849dbb0f76555f to 10.0.10.11 of eni-
eni-0ab045a4d6dce664a
awsd associate elastic ip successfully
awsd update route table rtb-0a7b4fec57feb1a21, replace route of dst 0.0.0.0/0 to eni-
0c4c085477aaff8c5
awsd update route successfully
```

**c. Verify on AWS that the public and internal networks' secondary IP addresses moved to the new primary FortiGate, and that the routing table changes to point to the secondary FortiGate's internal network ENI.**

## Deploying FortiGate-VM active-passive HA AWS between multiple zones manually with Transit Gateway integration

This guide provides sample configuration of a manual build of an AWS Transit Gateway (TGW) with two virtual private cloud (VPC) spokes and a security VPC. The security VPC contains two FortiGate-VMs to inspect inbound and outbound traffic.

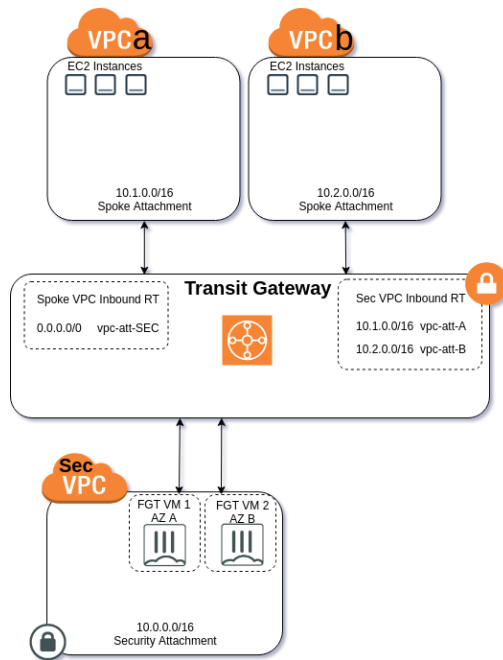
Before deploying FortiGate high availability (HA) for AWS with TGW integration, familiarity with the following AWS services is recommended:

- [Transit Gateway](#)
- [Elastic Cloud Compute \(EC2\)](#)
- [VPC](#)

If you are new to AWS, see [Getting Started with AWS](#).

This deployment consists of the following steps:

1. [Creating VPCs and subnets on page 130](#)
2. [Creating a Transit Gateway and related resources on page 131](#)
3. [Creating an Internet gateway on page 136](#)
4. [Creating VPC route tables on page 137](#)
5. [Deploying FortiGate-VM from AWS marketplace on page 138](#)
6. [Adding network interfaces and elastic IP addresses to the FortiGate-VMs on page 139](#)
7. [Configuring the FortiGate-VMs on page 141](#)
8. [Updating the route table and adding an IAM policy on page 142](#)
9. [Testing FortiGate-VM HA failover on page 143](#)



## Creating VPCs and subnets

Each VPC requires private subnets:

- Each spoke VPC must each have one private subnet.
- The security VPC hub must have ten subnets: five per availability zone (AZ). Each AZ contains a subnet for the following:
  - Management
  - FortiGate private interface
  - FortiGate public interface
  - FortiGate heartbeat interface
  - Transit Gateway (TGW) attachment

Create the spoke and security subnets in different AZs to demonstrate cross-AZ functionality. The example shows the following:

- Spoke 1 (A) has one subnet in the us-west-2a AZ.
- Spoke 2 (B) has one subnet in the us-west-2b AZ.
- The security hub has four subnets for each AZ in both the us-west-2a and us-west-2b AZs.

### To create VPCs and subnets:

1. In the AWS console, open the VPC service.
2. Select *Your VPCs* and click the *Create VPC* button.
3. In the *Name tag* field, enter the desired name.
4. In the *IPv4 CIDR block* and *IPv6 CIDR block* fields, specify the desired CIDR for the spoke VPC.
5. Click *Create*.

6. Repeat the process to create another spoke VPC and a security VPC.

| Create VPC Actions                                 |         |                       |           |             |
|--|---------|-----------------------|-----------|-------------|
| Filter by tags and attributes or search by keyword |         |                       |           |             |
| <input type="checkbox"/>                           | Name    | VPC ID                | State     | IPv4 CIDR   |
| <input type="checkbox"/>                           | VPC_B   | vpc-02ba18eb9afb20d76 | available | 10.2.0.0/16 |
| <input type="checkbox"/>                           | VPC_A   | vpc-0160b27ddcc2fc372 | available | 10.1.0.0/16 |
| <input type="checkbox"/>                           | Sec_VPC | vpc-0786d461404402bc4 | available | 10.0.0.0/16 |

7. Create subnets:

- In the AWS console, go to the VPC service.
- Select *Subnets*, then click the *Create Subnet* button.
- In the *Name tag* field, enter the desired name.
- In the *VPC* field, enter the VPC ID of the desired spoke or security VPC.
- From the *Availability Zone* dropdown list, select the desired AZ.
- In the *IPv4 CIDR block* field, enter the desired CIDR block. Using default /24-sized subnets is recommended.
- Click *Create*.
- Repeat the process until you have all of the subnets.

After completion of this process, the example has configured the following subnets:

- AZ A subnets in security VPC:
  - Public: 10.0.0.0/24
  - Internal: 10.0.1.0/24
  - Heartbeat: 10.0.2.0/24
  - Management: 10.0.3.0/24
  - TGW-Subnet: 10.0.4.0/24
- AZ B subnets in security VPC:
  - Public: 10.0.10.0/24
  - Internal: 10.0.11.0/24
  - Heartbeat: 10.0.12.0/24
  - Management: 10.0.13.0/24
  - TGW-Subnet: 10.0.14.0/24
- AZ A subnet in spoke 1 VPC: 10.1.1.0/24
- AZ B subnet in spoke 2 VPC: 10.2.1.0/24

## Creating a Transit Gateway and related resources

### To create a Transit Gateway and related resources:

- Create a Transit Gateway (TGW):
  - In the AWS console, open the VPC service.
  - Select *Transit Gateways*, then click the *Create Transit Gateway* button.
  - In the *Name tag* field, enter the desired name.
  - Deselect *Default route table association* and *Default route table propagation* to prevent undesired association into the security route.
  - Edit *Transit gateway CIDR blocks* to enter the same CIDR range as the TGW-Subnet that you configured in

[Creating VPCs and subnets on page 130.](#)

- f.** Configure other fields as desired, then click *Create*.
- g.** Wait for the TGW state to change from *Pending* to *Available* before proceeding.



2. Create two TGW route tables: one for the security VPC and another for the spokes:
  - a. In the AWS console, open the VPC service.
  - b. Select *Transit Gateway Route Tables*, then click the *Create Transit Gateway Route Table* button.
  - c. In the *Name tag* field, enter the desired name.
  - d. From the *Transit Gateway ID* dropdown list, select the Transit Gateway ID.
  - e. Click *Create*.
  - f. Repeat the process for the spoke route table.
3. Create three TGW attachments, one for each VPC:
  - a. In the AWS console, open the VPC service.
  - b. Select *Transit Gateway Attachments*, then click the *Create Transit Gateway Attachment* button.
  - c. From the *Transit Gateway ID* dropdown list, select the Transit Gateway ID.
  - d. In the *Attachment type* field, select *VPC*.
  - e. In the *Attachment name tag* field, enter the desired name.
  - f. In the *VPC ID* field, enter the security VPC ID for the first attachment. This is `TGW_Sec_VPC_Attachment` in the screenshot.

- g. For *Subnet IDs*, select the TGW-Subnet of each availability zone (AZ) for the security VPC.

VPC > Transit gateway attachments > Create transit gateway attachment

## Create transit gateway attachment [Info](#)

A transit gateway (TGW) is a network transit hub that interconnects attachments (VPCs and VPNs) within the same AWS account or across AWS accounts.

### Details

**Name tag - optional**  
Creates a tag with the key set to Name and the value set to the specified string.

SEC-VPC-TGW-SUB

**Transit gateway ID** [Info](#)

tgw-018513d9b014392ca (Internet-TGW)

**Attachment type** [Info](#)

VPC

### VPC attachment

Select and configure your VPC attachment.

☒ DNS support [Info](#)

☐ IPv6 support [Info](#)

**VPC ID**  
Select the VPC to attach to the transit gateway.

vpc-0fd3fa676a53c7463 (SEC\_VPC)

**Subnet IDs** [Info](#)  
Select the subnets in which to create the transit gateway VPC attachment.

|  |                                      |
|--|--------------------------------------|
| <input checked="" type="checkbox"/> us-west-2a | subnet-0d087c230a20d5a9c (SEC-TGW-A) |
| <input checked="" type="checkbox"/> us-west-2b | subnet-0ef1ac2b3c98b2592 (SEC-TGW-B) |
| <input type="checkbox"/> us-west-2c            | No subnet available                  |
| <input type="checkbox"/> us-west-2d            | No subnet available                  |

subnet-0d087c230a20d5a9c ✕    subnet-0ef1ac2b3c98b2592 ✕

- h. Repeat the process for the other two VPC IDs, spokes A and B. For the subnet VPC attachment, select the corresponding AZ for each, then the *Subnet ID* dropdown list shows the spoke subnet that you created.

- i. Wait for the State to become Available.

Create Transit Gateway Attachment Actions ▾

Filter by tags and attributes or search by keyword

| <input type="checkbox"/> | Name ▾                 | Transit Gateway attachment ID ▾ | Transit Gateway ▾ | Resource type ▾ | Resource ID ▾ | State ▾   |
|--------------------------|------------------------|---------------------------------|-------------------|-----------------|---------------|-----------|
| <input type="checkbox"/> | TGW-attach-A           | tgw-attach-0f19afc79167adf6c    | tgw-0a6e104f0...  | VPC             | vpc-0160b2... | available |
| <input type="checkbox"/> | TGW-attach-B           | tgw-attach-079a39bb958f2bfe4    | tgw-0a6e104f0...  | VPC             | vpc-02ba18... | available |
| <input type="checkbox"/> | TGW_Sec_VPC_Attachment | tgw-attach-0bd4ebf7075a4abc3    | tgw-0a6e104f0...  | VPC             | vpc-0786d4... | available |

4. Create TGW associations:

- In the AWS console, open the VPC service.
- Select *Transit Gateway Route Tables*, then select the spoke route table.
- On the *Associations* tab, click the *Create Association* button.
- From the *Choose attachment to associate* dropdown list, select the spoke 1 VPC.
- Click *Create association*.
- Repeat the process for spoke B, which will be the second association for the route table.
- Wait for both associations to achieve the *Associated* state before proceeding.
- Next, select the security route table.
- Repeat the same as above to add the security VPC attachment to the security TGW route table. Click *Create association*.



You should associate the security attachment using the TGW-Subnets to the security route table. The spoke attachments will be associated to the spoke route table.

- Add routes to the security TGW route table:
  - In the AWS console, open the VPC service.
  - Select *Transit Gateway Route Tables*, then select the security route table.
  - Add a static route for each spoke subnet and select spoke VPC attachments.
- Add routes to the spoke TGW route table:
  - In the AWS console, open the VPC service.
  - Select *Transit Gateway Route Tables*, then select the spoke route table.
  - Add a static route for 0.0.0.0/0 to the security VPC attachment.

- d. Add specific null routes: Spoke1(A) subnet, Spoke2(B) Subnet, and SEC-Public Subnets.

VPC > Transit gateway route tables > tgw-rtb-0b9e9052ad033e294

## tgw-rtb-0b9e9052ad033e294 / Spoke-Internal Info

Actions

### Details

|   |                    |                                       |                                       |
|---|--------------------|---------------------------------------|---------------------------------------|
| Transit gateway route table ID<br>tgw-rtb-0b9e9052ad033e294 | State<br>Available | Default association route table<br>No | Default propagation route table<br>No |
| Transit gateway ID<br>tgw-018513d9b014392ca                 |                    |                                       |                                       |

Associations
Propagations
Prefix list references
**Routes**
Tags

▼ Filter routes by CIDR (2)

**Exact CIDR**  
Select a valid IP4 or IPv6 CIDR.

**Longest prefix match**  
Enter a valid IP4 or IPv6 and press enter.

**Supernet of match**  
Select a valid IP4 or IPv6 CIDR.

**Subnet of match**  
Select a valid IP4 or IPv6 CIDR.

**Routes (3)**

1
⚙

|                          | CIDR        | Attachment ID                | Resource ID           | Resource type | Route type |
|--------------------------|-------------|------------------------------|-----------------------|---------------|------------|
| <input type="checkbox"/> | 0.0.0.0/0   | tgw-attach-01c04cf08e993db7e | vpc-0fd3fa676a53c7463 | VPC           | Static     |
| <input type="checkbox"/> | 10.1.1.0/24 | –                            | –                     | –             | Static     |
| <input type="checkbox"/> | 10.2.1.0/24 | –                            | –                     | –             | Static     |

## Creating an Internet gateway

### To create an Internet gateway:

1. In the AWS console, open the VPC service.
2. Click the *Create Internet Gateway* button.
3. In the *Name tag* field, enter the desired name.
4. Click *Create*.
5. Attach the Internet gateway to the security VPC by selecting the Internet gateway and selecting *Attach to VPC* from the *Actions* menu.
6. Select the security VPC in the VPC dropdown list and click the *Attach* button to save.

## Creating VPC route tables

### To create a VPC route table:

1. In the AWS console, open the VPC service.
2. Configure two spoke VPC route tables:
  - a. Select *Route Tables*, then click the *Create route table* button.
  - b. Configure the desired name, then select the spoke A VPC. Click the *Create* button.
  - c. Repeat the process for the spoke B VPC.
  - d. Select the spoke A VPC route table. On the *Routes* tab, click the *Edit routes* button.
  - e. Click *Add Route*.
  - f. In the *Destination* field, specify 0.0.0.0/0.
  - g. For the *Target*, specify the Transit Gateway (TGW). Click *Save Routes*.
  - h. On the *Subnet Associations* tab, click the *Edit subnet associations* button.
  - i. Select the spoke subnet that you just created, then click *Save*.

|                                     |       |                       |                          |   |     |
|-------------------------------------|-------|-----------------------|--------------------------|---|-----|
| <input checked="" type="checkbox"/> | VPC-A | rtb-0447130e8859c50df | subnet-05616fc1a92cce571 | - | Yes |
| <input type="checkbox"/>            | VPC-B | rtb-048232ec121dac120 | subnet-0e7f910e45fc879e8 | - | Yes |

Route Table: rtb-0447130e8859c50df

Summary Routes Subnet Associations Edge Associations Route Propagation Tags

Edit routes

View All routes

| Destination | Target                |
|-------------|-----------------------|
| 10.1.0.0/16 | local                 |
| 0.0.0.0/0   | tgw-0a6e104f04cbd7da4 |

- j. Repeat the process for the spoke B route table.
3. Configure the security VPC internal route table:
  - a. Click the *Create route table* button.
  - b. Configure Sec\_VPC\_Internal as the name. This will be the route for internal traffic targeting the TGW. Select the security VPC.
  - c. Click the *Create* button.
  - d. Select the security VPC internal route table. On the *Routes* tab, click the *Edit routes* button.
  - e. Click *Add Route*.
  - f. In the *Destination* field, enter 0.0.0.0/0. Use the TGW as the target.
  - g. Click *Save changes*.
  - h. On the *Subnet Associations* tab, click the *Edit subnet associations* button.
  - i. Select the internal/private subnets for both VPC availability zones (AZ) A and B, then click the *Save* button.
4. Configure the security VPC external route table:
  - a. Click the *Create route table* button.
  - b. Configure Sec\_VPC\_External as the name. This will be the Internet-facing route table. Select the security VPC.

- c. Click the *Create* button.
- d. Select the security VPC external route table. On the *Routes* tab, click the *Edit routes* button.
- e. Add the following routes:

| Destination | Target           |
|-------------|------------------|
| 0.0.0.0/0   | Internet gateway |
| 10.1.1.0/24 | TGW              |
| 10.2.1.0/24 | TGW              |

- f. On the *Subnet Associations* tab, click the *Edit subnet associations* button.
  - g. Add the management, public, and heartbeat subnets for security VPC AZs, then click the *Save* button.
5. Configure the route table for return traffic to the spoke VPCs from the FortiGate:
- a. Click the *Create route table* button.
  - b. Configure `Sec_VPC_TGW` as the name. Select the security VPC.
  - c. Click the *Create* button.
  - d. On the *Routes* tab, click the *Edit routes* button.
  - e. Add the following routes:

| Destination | Target |
|-------------|--------|
| 10.1.1.0/24 | TGW    |
| 10.2.1.0/24 | TGW    |

- f. On the *Subnet Associations* tab, click the *Edit subnet associations* button.
- g. Select the TGW subnets for both AZs A and B, then click the *Save* button.



You will add a route that targets the ENI ID of port2 of the primary FortiGate in a later step.

## Deploying FortiGate-VM from AWS marketplace

### To deploy the FortiGate-VM from the AWS marketplace:

1. On the AWS marketplace, find a FortiGate-VM listing and version available for selection. This example uses FortiGate-VM On-Demand 6.2.1, ami-0439b030915c59e67, on c5.xlarge instances. Available versions may change.



Deploying a high availability (HA) pair requires four network interfaces. Instances smaller than x.large do not support four network interfaces and do not work for this deployment type.

## 2. Launch the FortiGate-VM through Elastic Compute Cloud.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of

|                               |   |   |
|-------------------------------|---|---|
| Number of instances           | 1   | <a href="#">Launch into Auto Scaling Group</a>  |
| Purchasing option             | <input type="checkbox"/> Request Spot instances   |   |
| Network                       | vpc-0786d461404402bc4   Sec_VPC   | <a href="#">Create new VPC</a>                  |
| Subnet                        | subnet-0ad793f41977651f5   Public-A   us-west-2a<br>249 IP Addresses available                                  | <a href="#">Create new subnet</a>               |
| Auto-assign Public IP         | Enable  |   |
| Placement group               | <input type="checkbox"/> Add instance to placement group  |   |
| Capacity Reservation          | Open  | <a href="#">Create new Capacity Reservation</a> |
| IAM role                      | None  | <a href="#">Create new IAM role</a>             |
| CPU options                   | <input type="checkbox"/> Specify CPU options  |   |
| Shutdown behavior             | Stop  |   |
| Enable termination protection | <input type="checkbox"/> Protect against accidental termination   |   |
| Monitoring                    | <input type="checkbox"/> Enable CloudWatch detailed monitoring<br><a href="#">Additional charges apply.</a>     |   |
| EBS-optimized instance        | <input checked="" type="checkbox"/> Launch as EBS-optimized instance  |   |
| Tenancy                       | Shared - Run a shared hardware instance<br><a href="#">Additional charges will apply for dedicated tenancy.</a> |   |
| Elastic Inference             | <input type="checkbox"/> Add an Elastic Inference accelerator<br><a href="#">Additional charges apply.</a>      |   |

3. Deploy the VM with only one network interface with public IP address assignment enabled.
4. Repeat the steps for the second VM instance in a second availability zone.
5. To enable management access to the FortiGate-VMs and HA traffic flow, open the security group attached to the FortiGate-VMs:
  - a. In the AWS console, select *Security Groups*.
  - b. Click the *Create Security Group* button.
  - c. Add a rule with a source of 0.0.0.0/0 for all traffic types.
  - d. Assign the rule to all interfaces on both FortiGate-VMs. The next step in the process, [Adding network interfaces and elastic IP addresses to the FortiGate-VMs on page 139](#), explains creating additional network interfaces. You can tighten the security group later.

## Adding network interfaces and elastic IP addresses to the FortiGate-VMs

### To add network interfaces and elastic IP addresses to the FortiGate-VMs:

1. Add network interfaces:
  - a. In the AWS console, open the Elastic Compute Cloud (EC2) service.
  - b. Select *Network Interfaces*, then click the *Create Network Interface* button.
  - c. Provide a description of the interface, specify the private subnet in availability zone A and specify the security group created in [Deploying FortiGate-VM from AWS marketplace on page 138](#).
  - d. Click *Yes, Create*.
  - e. Click the newly created interface. From the *Actions* dropdown list, select *Change Source/Dest Check*. Disable *Source/Dest Check* and save.
  - f. From the *Actions* dropdown list, select *Attach*.

- g. From the dropdown list, select the first FortiGate-VM. Click *Attach*.
- h. Repeat the process for the second FortiGate-VM.
2. Repeat step 1 for the secondary FortiGate-VM. Each FortiGate-VM will be attached with four network interfaces:

| Port         | Purpose  |
|--------------|--|
| Port1 (eth0) | Public network IP address. Elastic IP address (EIP) only for primary FortiGate in high availability group. |
| Port2 (eth1) | Private network IP address   |
| Port3 (eth2) | Heartbeat network IP address   |
| Port4 (eth3) | Management network IP address. EIP on each FortiGate.  |

### Create Network Interface

Description  ⓘ

Subnet\*  ⓘ

IPv4 Private IP ☒ Auto-assign ⓘ ☐ Custom

Elastic Fabric Adapter ☐ ⓘ

Security groups\*  ⓘ

ⓘ
 

< 1 to 3 of 3 >

| <input type="checkbox"/>            | Group ID         | Group name         | Description  |
|-------------------------------------|------------------|--------------------|--|
| <input checked="" type="checkbox"/> | sg-085206d44...  | Fortinet FortiG... | This security group was generated by AWS Marketplace and is based on recommended settings for F... |
| <input type="checkbox"/>            | sg-0de336d7b...  | default            | default VPC security group   |
| <input type="checkbox"/>            | sg-0fat293e5a... | launch-wizard-1    | launch-wizard-1 created 2019-10-01T15:58:03.267-07:00  |

3. Add elastic IP addresses (EIPs):
  - a. In the AWS console, open the EC2 service.
  - b. Select *Elastic IPs*, then click the *Allocate new address* button.
  - c. Accept the defaults, then click the *Allocate* button.
  - d. Repeat steps a-c twice for a total of three EIPs:
    - One EIP is for port1 that will move to the secondary FortiGate-VM during failover.
    - Two EIPs are for high availability (HA) management ports.
4. Attach three EIPs as follows:
  - a. Port 1 of the primary FortiGate by selecting *Network Interface* as the *Resource Type* and its eth0 ENI network interface to associate.
  - b. Port 4 of the primary FortiGate by selecting *Network Interface* as the *Resource Type* and its eth3 ENI network interface to associate.
  - c. Port 4 of the secondary FortiGate by selecting *Network Interface* as the *Resource Type* and its eth 3ENI network interface to associate.

The primary FortiGate port 1 EIP will fail over to the secondary FortiGate in case of failure.

Port4 elastic IP addresses are not accessible until you form an HA cluster.



## Configuring the FortiGate-VMs

### To configure the FortiGate-VMs:

- Log in to the primary FortiGate-VM:
  - In the browser, enter `https://` followed by the port1 (eth0) public IP address.
  - Click *Advanced*, then proceed with the warning.
  - Enter admin and the instance ID as the username and password, respectively, for the primary FortiGate-VM, and proceed to change the default password.
- Configure the primary FortiGate-VM:
  - Go to *Network > Interfaces*. Confirm all four port IP address settings.
  - Go to *Network > Static Routes*. Set the static route for port1 and port2 to the corresponding gateway on each FortiGate-VM. Usually the last number is 1 for the same subnet (i.e. 10.0.0.1) on AWS.
  - Ensure that the 10.2.1.0/24 and 10.1.1.0/24 (or your internal subnet CIDR) route has been created to forward internal traffic out of port2.
  - Go to *System > HA*. Configure high availability (HA) settings. After enabling active-passive mode, you can only access the FortiGate-VM through the HA management port (elastic IP address on port4).
    - From the *Mode* dropdown list, select *Active-Passive*.
    - In the *Device priority* field, enter a value that will be higher than the one you configure for the secondary node.
    - Configure the *Group name* and *Password* fields.
    - Enable *Session pickup*.
    - For *Heartbeat interfaces*, select *port3*.
    - Enable *Management Interface Reservation*. From the *Interface* dropdown list, select *port4*. Specify the gateway for the same subnet.
    - Enable *Unicast Heartbeat*. Specify the port3 IP address of the peer FortiGate.
- Log in to and configure the secondary FortiGate-VM by repeating steps 1-2. When configuring device priority in HA settings, set a lower value than that of the primary node.
- Configure policies to forward internal traffic out from port1. You only need to configure such policies on the primary FortiGate-VM, as the policy configuration will synchronize between the FortiGate-VMs.

| ID              | Name     | Source | Destination | Schedule | Service | Action   | NAT        |
|-----------------|----------|--------|-------------|----------|---------|----------|------------|
| port1 → port2 ⓘ |          |        |             |          |         |          |            |
| 2               | Incoming | all    | all         | always   | ALL     | ✓ ACCEPT | ✗ Disabled |
| port2 → port1 ⓘ |          |        |             |          |         |          |            |
| 1               | outgoing | all    | all         | always   | ALL     | ✓ ACCEPT | ✓ Enabled  |

- You must configure a VDOM exception to prevent interface synchronization between the two FortiGates. Run the following commands in the FortiOS CLI:

```
config system vdom-exception
  edit 1
    set object system.interface
  next
  edit 2
    set object router.static
  next
  edit 3
    set object firewall.vip
  next
end
```

6. (Optional) You can configure an AWS SDN connector to allow population of dynamic objects such as policy objects. See [Access key-based SDN connector integration on page 167](#).

## Updating the route table and adding an IAM policy

### To update the route table and add an IAM policy:

1. Update the route table:
  - a. After configuring the internal network ports, you must route all internal traffic to the elastic network interface (ENI) of the primary FortiGate-VM port2. In the AWS console, open the Elastic Cloud Compute service.
  - b. Select *Instances*, then select the primary FortiGate-VM.
  - c. On the *Description* tab, select port2 (eth1) and copy the interface ID.
  - d. Save the content into a text editor.
  - e. In the AWS console, open the VPC service.
  - f. Select *Route Tables*, then select the Sec\_VPC\_TGW route table.
  - g. On the *Routes* tab, click the *Edit Routes* button.
  - h. Add the following route:

| Destination | Target  |
|-------------|---|
| 0.0.0.0/0   | Paste the ENI ID of port2 of the primary FortiGate. |

- i. Click *Save*.
- j. Ensure that the Sec\_VPC\_TGW route table has the following routes:

| Destination | Target   |
|-------------|--|
| 10.1.1.0/24 | Transit Gateway (TGW)                                |
| 10.2.1.0/24 | TGW  |
| 0.0.0.0/0   | ENI ID of port2 of the primary FortiGate.            |
| 10.0.0.0/16 | Local. Depends on the security VPC network settings. |



Check that the TGW subnets (security VPC TGW subnets) for both availability zones A and B are associated with this routing table.

2. Both firewalls need an IAM policy attached to make API calls to AWS to move the elastic IP address on port1 and network interface on port2 between primary and secondary FortiGate-VMs. Go to the IAM service and create a role with the following policy: {

