WHITE PAPER

Amazon Web Services (AWS) Reference Architecture

TABLE OF CONTENTS

Executive Summary

The public cloud, often referred to as "the cloud," continues to be a popular model of cloud computing. Cloud service providers leverage the internet and make resources such as infrastructure, storage, and servers available for businesses.

Third-party providers own and operate the shared physical hardware and offer it to companies based on their needs. Companies around the globe adopt the cloud to take advantage of core characteristics:

Cost-effectiveness. Cloud infrastructure means customers do not need to spend money on purchasing and maintaining equipment. This drastically reduces capital expenditure (CapEx) costs, saving companies the resources and time to invest in hardware, facilities, and utilities, or to build out a large data center to grow their business.

Scalability. Cloud-based solutions are ideal for businesses with growing and fluctuating bandwidth demands. If business demands increase, customers can easily increase their cloud capacity without investing in more physical infrastructure. This level of agility provides businesses a key advantage over competitors.

Disaster recovery. Data loss is a major concern for all organizations. Storing customer data in the cloud guarantees that data is always available, even if equipment such as laptops or PCs is damaged. Cloud-based services provide quick data recovery for emergency scenarios—from natural disasters to power outages.

Increased agility. Today, businesses need to be more dynamic to be productive. They need to continuously evolve and improve their processes, tools, technologies, and policies. Being agile enables businesses to make faster decisions, prioritize the work, and ensure customer satisfaction. With the cloud, businesses experience better delivery, better collaboration, and faster rollouts of new business initiatives.

The shared responsibility model

While cloud providers manage the security of the cloud, security in the cloud is the responsibility of their customers. Customers retain control of what security they choose to implement to protect their content and applications.

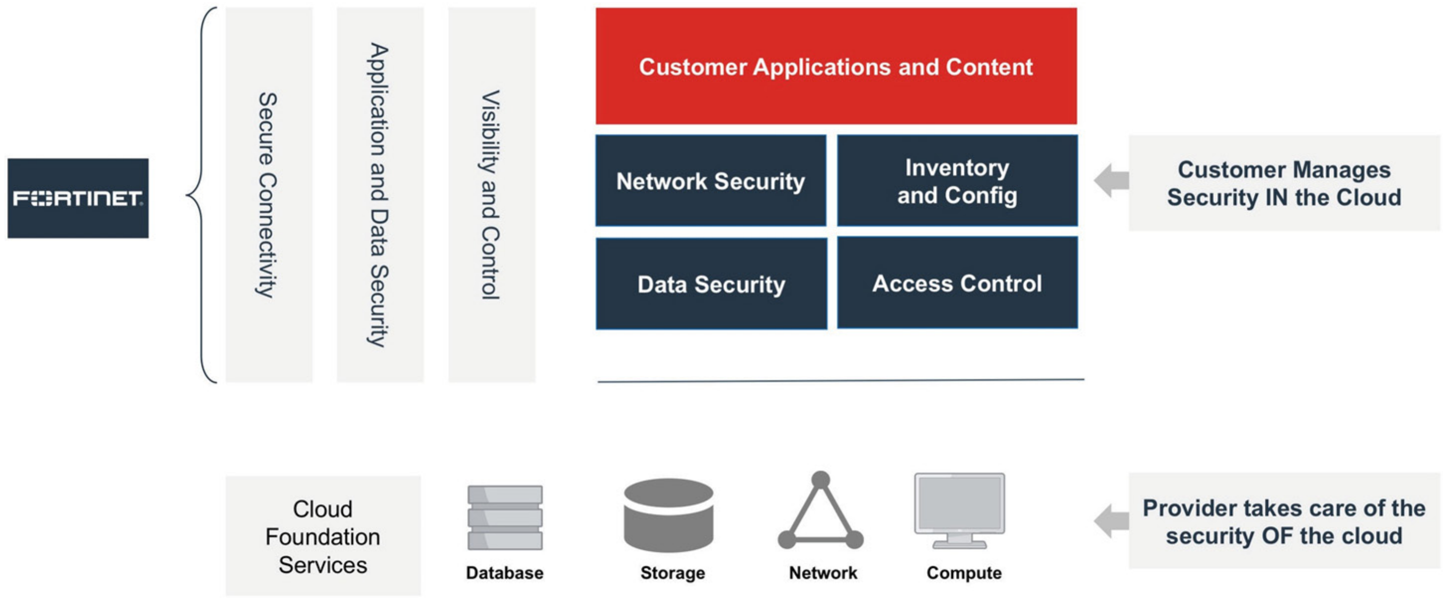

FORTINET
Secure Connectivity
Application and Data Security
Visibility and Control
Customer Applications and Content
Network Security
Inventory and Config
Data Security
Access Control
Customer Manages Security IN the Cloud
Cloud Foundation Services
Database
Storage
Network
Compute
Provider takes care of the security OF the cloud

Figure 1: The shared responsibility model

Cloud providers operate, manage, and control the components from the host operating system and virtualization layer down to the physical security of the facilities in which the service operates. However, customers retain ownership and control of their data and are responsible for configuring and deploying security baselines within their environments.

Amazon Web Services (AWS) is one of the most popular cloud providers that businesses use to run their applications. The Fortinet FortiGate Next-Generation Firewall (NGFW) on AWS leverages its powerful automation capabilities to help customers protect their workloads against sophisticated cyberattacks.

In the next section, the main AWS constructs that are relevant to this design guide are explained.

Terms to Know

- **Availability Zones:** One or more discrete data centers, each with redundant power, networking, and connectivity, that are housed in separate facilities.
- **AWS Region:** A physical location in the world where there are multiple Availability Zones.

AWS Networking and Service Primer

Because the main objective of this white paper is to explain design principles and guidelines of a secure AWS architecture, a good knowledge of AWS concepts and key AWS services is a prerequisite to understanding design topics discussed in this guide.

Regions and Availability Zones (AZ)

The AWS cloud infrastructure is built around AWS Regions and AZ. An AWS Region is a physical location in the world where there are multiple AZ. AZ consist of one or more discrete data centers, each with redundant power, networking, and connectivity, which are housed in separate facilities. These AZ offer customers the ability to operate production applications and databases that are more resilient, fault tolerant, and scalable than would be possible from a single data center.