```
"Version": "2012-10-17",
"Statement": [
  {
    "Action": [
      "ec2:Describe*",
      "ec2:AssociateAddress",
      "ec2:AssignPrivateIpAddresses",
      "ec2:UnassignPrivateIpAddresses",
      "ec2:ReplaceRoute"
    ],
    "Resource": "*"
  }
],
```

```

        "Resource": "*",
        "Effect": "Allow"
    }
]
}

```

3. Attach the AMI role to both FortiGate-VMs by selecting the FortiGate EC2 instance and selecting *Attach/Replace IAM Role* in the *Actions* menu.

## Testing FortiGate-VM HA failover

The following prerequisites are required for successful failover:

- Two FortiGates exist in the same virtual private cloud and different availability zones. The two FortiGates must also have the same FortiOS build (FGT\_VM64\_AWS or FGT\_VM64\_AWSONDEMAND) installed and the same instance shape. In this example, both FortiGate-VM instances were deployed as C5.xlarge.
- The high availability (HA) management port can resolve DNS and make API calls to AWS. The HA management port is not blocked by the security group and routed to the Internet gateway on all cluster members.
- If using FortiGate-VM BYOL instances, both FortiGate-VMs have valid licenses.
- Minimum sufficient IAM roles as shown in [Updating the route table and adding an IAM policy on page 142](#)

### To test FortiGate-VM HA failover:

1. To ensure that the FortiGate-VMs are in sync, run `get system ha status`:

```

master # get sys ha stat
HA Health Status: OK
Model: FortiGate-VM64-AWSONDEMAND
Mode: HA A-P
Group: 0
Debug: 0
Cluster Uptime: 1 days 1:50:18
Cluster state change time: 2019-01-31 18:20:47
Master selected using:
<2019/01/31 18:20:47> FGTAWS0006AB1961 is selected as the master because it has the
largest value of override priority.
<2019/01/31 18:20:47> FGTAWS0006AB1961 is selected as the master because it's the only
member in the cluster.
ses_pickup: enable, ses_pickup_delay=disable
override: disable
Master: FGTAWS0006AB1961, HA operating index = 0
Slave : FGTAWS000B29804F, HA operating index = 1

```

2. Enable debug mode on the secondary FortiGate:

```

diagnose debug enable
diagnose debug application awsd -1
Debug messages will be on for unlimited time.

```

3. Shut down the primary FortiGate. In the event of a successful failover, the secondary FortiGate CLI shows the following:

```

slave # Become HA master
send_vip_arp: vd root master 1 intf port1 ip 10.0.10.11
send_vip_arp: vd root master 1 intf port2 ip 10.0.11.11
awsd get instance id i-0b29804fd38976af4
awsd get iam role WikiDemoHARole
awsd get region us-west-2
awsd get vpc id vpc-0ade7ea6e64befbfc

```

```
awsd doing ha failover for vdom root
awsd associate elastic ip for port1
awsd associate elastic ip allocation eipalloc-06b849dbb0f76555f to 10.0.10.11 of eni
eni-0ab045a4d6dce664a
awsd associate elastic ip successfully
awsd update route table rtb-0a7b4fec57feb1a21, replace route of dst 0.0.0.0/0 to eni-
0c4c085477aaff8c5
awsd update route successfully
```

4. Verify on AWS that the public EIP on port1 and the Sec\_VPC\_Internal route table point to the new primary FortiGate port2 ENI.

# Deploying FortiGate-VM using Terraform

See the following:

- [Single FortiGate-VM deployment](#)
- [Active-passive HA cluster deployment in the same availability zone](#)
- [Active-passive HA cluster deployment across two availability zones](#)

See the [FortiGate-VM Terraform GitHub project](#) for more solutions.

## Support

For issues, see this GitHub project's [Issues](#) tab. For other questions related to the GitHub project, contact [github@fortinet.com](mailto:github@fortinet.com).

# Deploying FortiGate-VM on Snowball Edge

The following topics provide information on deploying FortiGate-VM on Snowball Edge. FortiOS 7.4.1 and later versions support this feature.

The Snowball Edge device used for this documentation was: Snowball Edge Compute Optimized with Amazon S3 compatible storage.

## Ordering an Snowball Edge device

See [Creating a Snowball Edge job](#) and [Before you order a Snowball Edge device](#) for information about ordering a Snowball Edge.

## Getting connected to the Snowball Edge device

This document assumes that you have installed and configured AWS OpsHub, Snowballedge client, and AWS CLI for your environment.

For information about downloading AWS OpsHub, see [AWS Snowball resources](#). For resources on how to use AWS OpsHub, see [Using AWS OpsHub for Snow Family to Manage Devices](#).



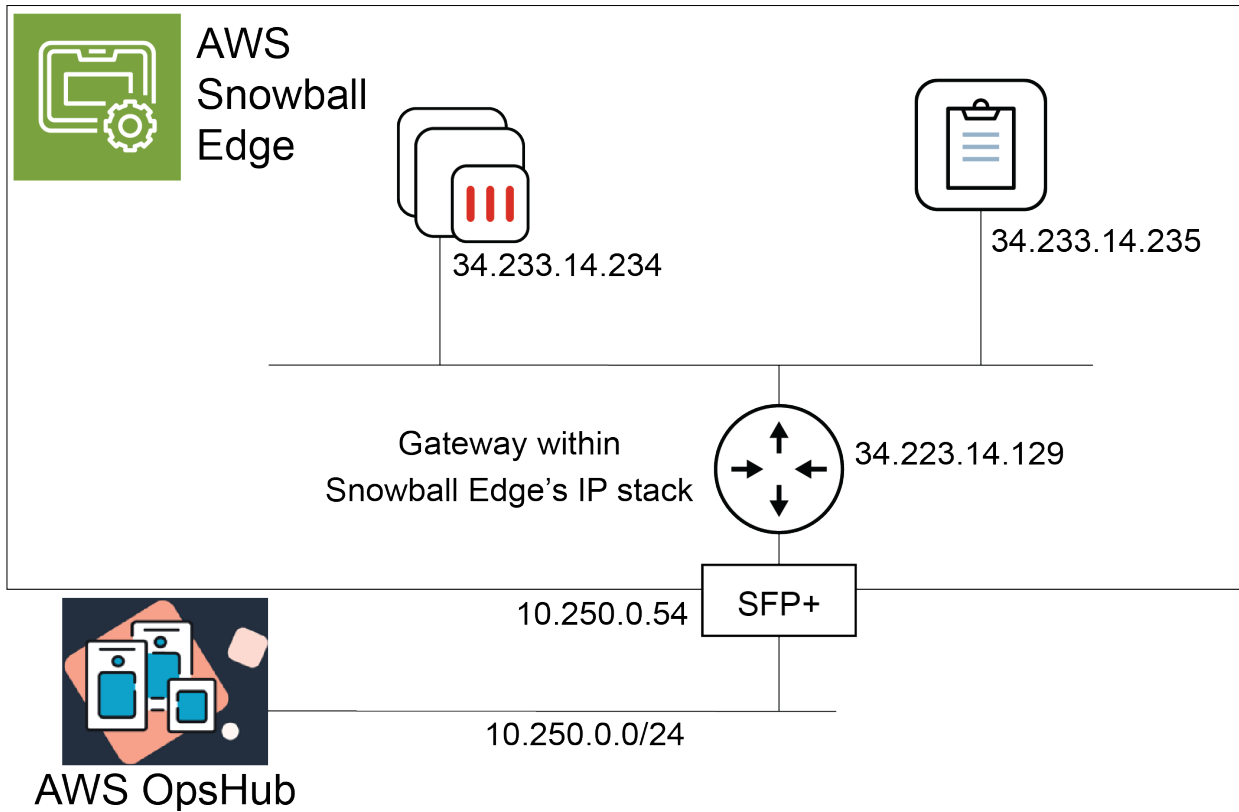
Ordering the Snowball Edge device with a marketplace AMI and FortiGate-VM preloaded is not currently supported.



Initial configuration of the Snowball Edge default gateway is important as the network addresses and certificates are used throughout the Snowball Edge device to access the Snowball Edge service APIs.

---

## Configuring the Snowball Edge networking



The following table provides the virtual network interface (VNI) entries information for the example deployment in the topology:

| EC2 instance    | Inside IP address | Outside IP address |
|-----------------|-------------------|--------------------|
| FortiGate-VM    | 34.233.14.234     | 10.250.0.69        |
| FortiManager-VM | 34.233.14.235     | 10.250.0.59        |

Most VNIs are created when creating the corresponding service and/or an instance that will use the VNI, for example, starting the S3-Snow service and creating a VM instance.

Directly attached network interfaces (DNI) are created after the VM instance is created.



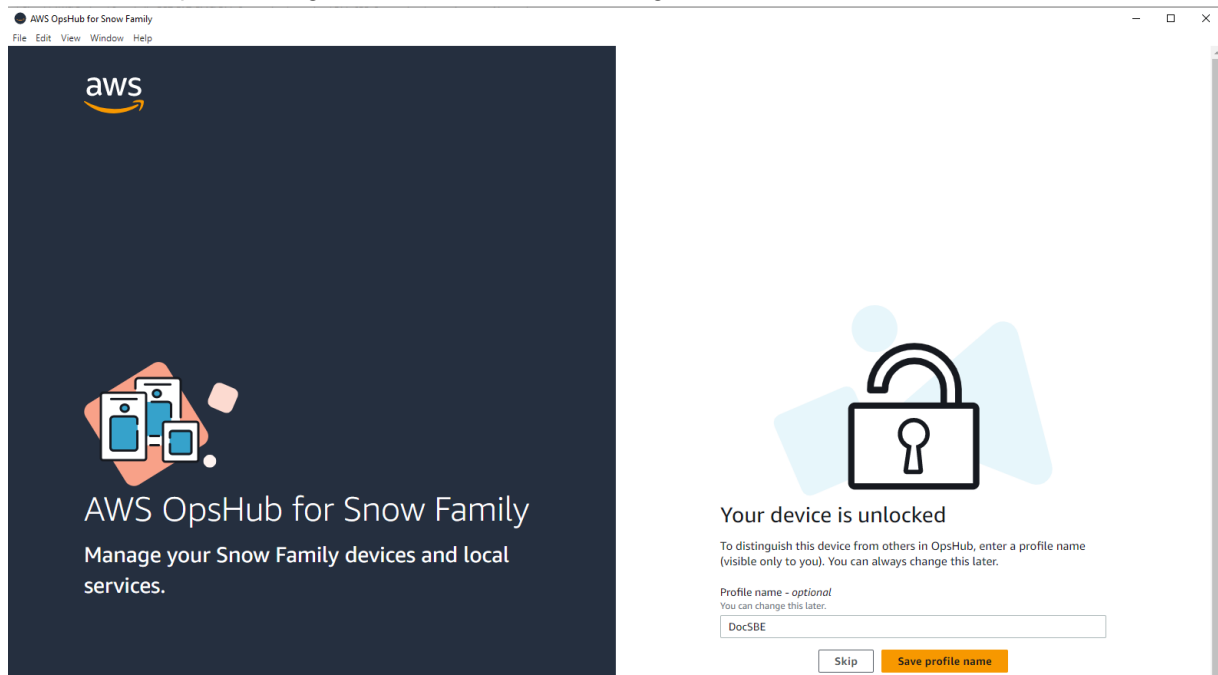
For more information, see the following resources:

- [Understanding Direct Network Interfaces on AWS Snow Family](#)
- [Network Configuration for Compute Instances](#)
- [Using Block Storage with Your Amazon EC2-compatible Instances](#)
- [Understanding Virtual Network Interfaces on AWS Snowball Edge](#)

## Configuring Snowball Edge via AWS OpsHub

To configure Snowball Edge via AWS OpsHub:

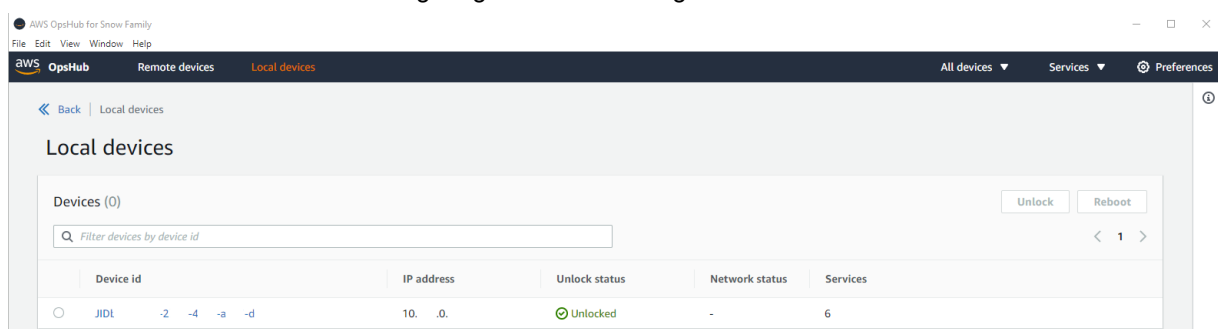
1. Launch AWS OpsHub and gain access to the Snowball Edge device.



Use the IP address configured physically on the device. Also use the unlock code and the Manifest file that AWS provides.

Saving profiles is recommended as the CLI client can use profiles for faster authentication and Snowball Edge management.

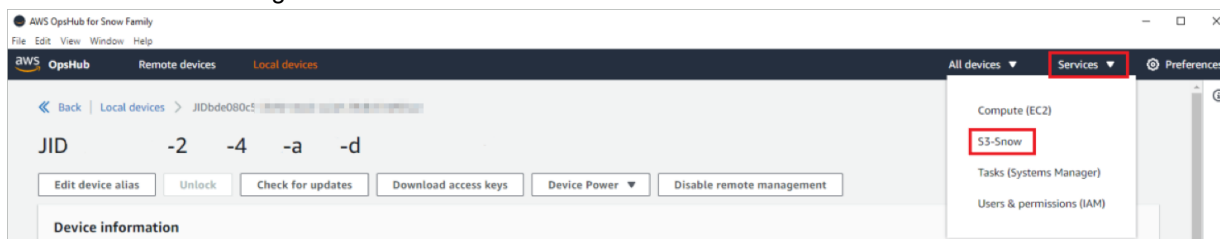
2. Select the device ID to continue configuring the Snowball Edge.



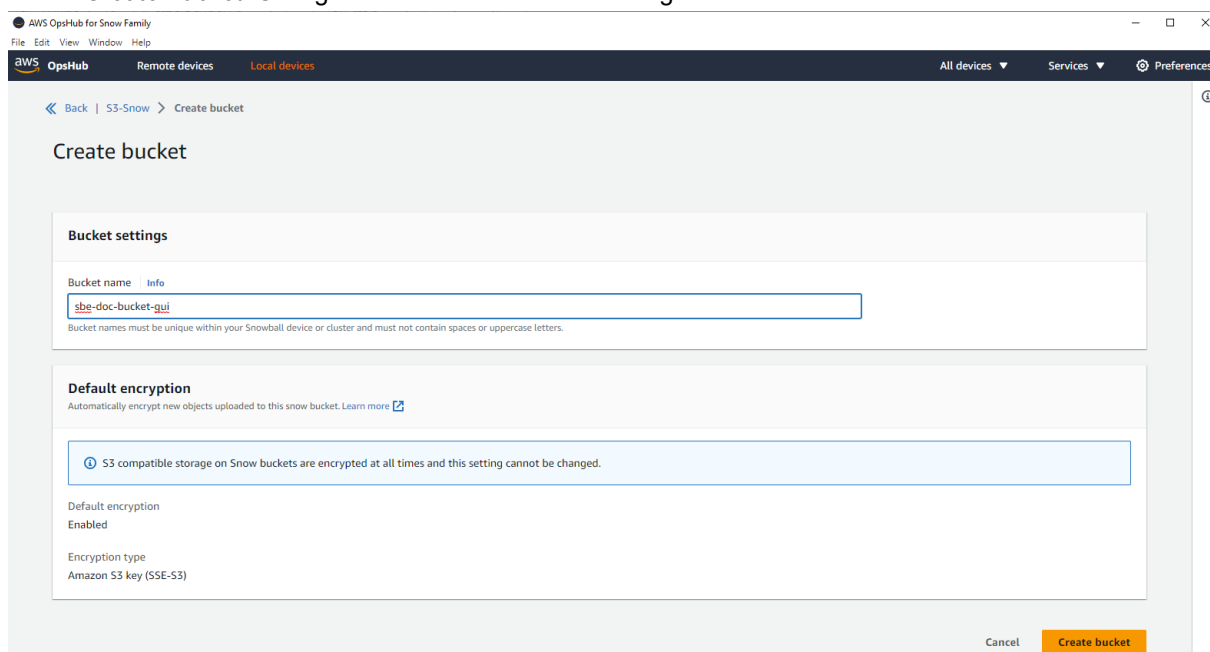
3. Configure the S3 Snow service:
  - a. In AWS OpsHub, select **Services** from the top right corner.
  - b. If the S3 Snow service is not configured, see [Set up Amazon S3 compatible storage on Snow Family devices](#).



- c. Select *S3-Snow* to configure the S3 Snow service and wait for it to become active.

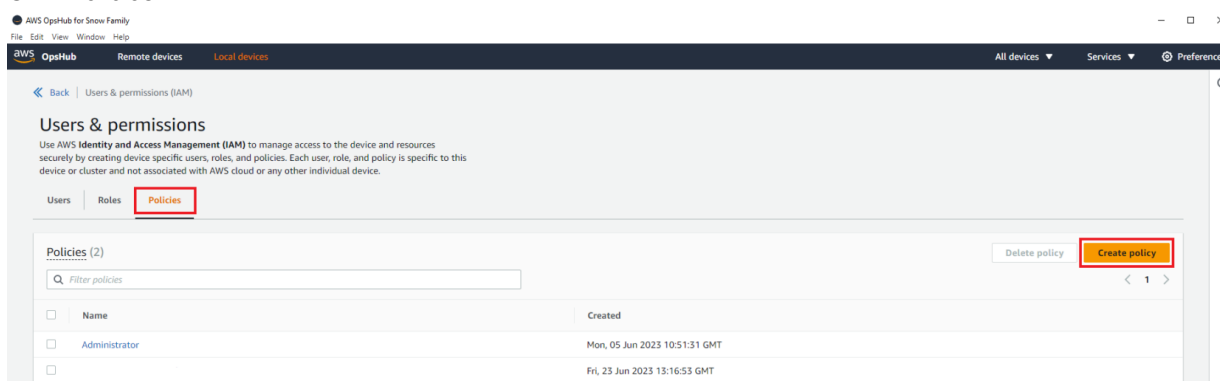


4. Select *Create Bucket*. Configure a bucket to house raw images.



5. Create policies:

- From the *Services* dropdown list, select *Users & permissions (IAM)* service page.
- Click *Policies*.



- c. In the *Policy* content pane, enter the following:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

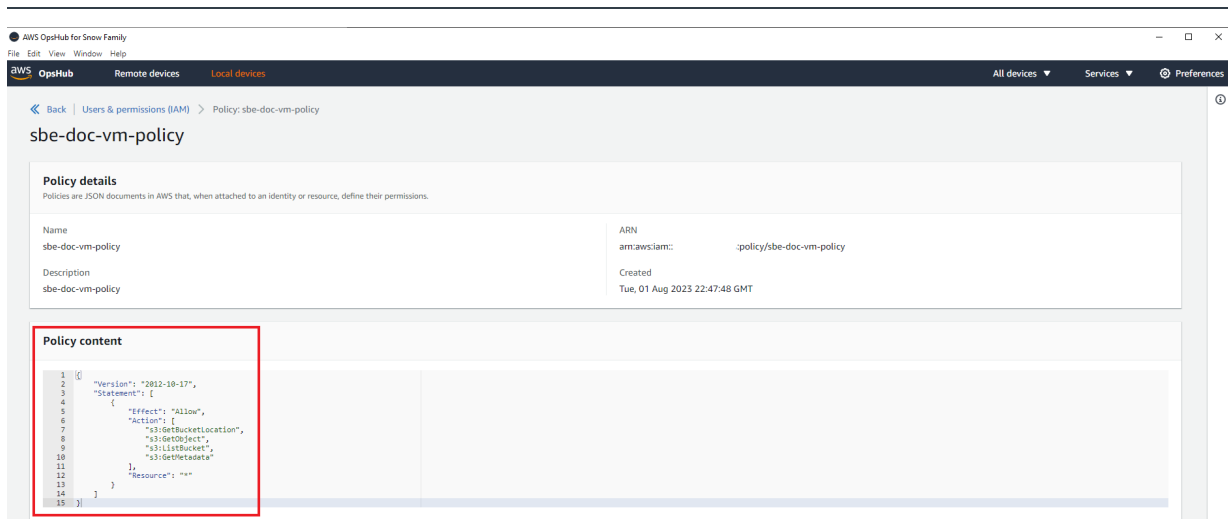
```

        "Effect": "Allow",
        "Action":
        "s3:GetBucketLocation",
            "s3:GetObject",
                "s3:ListBucket",
                "s3:GetMetadata"
        "Resource": "*"
    }
}

```



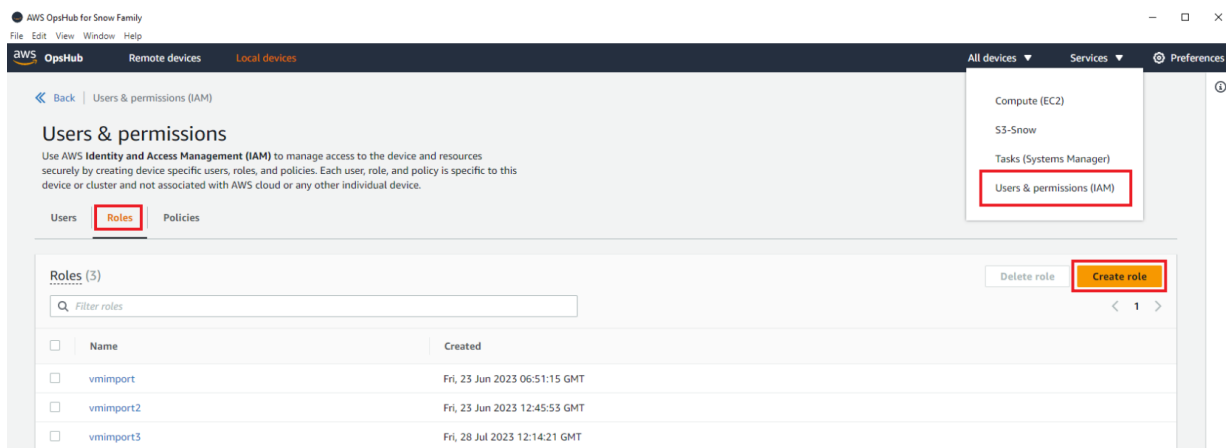
This example uses a less restrictive policy. Edit the policy to be more restrictive based on your needs.



6. (Optional) Construct your specific ARN for policies. When requiring a more restrictive policy, you can use a specific ARN. The ARN must be in the following format:  
`arn:aws:s3:snow:<AccountID>:snow/<JobID>/bucket/<BucketName>`. The following explains the ARN components:

| Component  | Description   |
|------------|---|
| AccountID  | Account ID that ordered the Snowball Edge device.   |
| JobID      | SB job ID.  |
| BucketName | Name of bucket where you will upload the raw image. |

7. Create a role:
  - a. From the *Services* dropdown list, select *Users & permissions (IAM)*.
  - b. Select *Roles*.

c. Select *Create role*.d. In the *Policies* field, select the policy that you created.e. In the *Assume role policy document* field, enter the following:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "vmie.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

**Create role**

**Role details**  
An IAM role is an IAM identity that you can create in your account that has specific permissions.

**Name**  
The name of the role to create  
sbe-doc-vm-import-role

**Description**  
A description of the role  
sbe-doc-vm-import-role

**Max session duration (optional)**  
The maximum session duration (in seconds) that you want to set for the specified role. If you do not specify a value for this setting, the default maximum of one hour is applied. This setting can have a value from 1 hour to 12 hours.

**Policies**  
Attaches the specified managed policies to the new role.  
sbe-doc-vm-policy

**Assume role policy document**  
The trust relationship policy document that grants an entity permission to assume the role.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "vmie.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

```

Cancel Create role

8. Create a user. Generating a secret key and secret for AWS CLI functionality requires users:

- From the **Services** dropdown list, select **Users & permissions (IAM)**.
- Select **Users**, then configure a user as desired. When creating a new access key for users, the secret key for the access key ID displays at the top of the page in AWS OpsHub.