Each Amazon Region is designed to be completely isolated from the other Amazon Regions. This achieves the greatest possible fault tolerance and stability. Each AZ is isolated, but the AZ in a Region are connected through low-latency links. AWS provides the flexibility to place instances and store data within multiple geographic regions as well as across multiple AZ within each AWS Region. Each AZ is designed as an independent failure zone. This means that AZ are physically separated within a typical metropolitan region and are in lower-risk flood plains.

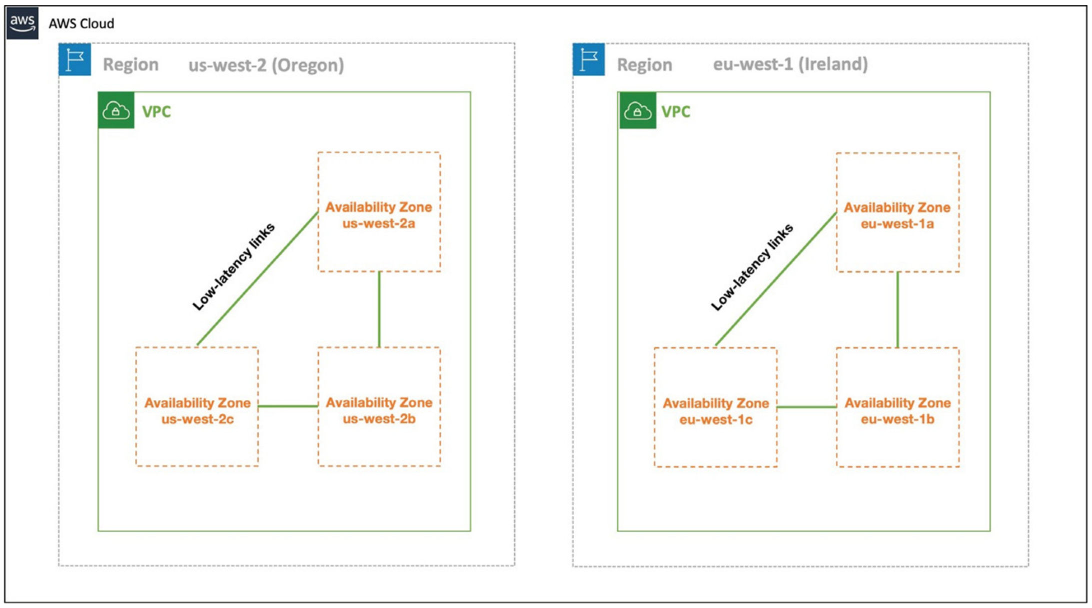

AWS Cloud
Region us-west-2 (Oregon)
VPC
Low-latency links
Availability Zone us-west-2a
Availability Zone us-west-2c
Availability Zone us-west-2b
Region eu-west-1 (Ireland)
VPC
Low-latency links
Availability Zone eu-west-1a
Availability Zone eu-west-1c
Availability Zone eu-west-1b

Figure 2: AWS Regions and Availability Zones

To take advantage of the fault tolerance and isolation offered by the AZ, it is necessary to distribute applications as well as security services that are deployed to protect them across multiple AZ.

Tech Tip #1
Note that Amazon VPC does not support modification of the primary CIDR.

Consider the overall network map before creating VPCs and defining CIDRs to them in the same manner when assigning CIDRs in physical data centers. This will help avoid problems such as IP address overlapping after instances are deployed in the AWS account.

Virtual Private Cloud (VPC)

Amazon Virtual Private Cloud (Amazon VPC) enables customers to launch AWS resources into a virtual network that they have defined. This virtual network closely resembles a traditional network that customers can operate in their own data center, with the benefits of using the scalable infrastructure of AWS. Amazon VPC lets customers provision a logically isolated section of the AWS Cloud where they can launch AWS resources in a virtual network that they define. Customers have complete control over their virtual networking environment, including selection of their own IP address range, creation of subnets, and configuration of route tables and network gateways. VPC networking supports both IPv4 and IPv6 addresses.

Most AWS accounts come with a default VPC that has a default subnet in each AZ. Customers can launch instances into their default VPC without knowing anything about Amazon VPC. However, AWS allows customers to create their own VPC known as nondefault VPC. This gives greater flexibility to customers who desire to customize configuration of VPC Classless Inter-Domain Routing (CIDR), subnetting, network configuration, etc.

Contrary to physical data center networks where extending CIDR blocks is a daunting task, VPC allows customers to define their CIDR block at the time of VPC creation and later extend the range by simply adding a secondary CIDR to take advantage of the dynamic nature of the cloud. Although customers are free to define and allocate any CIDR when creating a VPC, it is highly recommended to use CIDRs according to RFC 1918, "Address Allocation for Private Internets."

By default, every instance in a VPC has a route to all other instances within that VPC. However, internet connectivity is not enabled by default even if the defined CIDR is a publicly routable IP address range. To establish connection to the internet, an internet gateway must be created and then attached to the VPC. Instances that require internet connectivity can then be assigned public IP addresses. Elastic IP address can also be associated to an instance. Customers who need connectivity between their VPC and on-premises location can create a VPN gateway device and attach it to their VPC. IPsec VPN connections can then be established between the VPN gateway and the on-premises customer gateway.

Subnets

After creating a VPC, one or more subnets can be added in each AZ. To create a subnet, a CIDR block for the subnet that is a subset of the VPC CIDR block needs to be specified. Each subnet must reside entirely within one AZ and cannot span zones. If a subnet's traffic is routed to an internet gateway, the subnet is known as a public subnet. If a subnet does not have a route to the internet gateway, the subnet is known as a private subnet.

The CIDR block of a subnet can be the same as the CIDR block for the VPC (for a single subnet in the VPC), or a subset of the CIDR block for the VPC (for multiple subnets). The allowed block size is between /28 netmask and /16 netmask and subnets' CIDR within a VPC cannot overlap. For example, if the VPC CIDR block is 10.0.0.0/24, two /25 CIDR subnets can be created within that VPC. Note that the first four IP addresses and the last IP address in each subnet CIDR block are not available for customers to use and cannot be assigned to an instance. If, for example, a subnet CIDR is 10.0.0.0/24, the following five addresses are reserved:

- 10.0.0.0: Network address
- 10.0.0.1: Reserved by AWS for the VPC router
- 10.0.0.2: Reserved by AWS for the DNS server (base of the VPC CIDR plus two). AWS also reserves the base of each subnet range plus 2
- 10.0.0.3: Reserved by AWS for future use
- 10.0.0.255: Network broadcast address

Network ACL

A network access control list (ACL) is an optional layer of security for a VPC that acts as a stateless L4 firewall for controlling traffic in and out of one or more subnets. Since inter-subnet routing is available in a VPC by default, network ACL can be used to prevent a subnet from accessing other subnets in a VPC.

Security groups

A security group acts as a virtual Layer 4 firewall for instances to control both inbound and outbound traffic. Security groups are applied at an instance's network interface; therefore, each instance in a subnet could be assigned to a separate set of security groups. By default, AWS allows customers to apply up to five security groups to a network interface.

Contrary to network ACLs, security groups are stateful and do not support deny actions.

Security groups only function as Layer 4 firewalls and cannot inspect packets for application layer visibility. Therefore, an NGFW should be added as another security layer to detect and prevent advanced cyberattacks.

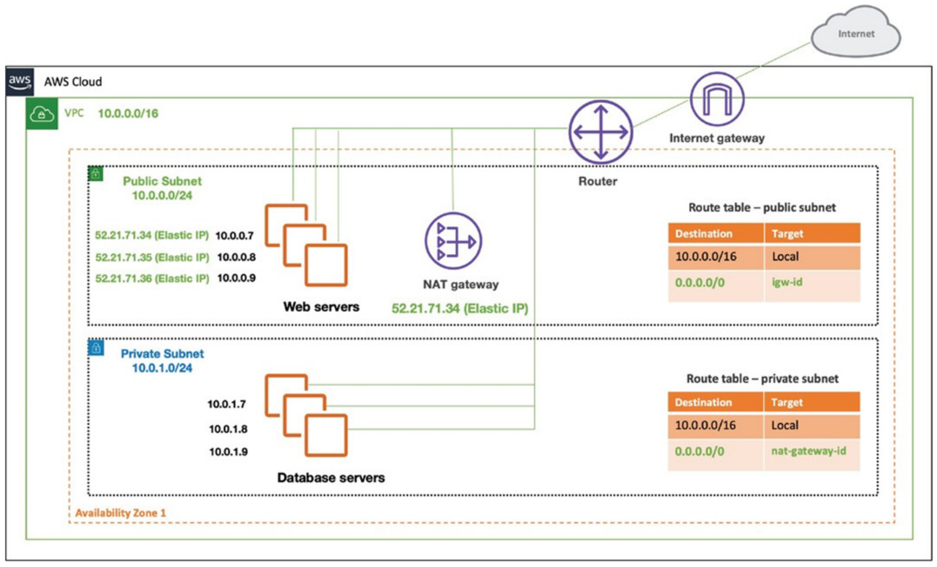
Route tables

A VPC has an implicit router and each subnet within the VPC must be associated with a route table, which specifies the allowed routes for outbound traffic leaving the subnet.

Default route table. When a VPC is created, it automatically comes with a default route table. The default route table controls the routing for all subnets that are not explicitly associated with any other route table. Routes in the default route table can be removed, added, and modified. However, the default route table cannot be deleted.

Custom route table. Customers can create additional route tables in addition to the main route table. By associating each new subnet with a custom route table, customers can ensure that they explicitly control each subnet route's outbound traffic.

All Amazon VPC route tables come with a default local route entry that cannot be removed.



Route table – public subnet

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	igw-id

Route table – private subnet

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	nat-gateway-id

Figure 3: Amazon VPC networking basics

What to Know Before Using a Network ACL

- All VPCs automatically come with a default NACL that allows all inbound and outbound IPv4 traffic.
- A network ACL contains a numbered list of rules that are evaluated in order, starting with the lowest numbered rule, to determine whether traffic is allowed in or out of any subnet associated with the network ACL.
- A network ACL has separate inbound and outbound rules, and each rule can either allow or deny traffic.
- Network ACLs are stateless; responses to allowed inbound traffic are subject to the rules for outbound traffic.

Main Attributes of a Security Group

- "Allow" rules can be applied but no "deny" rules are permitted.
- Separate rules can be specified for inbound and outbound traffic.
- All "allow" rules must be explicitly added to the security group. Otherwise, the default AWS action is to block all traffic.
- Security groups are stateful. If a request is sent from an instance, the response traffic for that request is allowed regardless of the inbound rules of the associated security group (and vice versa).
- No inbound traffic originating from another host to an instance is allowed until a security group's inbound rules are updated. This is due to the fact that AWS by default does not create any inbound rule.

Internet gateway

VPC routing is designed such that no instance in a VPC, in a private subnet or in a public subnet, can communicate over the internet without an attached internet gateway. An internet gateway performs network address translation for instances that have been assigned public IPv4 addresses.

After creating and attaching an internet gateway to a VPC, a route that directs internet-bound traffic to the internet gateway must be added to the route table associated with the public subnet, as shown in Figure 3. Also, to enable an instance to communicate over the internet, it must be assigned a public IP address or an elastic IP address that is associated with a private IP address. Because instances are only aware of their private IP addresses, the internet gateway is also required to provide logical one-to-one network address translation (NAT) on behalf of the instances, so that the return traffic is destined to the public IP address of an instance.

NAT gateway

A NAT gateway is a managed AWS service that can be used by customers to enable instances in a private subnet to connect to the internet. It also prevents unsolicited internet traffic to reach the private instances. When traffic goes to the internet, the source IP address is replaced with the NAT gateway's public IP address and similarly, when the response traffic goes to those instances, the NAT gateway translates the address, back to those instances' private IP addresses.

After a NAT gateway is created, the route table(s) associated with private subnet(s) must be updated to point internet-bound traffic to the NAT gateway. This enables instances in the private subnets to communicate with the internet. For example, the route table associated with the private subnet in Figure 3 has a route entry for the default route that has the NAT gateway as its target. This route ensures that all internet-bound traffic is routed through the NAT gateway.