**User details**

Name: sbe-doc  
Created: Wed, 02 Aug 2023 20:18:42 GMT  
ARN: arn:aws:iam::981356897718:user/sbe-doc

**Attached policies (1)**

Filter policies

| Name              | Arn  |
|-------------------|--|
| sbe-doc-vm-policy | arn:aws:iam::981356897718:policy/sbe-doc-vm-policy |

**Access keys (1)**

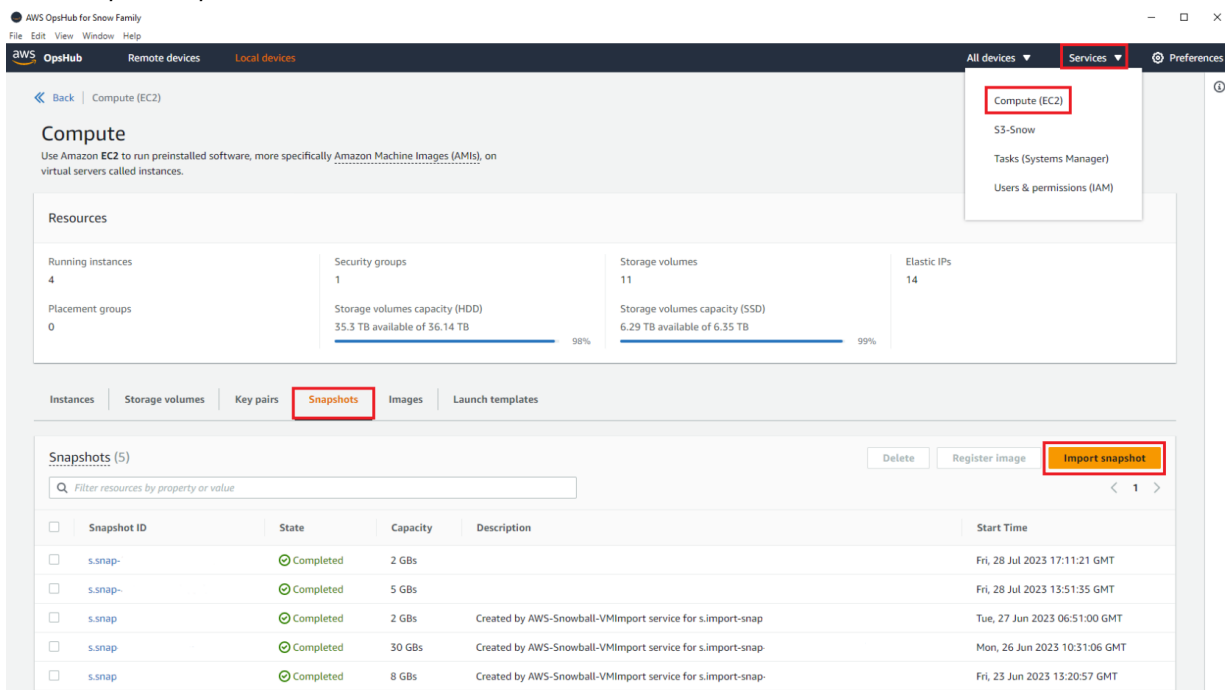
Access Key ID: AKIAEMHE4DCMZVG4DSNZ...  
Status: Active  
Created: Wed, 02 Aug 2023 20:19:43 GMT

## Generating a raw image and registering it in OpsHub

This section requires customers to obtain the FortiGate-VM qcow2 KVM image from the [Fortinet Support site](#). See [Downloading a firmware image](#). See [Converting between image formats](#) for information about creating a raw image.

### To generate a raw image and register it in OpsHub:

1. Generate a raw image:
  - a. Download FGT\_VM64\_KVM-v7.4.0.F-build2360-FORTINET.out.kvm.zip.
  - b. Unpack the zipped downloaded file containing the Qcow2.
  - c. Use `qemu-img convert -p -O raw fortios.qcow2 fgt_kvm.raw`
  - d. Place the raw image that was just created to the Snowball Edge device S3-Snow bucket that you created earlier.
2. Create a snapshot from the raw image:
  - a. From the *Services* dropdown list, select *Compute (EC2)*.
  - b. Select *Snapshots*.
  - c. Select *Import snapshot*.



- d. Select the previously created role.
- e. Select the bucket that you uploaded the raw image to.
- f. Select the raw image.

- g. Select **Submit** and wait for the snapshot import status to move to **Completed**.

**Import snapshot**

Device  
Device where import will be executed  
JID6 -2 -4 -a -d -10 .0

Import Description  
Text length should be less than 32768 characters  
sbe-doc-figt-image

Snapshot description  
Text length should be less than 32768 characters  
sbe-doc-figt-image

Role  
Role for AWS to assume to import snapshot  
sbe-doc-vm-import-role

Select S3 object  
s3://sbe-doc-bucket-gui/fortios.raw  
Raw image file, usually \*.raw file

Cancel Submit

3. From the **Snapshots** page, select the newly created snapshot, then select **Register image**.

**Compute**

Use Amazon EC2 to run preinstalled software, more specifically Amazon Machine Images (AMIs), on virtual servers called instances.

**Resources**

|                        |  |   |                   |
|------------------------|--|---|-------------------|
| Running instances<br>4 | Security groups<br>1   | Storage volumes<br>11   | Elastic IPs<br>14 |
| Placement groups<br>0  | Storage volumes capacity (HDD)<br>35.3 TB available of 36.14 TB<br>98% | Storage volumes capacity (SSD)<br>6.29 TB available of 6.35 TB<br>99% |                   |

Instances | Storage volumes | Key pairs | **Snapshots** | Images | Launch templates

**Snapshots (6)**

Delete Register image Import snapshot

| Snapshot ID              | State     | Capacity | Description        | Start Time                    |
|--------------------------|-----------|----------|--------------------|-------------------------------|
| s.snap-869b1148938ae4eb3 | Completed | 2 GBs    | sbe-doc-figt-image | Tue, 01 Aug 2023 23:03:18 GMT |

4. Configure the image as desired, then click *Submit*.

**Register image**

Name  
Text length should be less than 32768 characters  
sbe-doc-fgt-vm-image

Description  
Text length should be less than 32768 characters  
sbe-doc-fgt-vm-image

Root volume  
Name of root device, should be a valid path  
/dev/sda1

**Block device mappings**

| Device    | Snapshot ID                      | Size (GB) | Volume type                      | Delete on termination    |
|-----------|----------------------------------|-----------|----------------------------------|--------------------------|
| /dev/sda1 | s.snap-869b1148938ae4eb3 - 2 GiB | 2         | Performance Optimized SSD (sdp1) | <input type="checkbox"/> |

+ Add new volume

Cancel Submit

## Creating the FortiGate-VM Snowball Edge

The following assumes that you have already created a key pair, installed and configured Snowball Edge CLI and AWS CLI, and are familiar with working in Linux or PowerShell CLI.

### To create the FortiGate-VM Snowball Edge:

1. Launch an instance:
  - a. Go to Compute (EC2), then select *Launch Instance*.

**Compute**

Use Amazon EC2 to run preinstalled software, more specifically Amazon Machine Images (AMIs), on virtual servers called instances.

**Resources**

|                        |  |   |                   |
|------------------------|--|---|-------------------|
| Running instances<br>4 | Security groups<br>1   | Storage volumes<br>11   | Elastic IPs<br>14 |
| Placement groups<br>0  | Storage volumes capacity (HDD)<br>35.3 TB available of 36.14 TB<br>98% | Storage volumes capacity (SSD)<br>6.29 TB available of 6.35 TB<br>99% |                   |

**Instances (9)**

Filter resources by property or value

| Instance Name         | Instance type | State   | Private IP | Device      |
|-----------------------|---------------|---------|------------|-------------|
| s.i-8c7d01885c4061fbc | sbe-c.small   | Running | JIDb       | -2 -4 -a -d |

Launch instance

- b. From the *Image (AMI)* dropdown list, select the image that you created.

- c. Select *Create public IP address (VNI)*. This creates a virtual network interface and attaches it to port1 of the FortiGate.
- d. From the *Physical network interface* dropdown list, select the physical interface that is the Snowball Edge management interface.
- e. From the *IP Address assignment* dropdown list, select *DHCP*.
- f. Under *Key pair*, select *Use existing key pair*.
- g. From the *Use existing key pair* dropdown list, select the desired key pair.
- h. Read and select the checkbox for the acknowledgement.
- i. Click *Launch*.

Launch instance

Device

10.250.0.54

Image (AMI)

sbe-doc-fgt-vm-image

Instance type

sbe-c.large

Create public IP address (VNI)

Use existing IP address (VNI)

Do not attach IP address

Physical network interface

SFP+: s.ni-8bceb78d1d4b1f3d2

IP Address assignment

DHCP

Key pair

Create key pair

Use existing key pair

Do not attach key pair

Use existing key pair

sbe-doc-key-pair

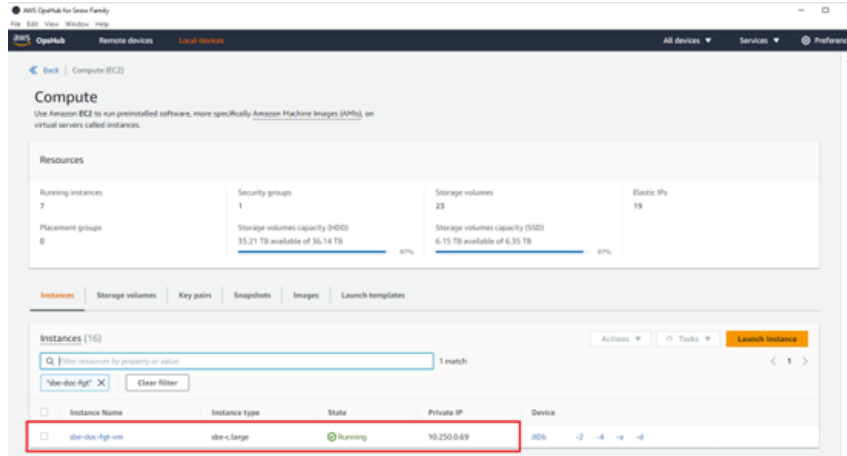
☒ I acknowledge that I have access to the selected private key file, and that without this file, I won't be able to log into my instance.

Cancel

Launch

2. After the instance state changes to *Running*, you can find the instance ID by selecting the FortiGate-VM instance in OpsHub. You will use the instance ID in the DNI CLI creation process.





For more information, see [Network Configuration for Compute Instances](#).

## Creating a DNI



OpsHub does not support DNI creation. Only the Snowballedge CLI supports DNI creation.

Following is an example of creating a DNI and attaching it to a running FortiGate-VM Snowball Edge instance.

### To obtain the physical interface ID:

Use `snowballedge describe-device` to copy and use the physical interface ID as needed. This example uses the physical interface ID:

```
snowballedge describe-device
{
  "DeviceId" : "JIDbxxxxxxxx-2xxx-4xxx-axxx-dxxxxxxxxxxxx",
  "UnlockStatus" : {
    "State" : "UNLOCKED"
  },
  "ActiveNetworkInterface" : {
    "IpAddress" : "10.xxx.xxx.54"
  },
  "PhysicalNetworkInterfaces" : [ {
    "PhysicalNetworkInterfaceId" : "s.ni-8bb5317c4fbba1682",
    "PhysicalConnectorType" : "QSFP",
    "IpAddressAssignment" : "STATIC",
    "IpAddress" : "0.0.0.0",
    "Netmask" : "0.0.0.0",
    "DefaultGateway" : "10.250.0.254",
    "MacAddress" : "00:8c:fa:ed:b8:00"
  }, {
    "PhysicalNetworkInterfaceId" : "s.ni-654321",
    "PhysicalConnectorType" : "SFP_PLUS",
```

```
    "IpAddressAssignment" : "STATIC",
    "IpAddress" : "10.250.0.54",
    "Netmask" : "255.255.255.0",
    "DefaultGateway" : "10.250.0.254",
    "MacAddress" : "00:8c:fa:ed:b8:01"
  }, {
    "PhysicalNetworkInterfaceId" : "s.ni-88686f9a4654fc2c1",
    "PhysicalConnectorType" : "RJ45",
    "IpAddressAssignment" : "STATIC",
    "IpAddress" : "10.120.100.10",
    "Netmask" : "255.255.255.0",
    "DefaultGateway" : "10.250.0.254",
    "MacAddress" : "00:8c:fa:ed:b7:ff"
  } ]
```

### To create and attach the DNI:

The following shows the format for the command for creating and attaching the DNI:

```
snowballedge create-direct-network-interface --instance-id <FortiGate-VM on Snowball Edge
instance ID> --physical-network-interface-id <Snowball Edge physical interface ID>
```

If the instance ID is s.i-123456 and the physical network interface ID is s.ni-654321, the command is as follows:

```
snowballedge create-direct-network-interface --instance-id s.i-123456 --physical-network-
interface-id s.ni-654321
```

The following shows an example of successful output for DNI creation around the SFP+ and attachment to the FortiGate-VM:

```
{
  "DirectNetworkInterface" : {
    "DirectNetworkInterfaceArn" : "arn:aws:snowball-device:::interface/s.ni-777777777777",
    "PhysicalNetworkInterfaceId" : "s.ni-654321",
    "InstanceId" : "s.i-123456",
    "Driver" : "mlx5_core",
    "MacAddress" : "06:13:48:48:8f:3e"
  }
}
```

### To create and associate the VNI:

Generally, the VNI is created and associated with the instance at the time of instance creation. However, if the VNI was not created and associated with the instance at the time of instance creation, you must create it and associate it with the instance.

```
snowballedge create-virtual-network-interface --ip-address-assignment DHCP --physical-
network-interface-id s.ni-654321
aws ec2 associate-address --public-ip <DHCP/Static IP address> --instance-id s.i-123456 --
endpoint http://Snowball_device_physical_local_ipaddress:8008
```

For information about VNIs, see [Understanding Virtual Network Interfaces on AWS Snowball Edge](#).

## Licensing the FortiGate-VM on Snowball Edge

The FortiGate-VM on Snowball Edge only supports the bring your own license licensing method. For information about licensing a FortiGate-VM, see [VM license](#).



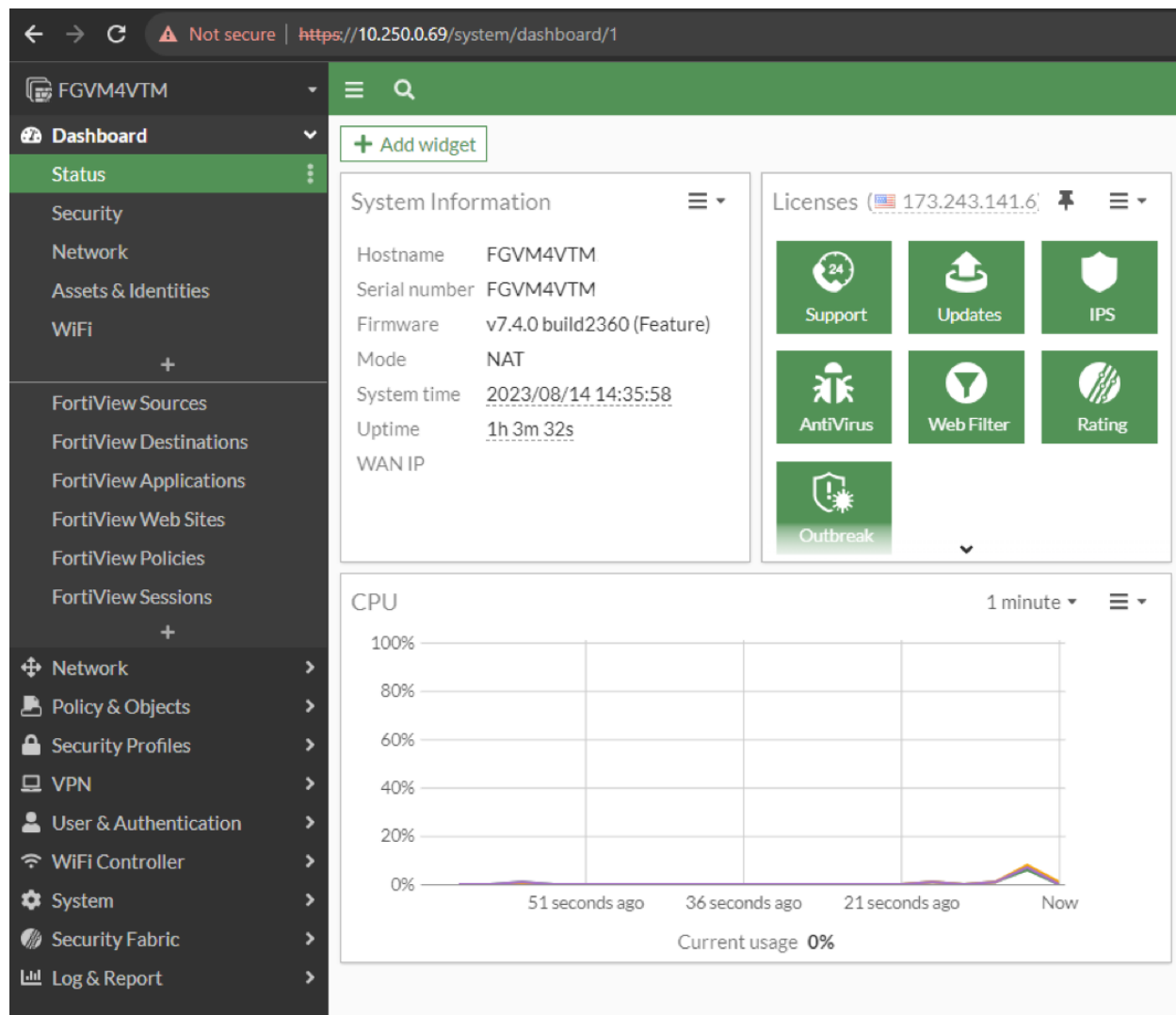
In a completely air-gapped solution, the FortiGate license expires after 30 days if the FortiGate cannot connect to FortiCloud and FortiGuard. Reach out to your sales associate to inquire about licensing a FortiGate-VM in a completely closed network environment.

---

## Accessing and configuring the FortiGate-VM

**To access and configure the FortiGate-VM:**

1. Use the virtual network interface IP address assigned at VM creation or address association to access and manage the FortiGate. The DNI is assigned to the FortiGate port2. For information about logging into a FortiGate, see [Getting started](#).



## 2. Configure port2 via the FortiOS CLI:

```
config system interface
  edit port2
    set allowaccess ping https ssh snmp http
    set mode dhcp
  next
end
```

## 3. Verify the port2 configuration and DHCP IP address:

```
config system interface
  edit port2
  get
    name: port2
    vdom : root
    vrf : 0
    cli-conn-status : 2
    fortilink : disable
    mode: dhcp
    client-options:
      distance: 5
      priority: 1
```

```
dhcp-relay-interface-select-method: auto
dhcp-broadcast-flag : enable
dhcp-relay-service : disable
dhcp-classless-route-addition: enable
ip: 10.250.0.58 255.255.255.0
end
```

4. Access the FortiGate-VM via the DNI IP address on port2.

## FortiManager

### Creating a FortiManager on Snowball Edge

This step is optional.

#### To create a FortiManager on Snowball Edge:

1. Download the FortiManager image: FMG\_VM64\_AWS-v7.4.0-buildXXXX-FORTINET.out.OpenXen.zip.
2. Repeat the steps in [Generating a raw image and registering it in OpsHub on page 153](#) to convert the FortiManager qcow2 into a raw image, upload the FortiManager raw image to the S3 bucket, create a snapshot from the raw image, and create an AMI from the FortiManager snapshot.
3. Launch the FortiManager instance with a virtual network interface (VNI):
  - a. From the *Image (AMI)* dropdown list, select the AMI that you created.
  - b. You can choose to create an outside IP address or use an existing VNI. This example uses an existing VNI, but it is likely that you will need to create a new VNI.
  - c. From the *Virtual network interface* dropdown list, select the desired VNI.

d. Click *Launch*.

Launch instance

Device

10.250.0.54

Image (AMI)

fmg\_aws\_74

Instance type

sbe-c.xlarge

☐ Create public IP address (VNI)

☒ Use existing IP address (VNI)

☐ Do not attach IP address

Physical network interface

SFP+: s.ni-8bceb78d1d4b1f3d2

Virtual network interface

10.250.0.59

Key pair

☐ Create key pair
 ☒ Use existing key pair
 ☐ Do not attach key pair

Use existing key pair

sbe-doc-key-pair

☒ I acknowledge that I have access to the selected private key file, and that without this file, I won't be able to log into my instance.

Cancel

Launch

| Instances   | Storage volumes | Key pairs     | Snapshots | Images      | Launch templates |
|---|-----------------|---------------|-----------|-------------|------------------|
| <div> <div>Instances (16)</div> <div> <div>Filter resources by property or value</div> <div>2 matches</div> </div> <div> <div>"sbe-doc" X</div> <div>Clear filter</div> </div> </div> |                 |               |           |             |                  |
| <input type="checkbox"/>  | Instance Name   | Instance type | State     | Private IP  | Device           |
| <input type="checkbox"/>  | sbe-doc-fmg-vm  | sbe-c.xlarge  | Running   | 10.250.0.59 | JIDb -2 -4 -a -d |
| <input type="checkbox"/>  | sbe-doc-fgt-vm  | sbe-c.small   | Running   | 10.250.0.69 | JIDb -2 -4 -a -d |

## Configuring the FortiManager as a licensing server for the FortiGate-VM

The following are the high-level steps for this configuration:

1. Request your entitlement file from Customer Support so that you can upload it to FortiManager.
2. Configure FortiManager for closed network license validation.
3. Configure FortiOS to use FortiManager for closed network license validation.



For information, see the following:

- [Operating as an FDS in a closed network](#)
- [Validating the FortiGate-VM license with FortiManager](#)
- [FortiManager performance and sizing in closed networks](#)
- [Configure a FortiManager without Internet connectivity to access a local FortiManager as FDS](#)

### To configure the FortiManager as a licensing server for the FortiGate-VM:

1. Access the FortiManager via the IP address of the VNI assigned to the FortiManager. In this example, the VNI was created with the 10.250.0.59 IP address.
2. License the FortiManager.
3. Run the following commands to enable license validation:

- a. Run the following commands on the FortiManager:

```
config fmupdate publicnetwork
  set status disable
end
```

- b. Upload the entitlement file.

- c. Run the following commands on the FortiGate-VM running on the Snowball Edge. For `set fmg-source-ip`, use the FortiGate's Snowball Edge inside IP address:

```
config system central-management
  set mode normal
  set type fortimanager
  set fmg 10.250.0.59
  config server-list
    edit 1
      set server-type update rating
      set server-address 10.250.0.59
    end
    set fmg-source-ip 34.223.14.234
    set include-default-servers disable
    set vdom root
  next
end
```

## Use case: Traffic inspection with DNI and VLANs

This example uses a new FortiGate-VM and two Linux VM images created with the image conversion, upload, snapshot, and AMI creation steps in [Generating a raw image and registering it in OpsHub on page 153](#).

The following table provides the virtual network interface (VNI) entry information for the example deployment in the topology:

| EC2 instance | Inside IP address | Outside IP address |
|--------------|-------------------|--------------------|
| FortiGate-VM | 34.233.14.234     | 10.250.0.69        |

### To set up this use case:

1. Create two DNIs for the FortiGate specifying the VLAN ID in each command:



Each DNI command run attaches the DNI in ascending order of ports on the FortiGate-VM. For example, the first command attaches the DNI to port2 (VLAN 10) and the second command attaches the DNI to port3 (VLAN 20).

```
snowballEdge create-direct-network-interface --endpoint https://10.250.0.54 --instance-id s.i-123456 --physical-network-interface-id s.ni-654321 --vlan 10
snowballEdge create-direct-network-interface --endpoint https://10.250.0.54 --instance-id s.i-123456 --physical-network-interface-id s.ni-654321 --vlan 20
```

2. Attach a DNI to the first Linux instance:

```
snowballEdge create-direct-network-interface --endpoint https://10.250.0.54 --instance-id s.i-789023 --physical-network-interface-id s.ni-654321 --vlan 10
```

3. Attach a DNI to the second Linux instance:

```
snowballEdge create-direct-network-interface --endpoint https://10.250.0.54 --instance-id s.i-567098 --physical-network-interface-id s.ni-654321 --vlan 20
```

### To configure the FortiGate:

```
config system interface
  edit "port1"
    set vdom "root"
    set mode dhcp
    set allowaccess ping https ssh http fgfm
    set type physical
    set snmp-index 1
  next
  edit "port2"
    set vdom "root"
    set ip 192.168.111.63 255.255.255.0
    set allowaccess ping https ssh snmp http telnet
    set type physical
    set snmp-index 6
  next
  edit "port3"
    set vdom "root"
    set ip 192.168.112.63 255.255.255.0
    set allowaccess ping https ssh snmp http telnet
    set type physical
    set snmp-index 7
  next
end
```



```
config firewall policy
  edit 1
    set uuid ca0d22da-362d-51ee-4b39-36a4d673891f
    set srcintf "port2"
    set dstintf "port3"
    set action accept
    set srcaddr "all"
    set dstaddr "all"
    set schedule "always"
    set service "ALL"
    set utm-status enable
    set ssl-ssh-profile "certificate-inspection"
    set av-profile "default"
    set ips-sensor "default"
    set application-list "default"
    set nat enable
  next
end
```

### To test the configuration:

1. From a Linux instance, attempt to transfer a virus test file between the instances. The transfer fails and generates a log that FortiOS blocked the transfer.
2. View logs by using the `execute log display` command. The following provides example output for this command:

```
1: date=2023-08-08 time=13:58:37 eventtime=1691528317272831227 tz="-0700"
logid="0211008192" type="utm" subtype="virus" eventtype="infected" level="warning"
vd="root" policyid=1 poluuid="ca0d22da-362d-51ee-4b39-36a4d673891f"
policytype="policy" msg="File is infected." action="blocked" service="HTTP"
sessionid=33993 srcip=192.168.111.64 dstip=192.168.112.65 srcport=59168 dstport=80
srccountry="Reserved" dstcountry="Reserved" srcintf="port2" srcintfrole="undefined"
dstintf="port3" dstintfrole="undefined" srcuuid="5d4592ce-33ce-51ee-06a6-2ea90a21e46e"
dstuuid="5d4592ce-33ce-51ee-06a6-2ea90a21e46e" proto=6
direction="incoming" filename="eicar.com" quarskip="Quarantine-disabled"
virus="EICAR_TEST_FILE" viruscat="Virus" dtype="av-engine"
ref="http://www.fortinet.com/ve?vn=EICAR_TEST_FILE" virusid=2172
url="http://192.168.112.65/eicar.com" profile="default" agent="curl/7.81.0"
httpmethod="GET"
analyticscksum="275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f"
analyticssubmit="false" crscore=50 craction=2 crlevel="critical"
```

## Troubleshooting

The following provides troubleshooting information for this configuration.

- If an instance is created but no IP address is assigned, a virtual network interface failed to be created. The failure could be due to the DHCP server not being accessible. These errors do not display in the GUI, but you can view them in the output of a CLI command.
- Use `snowballedge describe-device` to get interface and service information.

- Direct console connections to the instances are not supported. You can connect via SSH.
- After registering the image, OpsHub stops populating the AMI list. Close and open OpsHub to refresh the AMI list.
- To reset the network configuration, physical access is required to set all interfaces to 0.0.0.0.
- Disable and start the services to generate certificates. If the default gateway was changed and the Snowball Edge was restarted, it is most likely that the Snowball Edge services regenerated certificates.
- You cannot stop the S3 connector, but you can stop the service using the following commands:  
`snowballedge describe-service --service-id s3-snow`  
`snowballedge stop-service --service-id s3-snow`
- To view the DNI and VNI information, run `snowballedge describe-direct-network-interfaces`. The following shows example output for this command:

```
{
  "DirectNetworkInterfaces" : [ {
    ...
    {
      "DirectNetworkInterfaceArn" : "arn:aws:snowball-device:::interface/s.ni-
777777777777",
      "PhysicalNetworkInterfaceId" : "s.ni-654321",
      "InstanceId" : "s.i-123456",
      "Driver" : "ixgbev",
      "MacAddress" : "5e:12:37:b8:db:0b"
    } ]
  }
}
```

# SDN connector integration with AWS

## Access key-based SDN connector integration

See the *FortiOS Administration Guide*.