AWS offers two kinds of NAT devices—NAT instance and NAT gateway. However, NAT gateway is the recommended option as it is managed by AWS and saves customers from administration efforts.

Elastic network interfaces

An elastic network interface (ENI) is a logical networking component in a VPC that represents a virtual network card. Customers can create and configure network interfaces in their accounts and attach them to instances in their VPCs. An ENI can have multiple attributes.

Every instance comes with at least one network interface after it is launched. This primary interface cannot be detached. All other ENIs can be detached from an instance and attached to another instance if the ENI and the interface that the ENI is being attached to reside in the same AZ.

Source/destination check. The source/destination check attribute controls whether source/destination checking is enabled on the instance (the default value is enabled). Disabling this attribute enables an instance to manage network traffic that is not specifically destined for the instance. For example, an instance running a firewall service should set this value to disabled. In Figure 5, the source/destination check for the second ENI (private interface) of the FortiGate firewall has been disabled to allow internet-bound traffic to route through the firewall.

Possible ENI Attributes

- A primary private IPv4 address
- One or more secondary private IPv4 addresses
- One elastic IP address (IPv4) per private IPv4 address
- One public IPv4 address
- One or more security groups
- A MAC address
- A source/destination check flag

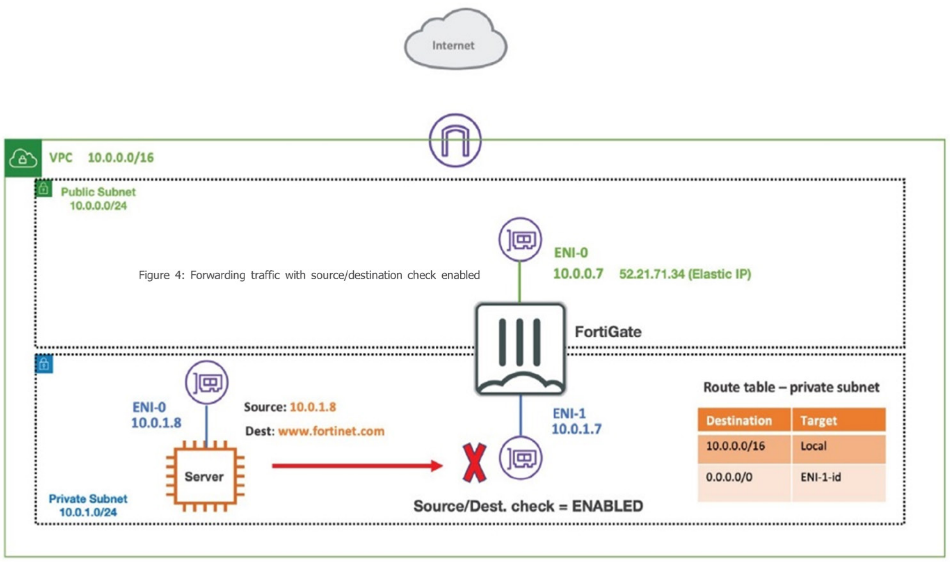

Internet
VPC 10.0.0.0/16
Public Subnet 10.0.0.0/24
Figure 4: Forwarding traffic with source/destination check enabled
ENI-0
10.0.0.7 52.21.71.34 (Elastic IP)
FortiGate
ENI-0
10.0.1.8
Source: 10.0.1.8
Dest: www.fortinet.com
Server
ENI-1
10.0.1.7
Route table – private subnet
Destination | Target
10.0.0.0/16 | Local
0.0.0.0/0 | ENI-1-id
Private Subnet 10.0.1.0/24
Source/Dest. check = ENABLED

Figure 4: Forwarding traffic with source/destination check enabled

Tech Tip #2

If ASN is not specified during creation of the virtual private gateway, AWS by default assigns ASN 64512.

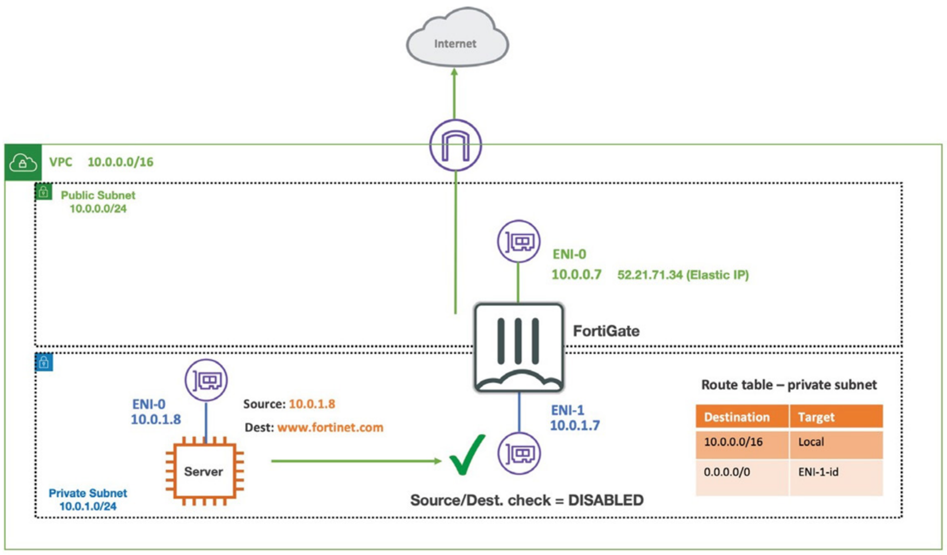


Figure 5: Forwarding traffic with source/destination check disabled

AWS VPN

AWS VPN (Site-to-Site) extends a customer's data center or branch office to AWS cloud via IPsec tunnels and supports connecting both virtual private gateway and AWS Transit Gateway. Customers can optionally run Border Gateway Protocol (BGP) over IPsec tunnels.

Virtual private gateway

A virtual private gateway is the VPN concentrator on the AWS side of the Site-to-Site VPN connection. A virtual private gateway can be attached to a VPC and acts as the VPN termination point in the VPC. An Autonomous System Number (ASN) can be assigned when customers create a virtual private gateway. However, the ASN cannot change after it is created.