## Configuring an AWS SDN connector using IAM roles

The following summarizes minimum sufficient Identity and Access Management (IAM) roles for this deployment:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:Describe*"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

For instances running in AWS (on demand or bring your own license), you can set up the AWS SDN connector using AWS IAM credentials.

IAM authentication is available only for FGT-AWS and FGT-AWSONDEMAND platforms.

### To configure AWS SDN connector using the GUI:

1. Configure the AWS software-defined network (SDN) connector:
  - a. Go to *Security Fabric > External Connectors*.
  - b. Click *Create New*, and select *Amazon Web Services (AWS)*.
  - c. Enable *Use metadata IAM*.
  - d. Configure other fields as desired.
  - e. Click *OK*.
2. Create a dynamic firewall address for the configured AWS SDN connector:
  - a. Go to *Policy & Objects > Addresses*.
  - b. Click *Create New*, then select *Address*.
  - c. Configure the address:
    - i. From the *Type* dropdown list, select *Dynamic*.
    - ii. From the *Sub Type* dropdown list, select *Fabric Connector Address*.
    - iii. From the *SDN Connector* dropdown list, select the connector that you created.

- iv. In the *Filter* field, configure the desired filter, such as `SecurityGroupId=sg-05f4749cf84267548` or `K8S_Region=us-west-2`.
- v. Configure other fields as desired, then click *OK*.

3. Ensure that the AWS SDN connector resolves dynamic firewall IP addresses:

- a. Go to *Policy & Objects > Addresses*.
- b. Hover over the address created in step 2 to see a list of IP addresses for instances that belong to the security group configured in step 2.

| <div> <div>+ Create New</div> <div>Edit</div> <div>Clone</div> <div>Delete</div> <div>Search</div> </div> |                                |                                 |                                     |            |      |
|---|--------------------------------|---------------------------------|-------------------------------------|------------|------|
| Name  | Type                           | Details                         | Interface                           | Visibility | Ref. |
| Address 11  |                                |                                 |                                     |            |      |
| FABRIC_DEVICE   | Subnet                         | 0.0.0.0/0                       |                                     | Visible    | 0    |
| FIREWALL  | RESS                           | 0.0.0.0/0                       |                                     | Hidden     | 0    |
| SSLVPN  | IP Range                       | 10.212.134.200 - 10.212.134.210 | SSL-VPN tunnel interface (ssl.root) | Visible    | 2    |
| all   | Subnet                         | 0.0.0.0/0                       |                                     | Visible    | 0    |
| aws-ec2   | Fabric Connector Address (AWS) |                                 |                                     | Visible    | 1    |

To configure AWS SDN connector using CLI commands:

1. Configure the AWS connector:

```
config system sdn-connector
edit "aws1"
    set status enable
    set type aws
    set use-metadata-iam enable
    set update-interval 60
next
end
```

2. Create a dynamic firewall address for the configured AWS SDN connector with the supported filter. The SDN connector resolves dynamic firewall address IP addresses:

```
config firewall address
edit "aws-ec2"
    set type dynamic
    set sdn "aws1"
    set filter "SecurityGroupId=sg-05f4749cf84267548"
    set sdn-addr-type public
next
edit "aws-eks1"
    set type dynamic
    set sdn "aws1"
    set filter "K8S_Region=us-west-2"
next
end
```

3. Confirm that the AWS SDN connector resolves dynamic firewall IP addresses using the configured filter:

```
config firewall address
edit "aws-ec2"
    set type dynamic
    set sdn "aws1"
    set filter "SecurityGroupId=sg-05f4749cf84267548"
    set sdn-addr-type public
config list
    edit "34.222.246.198"
    next
    edit "54.188.139.177"
    next
    edit "54.218.229.229"
```

```

        next
    end
next
edit "aws-eks1"
    set type dynamic
    set sdn "aws1"
    set filter "K8S_Region=us-west-2"
    config list
        edit "192.168.114.197"
        next
        edit "192.168.167.20"
        next
        edit "192.168.180.72"
        next
        edit "192.168.181.186"
        next
        edit "192.168.210.107"
        next
    end
next
end

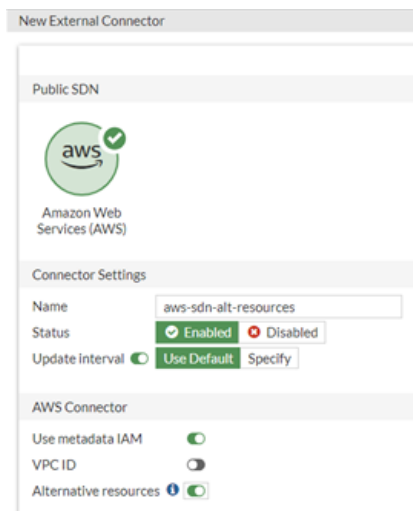
```

## SDN connector support for alternate resources

In 7.4.2 and later versions, the FortiOS AWS SDN connector supports querying AWS for resource elastic IP addresses (EIP) based on resource attributes such as the owner ID, resource descriptions, and tags.

**To configure an AWS SDN connector to query AWS alternate resources in the GUI:**

1. In FortiOS, go to *Security Fabric > External Connectors > Amazon Web Services (AWS)*.
2. Enable *Alternative resources*.



**To configure an AWS SDN connector to query AWS alternate resources in the CLI:**

```
config system sdn-connector
    edit "aws1"
        set status enable
        set type aws
        set access-key <accesskey>
        set secret-key <secretkey>
        set region "us-west-2"
        set vpc-id ''
        set alt-resource-ip enable
        set update-interval 30
    next
end
```

**To create an address object in the CLI:**

```
config firewall address
    edit "aws-alt-resource-ip-NetLB-PrivateIPs"
        set type dynamic
        set sub-type sdn
        set sdn <connectorname>
        set comment ''
        set associated-interface ''
        set color 0
        set filter "Description=ELB app/net-lb/afbb52e2591b3eee"
        set sdn-addr-type private | Public | ALL
        set fabric-object disable
    next
end
FGVM04TM##### (aws-alt-resource~ace) # set filter
<key1=value1>      [& <key2=value2>] [| <key3=value3>]
```

## Filter keys

The following lists filter keys for the AWS alternative resource:

- <InterfaceId>
- <InterfaceType>
- <SubnetId>
- <OwnerId>
- <Description>
- <VpcId>
- <Tag>
- <PrivateDnsName>
- <PublicDnsName>
- <SecurityGroupId>

For more information, see [DescribeNetworkInterfaces](#).

## Examples

The following provides examples of this feature in use.



Subnet ranges are 172.31.0.0/20 and 172.31.32.0/20 for private IP address examples.

### Query load balancer private EIP based on description

**To configure this address object in the GUI:**

1. In FortiOS, go to *Policy & Objects > Addresses*.
2. Click *Create New*, then select *Address*.
3. From the *Type* dropdown list, select *Dynamic*.
4. From the *Sub Type* dropdown list, select *Fabric Connector Address*.
5. From the *SDN Connector* dropdown list, select the AWS SDN connector.
6. For *SDN address type*, select *Private*.
7. In the *Filter* field, enter *Description=ELB app/net-lb/afbb52e2591b3eee*.
8. Click *OK*.

**To configure this address object in the CLI:**

```
FGVM04TM##### (address) # edit aws-alt-resource-ip-NetLB-PrivateIPs
FGVM04TM##### (aws-alt-resource~IPs) # show
config firewall address
    edit "aws-alt-resource-ip-NetLB-PrivateIPs"
        set uuid e94c3c1e-69c1-51ed-60d9-3e7d4d743b30
        set type dynamic
        set sdn "aws-alt-resource-ip"
        set filter "Description=ELB app/net-lb/afbb52e2591b3eee"
        config list
            edit "172.31.11.206"
            next
            edit "172.31.32.47"
            next
        end
    next
end
```

### Query load balancer public EIP based on description

**To configure this address object in the GUI:**

1. In FortiOS, go to *Policy & Objects > Addresses*.
2. Click *Create New*, then select *Address*.
3. From the *Type* dropdown list, select *Dynamic*.
4. From the *Sub Type* dropdown list, select *Fabric Connector Address*.
5. From the *SDN Connector* dropdown list, select the AWS SDN connector.

6. For *SDN address type*, select *Public*.
7. In the *Filter* field, enter `Description=ELB app/net-lb/afbb52e2591b3eee`.
8. Click *OK*.

**To configure this address object in the CLI:**

```
FGVM04TM##### (address) # edit aws-alt-resource-ip-NetLB-PublicIPs
FGVM04TM##### (aws-alt-resource~IPs) # show
config firewall address
    edit "aws-alt-resource-ip-NetLB-PublicIPs"
        set uuid 5c03cec0-69c2-51ed-4fb0-4e1425b62620
        set type dynamic
        set sdn "aws-alt-resource-ip"
        set filter "Description=ELB app/net-lb/afbb52e2591b3eee"
        set sdn-addr-type public
        config list
            edit "54.165.XX.XX"
            next
            edit "54.225.XX.XX"
            next
        end
    next
end
```

## Query dynamic address object based on interface type

**To configure this address object in the GUI:**

1. In FortiOS, go to *Policy & Objects > Addresses*.
2. Click *Create New*, then select *Address*.
3. From the *Type* dropdown list, select *Dynamic*.
4. From the *Sub Type* dropdown list, select *Fabric Connector Address*.
5. From the *SDN Connector* dropdown list, select the AWS SDN connector.
6. For *SDN address type*, select *Private*.
7. In the *Filter* field, enter `InterfaceType=gateway_load_balancer`.
8. Click *OK*.

**To configure this address object in the CLI:**

```
config firewall address
    edit "aws-alt-resource-interfacetype"
        set uuid 8faa639a-69f1-51ed-fa58-01bfd8fe3a72
        set type dynamic
        set sub-type sdn
        set sdn "aws-alt-resource-ip"
        set comment ''
        set associated-interface ''
        set color 0
        set filter "InterfaceType=gateway_load_balancer"
        set sdn-addr-type private
        config list
            edit "172.31.47.24"
```



```

        next
        edit "172.31.5.201"
        next
    end
    set fabric-object disable
next
end

```

## Query dynamic address object based on Workspace tag



Due to Workspace deployment, the Workspace interfaces must have the Workspace tag or you must assign the Workspace instance the tag.

### To configure this address object in the GUI:

1. In FortiOS, go to *Policy & Objects > Addresses*.
2. Click *Create New*, then select *Address*.
3. From the *Type* dropdown list, select *Dynamic*.
4. From the *Sub Type* dropdown list, select *Fabric Connector Address*.
5. From the *SDN Connector* dropdown list, select the AWS SDN connector.
6. For *SDN address type*, select *All*.
7. In the *Filter* field, enter *Tag.workspace=alt-resource*.
8. Click *OK*.

### To configure this address object in the CLI:

tag.keyname=valueoftag. Workspace is the key and alt-resource is the value.

```

FGVM04TM##### (address) # edit aws-alt-workspace-tag
FGVM04TM##### (aws-alt-workspace~tag) # show
config firewall address
    edit "aws-alt-workspace-tag"
        set uuid 1cfc1368-69c7-51ed-ff14-ed1ec99e8a73
        set type dynamic
        set sdn "aws-alt-resource-ip"
        set filter "Tag.workspace=alt-resource"
        set sdn-addr-type all
    config list
        edit "172.31.1.XX"
        next
        edit "23.21.70.XX"
        next
    end
next
end

```

## Query dynamic address object based on security group

### To configure this address object in the GUI:

1. In FortiOS, go to *Policy & Objects > Addresses*.
2. Click *Create New*, then select *Address*.
3. From the *Type* dropdown list, select *Dynamic*.
4. From the *Sub Type* dropdown list, select *Fabric Connector Address*.
5. From the *SDN Connector* dropdown list, select the AWS SDN connector.
6. For *SDN address type*, select *All*.
7. In the *Filter* field, enter `SecurityGroupId=sg-05011306bf07bc753`.
8. Click *OK*.

### To configure this address object in the CLI:

```
FGVM04TM22001205 (address) # edit aws-alt-workspace-secgroup
FGVM04TM22001205 (aws-alt-workspac~oup) # show
config firewall address
    edit "aws-alt-workspace-secgroup"
        set uuid da0363be-69cf-51ed-a282-01b64d23715f
        set type dynamic
        set sdn "aws-alt-resource-ip"
        set filter "SecurityGroupId=sg-05011306bf07bc753"
        set sdn-addr-type all
    config list
        edit "172.31.1.XX"
        next
        edit "23.21.70.XX"
        next
    end
next
end
```

## AWS EKS SDN connector

AWS SDN connectors support dynamic address groups based on AWS Kubernetes (EKS) filters. The following summarizes minimum permissions for this deployment:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "ec2:Describe*",
        "eks:DescribeCluster",
        "eks:ListClusters"
      ],
      "Resource": "*"
    }
  ]
}
```

```
]
}
```

Once you have the proper permissions for EKS, you must follow the steps at [Managing Users or IAM Roles for your Cluster](#) for EKS to properly pull data from the cluster. The following shows a successful pull of IP addresses from the EKS cluster:

```
awsd getting IPs from EKS cluster: dchao-cluster (us-west-2), endpoint:
https://F57B834C1ADA8ED7FA3CAFB36073D384.gr7.us-west-2.eks.amazonaws.com
kube url: https://F57B834C1ADA8ED7FA3CAFB36073D384.gr7.us-west-
2.eks.amazonaws.com/api/v1/services
kube host: F57B834C1ADA8ED7FA3CAFB36073D384.gr7.us-west-
2.eks.amazonaws.com:443:100.21.79.123
kube url: https://F57B834C1ADA8ED7FA3CAFB36073D384.gr7.us-west-
2.eks.amazonaws.com/api/v1/nodes
kube host: F57B834C1ADA8ED7FA3CAFB36073D384.gr7.us-west-
2.eks.amazonaws.com:443:100.21.79.123
k8s node ip: 172.31.34.72, nodename: ip-172-31-34-72.us-west-2.compute.internal
cluster: dchao-cluster, region: us-west-2, zone: us-west-2b
k8s node ip: 18.237.109.243, nodename: ip-172-31-34-72.us-west-2.compute.internal
cluster: dchao-cluster, region: us-west-2, zone: us-west-2b
kube url: https://F57B834C1ADA8ED7FA3CAFB36073D384.gr7.us-west-
2.eks.amazonaws.com/api/v1/pods
kube host: F57B834C1ADA8ED7FA3CAFB36073D384.gr7.us-west-
2.eks.amazonaws.com:443:100.21.79.123
k8s pod ip: 172.31.34.72, podname: aws-node-7kbm5, namespace: kube-system
cluster: dchao-cluster, region: us-west-2, zone: us-west-2b
k8s pod ip: 172.31.45.127, podname: coredns-6f647f5754-85m88, namespace: kube-system
cluster: dchao-cluster, region: us-west-2, zone: us-west-2b
k8s pod ip: 172.31.38.147, podname: coredns-6f647f5754-87ch7, namespace: kube-system
cluster: dchao-cluster, region: us-west-2, zone: us-west-2b
k8s pod ip: 172.31.34.72, podname: kube-proxy-ks9pw, namespace: kube-system
cluster: dchao-cluster, region: us-west-2, zone: us-west-2b
```

After configuring the above, follow the instructions in the [FortiOS Cookbook](#) to complete configuration.

## Populating threat feeds with GuardDuty

AWS GuardDuty is a managed threat detection service that monitors malicious or unauthorized behaviors/activities related to AWS resources. GuardDuty provides visibility of logs called "findings", and Fortinet provides a Lambda script called "[aws-lambda-guardduty](#)", which translates feeds from AWS GuardDuty findings into a list of malicious IP addresses in an S3 location, which a FortiGate-VM can consume as an external threat feed after being configured to point to the list's URL. To use this feature, you must subscribe to GuardDuty, CloudWatch, S3, and DynamoDB.

Installing and configuring GuardDuty requires knowledge of:

- CLI
- AWS Lambda function, DynamoDB, S3 bucket, and IAM
- Node.js

The Lambda script is available to download on [GitHub](#).

## Security implications

It is highly recommended that you create a dedicated AWS IAM role to run this Lambda function. The role should have limited permissions to restrict operation on a dedicated S3 bucket resource for only this project.

It is never suggested to attach a full control policy such as AmazonS3FullAccess, which has full permissions to all resources under your Amazon AWS account, to the role which runs the Lambda function. Allowing full-access permissions to all resources may put your resources at risk.

Following is a list of permissions required for the IAM role to run this project across the required AWS services:

| AWS service | Permission  |
|-------------|---|
| S3          | ListBucket, HeadBucket, GetObject, PutObject, PutObjectAcl                  |
| DynamoDB    | DescribeStream, ListStreams, Scan, GetShardIterator, GetRecords, UpdateItem |

## Parameters

GuardDuty findings give visibility on the following:

- Severity: high/medium/low (associated with scores)
- Where the behavior/activity occurred: Region, resource ID, account ID
- When: last seen date/time
- Count
- Detailed information
  - Affected resource: type/instance ID/image ID/port/resource type/image description/launch time/tags/network interfaces (public IP, private IP, subnet ID, VPC ID, security groups)
  - Action: type/connection direction
  - Actor
  - Additional

For more information about Amazon GuardDuty, see the [Amazon GuardDuty official website](#).

There are five configurable environment variables in the Lambda function:

| Variable name    | Type    | Description   |
|------------------|---------|---|
| MIN_SEVERITY     | Integer | The minimum severity to block an IP address. Defaults to 3. Value ranges from 1 to 10 by <a href="#">AWS GuardDuty definition</a> . |
| S3_BUCKET        | Text    | S3 bucket name to store the IP block list file. No default value. Must specify.   |
| S3_BLOCKLIST_KEY | Text    | Path to the IP block list file within the S3 bucket. No default value. Must specify. The relative file path to the S3 bucket.       |
| REGION           | Text    | AWS region to run Lambda, DynamoDB services. Must specify.  |

| Variable name  | Type | Description  |
|----------------|------|--|
| DDB_TABLE_NAME | Text | DynamoDB table name which stores malicious IP addresses from findings. Must specify. |

## Installation

You can follow the following installation steps to setup this Lambda function:

### Prerequisites

See the following for a list of tools required to deploy this project before installation. Some prerequisites are platform-specific. Choose the right one for your OS (such as Windows, Linux, or macOS).

- [Node.js](#) (6.5.0 or later)
- [npm](#). Although npm comes with Node.js, check [here](#) for how to install npm and manage the npm version.
- [AWS account](#)
- [Git](#) (latest version)
- [\\*Git Bash](#) (latest version). Git Bash is a solution for Windows platform users to run the following installation steps. The article [Use git, ssh and npm on windows with Git Bash](#) gives more information about setting up Git Bash on Windows.

### Preparing the deployment package

When you have all prerequisites ready, you can continue the installation as follows. The commands in each steps are intended to run in Terminal or Git Bash only.

You must create a deployment package from the local Git project repository, which will be uploaded for the Lambda function creation in a later step.

#### To prepare the deployment package:

1. Clone this project into the "guardduty" folder in your current local directory, and enter the project directory:  

```
$ git clone https://github.com/fortinet/aws-lambda-guardduty.git guardduty  
$ cd guardduty
```
2. Install project dependencies:  

```
$ npm install
```
3. Build this project locally to create a deployment package .zip file. The file will be located in ./dist/aws\_lambda\_guardduty.zip:  

```
$ npm run build
```

### Setting up the S3 bucket

This project needs one S3 bucket. The example in the following steps creates an S3 bucket named "my-aws-lambda-guardduty". The example uses the bucket name in some configuration steps. Due to bucket naming limitations in S3, each bucket should have a globally unique name. Therefore, your bucket should have a different name than the example's. Write down your bucket name, since it is used in other configuration steps.

Create the S3 bucket to store the IP block list. In this example, the bucket is named `my-aws-lambda-guardduty`. This bucket is required to run this project. Although bucket creation is region-specific, once created, the bucket can be accessed from any region. Do not grant the bucket public access permissions. The Lambda function points to this bucket through its `S3_BUCKET` environment variable.

## Setting up the DynamoDB table

One DynamoDB table with the stream feature enabled is required to store records of malicious IP addresses from GuardDuty findings. DynamoDB tables and Lambda functions are region-specific so you must create the table and the Lambda function in the same AWS region. A DynamoDB trigger on this table is created to cause the Lambda function to execute. Since the Lambda function has not been created yet, instructions to create the trigger are provided later in [Setting up the DynamoDB stream trigger](#).

1. Create the DynamoDB table. In this example, the table is named `my-aws-lambda-guardduty-db`.
  - a. For the primary key, do the following:
    - i. Input the value `finding_id`. This value is case-sensitive.
    - ii. From the data type dropdown list, select `String`.
  - b. Add a sort key:
    - i. Input the value `ip`. This value is case-sensitive.
    - ii. From the data type dropdown list, select `String`.
  - c. Check used default settings for *Table settings*.
  - d. Click *Create*.
2. Enable the Stream feature on the table.
  - a. On the *Overview* tab, click *Manage Stream*, select *Keys only*, then click *Enable* to save.
  - b. Write down the *Latest stream ARN*. This ARN is used in the IAM policy creation step.

## Setting up the IAM role and policies

An IAM role is created to run the Lambda function. Three policies attach to the IAM role. The first one is a user-managed policy which grants permissions to operation on the S3 bucket `my-aws-lambda-guardduty`. The second one is a user-managed policy which grants permission to operation on the DynamoDB table `my-aws-lambda-guardduty-db`. The third one is an AWS-managed policy which allows the Lambda function to write logs to CloudWatch.

1. Create a policy to operate on the S3 bucket.
  - a. Choose S3 as its service.
  - b. In *Access level*, add *ListBucket* on *List*, *HeadBucket* and *GetObject* on *Read*, *PutObject* on *Write*, and *PutObjectAcl* on *Permissions management*.
  - c. In *Resources*, choose *Specific*.
    - i. For the *bucket* resource type, add the `my-aws-lambda-guardduty` S3 bucket ARN (for example, `arn:aws:s3:::my-aws-lambda-guardduty`) to restrict access to any file in the specific bucket only.
    - ii. For the *object* resource type, add the `my-aws-lambda-guardduty` S3 bucket ARN and a `/*` wildcard (for example, `*arn:aws:s3:::my-aws-lambda-guardduty/*`) to restrict access to any file in the specific bucket only.
  - d. Click *Review Policy*, then *Save Changes*. The policy in JSON form looks like the following code snippet:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
```

```

        "Effect": "Allow",
        "Action": [
            "s3:PutObject",
            "s3:GetObject",
            "s3:ListBucket",
            "s3:PutObjectAcl"
        ],
        "Resource": [
            "arn:aws:s3::my-aws-lambda-guarddduty",
            "arn:aws:s3::my-aws-lambda-guarddduty/*"
        ]
    },
    {
        "Sid": "VisualEditor1",
        "Effect": "Allow",
        "Action": "s3:HeadBucket",
        "Resource": "*"
    }
]
}

```

**2. Create a policy to operate on the DynamoDB table.**

- a. Choose DynamoDB as its service.
- b. In *Access level*, add *ListStreams* on *List*, *DescribeStream*, *GetRecords*, *GetShardIterator*, *Scan* on *Read*, and *UpdateItem* on *Write*.
- c. In *Resources*, choose *Specific*.
  - i. For the *stream* resource type, add the my-aws-lambda-guarddduty-db latest stream ARN (for example, arn:aws:dynamodb:us-east-1:888888888888:table/my-aws-lambda-guarddduty-db/2018-07-20T10:30:10.888). Replace the Stream label content with the \* wildcard to allow for access to any stream resource of the my-aws-lambda-guarddduty-db table.
  - ii. For the *table* resource type, add the my-aws-lambda-guarddduty-db DynamoDB table ARN (for example, arn:aws:dynamodb:us-east-1:888888888888:table/my-aws-lambda-guarddduty-db) to restrict access to the specific table only.
- d. Click *Review Policy*, then *Save Changes*. The policy in JSON form looks like the following code snippet:

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "dynamodb:GetShardIterator",
                "dynamodb:Scan",
                "dynamodb:UpdateItem",
                "dynamodb:DescribeStream",
                "dynamodb:GetRecords"
            ],
            "Resource": [
                "arn:aws:dynamodb:us-east-1:888888888888:table/my-aws-lambda-guarddduty-db/stream/*",
                "arn:aws:dynamodb:us-east-1:888888888888:table/my-aws-lambda-guarddduty-db"
            ]
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",

```

```

        "Action": "dynamodb:ListStreams",
        "Resource": "*"
    }
}
]
}

```

3. Create an IAM role to run the Lambda function.
  - a. Choose the Lambda service that will use this role.
  - b. Attach the two user-managed policies created in the previous steps to this role.
  - c. Attach the AWS-managed policy `AWSLambdaBasicExecutionRole` to this role.

## Creating the Lambda function

The Lambda function is created with the deployment package generated in [Preparing the deployment package on page 177](#). This package is uploaded directly to this Lambda function. The Lambda function has five configurable environment variables for severity, AWS region, DynamoDB table name, and IP block list file entry point.

1. Create a function that authors from scratch.
  - a. Give the function a unique name.
  - b. For its *Runtime*, select *Node.js 6.10*.
  - c. For *Role*, select *Choose an existing role*. Select the role created in [Setting up the IAM role and policies on page 178](#).
2. Set up the function code.
  - a. For code entry type, select *Upload a .ZIP file*. The *Function package* field appears.
  - b. For *Function package*, click *Upload* to upload the deployment package .zip file generated in [Preparing the deployment package](#).
  - c. For *Handler*, enter *index.handler*.
3. Set up the environment variables. Note values for key fields are case-sensitive and should all be in upper case.
  - a. Add a key `MIN_SEVERITY` and input a value of 3.
  - b. Add a key `S3_BUCKET` and paste the name of the S3 bucket created in [Setting up the S3 bucket on page 177](#). In this example, the S3 bucket name is `my-aws-lambda-guardduty`.
  - c. Add a key `S3_BLOCKLIST_KEY` and input a value of `ip_blocklist` or a different name as desired.
  - d. Add a key `REGION` and input the AWS region where your Lambda function and DynamoDB table are situated. For example, the region of US East (N. Virginia) is `us-east-1`. For information about AWS Regions, see [AWS Regions and Endpoints](#).
  - e. Add a key `DDB_TABLE_NAME` and input the name of the DynamoDB table created in [Setting up the DynamoDB table on page 178](#). In this example, the DynamoDB table name is `my-aws-lambda-guardduty-db`.
4. Save the Lambda function.

## Setting up the DynamoDB stream trigger

You must add a trigger to the DynamoDB table created in [Setting up the DynamoDB table on page 178](#). This trigger is the key that causes the Lambda function to generate a full IP block list to a static file in the S3 bucket.

The following describes how to create a trigger on a DynamoDB table

1. In DynamoDB, click the table to toggle on its detail window.
2. On the *Triggers* tab, click *Create Trigger*, then *Existing Lambda function* from the dropdown list.
3. From the *Function* dropdown list, select the Lambda function created in [Creating the Lambda function on page 180](#).
4. Leave the *Batch size* value at its default, which is normally 100.



5. Select the *Enable trigger* checkbox.
6. Click *Create*.

At this point, installation is complete, although the AWS CloudWatch and GuardDuty services need additional configuration to work with the Lambda function.

## Setting up CloudWatch

In this section, a CloudWatch event rule is created to invoke the Lambda function based on events happening in GuardDuty findings. If you have not subscribed to GuardDuty yet, you must subscribe to it before moving on. For information about GuardDuty, see [Amazon GuardDuty](#).

### To create a new event rule:

1. For *Event Source*, choose *Event Pattern*, and select *Events by Service* from the dropdown list.
2. For *Service Name*, select *GuardDuty* from the dropdown list.
3. For *Event Type*, select *GuardDuty Finding* from the dropdown list.
4. Check that the *Event Pattern Preview* looks like the following code snippet:

```
{
  "source": [
    "aws.guarddduty"
  ],
  "detail-type": [
    "GuardDuty Finding"
  ]
}
```

5. For the targets, click *Add Target\** and select *Lambda function* from the dropdown list.
6. For the *Function*, select the Lambda function you created from the dropdown list.
7. Click *Configure rule details*.
8. Name the rule as desired.
9. For *State*, select the *Enabled* checkbox.
10. Click *Create Rule*.

## Testing the setup

When all services have been created and configured properly, execute this simple test to verify your work.

### To test the setup:

1. Create and run the test event from the Lambda function:
  - a. From the *Test Event* dropdown list, select *Configure test events*.
  - b. Select *Create new test event* to add a test event with the content as the following code snippet:

```
{
  "id": "fa9fa4a5-0232-188d-dalc-af410bcfc344",
  "detail": {
    "service": {
      "serviceName": "guarddduty",
      "action": {
        "networkConnectionAction": {
```

```

        "connectionDirection": "INBOUND",
        "remoteIpDetails": {
            "ipAddressV4": "192.168.123.123"
        }
    },
    "additionalInfo": {
        "threatListName": "GeneratedFindingThreatListName">
    },
    "eventLastSeen": "2018-07-18T22:12:01.720Z"
},
"severity": 3
}
}

```

- c. From the *Test Event* dropdown list again, select the event you have just created, then click *Test* to execute this Lambda function with the given event.

## 2. Verify the test result.

- a. If everything was set up correctly, you will see *Execution result: succeeded* on the top of the page of this Lambda function.
- b. Check and see a record with *finding\_id - fa9fa4a5-0232-188d-da1c-af410bcfc344* and *ip - 192.168.123.123* is in the DynamoDB table - *my-aws-lambda-guardduty-db*.
- c. Check and see the file *ip\_blocklist* resides in the S3 bucket *my-aws-lambda-guardduty*.
- d. Check that the *ip\_blocklist* file has a *Read object* permission for *Everyone* under the *Public access* section.
- e. Check that the *ip\_blocklist* is accessible through its link in browser (e.g. [https://s3-us-east-1.amazonaws.com/\\*\\*my-aws-lambda-guardduty\\*\\*\\*/ip\\_blocklist](https://s3-us-east-1.amazonaws.com/**my-aws-lambda-guardduty***/ip_blocklist))
- f. Check that the *ip\_blocklist* file contains *192.168.123.123* in a single line in its content.

## (Optional) Generating sample findings in GuardDuty

Amazon GuardDuty monitors your AWS infrastructures on a continuous basis to detect malicious or unauthorized behavior and creates records based on such findings. If you have just subscribed to GuardDuty for the first time, you will see no findings in the list. You can click *Generate sample findings* under *Settings* and get some samples. Then several dummy findings marked as "[SAMPLE]" are created. As long as you have set up the Lambda function and CloudWatch correctly, some of those sample findings trigger the CloudWatch event rule to run the Lambda function. A few new IP addresses eventually appear in the *ip\_blocklist*.

## Setting up the FortiGate(s)

As a FortiGate-VM feature, GuardDuty integration introduces the ability to dynamically import external block lists from an HTTP server. You can use the block lists to enforce your organization's specialized security requirements. This can include long term policies, such as always blocking access to certain websites, or short term requirements to block access to known compromised locations. Since these lists are dynamically imported, the FortiGate-VM instantly imports any changes made to the list.

In this example, the FortiGate-VM integrates with AWS GuardDuty to populate a list, which is treated as a "threat feed". You can use a threat feed to deny access to a source or destination IP address in web filter and DNS filter profiles, SSL inspection exemptions, and as a source/destination in proxy policies. The block list is stored as an external resource, which is dynamically imported to the FortiGate-VM at a configured interval/refresh rate to maintain an updated list. The administrator can configure multiple threat feeds in each profile.

1. To configure a threat feed, go to *Security Fabric > External Connectors*, then click *Create New*, then *IP Address* under *Threat Feeds*.
2. The following example creates an IP address connector. The resource name appears as an external IP blocklist in DNS filter profiles and as a source/destination in proxy policies. Configure the following:
  - a. *URI of external resource*: link to an external resource file. The file should be a plain text file with one IP address on each line. In this example, the IP address is `https://s3-us-east-1.amazonaws.com/***my-aws-lambda-guardduty***/ip_blocklist`. The file size is up to 10 MB or 128000 lines of text, whichever is more restrictive.
  - b. *Refresh Rate*: time interval to refresh the external resource. The rate can be between 1 to 43200 minutes.
3. Go to *Policy & Objects > Firewall Policy* and create/edit a policy. In the *Source* and *Destination* fields, you should be able to add the new feed.

## Cleanup

Since test events and sample findings can update the `ip_blocklist` with sample IP addresses, cleaning up the `ip_blocklist` is recommended for production use. This cleanup step removes the `ip_blocklist` from the S3 bucket and clears the DynamoDB table.

### To clean up the `ip_blocklist`:

1. Delete all records from the DynamoDB table. In this example, the DynamoDB table is `my-aws-lambda-guardduty-db`.
2. Delete the `ip_blocklist` file in the `my-aws-lambda-guardduty` bucket.

## Pipelined automation using AWS Lambda

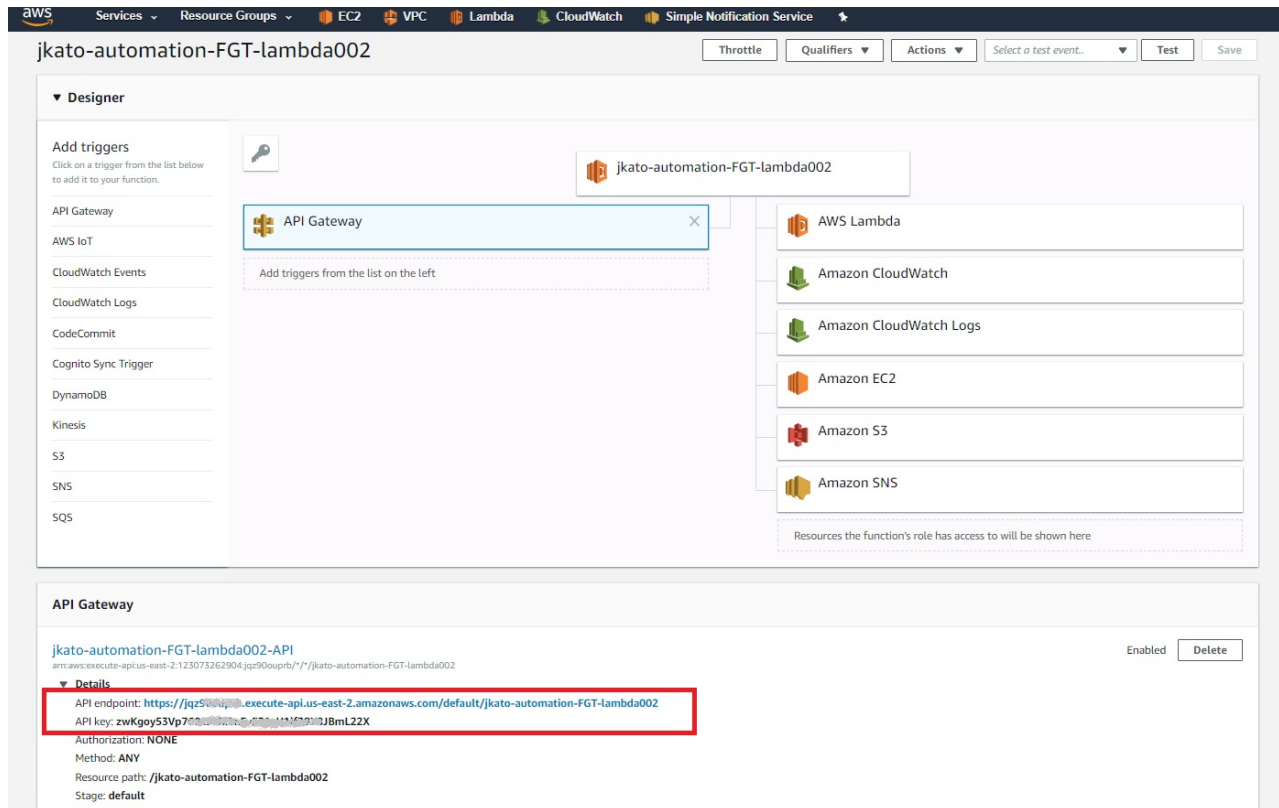
With automation stitches, you can decrease response times to security events by automating activities between different device components in the Security Fabric. You can monitor events from any source in the Security Fabric and set up action responses to any destination.

FortiGate (both physical and virtual instances) supports AWS Lambda as an automated workflow.

- [Creating an automation stitch on page 183](#)
- [Configuring an example automation stitch on page 184](#)

## Creating an automation stitch

1. In FortiOS, go to *Security Fabric > Automation*.
2. Under *Action*, select *AWS Lambda*.
3. In the *Name* field, enter the desired name for the stitch.
4. Under *AWS Lambda*, configure the following:
  - a. In the *Name* field, enter the name of the action.
  - b. For the *API Gateway* field, see the Lambda code configuration page, which shows the API gateway URL once added to the Lambda code.
  - c. For the *API Key* field, see the Lambda code configuration page as above.



The screenshot shows the AWS Lambda console for the function 'jkato-automation-FGT-lambda002'. The 'Designer' tab is active, showing the 'API Gateway' trigger. The 'Details' section for the API Gateway is expanded, showing the following information:

- API endpoint: <https://jqz5v8v9p.execute-api.us-east-2.amazonaws.com/default/jkato-automation-FGT-lambda002>
- API key: `zwKgoy53Vp70...`
- Authorization: `NONE`
- Method: `ANY`
- Resource path: `/jkato-automation-FGT-lambda002`
- Stage: `default`



You must specify an AWS role that is sufficiently privileged to run the Lambda code and access CloudWatch/CloudWatch logs.

## Configuring an example automation stitch

Let's try creating an example automation stitch with a simple pipeline. The example pipeline is as follows:

1. When an event log is created due to a successful login to the FortiGate,
2. Pick up one of the key-value pairs that the FortiGate sends to the API gateway
3. Invoke its AWS Lambda script, and, as an action, output the value on CloudWatch

Other actions you may want to configure include quarantining an EC2 instance by applying a different security group, renaming an EC2 tag, and so on. You can configure a variety of actions as fits your deployment scenario.

For this example, do the following:

1. Create an automation stitch by completing all steps in [Creating an automation stitch](#).
2. Under *Trigger*, select *Event Log*.
3. In the *Event* dropdown list, select *Admin login successful*.
4. You will need to know what elements FortiGate sends with the event log and what to pick on the Lambda script. Now let's make the example event happen by logging into the FortiGate successfully as an admin user. Log out of the FortiGate, then log in again. You will see the corresponding event log.

## 5. Go to **Log & Report > System Events**. Find the desired event log.

The screenshot shows the FortiGate System Events log. The log entry for 'Administrator admin logged in successfully from https(208.xx.yy.1)' is highlighted. The 'Log Details' pane on the right provides additional information about the event.

| Log Details   |
|---|
| <b>General</b>  |
| Date: 08/29/2018  |
| Time: 14:22:00  |
| Virtual Domain: root  |
| Log Description: Admin login successful                                     |
| <b>Source</b>   |
| IP: 208.xx.yy.1   |
| User: admin   |
| <b>Destination</b>  |
| IP: 10.10.1.12  |
| <b>Action</b>   |
| Action: login   |
| Status: success   |
| Reason: none  |
| <b>Security</b>   |
| Level: [Progress Bar]   |
| <b>Event</b>  |
| Profile Name: super_admin   |
| User Interface: https(208.xx.yy.1)  |
| Message: Administrator admin logged in successfully from https(208.xx.yy.1) |
| <b>Other</b>  |
| Log event original timestamp: 1535577720                                    |
| Method: https   |
| Log ID: 32001   |

## 6. Download the log as a file. You can filter logs.

## 7. Open the SystemEventLog-disk-<date/time/number>.log file in a text editor. It should look as follows:

```
date=2018-08-29 time=15:56:13 logid="0100032001" type="event" subtype="system"
level="information" vd="root" eventtime=1535583373 logdesc="Admin login successful"
sn="15355xyz73" user="admin" ui="https(208.xx.yy.1)" method="https"
srcip=208.xx.yy.1 dstip=192.168.1.15 action="login" status="success" reason="none"
profile="super_admin" msg="Administrator admin logged in successfully from https
(208.xx.yy.1)"
```

You have a rough idea about what elements can be picked. Raw JSON data will look as follows:

```
{ email: 'your_email@xyz.com',
  data:
    { stitch: 'Your Stitch Name',
      actions: [ [Object] ],
      eventtype: 'login',
      sn: 'Serial Number of your FortiGate',
      time: 1535587464,
      rawlog:
        { date: '2018-08-29',
          time: '17:04:24',
          logid: '0100032001',
          type: 'event',
          subtype: 'system',
          level: 'information',
          vd: 'root',
          eventtime: '1535587464',
          logdesc: 'Admin login successful',
          sn: 'xyz',
          user: 'admin',
          ui: 'https(FortiGate IP address)',
          method: 'https',
          srcip: 'FortiGate IP address',
          dstip: '10.10.1.12',
          action: 'login',
          status: 'success',
          reason: 'none',
          profile: 'super_admin',
```

```

    msg: 'Administrator admin logged in successfully from https(FortiGate IP
        address)'
  }
}
}

```

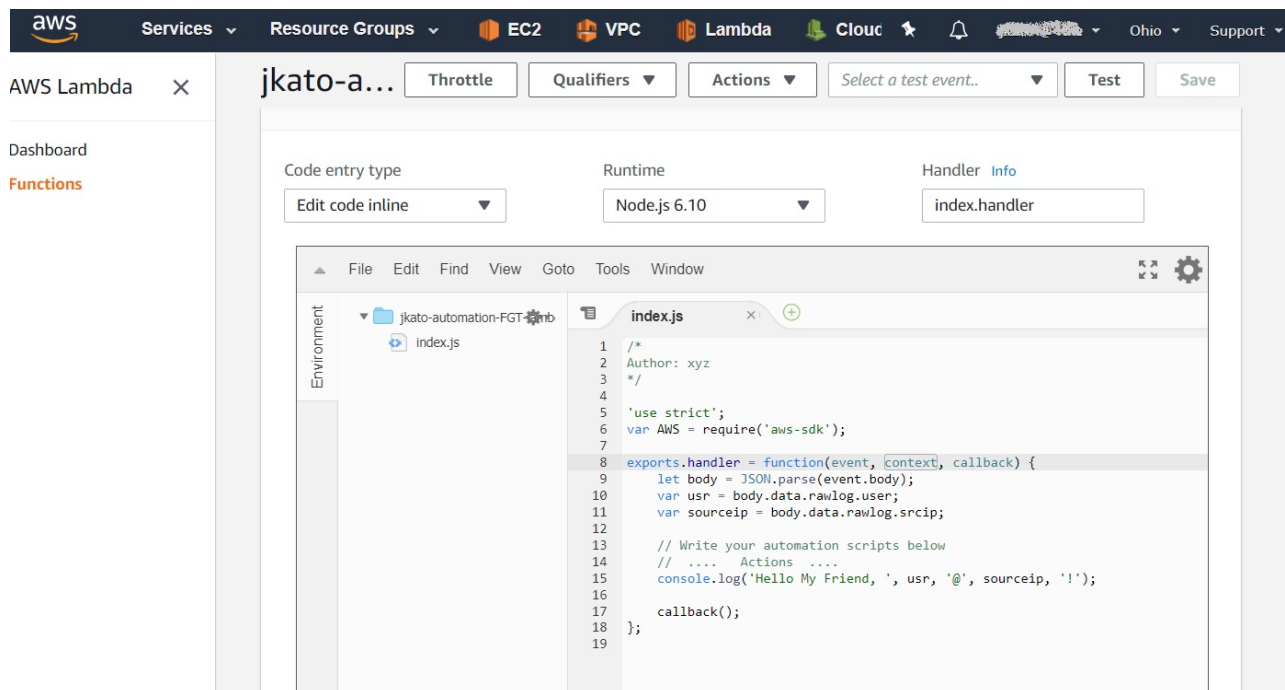
8. You can pick available key-value pairs in your AWS Lambda code. In this particular event log, useful keys include `stitch / date / time / vd / logdesc / user / ui / method / srcip / dstip / action / status / profile / msg`.
9. You can see all JSON logs sent by FortiGate on CloudWatch Log by entering the following line in the Lambda code:  
`console.log(JSON.parse(event.body));`
10. Now, as an example, let's pick `user: 'admin'` and `srcip: '208.xx.yy.1'`. Here is the Lambda script:

```

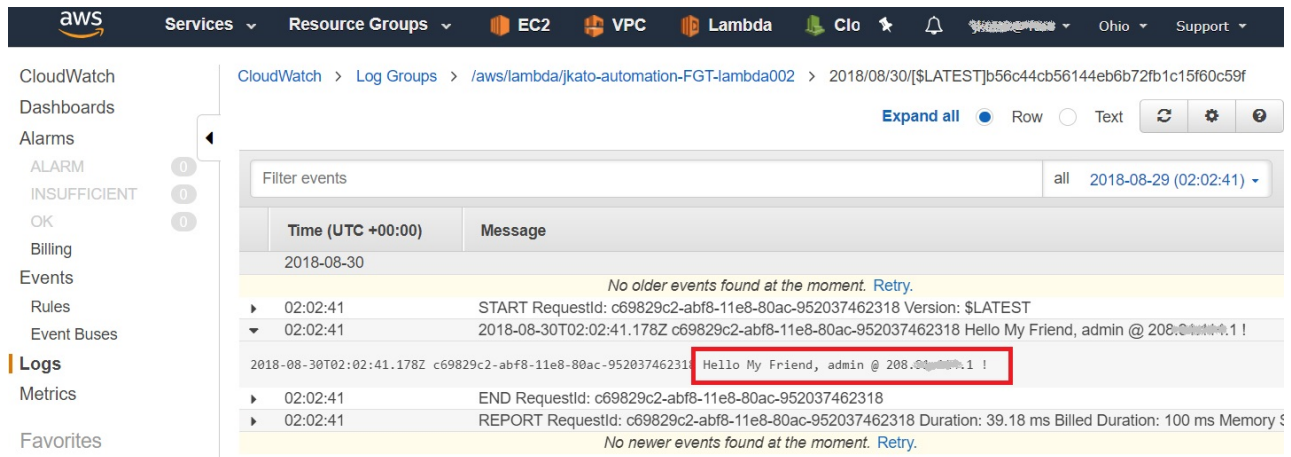
'use strict';
var AWS = require('aws-sdk');
exports.handler = function(event, context, callback) {
  let body = JSON.parse(event.body);
  var usr = body.data.rawlog.user;
  var sourceip = body.data.rawlog.srcip;
  // Write your automation scripts below
  // .... Actions ....
  console.log('Hello My Friend, ', usr, '@', sourceip, '!');
  callback();
};

```

This is what the Lambda script will look like:



11. Save the script.
12. Log out of the FortiGate, then log in again as an administrator. This triggers the event log. The Lambda code is invoked, and CloudWatch Log shows something like the following:



## Configuring FortiGate-VM load balancer using dynamic address objects

FortiOS supports using dynamic firewall addresses in real servers under a virtual server load balancing configuration. Combined with support for the autoscaling group filter (see [Access key-based SDN connector integration on page 167](#)), this enables you to use the FortiGate as a load balancer in AWS for an autoscaling deployment. You do not need to manually change each server's IP address whenever a scale in/out action occurs, as FortiOS dynamically updates the IP addresses following each scale in/out action.

Consider a scenario where the FortiGate-VM is deployed on AWS and load balancing for three servers. The SDN connector configured in FortiOS dynamically loads the server IP addresses. If a scale in action occurs, the load balancer dynamically updates to load balance to the two remaining servers.

The following instructions assume the following:

1. An AWS SDN connector is configured and up.
2. An AWS dynamic firewall address with a filter is configured.

### To configure a dynamic address object in a real server under virtual server load balance:

CLI commands introduced in FortiOS 7.4 are shown bolded.

```
config firewall vip
edit "0"
set id 0
set uuid 0949dfbe-7512-51ea-4671-d3a706b09657
set comment ''
set type server-load-balance
set extip 0.0.0.0
set extintf "port1"
set arp-reply enable
set server-type http
set nat-source-vip disable
set gratuitous-arp-interval 0
set http-ip-header disable
set color 0
set ldb-method static
```



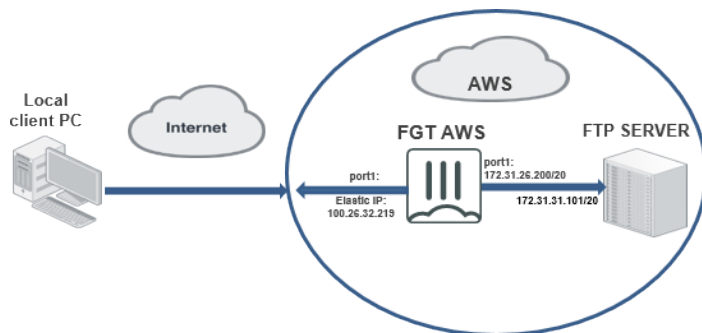
```

set http-redirect disable
set persistence none
set extport 80
config realservers
  edit 1
    set type address
    set address "aws addresses"
    set port 8080
    set status active
    set holddown-interval 300
    set healthcheck vip
    set max-connections 0
    unset client-ip
  next
end
set http-multiplex disable
set max-embryonic-connections 1000
next
end

```

## Accessing a cloud server using an SDN connector via VPN

This guide provides a sample configuration that allows a local client PC to access an FTP server deployed inside the AWS cloud by using an AWS SDN connector via SSL VPN.



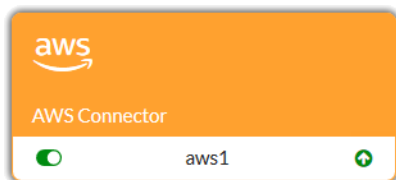
In this topology, a FortiGate-VM for AWS is deployed inside the AWS cloud. The FortiGate-VM can dynamically resolve the FTP server's private IP address in the AWS cloud through an AWS SDN connector. A local client PC with FortiClient installed can establish an SSL VPN tunnel to the FortiGate-VM inside the AWS cloud, then access the FTP server through the SSL VPN tunnel.

### To configure the FortiGate-VM:

1. Configure the AWS SDN connector:
  - a. In FortiOS, go to *Security Fabric > Fabric Connectors*.
  - b. Click *Create New*.
  - c. Select *Amazon Web Services (AWS)*.
  - d. In the *AWS region name* field, enter *us-east-1*.
  - e. Leave the *AWS VPC ID* field blank if no VPC ID is specified.
  - f. Configure other fields as required. Click *OK*.



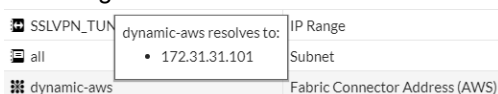
- g. Go to *Security Fabric > Fabric Connectors*. Click the refresh icon for the configured connector. The green arrow means that the connector is connected.



2. Create an SDN connector firewall address to associate the configured SDN connector:
  - a. Go to *Policy & Objects > Addresses*.
  - b. Click *Create New*, then select *Address*.
  - c. From the *Type* dropdown list, select *Fabric Connector Address*.
  - d. From the *SDN Connector* dropdown list, select the connector created in step 1.
  - e. For *SDN address type*, select *Private*.
  - f. In the *Filter* field, enter *Tag.Name=publicftp*. This is the name of the FTP server in the AWS cloud.
  - g. From the *Interface* dropdown list, select *any*.
  - h. Click *OK*. The following shows the FTP server as seen in the AWS management console.

| Name      | Instance ID         | Instance | Availability | Instance | Status Checks     | IPv4 Public IP | Private IP Address |
|-----------|---------------------|----------|--------------|----------|-------------------|----------------|--------------------|
| publicftp | i-0fe5a1ef16bb94796 | t2.micro | us-east-1c   | running  | 2/2 checks passed | 54.210.36.196  | 172.31.31.101      |

3. After the update interval (60 seconds by default), check the resolved firewall address:
  - a. Go to *Policy & Objects > Addresses*.
  - b. Hover over the address created in step 2. In this example, it shows the firewall address (172.31.31.101) that the configured SDN connector resolves to.



4. Configure SSL VPN to access the FTP server:
  - a. Configure the user and user group:
    - i. Go to *User & Device > User Definition*.
    - ii. Create a new local user.
    - iii. Go to *User & Device > User Groups*.
    - iv. Create a group that includes the new local user.
  - b. Configure SSL VPN settings:
    - i. Go to *VPN > SSL-VPN Settings*.
    - ii. In the *Listen on Interface* field, select the proper interface. This example selects port1.
    - iii. In the *Listen on Port* field, enter 10443.
    - iv. From the *Server Certificate* dropdown list, select the desired certificate.



Self-signed certificates are provided by default to simplify initial installation and testing. Acquiring a signed certificate for your installation is **HIGHLY** recommended.

Continuing to use these certificates can result in your connection being compromised, allowing attackers to steal your information, such as credit card details.

For more information, review [Use a non-factory SSL certificate for the SSL VPN portal](#) and learn how to [Purchase and import a signed SSL certificate](#).

- v. Under *Authentication/Port Mapping*, set the default full-access portal for *All Other Users/Groups*.
- vi. Create a new authentication/portal mapping for the group created in step a, mapping to the full-access portal.
- c. Configure the SSL VPN firewall policy:
  - i. Go to *Policy & Objects > IPv4 Policy*.
  - ii. From the *Incoming Interface* dropdown list, select the SSL VPN tunnel interface (ssl.root).
  - iii. From the *Outgoing Interface* dropdown list, select *port1*.
  - iv. In the *Source* field, select *all* and the group configured in step a.
  - v. In the *Destination* field, select the address created in step 2.
  - vi. From the *Schedule* dropdown list, select *always*.
  - vii. In the *Service* field, select *ALL*.
  - viii. For *Action*, select *Accept*.
  - ix. Click *OK*.

### To establish an SSL VPN connection from the local client PC:

This example assumes that you are not using EMS to manage endpoints. If you are using EMS, use a licensed FortiClient endpoint for the following configuration, skipping the installation step.

1. Download VPN-only FortiClient from [FortiClient.com](https://fortinet.com). Install onto the local client PC.
2. In FortiClient, on the *Remote Access* tab, add a new connection.
3. For *VPN*, select *SSL-VPN*.
4. In the *Remote Gateway* field, enter the IP address of the listening FortiGate interface. In this example, it is 100.26.32.219, the FortiGate-VM port1 public IP address.
5. Select *Customize port*, then enter 10443.
6. Save the configuration.
7. Use the credentials configured in step 4a above to connect to the SSL VPN tunnel. After connection, traffic to the SDN connector resolved IP address (172.31.31.101) goes through the tunnel. Other traffic goes through the local gateway. The client PC side shows the routing entry for the SSL VPN tunnel:

| Destination          | Gateway               | Genmask                | Flags      | Metric   | Ref      | Use      | Iface       |
|----------------------|-----------------------|------------------------|------------|----------|----------|----------|-------------|
| 0.0.0.0              | 172.16.200.1          | 0.0.0.0                | UG         | 0        | 0        | 0        | eth1        |
| <b>172.31.31.101</b> | <b>10.212.134.200</b> | <b>255.255.255.255</b> | <b>UGH</b> | <b>0</b> | <b>0</b> | <b>0</b> | <b>ppp0</b> |

The FortiGate-VM shows the logged in user and the assigned SSL VPN tunnel virtual IP address.

|          |                     |               |                        |
|----------|---------------------|---------------|------------------------|
| Refresh  |                     | FGT-AWS-3     |                        |
| Username | Last Login          | Remote Host   | Active Connections     |
| usera    | 2019/04/04 15:42:22 | 208.91.115.10 | Tunnel: 10.212.134.200 |