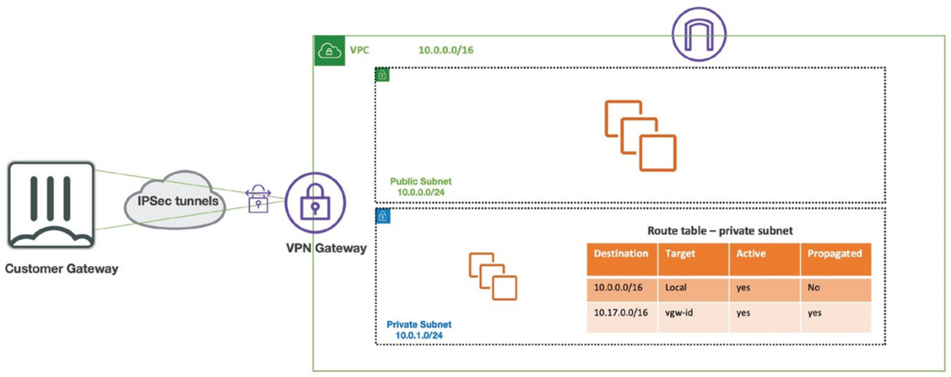
After a virtual private gateway is created and attached to a VPC, and a VPN connection is established, the virtual private gateway can function as a gateway to the customer's on-premises data center. It can either be statically added to the route table as the target for remote routes or is automatically added to the route table once remote routes are learned via BGP. For example, in the route table shown in Figure 6, the route to 10.17.0.0/16 has been dynamically learned and populated.

Although a virtual private gateway can be used to create multiple Site-to-Site VPN connections, network performance might be impacted, as it has a limited bandwidth and does not dynamically scale to address multi-GB throughput requirements of many organizations.

Customer gateway

A customer gateway is a physical appliance or a software application that runs in the customer's data center. AWS requires customers to create a customer gateway object in their AWS account before a VPN connection can be established. Customer gateways must have a publicly routable IP address to create a

VPN connection. Note that an existing ASN assigned to the network can be used when creating a customer gateway device. Alternatively, one can use a private ASN within the 64512-65534 range. Once created, administrators need to configure the customer gateway to establish a VPN connection. Fortinet has worked with the AWS team to provide a configuration template directly downloadable from AWS.



Route table – private subnet

Destination	Target	Active	Propagated
10.0.0.0/16	Local	yes	No
10.17.0.0/16	vgw-id	yes	yes

Figure 6: AWS Site-to-Site VPN connection

As shown in Figure 6, each Site-to-Site VPN connection has two tunnels, with each tunnel using a unique virtual private gateway public IP address. It is important to configure both tunnels for redundancy so that when one tunnel becomes unavailable (e. g., down for maintenance), network traffic is automatically routed to the available tunnel for that specific Site-to-Site VPN connection.

Transit gateway

Until recently, there were limited options for organizations that wanted to interconnect their VPCs. They could create point-to-point peering to manage networking at each VPC, which added a great deal of complexity. Or they could create IPsec tunnels from each VPC to third-party router and firewall appliances in a shared VPC. This results in a hub-and-spoke topology called "transit VPC."

Tech Tip #3

Virtual private gateways can only scale to support up to 1.25 Gbps network throughput. Also, they do not support equal-cost multi-path (ECMP) routing.

Tech Tip #4

To establish IPsec tunnels, customer gateways must initiate the tunnel.

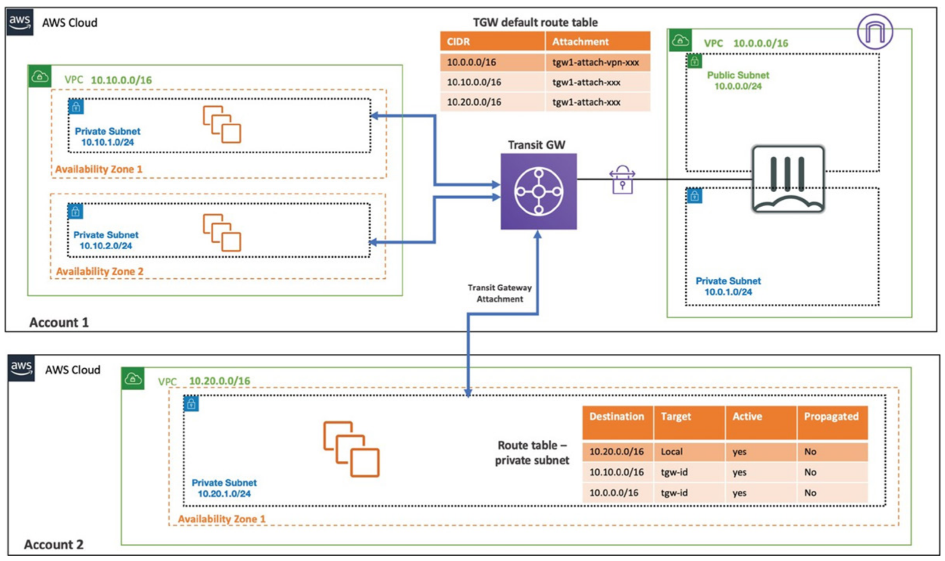
While transit VPC deployments, such as Fortinet Transit VPC, have become a preferred approach to address inter-VPC connectivity and security requirements, an AWS virtual private gateway—deployed at each VPC spoke to terminate VPN connections—has serious bandwidth restrictions, thus limiting network performance.

The AWS Transit Gateway resolves this issue through a distributed service that allows connectivity at scale. Because it supports equal-cost multi-path (ECMP) routing, traffic can be equally distributed over two or more VPN connections that propagate the same IP prefix.

A transit gateway works across AWS accounts but can only attach to VPCs that are within the same Region.

Key Characteristics of Transit Gateways

- Customers can attach a VPC or VPN connection to a transit gateway.
- A transit gateway has a default route table and can optionally have additional route tables.
- By default, the VPCs and VPN connections that attach to a transit gateway are associated with the default transit gateway route table.
- Each transit gateway attachment is associated with exactly one route table.
- Each route table can be associated with zero to many attachments.
- Only one transit gateway attachment can be attached to an Availability Zone within a VPC.
- A VPC or VPN connection can dynamically propagate routes to a transit gateway route table.


AWS Cloud
VPC 10.10.0.0/16
Private Subnet 10.10.1.0/24
Availability Zone 1
Private Subnet 10.10.2.0/24
Availability Zone 2
Transit GW
Transit Gateway Attachment
VPC 10.0.0.0/16
Public Subnet 10.0.0.0/24
Private Subnet 10.0.1.0/24
Account 1
AWS Cloud
VPC 10.20.0.0/16
Private Subnet 10.20.1.0/24
Availability Zone 1
Account 2

TGW default route table

CIDR	Attachment
10.0.0.0/16	tgw1-attach-vpn-xxx
10.10.0.0/16	tgw1-attach-xxx
10.20.0.0/16	tgw1-attach-xxx

Route table – private subnet

Destination	Target	Active	Propagated
10.20.0.0/16	Local	yes	No
10.10.0.0/16	tgw-id	yes	No
10.0.0.0/16	tgw-id	yes	No

Figure 7: AWS transit gateway architecture

The transit gateway in Figure 7 is attached to two VPCs in two different AWS accounts. All attachments are associated with the default transit gateway route table. The route table in the VPC for the second account has been updated to include routes to the transit gateway. One transit gateway attachment object of type VPN in account one has been created to connect the transit gateway to the FortiGate NGFW in the hub VPC. Also, two transit gateway attachments, one per each AZ, have been created to ensure high availability of the connection between the spoke VPC in account one and the transit gateway.

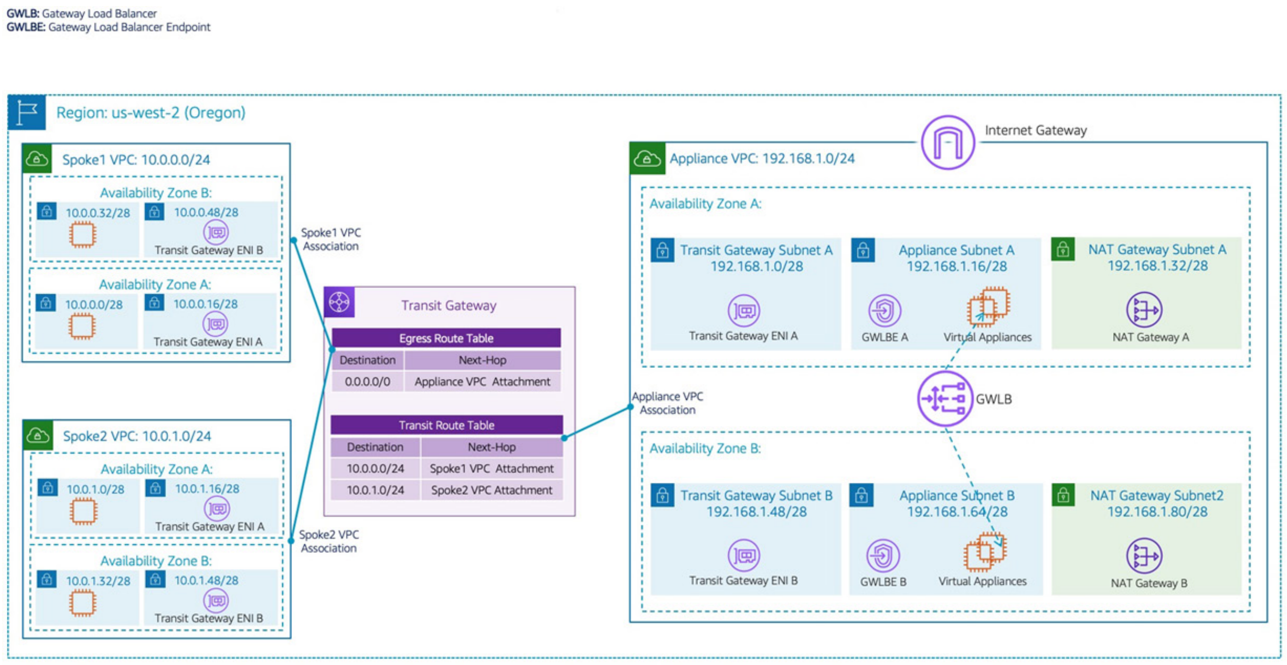
Additional AWS Services mentioned in this guide:

AWS Event Bridge

Amazon Event Bridge is a serverless event bus that makes it easier to build event-driven applications at scale using events generated from applications, integrated Software-as-a-Service (SaaS) applications, and AWS services. Event Bridge delivers a stream of real-time data from event sources such as Zendesk or Shopify to targets like AWS Lambda and other SaaS applications. You can set up routing rules to determine where to send your data to build application architectures that react in real time to your data sources with event publisher and consumer completely decoupled.

AWS Gateway Load Balancer

Gateway Load Balancer (GWLB) helps you easily deploy, scale, and manage your third-party virtual appliances. It gives you one gateway for distributing traffic across multiple virtual appliances while scaling them up or down, based on demand. This decreases potential points of failure in your network and increases availability.



Egress Route Table

Destination	Next-Hop
0.0.0.0/0	Appliance VPC Attachment

Transit Route Table

Destination	Next-Hop
10.0.0.0/24	Spoke1 VPC Attachment
10.0.1.0/24	Spoke2 VPC Attachment

Gateway Load Balancer shown here with transit gateway for multi-VPC inspection capabilities, across accounts

GWLB

When you create a VPC endpoint service, you specify the GWLB. Other AWS principals access the endpoint service by creating a GWLB endpoint. This allows for connectivity from VPCs to the GWLB.

EKS

Amazon Elastic Kubernetes Service (Amazon EKS) is a managed Kubernetes service that makes it easy for you to run Kubernetes on AWS and on-premises. Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications. Amazon EKS is certified Kubernetes-conformant, so existing applications that run on upstream Kubernetes are compatible with Amazon EKS.