```
execute vpn sslvpn list
SSL VPN Login Users:
  Index  User   Auth Type  Timeout  From      HTTP in/out  HTTPS
in/out
  0      usera  1(1)      284      208.91.115.10  0/0          0/0
SSL VPN sessions:
  Index  User   Source IP  Duration  I/O Bytes  Tunnel/Dest IP
  0      usera  208.91.115.10  76      1883/1728  10.212.134.200
```

**To run diagnose commands:**

1. To show SDN connector status, run the `diagnose sys sdn status` command. The output should be as follows:

| SDN Connector | Type | Status    |
|---------------|------|-----------|
| aws1          | aws  | connected |

2. To debug the SDN connector to resolve the firewall address, run the `diagnose debug application awsd -1` command. The output should be as follows:

```
...
awsd checking firewall address object dynamic-aws, vd 0
address change, new ip list:
  172.31.31.101
awsd sdn connector aws1 finish updating IP addresses
...
```

3. To restart the AWS SDN connector daemon, run the `diagnose test application awsd 99` command.

## SDN connector support for AWS STS

This enhancement enables the AWS SDN connector to use the AWS security token service (STS) API to connect to multiple AWS accounts concurrently. This allows a single AWS SDN connector to retrieve dynamic objects from multiple accounts, instead of needing to create an SDN connector for each account. This is especially useful for large organizations who may have hundreds of AWS accounts and require seamless integration.

For FortiOS 7.2.1 and later versions, the SDN connector supports using external ID, which allows the target account owner to permit the source account to assume the role only under specific circumstances. This enhances security. See [How to use an external ID when granting access to your AWS resources to a third party](#) for more details.

The example demonstrates a source account, the AWS account that FortiOS is connected to, accessing a target account. The target account must explicitly allow an external ID string in its role definition. The role definition has a trust policy that allows the source account on the condition that it connects with the specified external ID. You can configure these definitions on the target account in AWS.

This example uses two AWS accounts:

- **Target account:** 601xxxxxx685
- **Source account:** 269xxxxxx203

**To configure SDN connector support for AWS STS:**

1. Log in to the AWS console using the target account.
2. Create an Identity > Access Management (IAM) role on the target account:
  - a. Go to *IAM > Roles > Create role > Another AWS account*.
  - b. In the *Account ID* field, enter the source account. In this example, the source account is 269xxxxxx203.
  - c. Enable *Require external ID (Best practice when a third party will assume this role)*.
  - d. In the *External ID* field, enter the desired external ID. In this example, the external ID is external-id-demo-123456.
  - e. Click *Next*.

- f. Continue with the configuration until the *Review* step. In the *Role name* field, enter the desired role name. In this example, the role name is CrossAccountSTS.
3. Create an inline policy on the target account:
  - a. Go to *IAM > Roles*.
  - b. Select the role that you created.
  - c. Click *Add inline policy > JSON*.
  - d. Paste the following in to the text box:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeRegions",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeVpcEndpoints"
      ],
      "Resource": "*"
    }
  ]
}
```



- e. Continue to create the policy. Name the policy as desired. In this example, the policy name is CrossAccountPolicy.



You can also create a standalone policy in *IAM > Policies*, and attach the policy to the IAM role, instead of adding an inline policy as this procedure describes.

4. Log in to the AWS console using the source account.

5. Create an IAM role on the source account:
  - a. Go to *IAM > Roles > Create role > AWS service > EC2*.
  - b. Under *Permissions*, configure the desired permissions. In this example, this role is configured with *AmazonEC2FullAccess*.
  - c. Click *Next*.
  - d. Continue with the configuration until the *Review* step. In the *Role name* field, enter the desired role name.
6. Create an inline policy on the source account:
  - a. Go to *IAM > Roles*.
  - b. Select the role that you created.
  - c. Click *Add inline policy > JSON*.
  - d. Paste the following in to the text box:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sts:AssumeRole"
      ],
      "Resource": [
        "arn:aws:iam::601xxxxxx685:role/CrossAccountSTS"
      ]
    }
  ]
}
```

- e. Continue to create the policy. Name the policy as desired. The resource should be the Amazon resource name (ARN) of the IAM role that you created in the target account. You can find the ARN by logging in to the AWS portal under the target account and going to the IAM web portal.



You can also create a standalone policy in *IAM > Policies*, and attach the policy to the IAM role, instead of adding an inline policy as this procedure describes.

---

7. Launch a FortiGate-VM under the source account.
8. Assign the IAM role that you created in step 5 to the FortiGate-VM.
9. Configure FortiOS:
  - a. Configure the AWS SDN connector to be able to access the target account:

```
config system sdn-connector
  edit "aws1"
    config external-account-list
      edit "arn:aws:iam::601xxxxxx685:role/cross-account-with-external-id-demo"
        set external-id "external-id-demo-123456"
        set region-list "us-east-1"
      next
    end
```

```

    next
end

```



The `use-metadata-iam`, `access-key`, and `secret-key` properties are only for STS credential setup when `external-account-list` is enabled.

To retrieve an IP address in the source AWS account, you must also add the source account to the `external-account-list` property.

- b. Configure a dynamic address. This address checks whether the FortiGate-VM can retrieve the instance address in the target account:

```

config firewall address
    edit "sdn1"
        set type dynamic
        set sdn "aws1"
        set filter "InstanceId=i-02c5141c75e6aed4f"
    next
end

```

- c. Confirm that the FortiGate-VM can retrieve the dynamic IP address from the target account:

```

show firewall address sdn1
config firewall address
    edit "sdn1"
        set type dynamic
        set sdn "aws1"
        set filter "InstanceId=i-02c5141c75e6aed4f"
        set sdn-addr-type all
    config list
        edit "172.31.24.149"
        next
        edit "54.172.135.95"
        next
    end
    next
end

```

# VPN for FortiGate-VM on AWS

## Connecting a local FortiGate to an AWS VPC VPN

This recipe provides sample configuration of a site-to-site VPN connection from a local FortiGate to an AWS VPC VPN via IPsec with static routing.

Instances that you launch into an Amazon VPC can communicate with your own remote network via a site-to-site VPN between your on-premise FortiGate and AWS VPC VPN. You can enable access to your remote network from your VPC by configuring a virtual private gateway (VPG) and customer gateway to the VPC, then configuring the site-to-site VPC VPN.

The following prerequisites must be met for this configuration:

- An AWS VPC with some configured subnets, routing tables, security group rules, and so on
- An on-premise FortiGate with an external IP address

This recipe consists of the following steps:

1. [Create a VPG.](#)
2. [Create a customer gateway.](#)
3. [Create a site-to-site VPN connection on AWS.](#)
4. [Configure the on-premise FortiGate.](#)

### To create a VPG:

A VPG is the VPN concentrator on the Amazon side of the site-to-site VPN connection. You can create a VPG and attach it to the VPC from which you want to create the site-to-site VPN connection.

1. In the AWS management console, go to *Virtual Private Gateways*, then click *Create Virtual Private Gateway*.
2. In the *Name tag* field, enter the desired gateway name.
3. For static route configuration, the ASN is not important, as the ASN is for BGP routing. By default, the VPG is created with the default ASN, 64512. You cannot change the ASN once the VPG has been created.
4. After creating the VPG, select it from the list of VPGs, and click *Actions > Attach to VPC*.
5. On the *Attach to VPC* page, select the ID for the desired VPC from the *VPC* dropdown list.

### To create a customer gateway:

In this example, the customer gateway refers to the on-premise FortiGate for the VPC VPN to connect to.

1. Go to *Customer Gateways*, then click *Create Customer Gateway*.
2. In the *Name* field, enter the desired gateway name.
3. For *Routing*, select *Static*.
4. In the *IP Address* field, enter the on-premise FortiGate's external address.

### To create a site-to-site VPN connection on AWS:

AWS VPC VPN supports the following:

- Internet Key Exchange version 2 (IKEv2)
- NAT traversal
- Four-byte ASN (in addition to two-byte ASN)
- Reusable IP addresses for customer gateways
- Additional encryption options including AES 256-bit encryption, SHA-2 hashing, and additional Diffie-Hellman groups
- Configurable tunnel options
- Custom private ASN for the Amazon side of a BGP session

This example describes creating an IPsec site-to-site VPN.

1. Go to *VPN Connections*, then click *Create VPN Connection*.
2. In the *Name tag* field, enter the desired VPN connection name.
3. From the *Virtual Private Gateway* dropdown list, select the VPG ID for the VPG created earlier.
4. For *Routing Options*, select *Static*.
5. In the *IP Prefixes* field, enter the CIDR of the networks behind your on-premise FortiGate.
6. Leave the tunnel options blank. You will obtain this information from a configuration file download.

### To configure the on-premise FortiGate:

1. After creating the VPN, select it in the VPN list, then click *Download Configuration*. This document contains information needed to configure the FortiGate correctly.
2. You can configure the FortiGate using this downloaded configuration file. The example FortiGate has port1 with an external IP address of 35.188.119.246 and an internal IP address of 10.6.30.2/24. Port2 has an internal IP address of 10.1.100.3/24. The downloaded configuration file resembles the following. The most important information here is the `remote-gw` value, which in this case is 3.95.86.157, and the `psksecret` value.

```
! -----
! IPsec Tunnel #1
! -----
! #1: Internet Key Exchange (IKE) Configuration
!
! A policy is established for the supported ISAKMP encryption,
! authentication, Diffie-Hellman, lifetime, and key parameters.
! Please note, these sample configurations are for the minimum requirement of AES128, SHA1, and DH Group 2.
! Category "VPN" connections in the GovCloud region have a minimum requirement of AES128, SHA2, and DH Group 14.
! You will need to modify these sample configuration files to take advantage of AES256, SHA256, or other DH groups like 2, 14-18, 22, 23, and 24.
! Higher parameters are only available for VPNs of category "VPN," and not for "VPN-Classic".
!
!
! The address of the external interface for your customer gateway must be a static address.
! Your customer gateway may reside behind a device performing network address translation (NAT).
! To ensure that NAT traversal (NAT-T) can function, you must adjust your firewall rules to unblock UDP port 4500. If not behind NAT, we recommend disabling NAT-T.
!
! Configuration begins in root VDOM.

config vpn ipsec phase1-interface
edit vpn-032037fa39969e15b-0 ! Name must be shorter than 15 chars, best if shorter than 12
set interface "wan1"

! The IPsec Dead Peer Detection causes periodic messages to be
! sent to ensure a Security Association remains operational

set dpd enable
set local-gw 35.188.119.246
set dhgrp 2
set proposal aes128-shal
set keylife 28800
set remote-gw 3.95.86.157
set psksecret N1IIFTQOfiVuRWkQui_A5IjNT_4lVTcP
set dpd-retryinterval 10
next
end
```

Run the following commands in the FortiOS CLI to configure the FortiGate, using the `remote-gw` and `psksecret` values from the downloaded configuration file as shown. When setting the destination for the static route, use the VPC's IPv4 CIDR:

```
config vpn ipsec phase1-interface
edit "examplephase1"
set interface "port1"
set keylife 28800
set peertype any
```



```

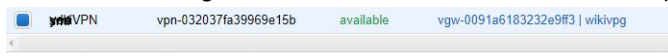
        set proposal aes128-sha1
        set dhgrp 2
        set remote-gw 3.95.86.157
        set psksecret NlITFTQJfiVuRWkQui_A5IjNT_4lVTtP
        set dpd-retryinterval 10
    next
end
config vpn ipsec phase2-interface
    edit "examplephase2"
        set phase1name "examplephase1"
        set proposal aes128-sha1
        set dhgrp 2
        set keylifeseconds 3600
    next
end
config router static
    edit 1
        set dst 10.0.0.0 255.255.0.0
        set device "examplephase1"
    next
end
config firewall policy
    edit 1
        set srcintf "examplephase1"
        set dstintf "port2"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set srcintf "port2"
        set dstintf "examplephase1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end

```

3. Run the `diagnose vpn tunnel up examplephase2` command if the tunnel is not up automatically already.
4. Check in the FortiOS GUI in **VPN > IPsec Tunnels** that the tunnel is up.

| <div> <div>+ Create New</div> <div>Edit</div> <div>Delete</div> <div>Print Instructions</div> <div>Search</div> <div>Q</div> </div> |                   |        |   |
|---|-------------------|--------|---|
| Tunnel  | Interface Binding | Status |   |
| Custom 1  |                   |        |   |
| examplephase1   | port1             | Up     | 4 |

5. In the AWS management console, check that the tunnel is up:



VPN Connection: vpn-032037fa39969e15b

Details Tunnel Details Static Routes Tags

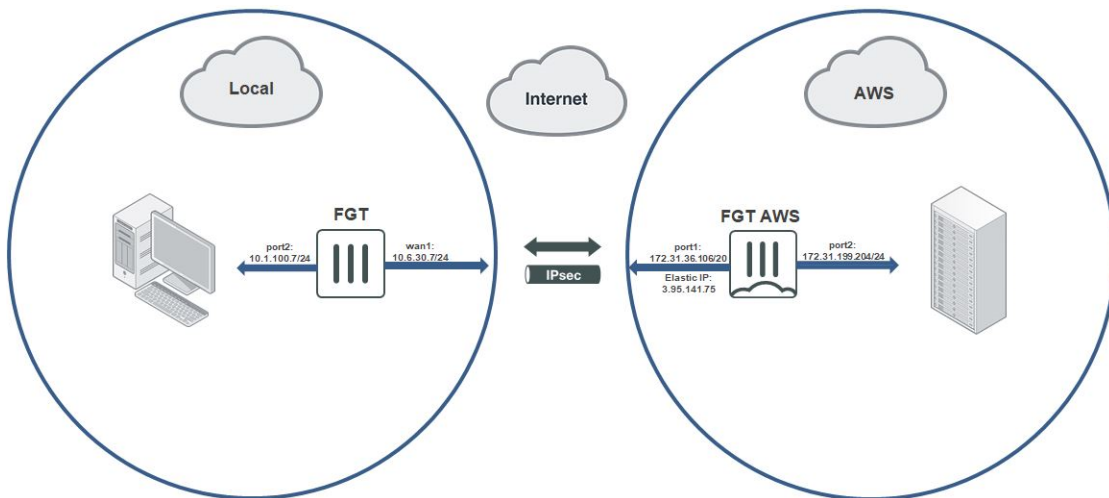
| Outside IP Address | Inside IP CIDR   | Status | Status Last Changed                   |
|--------------------|------------------|--------|---------------------------------------|
| 3.95.86.157        | 169.254.45.88/30 | UP     | February 15, 2019 at 6:11:14 PM UT... |
| 54.175.28.83       | 169.254.47.44/30 | DOWN   | February 15, 2019 at 5:17:42 PM UT... |

6. After the tunnel is up, you must edit a custom route table and security group rules to achieve connectivity between a resource behind the FortiGate to a resource on the AWS cloud.
7. On AWS, there are two tunnels for each created VPN. This example only shows connecting to one tunnel, but you can create the second tunnel in FortiOS as well. The second tunnel is for redundancy. If one tunnel goes down, the FortiGate can reach AWS resources using the other tunnel.

## Connecting a local FortiGate to an AWS FortiGate via site-to-site VPN

This guide provides sample configuration of a site-to-site VPN connection from a local FortiGate to an AWS FortiGate via site-to-site IPsec VPN with static routing. You can access resources that are protected behind a FortiGate on AWS from your local environment by using a site-to-site VPN.

The following depicts the network topology for this sample deployment:



The following prerequisites must be met for this configuration:

- A FortiGate located on AWS with some resources behind it. In this example, the AWS FortiGate has port1 connected to WAN and port2 connected to local LAN.
- An on-premise FortiGate. For your local environment, determine if your FortiGate has a publicly accessible IP address or if it is behind NAT. In this example, the on-premise FortiGate is behind NAT.

This recipe consists of the following steps:

1. [Create a VPN on the local FortiGate to the AWS FortiGate.](#)

2. [Create a VPN on the AWS FortiGate to the local FortiGate.](#)
3. [Establish a connection between the FortiGates.](#)

**To create a VPN on the local FortiGate to the AWS FortiGate:**

1. In FortiOS on the local FortiGate, go to *VPN > IPsec Wizard*.
2. On the *VPN Setup* tab, configure the following:
  - a. In the *Name* field, enter the desired name.
  - b. For *Template Type*, select *Site to Site*.
  - c. For *Remote Device Type*, select *FortiGate*.
  - d. For *NAT Configuration*, select the appropriate option. In this example, since the local FortiGate is behind NAT, *This site is behind NAT* is selected. Click *Next*. For non-dialup situations where the local FortiGate has an external IP address, select *No NAT between sites*.
3. On the *Authentication* tab, configure the following:
  - a. For *Remote Device*, select *IP Address*.
  - b. In the *IP Address* field, enter the AWS FortiGate's elastic IP address. In this example, it is 3.95.141.75.
  - c. For *Outgoing Interface*, allow FortiOS to detect the interface via routing lookup.
  - d. For *Authentication Method*, select *Pre-shared Key*.
  - e. In the *Pre-shared Key* field, enter the desired key. Click *Next*.
4. On the *Policy & Routing* tab, configure the following:
  - a. For *Local Interface*, select the desired local interface. In this example, port2 is selected. The *Local Subnets* field should then auto-populate.
  - b. In the *Remote Subnets* field, enter the remote subnet on the other side of the AWS FortiGate. In this example, it is 172.31.199.0/24.
  - c. For *Internet Access*, select *None*.
5. Click *Create*. The IPsec Wizard creates the following:
  - a. Firewall addresses for local and remote subnets
  - b. Firewall address groups containing the above firewall addresses
  - c. phase-1 and phase-2 interfaces
  - d. Static route and blackhole route
  - e. Two firewall policies: one for traffic to the tunnel interface and one for traffic from the tunnel interface

**To create a VPN on the AWS FortiGate to the local FortiGate:**

1. In FortiOS on the AWS FortiGate, go to *VPN > IPsec Wizard*.
2. On the *VPN Setup* tab, configure the following:
  - a. In the *Name* field, enter the desired name.
  - b. For *Template Type*, select *Site to Site*.
  - c. For *Remote Device Type*, select *FortiGate*.
  - d. For *NAT Configuration*, select *This site is behind NAT*. This is the correct configuration since the AWS FortiGate has an elastic IP address. Click *Next*.
3. On the *Authentication* tab, configure the following:
  - a. For *Incoming Interface*, select the WAN-facing incoming interface. In this example, it is port1.
  - b. For *Authentication Method*, select *Pre-shared Key*.
  - c. In the *Pre-shared Key* field, enter the same key configured on the local FortiGate. Click *Next*.

4. On the *Policy & Routing* tab, configure the following:
  - a. For *Local Interface*, select the desired local interface. In this example, port2 is selected. The *Local Subnets* field should then auto-populate.
  - b. In the *Remote Subnets* field, enter the remote subnet on the other side of the local FortiGate. In this example, it is 10.1.100.0/24.
  - c. For *Internet Access*, select *None*.
5. Click *Create*. The IPsec Wizard creates the following:
  - a. Firewall addresses for local and remote subnets
  - b. Firewall address groups containing the above firewall addresses
  - c. phase-1 and phase-2 interfaces
  - d. Static route and blackhole route
  - e. Two firewall policies: one for traffic to the tunnel interface and one for traffic from the tunnel interface

#### To establish a connection between the FortiGates:

1. The tunnels are down until you initiate a connection from the local FortiGate to the AWS FortiGate. In FortiOS on the local FortiGate, go to *Monitor > IPsec Monitor*.
2. Right-click the phase-2 interface, and select *Bring Up*.
3. In FortiOS on the AWS FortiGate, go to *Monitor > IPsec Monitor* and verify that the connection is up.

| FortiGate VM64-AWS FGVM080000168945 |   |           |                    |                |            |               |                   |
|-------------------------------------|---|-----------|--------------------|----------------|------------|---------------|-------------------|
| Dashboard                           | > | Refresh   | Reset Statistics   | Bring Up       | Bring Down |               |                   |
| Security Fabric                     | > | Name      | Type               | Remote Gateway | Peer ID    | Incoming Data | Outgoing Data     |
| FortiView                           | > | toLOCAL_1 | Dialup - FortiGate | 208.91.114.1   |            | 0 B           | 0 B               |
| Network                             | > |           |                    |                |            | Phase 1       | Phase 2 Selectors |
|                                     |   |           |                    |                |            | toLOCAL       | toLOCAL           |



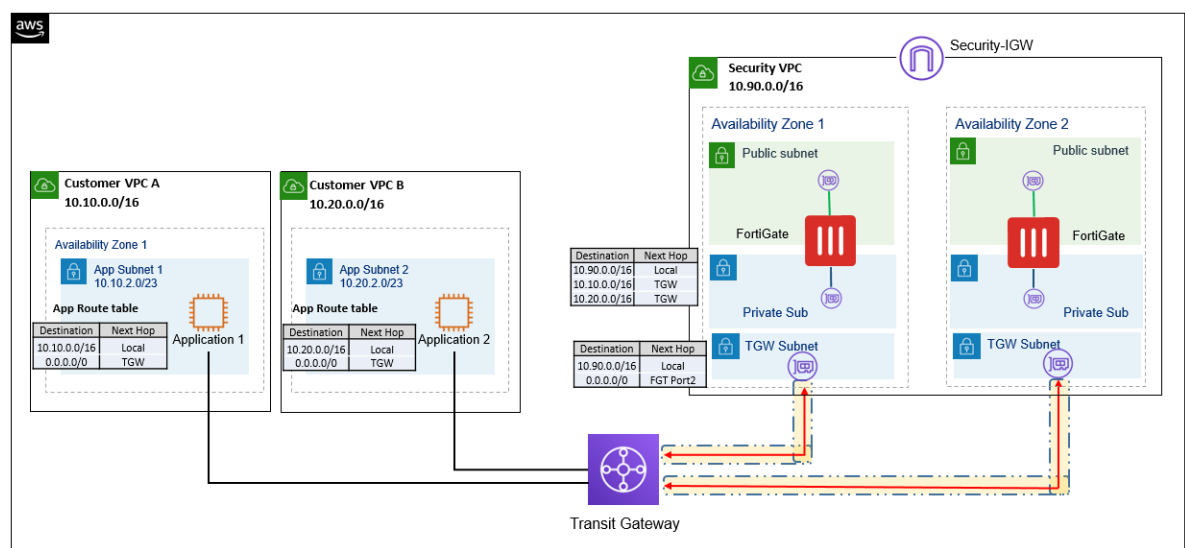
The elastic IP address can be considered as one to one to the FortiGate's IP address, even though the port IP address may be an internal IP address.

# SD-WAN

## SD-WAN cloud on-ramp

See [SD-WAN cloud on-ramp](#).

## SD-WAN Transit Gateway Connect



This guide assumes that the customer and security virtual private clouds (VPC) and the FortiGate instances that the diagram shows are already in place and application instances are already created. This guide does not cover the steps for creating those resources.

| VPC      | Description  |
|----------|--|
| Customer | Where the customer workloads will be deployed. Each availability zone (AZ) has an Application subnet, where the application workloads are deployed. This VPC does not have an Internet gateway and all North-South traffic is routed through the FortiGate instances in the Security subnet via the Transit Gateway (TGW). |
| Security | Where FortiGates are deployed. All North-South traffic is routed through the FortiGate. This routing is achieved by the following: <ul style="list-style-type: none"><li>• Sharing BGP routes using the TGW Connect attachment</li><li>• Configuring BGP connect peers between the FortiGate and the TGW</li></ul>         |

## Creating the TGW and related resources

A transit gateway (TGW) is a transit hub that connects two virtual private clouds (VPC) or a VPC to an on-premise network. This scenario connects multiple Application VPCs to the Security VPC via a TGW. This ensures that any access to and from the Application VPC is routed via the Security VPC, where the FortiGates can inspect it.

### To create the TGW and related resources:

#### 1. Create the TGW:

- a. In the AWS management console, go to *VPC Dashboard > Transit Gateways > Transit Gateways*.
- b. Click *Create Transit Gateway*.
- c. Configure the *Name tag* and *Amazon side ASN* fields with the desired values.
- d. Enable *DNS support*, *VPN ECMP support*, *Default route table association*, and *Default route table propagation*.
- e. In the *CIDR* field, enter the desired CIDR. In this example, the value is 10.100.0/16.

Create Transit Gateway

A Transit Gateway (TGW) is a network transit hub that interconnects attachments (VPCs and VPNs) within the same account or across accounts.

Name tag  ⓘ

Description  ⓘ

Configure the Transit Gateway

Amazon side ASN  ⓘ

DNS support ☒ enable ⓘ

VPN ECMP support ☒ enable ⓘ

Default route table association ☒ enable ⓘ

Default route table propagation ☒ enable ⓘ

Multicast support ☐ enable ⓘ

Configure sharing options for cross account

Auto accept shared attachments ☐ enable ⓘ

Transit Gateway CIDR blocks

Add or remove IPv4 or IPv6 CIDR blocks for your Transit Gateway. [Learn more.](#)

CIDR ⓘ

ⓘ

Add CIDR

2. Creating a TGW creates a TGW default route table. You can use this table as the default association and propagation route table for the TGW. Confirm that the table was created by going to *VPC Dashboard > Transit Gateways > Transit Gateway Route Tables*.
3. You must create a TGW attachment to link separate VPCs and subnets to the newly created TGW. The two resources can belong to the same or different AWS accounts. This example assumes that both VPCs are in the same AWS account. Create the TGW attachment:
  - a. Go to *VPC Dashboard > Transit Gateways > Transit Gateway Attachments*.
  - b. Click *Create Transit Gateway Attachment*.
  - c. From the *Transit Gateway ID* dropdown list, select the TGW you created in step 1.
  - d. From the *Attachment type* dropdown list, select *VPC*.
  - e. From the *VPC ID* dropdown list, select the VPC to attach to the TGW.
  - f. In the *Subnet IDs* field, select the required subnet in the correct availability zone (AZ).

- g. Configure other fields as desired.
  - h. Click *Create Attachment*.
4. Repeat step 3 for the remaining three attachments. For the Security VPC attachment, ensure that you select the TGW subnet in both AZs.
  5. You must create a TGW connect attachment to help form a Generic Routing Encapsulation (GRE) tunnel on top of a VPC attachments. You then create Border Gateway Protocol (BGP) peers for each FortiGate in each AZ. By BGP peering the TGW and the FortiGate, you can send Customer VPC routes to the FortiGates in the Security VPC. Create the TGW connect attachment:
    - a. Go to *VPC Dashboard > Transit Gateways > Transit Gateway Attachments*.
    - b. Click *Create Transit Gateway Attachment*.
    - c. From the *Transit Gateway ID* dropdown list, select the TGW you created in step 1.
    - d. From the *Attachment type* dropdown list, select *Connect*.
    - e. Configure the *Attachment name tag* field as desired.
    - f. From the *Transport Attachment ID* dropdown list, select the TGW attachment over which to create the Connect attachment. This example selects the Security VPC attachment.
    - g. Click *Create Attachment*.
  6. You must add peering connections for each AZ. Create the peers:
    - a. Go to *VPC Dashboard > Transit Gateways > Transit Gateway Attachments*.
    - b. Select the newly created connect attachment.
    - c. On the *Connect peers* tab, click *Create Connect peer*.
    - d. In the *Peer GRE address* field, enter the FortiGate port 2 IP address. You must create new connect peers for FortiGates in other AZs.
    - e. In the *BGP Inside CIDR blocks IPv4* field, configure a unique /29 block in the 169.254.x.x /16 CIDR range for each connect peer.
    - f. In the *BGP Inside CIDR blocks IPv6* field, configure a unique /125 block in the fd00: : /8 CIDR range for each connect peer if applicable.
    - g. In the *Peer ASN* field, enter an existing ASN assigned in the network, or assign a private ASN in the range 64512-65534. This setup uses eBGP and the peer ASN must differ from the AWS default.
    - h. Click *Create*.

## Configuring BGP

### To configure BGP:

1. Configure the generic routing encapsulation (GRE) interface in the FortiOS CLI on both FortiGates. Configuring a GRE tunnel interface enables you to form a GRE tunnel between the FortiGate and the transit gateway (TGW) to exchange border gateway protocol (BGP) routes:

```
config system gre-tunnel
  edit "tgwc"
    set interface "port2"
    set remote-gw <TGW GRE address>
    set local-gw <FortiGate port2 IP address>
  next
end
```

You can find the TGW GRE address on the virtual private cloud (VPC) Dashboard in *Transit Gateways > Transit Gateway Attachments* in the AWS management console. Select the Transit Gateway Connect attachment, then the *Connect peers* tab. The remote gateway IP address is unique for each connect peer. The following shows commands for the first FortiGate in the example scenario:

```

config system gre-tunnel
  edit "tgwc"
    set interface "port2"
    set remote-gw 10.100.0.32
    set local-gw 10.90.208.174
  next
end

```

The following shows commands for the second FortiGate in the example scenario:

```

config system gre-tunnel
  edit "tgwc"
    set interface "port2"
    set remote-gw 10.100.0.236
    set local-gw 10.90.46.172
  next
end

```

**2. Configure the tunnel interface IP address:**

- a. Go to *Network > Interfaces* in FortiOS.
- b. Select the newly created GRE interface, then select *Edit*.
- c. In the *IP* field, enter the peer BGP address. You can find this value on the AWS management console in *VPC Dashboard > Transit Gateways > Transit Gateway Attachments*, selecting the Security VPC Connect attachment, and going to the *Connect peers* tab.
- d. In the *Remote IP* field, enter the TGW BGP 1 address. The mask is /29. You can find this value on the AWS management console in *VPC Dashboard > Transit Gateways > Transit Gateways*, selecting the Security VPC Connect attachment, and going to the *Connect peers* tab.

**3. Configure BGP neighbors on both FortiGates.** You can find this value on the AWS management console in *VPC Dashboard > Transit Gateways > Transit Gateway Attachments*, selecting the TGW Connect attachment, and going to the *Connect peers* tab.

| Peer ASN | Peer BGP address | Transit Gateway BGP 1 address | Transit Gateway BGP 1 Status | Transit Gateway BGP 2 address | Transit Gateway BGP 2 Status |
|----------|------------------|-------------------------------|------------------------------|-------------------------------|------------------------------|
| 7116     | 169.254.102.1    | 169.254.102.2                 | UP                           | 169.254.102.3                 | UP                           |
| 7115     | 169.254.120.1    | 169.254.120.2                 | UP                           | 169.254.120.3                 | UP                           |

```

config router bgp
  set as 7115
  config neighbor
    edit "<TGW BGP 1 address>"
      set capability-default-originate enable
      set ebgp-enforce-multihop enable
      set soft-reconfiguration enable
      set remote-as 64512
    next
    edit "<TGWP BGP 2 address>"
      set capability-default-originate enable
      set ebgp-enforce-multihop enable
      set soft-reconfiguration enable
      set remote-as 64512
    next
  end
config network
  edit 1
    set prefix 10.90.32.0 255.255.240.0
  next
end
config redistribute "connected"
end

```



```

config redistribute "rip"
end
config redistribute "ospf"
end
config redistribute "static"
end
config redistribute "isis"
end

```

#### 4. Configure static routes on each FortiGate to forward packets to the TGW subnet.

+ Create New

Edit

Clone

Delete

Search

Q

| Destination   | Gateway IP | Interface | Status  |
|---------------|------------|-----------|---------|
| IPv4 2        |            |           |         |
| 10.90.32.0/20 | 10.90.32.1 | port2     | Enabled |
| 10.100.0.0/16 | 10.90.32.1 | port2     | Enabled |

+ Create New

Edit

Clone

Delete

Search

Q

⚙

| Destination    | Gateway IP  | Interface | Status  |
|----------------|-------------|-----------|---------|
| IPv4 2         |             |           |         |
| 10.90.208.0/20 | 10.90.208.1 | port2     | Enabled |
| 10.100.0.0/16  | 10.90.208.1 | port2     | Enabled |

#### 5. Configure the desired firewall rules on each FortiGate.

+ Create New

Edit

Delete

Policy Lookup

Search

|  | Name          | Source | Destination | Schedule | Service | Action | NAT     | Security Profiles | Log |
|--|---------------|--------|-------------|----------|---------|--------|---------|-------------------|-----|
|  | port2 → port1 |        |             |          |         |        |         |                   |     |
|  | outbound      | all    | all         | always   | ALL     | ACCEPT | Enabled | no-inspection     | UTM |
|  | port2 → tgwc  |        |             |          |         |        |         |                   |     |
|  | totgwc        | all    | all         | always   | ALL     | ACCEPT | Enabled | no-inspection     | UTM |
|  | tgwc → port1  |        |             |          |         |        |         |                   |     |
|  | tgwcoutbound  | all    | all         | always   | ALL     | ACCEPT | Enabled | no-inspection     | UTM |
|  | tgwc → port2  |        |             |          |         |        |         |                   |     |
|  | fromtgwc      | all    | all         | always   | ALL     | ACCEPT | Enabled | no-inspection     | UTM |
|  | Implicit      |        |             |          |         |        |         |                   |     |

## Verifying the configuration

### To verify the configuration:

1. In the FortiOS CLI, enter the following commands to verify the routes received and advertised via BGP between the FortiGate and TGW. See [Technical Tip: How to check BGP advertised and received routes on a FortiGate](#) for details:

```

get router info bgp neighbors <neighbor_IP> received-routes
get router info bgp neighbors <neighbor_IP> advertised-routes

```

In a successful scenario, Customer VPC routes should be visible to the FortiGate via the TGW. You should be able to verify this on both FortiGate instances.

```

FGTAWSLSSUEPZXDC # get router info bgp neighbors 169.254.120.2 received-routes
VRF 0 BGP table version is 4, local router ID is 169.254.120.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network        Next Hop        Metric LocPrf Weight RouteTag Path
*> 10.10.0.0/16    169.254.120.2    100          0          0 64512 i <-/->
*> 10.20.0.0/16    169.254.120.2    100          0          0 64512 i <-/->
*> 10.30.0.0/16    169.254.120.2    100          0          0 64512 i <-/->
*> 10.90.0.0/16    169.254.120.2    100          0          0 64512 i <-/->
*> 10.90.208.0/20  169.254.120.2    100          0          0 64512 7116 i <-/->

Total number of prefixes 5

FGTAWSLSSUEPZXDC # get router info bgp neighbors 169.254.120.2 advertised-routes
VRF 0 BGP table version is 4, local router ID is 169.254.120.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network        Next Hop        Metric LocPrf Weight RouteTag Path
*> 0.0.0.0/0       169.254.120.1    100 32768      0 i <-/->
*> 10.10.0.0/16    169.254.120.1          0          0 64512 i <-/->
*> 10.20.0.0/16    169.254.120.1          0          0 64512 i <-/->
*> 10.30.0.0/16    169.254.120.1          0          0 64512 i <-/->
*> 10.90.0.0/16    169.254.120.1          0          0 64512 i <-/->
*> 10.90.32.0/20   169.254.120.1    100 32768      0 i <-/->
*> 10.90.208.0/20  169.254.120.1          0          0 64512 7116 i <-/->

Total number of prefixes 7

```

2. Verify the TGW BGP status. On the AWS management console, go to *VPC Dashboard > Transit Gateways > Transit Gateway Attachments*. Select the TGW Connect attachment, then go to the *Connect peers* tab. Confirm that the TGW BGP 1 and 2 Status display as UP.
3. Verify the TGW BGP status for both connect peers in the TGW route table. On the AWS management console, go to *VPC Dashboard > Transit Gateways > Transit Gateway Route Tables*. Select the default TGW route table, and go to the Routes tab. You should see several propagated routes with the Connect resource type.

## Cloud WAN

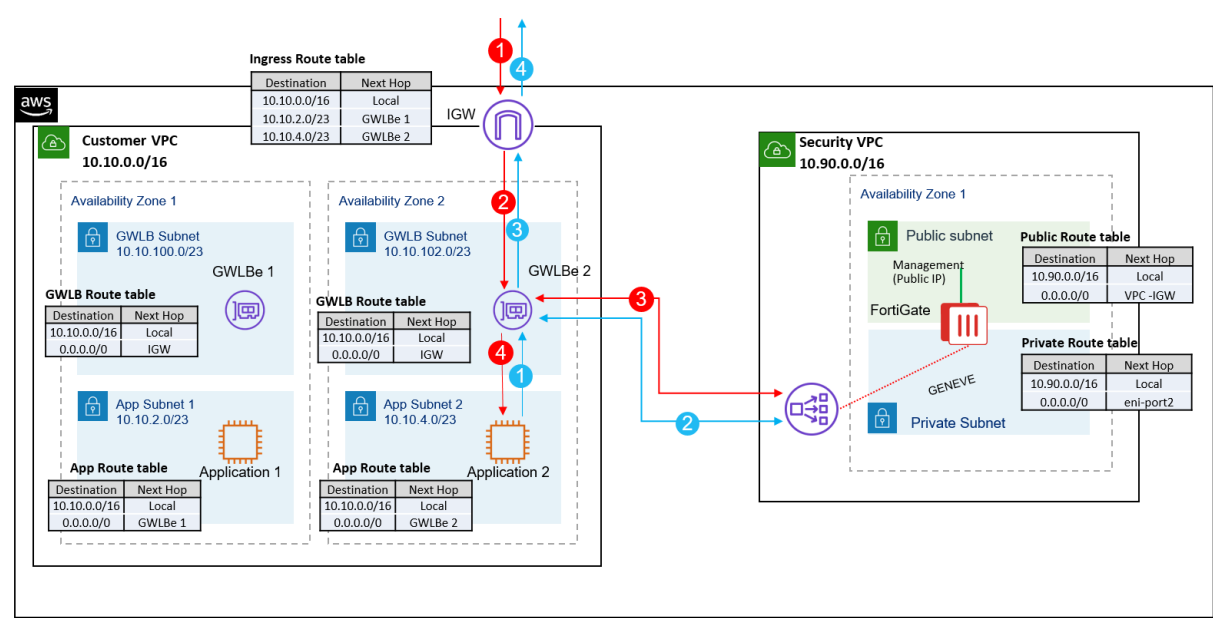
See [Technical Tip: AWS Cloud WAN unleashed with Fortinet SD-WAN](#).

# Security inspection with GWLB integration

The following deployment scenarios describe configuring security inspection with AWS Gateway Load Balancer (GWLB):

[Multitenancy support with AWS GWLB on page 221](#) describes configuring multitenancy support with GWLB integration:

## North-south security inspection to customer VPC



This guide assumes that the following are already created and in place as the diagram shows:

- Customer VPC and subnets in all zones to be load balanced
- Security VPC and subnets in all zones to be load balanced
- FortiGate with at least one management network interface and elastic IP address assigned
- Application instances

The guide describes configuring additional network interfaces to handle data traffic. The following describes the two VPCs in this deployment:

| VPC      | Description  |
|----------|--|
| Customer | Where customer workloads are deployed. The customer VPC has four subnets (two in each availability zone (AZ)). Each AZ has an application-purposed subnet and a GWLB endpoint subnet: <ul style="list-style-type: none"><li>• <b>Application-purposed subnet:</b> deploy application workloads where the</li></ul> |

| VPC      | Description   |
|----------|---|
|          | <p>FortiGate must inspect the traffic.</p> <ul style="list-style-type: none"> <li>• <b>GWLB endpoint subnet:</b> deploy the GWLB endpoint so that traffic is redirected to the GWLB, which then redirects the traffic to the FortiGate for inspection.</li> </ul> |
| Security | Where the FortiGate is deployed. You create the GWLB in this VPC.   |

The following describes the traffic flow in this deployment:

| Traffic flow     | Description   |
|------------------|---|
| Inbound traffic  | With this configuration, the FortiGate inspects traffic that is destined for the application instances. The Internet gateway in the customer VPC is associated with an ingress route table. The route table directs the traffic for the application subnets through the GWLB endpoints (GWLBs) in its dedicated subnets. The traffic then goes through the GWLB in the security VPC, where it is encapsulated with Geneve protocol and sent to the FortiGate. The FortiGate inspects the traffic and redirects it to the application instances. |
| Outbound traffic | The route tables that the application subnets are associated with have a default route through the GWLB endpoints in their AZ. The traffic originating from the application instances is forwarded to the FortiGate through the GWLB. After inspection, the FortiGate sends the traffic to the Internet. You set static routes for all of these traffic redirects after deployment. See <a href="#">Post-deployment configuration on page 210</a> .   |

### To add support for IPv6:

See [New – Gateway Load Balancer support for IPv6](#) for extending a current or new deployment to support IPv6.

The following provides an overview of steps that you must complete:

1. VPCs subnets that the GWLB exists in and that you will deploy the GWLBs to must have IPv6 enabled and a CIDR assigned. This means that your VPC and subnets must have IPv6 enabled before configuring GWLB IPv6 settings.
2. GWLB must be in dual stack mode.
3. Endpoint services must support IPv4 and IPv6.
4. Endpoint must be in dual stack mode.
5. FortiGates are not assigned an IPv6 address as they use IPv4 to send and receive traffic from the GENEVE tunnel.

## Creating the GWLB and registering targets

### To create the GWLB and register targets:

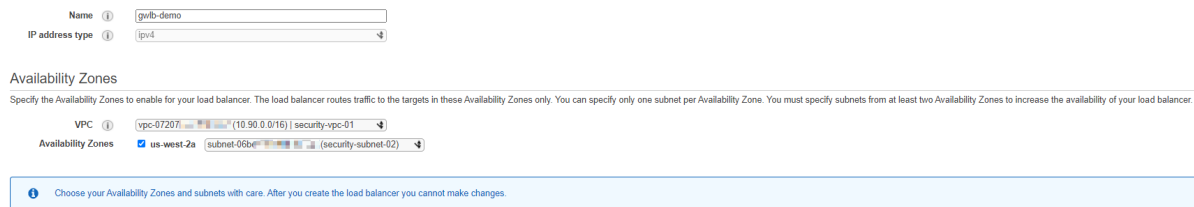
1. Go to *Compute > EC2 Dashboard > Load Balancing > Load Balancers*.
2. Click *Create Load Balancer*, then *Gateway Load Balancer*.
3. Configure the gateway load balancer (GWLB):
  - a. From the *IP address type* dropdown list, select *ipv4*.
  - b. From the *VPC* dropdown list, select the security VPC, where the FortiGate is deployed.

- c. From the *Availability Zones* dropdown list, select the AZ and subnet where the FortiGate is deployed. This example selects the private subnet where the FortiGate port2 is mapped to. In this example, you can enable multiple VDOMs (only available on BYOL instances) or split-task VDOMs (available on BYOL and on-demand instances), and port2 is mapped to the traffic-handling VDOM. You then create the Geneve interface on port2 to handle the traffic that has been redirected via the GWLB. See [Post-deployment configuration on page 210](#)

Step 1: Configure Load Balancer

#### Basic Configuration

To configure your Gateway Load Balancer, provide a name and confirm your VPC and subnet selections. This type of load balancer consists of an IP listener that receives all connection requests and routes them to the target group you will specify in the steps that follow.



4. Under *IP Listeners Routing*, click *Create Target Group* to configure a target group:
  - a. For *Target Type*, select *IP Address*.
  - b. In the *Target Group Name* field, enter the desired name.
  - c. For *Protocol*, select *GENEVE*.
  - d. In the *Port* field, enter 6081.
  - e. For *VPC*, select the VPC where you have deployed or will deploy the GWLB. In this example, the desired VPC is the security VPC.
  - f. Under *Health Checks*, configure the following:
    - i. For *Protocol*, select *TCP*.
    - ii. Override the *Advanced Health Check Settings > Port* setting to 443.
5. Register the targets during target group creation:
  - a. In the *IP* field, enter the FortiGate IP address. In this example, you would enter the FortiGate port2 IP address.
  - b. Click *Include as pending below*.
  - c. Click *Create Target Group*.
6. Ensure that cross-zone LB is enabled:
  - a. Go to *Compute > EC2 Dashboard > Load Balancing > Load Balancers*.
  - b. Select the newly created LB.
  - c. On the *Attributes* tab, edit the attributes and ensure that cross-zone LB is enabled.

## Creating the LB endpoint

The LB endpoint is a listener that forwards traffic from the customer VPC to the GWLB and subsequently to the target group that you created in [Creating the GWLB and registering targets on page 208](#). Before you create the LB endpoint, you must deploy an endpoint service in the region where your endpoint will be.

### To create an endpoint service:

1. Go to *VPC Dashboard > Virtual Private Cloud > Endpoint services*.
2. Click *Create Endpoint Service*.
3. For *Associate Load Balancers*, select the GWLB that you created in [Creating the GWLB and registering targets on page 208](#).
4. Enable endpoint acceptance if desired. This example does not require it.
5. Click *Create service*.

**To create the LB endpoint:**

1. Go to *VPC Dashboard > Virtual Private Cloud > Endpoint Services*.
2. Select the newly created endpoint service.
3. Copy the service name of the service on the *Details* tab.
4. Go to *VPC Dashboard > Virtual Private Cloud > Endpoints*.
5. Create the endpoint:
  - a. Click *Create Endpoint*.
  - b. For *Service category*, select *Other endpoint Services*.
  - c. In the *Service Name* field, paste the service name that you copied in step 3.
  - d. Click *Verify service*.
  - e. From the *VPC* dropdown list, select the VPC where you need to deploy the endpoint.
  - f. From the *Subnets* dropdown list, select the subnet where you need to deploy the endpoint.
  - g. Click *Create Endpoint*.

## VPC route tables


This example has two VPCs and multiple subnets within each VPC:

- **Customer VPC (10.10.0.0/16)**: place protected resources whose traffic must be analyzed.
- **Security VPC (10.90.0.0/16)**: place FortiGates here.

Application subnets are placed in different AZs.

Configure the ingress route table as follows:



- Subnet 1 (10.10.2.0/23) is mapped to the GWLB endpoint placed in AZ 1 subnet.
- Subnet 2 (10.10.4.0/23) is mapped to the GWLB endpoint placed in AZ 2 subnet.

Route Table: rtb-

Summary Routes Subnet Associations Edge Associations Route Propagation Tags

Edit routes

View All routes

| Destination  | Target   | Status | Propagated |
|--------------|--|--------|------------|
| 10.10.0.0/16 | local  | active | No         |
| 10.10.2.0/23 | vpce-  Availability Zone 1 - Subnet | active | No         |
| 10.10.4.0/23 | vpce-  Availability Zone 2 - Subnet | active | No         |

- The Internet gateway is assigned on the route table *Edge Associations* tab. This allows traffic to flow into the VPC and then be redirected into their respective subnets via the routes that you created above.

## Post-deployment configuration

You must create a Geneve interface on the FortiGate to handle traffic between the FortiGate and GWLB.

**To create the Geneve interface:**

1. Go to *EC2 Dashboard > Network & Security > Network Interfaces*. Copy the *Primary private IPv4 address* value for the GWLB interface created in the security VPC. There is one GWLB interface for each availability zone (AZ). Ensure to use the IPv4 for the GWLB interface in the same zone as the FortiGate being configured.

2. This example creates separate VDOMs via the split VDOM feature to handle traffic from the application VPC.

Enable the multi-VDOM feature:

```
config system global
    set vdom-mode multi-vdom
end
```



FortiOS prompts you to log in again to enable the multi-VDOM feature.

---

3. Create a new VDOM:

```
config vdom
    edit FG-traffic
end
```

4. Enable probe response on port 2 on both FortiGate instances. This allows LB health check to function:

```
config vdom
    edit FG-traffic
    config system interface
        edit "port2"
            set vdom "FG-traffic"
            set alias private
            set mode dhcp
            set allowaccess ping https ssh fgfm probe-response
            set defaultgw disable
        next
    end
end
```

5. Create Geneve interfaces:

```
config vdom
    edit "FG-traffic"
        config system geneve
            edit "awsgeneve"
                set interface "port2"
                set type ppp
                set remote-ip <GWLB_interface_ip (from step 1)>
            next
        end
    next
end
```

6. Setting a higher priority on static routes for Geneve interfaces is recommended to avoid unintended functionality:

```
config router static
    edit 2
        set priority 100
        set device "awsgeneve"
    next
end
```

7. Configure IPv6 routing:

```
config router static6
    edit 2
        set device "awsgeneve"
    next
end
```

8. In a scenario where the load balancer is in a different subnet than the FortiGate interface, configure the following static route to avoid health check failures:

```
config router static
  edit 3
    set device port2
    set dst <loadbal_subnet>
    set gateway <local_gateway>
  next
end
```

### To configure egress routes:

If the current VDOM has multiple interfaces, you must add egress routes to ensure that traffic entering through the Geneve interfaces egress through the same interface.

The following provides commands for configuring egress routes for IPv4:

```
config router policy
  edit 1
    set input-device "awsgeneve"
    set src "0.0.0.0/0.0.0.0"
    set dst "10.10.2.0/255.255.254.0"
    set output-device "awsgeneve"
  next
end
```

The following provides commands for configuring egress routes for IPv6:

```
config router policy6
  edit 1
    set input-device "awsgeneve"
    set src ":::0"
    set dst <IPv6 Subnet associated to Customer APP Subnet>
    set output-device "awsgeneve"
  next
end
```

## Validating the configuration

Since traffic between the Internet and the application EC2 instance flows through the FortiGate Geneve interface, this example creates a FortiOS firewall policy that allows communication from the Geneve interface to the Geneve interface. The following shows an example policy.



This policy facilitates easy debugging and inclusion of IPv6 source and destination addresses. You should not configure this policy in a production environment.

---

### To configure the policy:

```
config firewall policy
  edit 1
    set name "test_policy"
    set srcintf "awsgeneve"
    set dstintf "awsgeneve"
```