Auto-Scale

AWS Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost. Using AWS Auto Scaling, it's easy to set up application scaling for multiple resources across multiple services in minutes. The service provides a simple, powerful user interface that lets you build scaling plans for resources including Amazon EC2 instances and Spot Fleets, Amazon ECS tasks, Amazon DynamoDB tables and indexes, and Amazon Aurora Replicas. AWS Auto Scaling makes scaling simple with recommendations that allow you to optimize performance, costs, or balance between them. If you're already using Amazon EC2 Auto Scaling to dynamically scale your Amazon EC2 instances, you can now combine it with AWS Auto Scaling to scale additional resources for other AWS services. With AWS Auto Scaling, your applications always have the right resources at the right time.

AWS Network Firewall

AWS Network Firewall is a managed service that makes it easy to deploy essential network protections for all of your Amazon Virtual Private Clouds (VPCs). The service can be set up with just a few clicks and scales automatically with your network traffic, so you don't have to worry about deploying and managing any infrastructure. AWS Network Firewall's flexible rules engine lets you define firewall rules that give you fine-grained control over network traffic, such as blocking outbound Server Message Block (SMB) requests to prevent the spread of malicious activity. You can also import rules you've already written in common open source rule formats as well as enable integrations with managed intelligence feeds sourced by AWS partners. AWS Network Firewall works together with AWS Firewall Manager so you can build policies based on AWS Network Firewall rules and then centrally apply those policies across your VPCs and accounts.

AWS Web Application Firewall (WAF)

AWS WAF is a web application firewall that helps protect your web applications or APIs against common web exploits and bots that may affect availability, compromise security, or consume excessive resources. AWS WAF gives you control over how traffic reaches your applications by enabling you to create security rules that control bot traffic and block common attack patterns, such as SQL injection or cross-site scripting. You can also customize rules that filter out specific traffic patterns.

Fortinet Security Solutions for AWS

FortiGate Next-Generation Firewall VM

By combining stateful inspection with a comprehensive suite of powerful security features, FortiGate NGFW technology delivers complete content and network protection. This solution is Security Competency certified, and 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, intrusion prevention system (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 (SR-IOV) for higher and more consistent performance.

FortiManager

FortiManager virtual security management appliances for AWS offer the same powerful network security management features as FortiManager hardware-based appliances, with the addition of a stackable license model that enables easy growth with your network environment. Fortinet virtual appliances allow you to deploy a mix of hardware and virtual appliances, operating together and managed from a common, centralized FortiManager platform.

FortiADC-VM on AWS

FortiADC provides unmatched application acceleration, load balancing, and web security, regardless of whether it is used for applications within a single data center or serves multiple applications for millions of users around the globe. FortiADC includes application acceleration, WAF, IPS, SSL inspection, link load balancing, and user authentication in one solution to deliver availability, performance, and security in a single, all-inclusive license.

Fortinet Managed Rules for AWS WAF: Complete OWASP Top 10

The Complete OWASP Top 10 Ruleset provides a comprehensive package for web application protection through the AWS WAF. Fortinet offers these rules to help cover the entire list of OWASP Top 10 web application threats. Includes protection for SQL injection, cross-site scripting, general and known exploits, malicious bots, and Common Vulnerabilities and Exposures (CVE). These rules provide FortiGuard threat intelligence for AWS Security Services.

Benefits of FortiGate-VM on AWS

- Protect against known exploits and malware using continuous threat intelligence provided by FortiGuard Labs security services
- Identify thousands of applications including cloud applications for deep inspection into network traffic
- Protect against unknown attacks using dynamic analysis and provide automated mitigation to stop targeted attacks
- Automate incident response and threat intelligence from AWS GuardDuty threat-detection service
- Natively integrate with AWS accounts using the Fortinet fabric connector to dynamically enforce security policies across customers' multi-cloud environments

FortiWeb

Using a multilayered and correlated approach, FortiWeb intelligently and accurately protects your web applications from the OWASP Top 10 threats. Combined with Fortinet Web Application Security Service from FortiGuard Labs, FortiWeb keeps your applications safe from vulnerability exploits, bots, malware uploads, denial-of-service (DoS) attacks, advanced persistent threats (APTs), and zero-day attacks. Available as hardware, VMs, and as-a-Service, FortiWeb provides world-class WAF protections in the deployment model your enterprise needs.

FortiCNP

FortiCNP, Fortinet's cloud-native protection solution, helps security practitioners prioritize risk management activities based on a broad set of security signals across their cloud environment. FortiCNP includes cloud security posture management (CSPM) services that continuously discover, monitor, and detect cloud risk across multi-cloud environments. FortiCNP collects information from cloud-native vulnerability scanning, permissions analysis, threat detection, and other security services as well as Fortinet Cloud Security products to calculate an aggregate risk score for cloud resources, so customers can prioritize risk management work based on the context-rich Resource Risk Insights (RRI) that FortiCNP produces. Unlike traditional cloud security posture management (CSPM) and cloud workload protection platform (CWPP) products, by integrating with cloud-native security services and other Fortinet Security Fabric products, FortiCNP provides broad security visibility across your cloud footprint and helps you prioritize security workflows for effective risk management.

FortiGuard Security Services

FortiGuard Labs offers real-time intelligence on the threat landscape, delivering comprehensive security updates across the full range of Fortinet solutions.

Composed of security threat researchers, engineers, and forensic specialists, FortiGuard Labs collaborates with the world's leading threat-monitoring organizations and other network and security vendors, as well as law enforcement agencies.

Fortinet Licensing

With a multitude of deployment methods supported across various private and public cloud deployments, Fortinet Security Solutions for AWS supports on-demand pay-as-you-go (PAYG) and bring-your-own-license (BYOL) models.

BYOL is annual perpetual licensing available for purchase from resellers or distributors. It provides the same ordering practice across all private and public clouds, no matter what the platform is. Customers must activate a license the first time they access the instance before using various features. It is ideal for migration use cases, where an existing private cloud deployment is migrated to a public cloud deployment. When using an existing license, the only additional cost would be the price for the AWS instances.

PAYG is a highly flexible option for both initial deployments and growing them as needed. With a wide selection of supported instance types, there is a solution for every use case. This license offers FortiOS with the unified threat management (UTM) bundle. To purchase PAYG, subscribe to the product on the AWS Marketplace.