```

set srcaddr "all"
set dstaddr "all"
set srcaddr6 "all"
set dstaddr6 "all"
set action accept
set schedule "always"
set service "ALL"
next
end

```

### To run a packet sniffer on the Geneve interface created to handle GWLB traffic:

In this example, the VDOM name is FG-traffic. When multiple VDOM mode (available only on BYOL instances) is enabled, substitute the name of your VDOM here for FG-traffic.

#### 1. Run a packet sniffer:

```

config vdom
edit FG-traffic
diagnose sniffer packet awsgeneve

```

#### 2. While the packet capture is running, attempt to access/ping a resource in the application subnet (protected subnet). The ping should succeed. The following shows the FortiGate packet capture for this access attempt:

```

FGVM04# # config vdom
FGVM04# (vdom) # edit FG-traffic
current vf=FG-traffic:3
FGVM04# (FG-traffic) # diag sniffer packet awsgeneve
Using Original Sniffing Mode
interfaces=[awsgeneve]
filters=[none]
pcap_lookupnet: awsgeneve: no IPv4 address assigned
1.757414 37.252.69.104.50605 -> 10.10.3.30.445: syn 3512577133
1.757455 37.252.69.104.50605 -> 10.10.3.30.445: syn 3512577133
1.758454 10.10.3.30.445 -> 37.252.69.104.50605: rst 0 ack 3512577134
1.758465 10.10.3.30.445 -> 37.252.69.104.50605: rst 0 ack 3512577134
3.671619 194.147.140.98.43814 -> 10.10.3.30.33759: syn 533814003
3.671660 194.147.140.98.43814 -> 10.10.3.30.33759: syn 533814003
3.672472 10.10.3.30.33759 -> 194.147.140.98.43814: rst 0 ack 533814004
3.672482 10.10.3.30.33759 -> 194.147.140.98.43814: rst 0 ack 533814004
3.841786 194.147.140.98.43814 -> 10.10.3.30.33759: rst 533814004
3.841799 194.147.140.98.43814 -> 10.10.3.30.33759: rst 533814004
4.857755 173.180.116.40 -> 10.10.3.30: icmp: echo request
4.857798 173.180.116.40 -> 10.10.3.30: icmp: echo request
4.859288 10.10.3.30 -> 173.180.116.40: icmp: echo reply

```

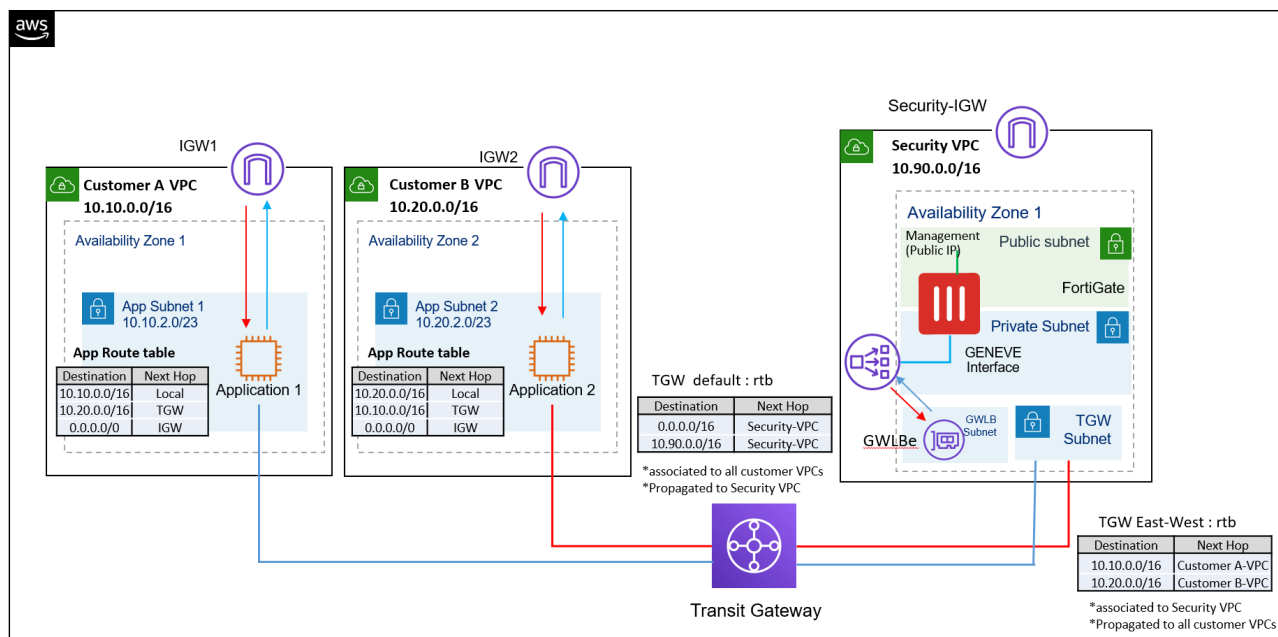
#### 3. While the packet capture is running, attempt to access/ping an Internet resource from the protected resource. The ping should succeed. The following shows the FortiGate packet capture for this access attempt:

```

FGVM04# (vdom) # edit FG-traffic
current vf=FG-traffic:3
FGVM04# (FG-traffic) # diag sniffer packet awsgeneve
Using Original Sniffing Mode
interfaces=[awsgeneve]
filters=[none]
pcap_lookupnet: awsgeneve: no IPv4 address assigned
6.246479 173.180.116.40.56015 -> 10.10.3.30.22: psh 3989997602 ack 3873790126
6.246504 173.180.116.40.56015 -> 10.10.3.30.22: psh 3989997602 ack 3873790126
6.247445 10.10.3.30.22 -> 173.180.116.40.56015: ack 3989997650
6.247458 10.10.3.30.22 -> 173.180.116.40.56015: ack 3989997650
6.247463 10.10.3.30.22 -> 173.180.116.40.56015: psh 3873790126 ack 3989997650
6.247465 10.10.3.30.22 -> 173.180.116.40.56015: psh 3873790126 ack 3989997650
6.315274 173.180.116.40.56015 -> 10.10.3.30.22: ack 3873790174
6.315303 173.180.116.40.56015 -> 10.10.3.30.22: ack 3873790174
6.372304 173.180.116.40.56015 -> 10.10.3.30.22: psh 3989997650 ack 3873790174
6.372314 173.180.116.40.56015 -> 10.10.3.30.22: psh 3989997650 ack 3873790174

```

## East-west security inspection between two customer VPCs



This document illustrates east-west security inspection for traffic flowing between two customer virtual private clouds (VPC). Though you can configure AWS resources and FortiGates to route and inspect all traffic (not only east-west traffic), this document focuses on the configuration of security inspection specifically for east-west traffic between two customer VPCs leveraging transit gateway (TGW) and gateway load balancer (GWLB).



Route tables on page 217 illustrates the AWS VPC, TGW, and GWLB route table configuration to achieve inspection of traffic flowing between the Application subnets via the FortiGate in the security VPC.

This guide assumes that you have already created the following and they are in place as the diagram shows:

- Customer A and B VPCs
- Security VPC
- FortiGate with at least one management network interface and elastic IP address assigned
- Application instances

The guide describes configuring additional network interfaces to handle data traffic. The following describes the two VPC types in this deployment:

| VPC      | Description  |
|----------|--|
| Customer | Where customer workloads are deployed. The customer VPCs each have one availability zone (AZ) with an application-purposed subnet where you deploy application workloads where the FortiGate must inspect the traffic. |
| Security | Where the FortiGate is deployed. You create the GWLB in this VPC. The security VPC AZ also includes the following subnets:   |

| VPC | Description   |
|-----|---|
|     | <ul style="list-style-type: none"> <li>• <b>GWLB endpoint subnet:</b> deploy the GWLB endpoint so that traffic is redirected to the GWLB, which then redirects the traffic to the FortiGate for inspection.</li> <li>• <b>TGW subnet:</b> deploy the TGW and associated resources, which allows connection of the customer VPCs to the security VPC.</li> </ul> |

## Creating the GWLB and registering targets

For this deployment, you create the GWLB in the security subnet.

### To create the GWLB and register targets:

1. Go to *Compute > EC2 Dashboard > Load Balancing > Load Balancers*.
2. Click *Create Load Balancer*, then *Gateway Load Balancer*.
3. Configure the GWLB:
  - a. From the *IP address type* dropdown list, select *ipv4*.
  - b. From the *VPC* dropdown list, select the security VPC, where the FortiGate is deployed.
  - c. From the *Availability Zones* dropdown list, select the AZ and subnet where the FortiGate is deployed. This example selects the private subnets for the respective AZs where the FortiGate port2 is mapped to. In this example, you can enable multiple VDOMs (only available on BYOL instances) or split-task VDOMs (available on BYOL and on-demand instances), and port2 is mapped to the traffic-handling VDOM. You then create the Geneve interface on port2 to handle the traffic that has been redirected via the GWLB. See [Post-deployment configuration on page 218](#).

#### Step 1: Configure Load Balancer

##### Basic Configuration

To configure your Gateway Load Balancer, provide a name and confirm your VPC and subnet selections. This type of load balancer consists of an IP listener that receives all connection requests.

Name ⓘ

IP address type ⓘ

##### Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only. You can specify only one subnet per Availability Zone.

VPC ⓘ

Availability Zones

☒ us-west-2a

☒ us-west-2b

4. Configure routing:
  - a. From the *Target group* dropdown list, create a new target group with the desired name.
  - b. For *Target type*, select *IP*.
  - c. Ensure that *Protocol:Port* displays as *GENEVE: 6081*.
  - d. From the *Protocol* dropdown list, select *HTTPS*.
  - e. From the *Port* dropdown list, select the desired port. This example uses port 443. Ensure that your security group configuration allows traffic on that port.
5. Register the targets:
  - a. In the *IP* field, enter the FortiGate IP address. In this example, you would enter the FortiGate port2 IP address.
  - b. Click *Add to list*, then *Next*.
  - c. Click *Review and Create*.

6. Ensure that cross-zone LB is enabled:
  - a. Go to *Compute > EC2 Dashboard > Load Balancing > Load Balancers*.
  - b. Select the newly created LB.
  - c. On the *Description* tab, ensure that cross-zone LB is enabled.

## Creating the LB endpoint

The LB endpoint is a listener that forwards traffic from the customer VPC to the GWLB and subsequently to the target group that you created in [Creating the GWLB and registering targets on page 215](#). You must create an endpoint for each AZ. Before you create the LB endpoint, you must deploy an endpoint service in the region where your endpoint will be.

### To create an endpoint service:

1. Go to *VPC Dashboard > Virtual Private Cloud > Endpoint services*.
2. Click *Create Endpoint Service*.
3. For *Associate Load Balancers*, select the GWLB that you created in [Creating the GWLB and registering targets on page 215](#).
4. Enable endpoint acceptance if desired. This example does not require it.
5. Click *Create service*.

### To create the LB endpoint:

1. Go to *VPC Dashboard > Virtual Private Cloud > Endpoint Services*.
2. Select the newly created endpoint service.
3. Copy the service name of the service on the *Details* tab.
4. Create the endpoint for the first AZ:
  - a. Go to *VPC Dashboard > Virtual Private Cloud > Endpoints*.
  - b. Click *Create Endpoint*.
  - c. For *Service category*, select *Find service by name*.
  - d. In the *Service Name* field, paste the service name that you copied in step 1.
  - e. Click *Verify*.
  - f. From the *VPC* dropdown list, select the VPC where you need to deploy the endpoint.
  - g. From the *Subnets* dropdown list, select the subnet where you need to deploy the endpoint. This example selects the GWLB endpoint subnet created in each AZ in the security subnet.
  - h. Click *Create Endpoint*.
5. Repeat the process to create the endpoint for the second AZ.

## Creating the transit gateway

A transit gateway (TGW) is a transit hub used to connect two VPCs or a VPC to an on-premise network. This example connects the application VPC to the security VPC via a TGW. This ensures that any access to and from the application VPC is routed via the security VPC, where the FortiGates can inspect it.

**To create the TGW:**

1. Go to *VPC Dashboard > Transit Gateways > Transit Gateways*.
2. Click *Create Transit Gateway*.
3. Configure the TGW as needed. Creating a TGW creates a TGW default route table. The table is used as the default association and propagation route table for this gateway. You can access this table in *VPC Dashboard > Transit Gateways > Transit Gateway Route Tables*.

**To create the TGW attachment:**

You can create a gateway attachment to link separate VPCs and subnets to the newly created TGW. The two resources can be in the same or different AWS accounts. This example assumes that both VPCs are in the same AWS account.

1. Go to *VPC Dashboard > Transit Gateways > Transit Gateway Attachments*.
2. Click *Create Transit Gateway Attachment*.
3. From the *Transit Gateway ID* dropdown list, select the TGW that you created.
4. From the *Attachment type* dropdown list, select *VPC*.
5. From the *VPC ID* dropdown list, select the VPC that you want to attach to the TGW.
6. Under *Subnet IDs*, select the required subnet in the desired AZ.
7. Configure other fields as desired.
8. Click *Create Attachment*.
9. Repeat the process for the remaining two VPC attachments. The security VPC is attached to the TGW, with only the TGW subnets in each AZ selected. This ensures that traffic can be routed seamlessly to and from the GWLB endpoint. You must attach each subnet/AZ to the TGW separately.

## Route tables

This example has three VPCs and multiple subnets within each VPC:

- **Customer A VPC (10.10.0.0/16) and Customer B VPC (10.20.0.0/16):** place protected resources whose traffic must be analyzed here.
- **Security VPC (10.90.0.0/16):** place FortiGates here.



Application subnets are placed in different availability zones.

## East-west egress route table



Ensure that the east-west egress route table is configured as follows:

- Link the customer application subnets on VPC A and B to the security VPC via the TGW.
- Route traffic from customer A VPC so that it must go through the security VPC to reach the customer B VPC.

The following shows the customer A VPC route table. You must configure the customer B VPC route table similarly:

| Destination  | Target  | Status | Propagated |
|--------------|---|--------|------------|
| 10.10.0.0/16 | local   | active | No         |
| 0.0.0.0/0    | igw-  Internet Access via IGW                  | active | No         |
| 10.20.0.0/16 | tgw-  CustomerB VPC access via Transit Gateway | active | No         |



Ensure that the TGW is attached to the designated TGW subnet in each AZ of the security VPC. Designated TGW VPCs allow you to configure forward and reverse routes to and from the FortiGate without causing routing loops. The following shows the TGW subnet route table for the forward route:

| Destination                        | Target  | Status                           | Propagated |
|------------------------------------|---|----------------------------------|------------|
| 10.90.0.0/16                       | local   | active                           | No         |
| 10.10.0.0/16 <b>Customer A VPC</b> | vpce-  | <b>GWLB endpoint -</b><br>active | No         |
| 10.20.0.0/16 <b>Customer B VPC</b> | vpce-  | <b>Security VPC</b><br>active    | No         |



An ideal configuration would have multiple GWLB endpoints in each AZ and selectively route traffic for high availability. Due to the way routes are configured on AWS, you can configure a single GWLB endpoint for multiple FortiGates.

The following shows the GWLB endpoint subnet route table for the reverse route:

| Destination                        | Target   | Status                           | Propagated |
|------------------------------------|--|----------------------------------|------------|
| 10.90.0.0/16                       | local  | active                           | No         |
| 10.10.0.0/16 <b>Customer A VPC</b> | tgw-  | <b>Transit Gateway</b><br>active | No         |
| 10.20.0.0/16 <b>Customer B VPC</b> | tgw-  | active                           | No         |

## Configuring TGW route tables

Since traffic from customer VPC A and customer VPC B must be routed via the security subnet and cannot be forward directly, you must configure the following on the TGW route table for east-west traffic.

### To configure TGW route tables:

1. Go to *VPC Dashboard > Transit Gateways > Transit Gateway Route Tables*.
2. Delete the automatically generated route table and its associations. You will create two new TGW route tables.
3. Create the TGW default route table:
  - a. On the *Associations* tab, associate the route table with Customer A and Customer B VPCs.
  - b. On the *Propagations* tab, propagate the route table to the security VPC.
  - c. On the *Routes* tab, add a default route to send all traffic to the security VPC.
4. Create the east-west route table:
  - a. On the *Associations* tab, associate the route table with the security VPC.
  - b. On the *Propagations* tab, propagate the route table to Customer A and Customer B VPCs.
  - c. On the *Routes* tab, define customer A and B VPC routes.

## Post-deployment configuration

You must create a Geneve interface on the FortiGate to handle traffic between the FortiGate and GWLB.

### To create the Geneve interface:

1. Go to *EC2 Dashboard > Network & Security > Network Interfaces*. Copy the *Primary private IPv4 address* value for the GWLB interface created in the security VPC.

2. This example creates separate VDOMs via the split VDOM feature to handle traffic from the application VPC. Enable probe response on port 2 on both FortiGate instances. This allows LB health check to function:

```
config system global
  config system interface
    edit "port2"
      set vdom "FG-traffic"
      set alias private
      set mode dhcp
      set allowaccess ping https ssh fgfm probe-response
      set defaultgw disable
    next
  end
end
```

3. Create Geneve interfaces:

```
config vdom
  edit "FG-traffic"
    config system geneve
      edit "awsgeneve"
        set interface "port2"
        set type ppp
        set remote-ip <GWLB_interface_ip (from step 1)>
      next
    end
  next
end
```

4. Setting a higher priority on static routes for Geneve interfaces is recommended to avoid unintended functionality:

```
config router static
  edit 2
    set priority 100
    set device "awsgeneve"
  next
end
```

5. In a scenario where the load balancer is in a different subnet than the FortiGate interface, configure the following static route to avoid health check failures:

```
config router static
  edit 3
    set device port2
    set dst <loadbal_subnet>
    set gateway <local_gateway>
  next
end
```

### To configure egress routes:

If the current VDOM has multiple interfaces, you must add egress routes to ensure that traffic entering through the Geneve interfaces egress through the same interface.

```
config router policy
  edit 1
    set input-device "awsgeneve"
    set src "0.0.0.0/0.0.0.0"
    set dst "10.10.2.0/255.255.254.0"
    set output-device "awsgeneve"
  next
end
```

## Validating the configuration

Since traffic between the Internet and the application EC2 instance flows through the FortiGate Geneve interface, this example creates a FortiOS firewall policy that allows communication from the Geneve interface to the Geneve interface. The following shows an example policy.



This policy facilitates easy debugging. You should not configure this policy in a production environment.

### To configure the policy:

```
config firewall policy
  edit 1
    set name "test_policy"
    set srcintf "az2"
    set dstintf "az2"
    set srcaddr "all"
    set dstaddr "all"
    set action accept
    set schedule "always"
    set service "ALL"
  next
end
```

### To run a packet sniffer on the Geneve interface created to handle GWLB traffic:

In this example, the VDOM name is FG-traffic. When multiple VDOM mode (available only on BYOL instances) is enabled, substitute the name of your VDOM here for FG-traffic.

#### 1. Run a packet sniffer:

```
config vdom
  edit FG-traffic
    diagnose sniffer packet awsgeneve
```

#### 2. While the packet capture is running, attempt to access/ping a resource in customer B VPC from a resource in customer A VPC. The ping should succeed. The following shows the FortiGate packet capture for this access attempt:

```
FGTAW5 [redacted] (FG-traffic) # diag sniffer packet awsgeneve
Using Original Sniffing Mode
interfaces=[awsgeneve]
filters=[none]
pcap_lookupnet: awsgeneve: no IPv4 address assigned
9.216818 10.10.2.114 -> 10.20.2.147: icmp: echo request
9.216871 10.10.2.114 -> 10.20.2.147: icmp: echo request
9.218211 10.20.2.147 -> 10.10.2.114: icmp: echo reply
9.218220 10.20.2.147 -> 10.10.2.114: icmp: echo reply
10.217660 10.10.2.114 -> 10.20.2.147: icmp: echo request
10.217672 10.10.2.114 -> 10.20.2.147: icmp: echo request
```



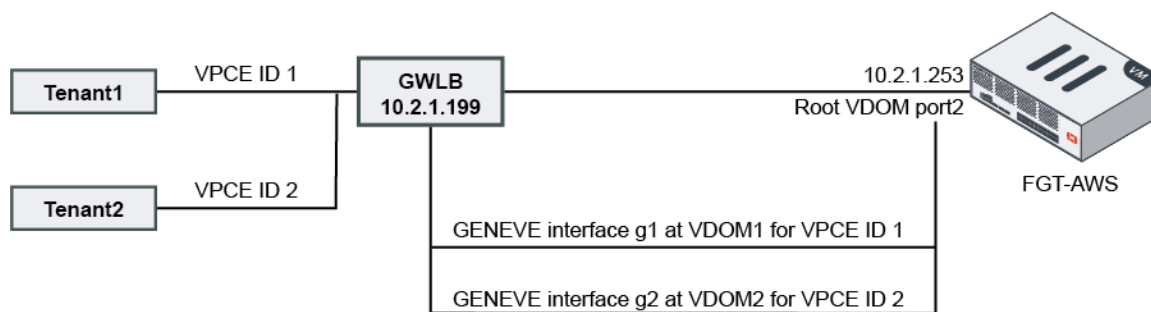
## Multitenancy support with AWS GWLB

To better support multitenancy with AWS gateway load balancer (GWLB), this enhancement adds support to identify incoming traffic using virtual private cloud (VPC) endpoint IDs in the GENEVE header to forward traffic to the appropriate virtual domain (VDM) tenant.

The VPC endpoint (VPCE) to VDOM mapping is configured under the following CLI commands:

```
config aws vpce
  edit <id>
    set name <VPCE name>
    set endpoint-id <VPCE ID>
    set vdom <VDOM name>
  next
end
```

This guide assumes that you have previously configured a GWLB environment. The following shows the topology for this deployment:



This feature is available with FortiOS 7.0.4 and later versions.

### To configure multitenancy support with AWS GWLB:

#### 1. Configure the GENEVE interface in VDOM 1:

```
config system geneve
  edit "g1"
    set interface "port2"
    set type ppp
    set remote-ip 10.2.1.199
  next
end
```

#### 2. Configure the GENEVE interface in VDOM 2:

```
config system geneve
  edit "g2"
    set interface "port2"
    set type ppp
    set remote-ip 10.2.1.199
  next
end
```

**3. Configure a static route and firewall policy in VDOM 1:**

```
config router static
  edit 1
    set device "g1"
  next
end
config firewall policy
  edit 1
    set srcintf "g1"
    set dstintf "g1"
    set action accept
    set srcaddr "all"
    set dstaddr "all"
    set schedule "always"
    set service "ALL"
  next
end
```

**4. Configure a static route and firewall policy in VDOM 2:**

```
config router static
  edit 1
    set device "g2"
  next
end
config firewall policy
  edit 1
    set srcintf "g2"
    set dstintf "g2"
    set action accept
    set srcaddr "all"
    set dstaddr "all"
    set schedule "always"
    set service "ALL"
  next
end
```

**5. Configure the AWS VPCE in the global VDOM:**

```
config aws vpce
  edit 1
    set name "tenant1"
    set endpoint-id "fac3dcc5b40ca0b9"
    set vdom "vdom1"
  next
  edit 2
    set name "tenant2"
    set endpoint-id "07392059b988e86af"
    set vdom "vdom2"
  next
end
```

- 6.** Ensure that the FortiGate routes traffic from different VPCE IDs to different VDOMs as desired. The following shows an example of the desired output:

```
diagnose sniffer packet any icmp 4
Using Original Sniffing Mode
interfaces=[any]
filters=[icmp]
5.330846 g1 in 10.1.1.10 -> 8.8.8.8: icmp: echo request
5.330882 g1 out 10.1.1.10 -> 8.8.8.8: icmp: echo request
5.339186 g1 in 8.8.8.8 -> 10.1.1.10: icmp: echo reply
5.339210 g1 out 8.8.8.8 -> 10.1.1.10: icmp: echo reply
7.785495 g2 in 10.1.2.10 -> 8.8.8.8: icmp: echo request
7.785533 g2 out 10.1.2.10 -> 8.8.8.8: icmp: echo request
7.794251 g2 in 8.8.8.8 -> 10.1.2.10: icmp: echo reply
7.794273 g2 out 8.8.8.8 -> 10.1.2.10: icmp: echo reply
```

# Change log

| Date       | Change description   |
|------------|--|
| 2023-05-11 | Initial release.   |
| 2023-08-31 | Updated: <ul style="list-style-type: none"><li>• <a href="#">Instance type support on page 7</a></li><li>• <a href="#">Region support on page 13</a></li></ul> Added <a href="#">Deploying FortiGate-VM on Snowball Edge on page 146</a> . |
| 2023-09-06 | Updated <a href="#">Deploying FortiGate-VM on Snowball Edge on page 146</a> and subtopics.   |
| 2023-10-30 | Added: <ul style="list-style-type: none"><li>• <a href="#">SD-WAN on page 201</a></li><li>• <a href="#">Cloud WAN on page 206</a></li></ul>  |
| 2023-12-20 | Added <a href="#">SDN connector support for alternate resources on page 169</a> .  |
| 2024-01-22 | Updated <a href="#">Instance type support on page 7</a> .  |
| 2024-03-14 | Updated <a href="#">Instance type support on page 7</a> .  |



[www.fortinet.com](http://www.fortinet.com)

Copyright© 2024 Fortinet, Inc. All rights reserved. Fortinet®, FortiGate®, FortiCare® and FortiGuard®, and certain other marks are registered trademarks of Fortinet, Inc., and other Fortinet names herein may also be registered and/or common law trademarks of Fortinet. All other product or company names may be trademarks of their respective owners. Performance and other metrics contained herein were attained in internal lab tests under ideal conditions, and actual performance and other results may vary. Network variables, different network environments and other conditions may affect performance results. Nothing herein represents any binding commitment by Fortinet, and Fortinet disclaims all warranties, whether express or implied, except to the extent Fortinet enters a binding written contract, signed by Fortinet's Chief Legal Officer, with a purchaser that expressly warrants that the identified product will perform according to certain expressly-identified performance metrics and, in such event, only the specific performance metrics expressly identified in such binding written contract shall be binding on Fortinet. For absolute clarity, any such warranty will be limited to performance in the same ideal conditions as in Fortinet's internal lab tests. Fortinet disclaims in full any covenants, representations, and guarantees pursuant hereto, whether express or implied. Fortinet reserves the right to change, modify, transfer, or otherwise revise this publication without notice, and the most current version of the publication shall be applicable.