SaaS offerings are also available on AWS Marketplace and are metered based on consumption. FortiCNP is metered by protected hosts per month, while FortiWeb Cloud WAF-as-a-Service is metered by cumulative throughput per application. SaaS licensing is available on AWS Marketplace, or through your preferred reseller.

Instance type support

FortiGate-VM supports the following instance types on AWS:

Instance Category	Instance type	vCPU	Max NIC (enabled by AWS)	FortiGate minimum order (BYOL) to consume all instance CPU
General purpose	T2.small	1	2	FG-VMO1 or FG-VM01v
Compute optimized	C4.large	2	3	FG-VMO2 or FG-VM02v
	C4.xlarge	4	4	FG-VMO4 or FG-VM04v
	C4.2xlarge	8	4	FG-VMO8 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-VMO2 or FG-VM02v
	C5.xlarge	4	4	FG-VMO4 or FG-VM04v
	C5.2xlarge	8	4	FG-VMO8 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-VMO2 or FG-VM02v
	C5d.xlarge	4	4	FG-VMO4 or FG-VM04v
	C5d.2xlarge	8	4	FG-VMO8 or FG-VM08v
	C5d.4xlarge	16	8	FG-VM16 or FG-VM161v
	C5d.9xlarge	36	8	FG-VMUL or FG-VMULv
	C5d.18xlarge	72	15	
	C6i.large	2	3	FG-VMO2 or FG-VM02v
	C6i.xlarge (recommended by default)	4	4	FG-VMO4 or FG-VM04v
	C6i.2xlarge	8	4	FG-VMO8 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	

Table 1: FortiGate-VM instance type support on AWS

FortiWeb-VM Instance Type Support:

Instance type	vCPU
C5.large	2
C5.xlarge	4
C5.2xlarge	8
R5.xlarge	4
R5.2xlarge	8

FortiADC-VM Instance Type Support:

Instance type	vCPU
C5.large	2
C5.xlarge	4
C5.2xlarge	8
C5.4xlarge	16
M5.large	2
M5.xlarge	4
M5.2xlarge	8
M5.4xlarge	16

FortiManager-VM Instance Type Support:

Instance type	vCPU
M5.large	2
M5.xlarge	4
M5.2xlarge	8
M5.4xlarge	16
M5.8xlarge	32
M5.12xlarge	48
M5.16xlarge	64
M5.24xlarge	96

Fortinet strongly recommends utilizing C5, M5, C6i, or C6g instance types to take advantage of AWS enhanced networking to achieve the maximum network throughput. The C6g instances are powered by AWS Graviton, and currently only available for FortiGate.

The following table shows example BYOL models available to order:

Model Name	vCPU – minimum	vCPU – maximum
FG-VM01 or 01v	1	1
FG-VM02 or 02v	1	2
FG-VM04 or 04v	1	4
FG-VM08 or 08v	1	8
FG-VM16 or 16v	1	16
FG-VMUL or ULv	1	72

Table 2. FortiGate-VM Models (BYOL).

FortiGate launch on AWS

The most basic deployment consists of one FortiGate with two network interfaces, one public interface, and a private interface.

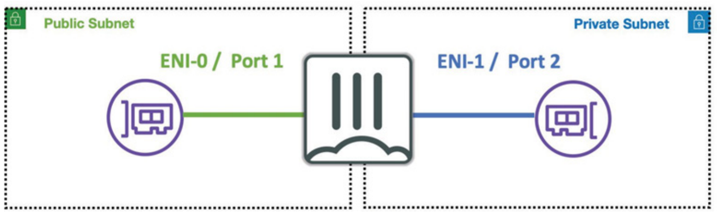


Figure 8: FortiGate-VM on AWS

Enter the following information to create a connector that can connect to an AWS VPC:

- AWS access key ID/secret access key—AWS credentials needed for API access
- AWS Region name
- AWS VPC ID
- Update interval—indicates how often the connector should update the dynamic address object

Bootstrapping FortiGate at initial boot-up

AWS user data is the set of commands/data that can be injected to an Amazon Elastic Compute Cloud (EC2) instance at launch time. It is up to the guest OS of the EC2 instance to consume this data. FortiGate-VM on AWS supports this feature to automate the day-zero configuration and other necessary bootstrapping of the FortiGate instance.

Customers can create an Amazon S3 bucket and upload the license file for BYOL instances, as well as the day-zero configuration file to the S3 bucket. Note that an identity and access management (IAM) role that allows access to the S3 bucket needs to be attached to the FortiGate instance at launch time. The following is an example of user data that is passed to the FortiGate instance to bootstrap the instance at launch time:

```
{
“bucket” : ”unique-bucket-name”,
“region” : “us-west-2”,
“license” : “/FGVM020000000000 .lic”,
“config” : “/fgtconfig-init .txt”,
}
```

Fortinet Fabric Connector

In increasingly dynamic network environments, security solutions must be more tightly coordinated with networking and other IT infrastructure to provide agility in the face of fast-paced and rapidly changing operations.

The adoption of hybrid cloud, which is isolated sets of public clouds, private clouds, and physical entities, requires different security management methodologies, which have become burdens to administrators. Additionally, inconsistent security management with an assortment of security solutions at different sites and tenants makes maintaining security posture across different clouds a daunting task for any organization.

Fortinet Fabric Connectors feature APIs and other interfaces to make them highly extensible platforms. They provide out-of-the-box or built-in integration mechanisms and orchestration of FortiGate or FortiManager with key software- defined network (SDN) and public cloud solutions—including AWS.

Dynamic address objects

Fortinet Fabric Connectors for SDN (private clouds) and cloud (public clouds) enable either FortiGate as a standalone system or FortiManager, which manages multiple FortiGate NGFWs, to integrate with the third-party SDN or cloud platforms. In the case of FortiManager, it synchronizes dynamic address group objects that are protected by the FortiGate firewall policy.

Regardless of how objects change their forms and locations in elastic and volatile fashions, FortiGate identifies them as address objects that can be used as sources and destinations. It then applies appropriate firewall policies automatically without the administrator’s manual intervention.

FortiManager is an optional component. The remainder of this section assumes customers directly use FortiGate to set up an address object and firewall policies.

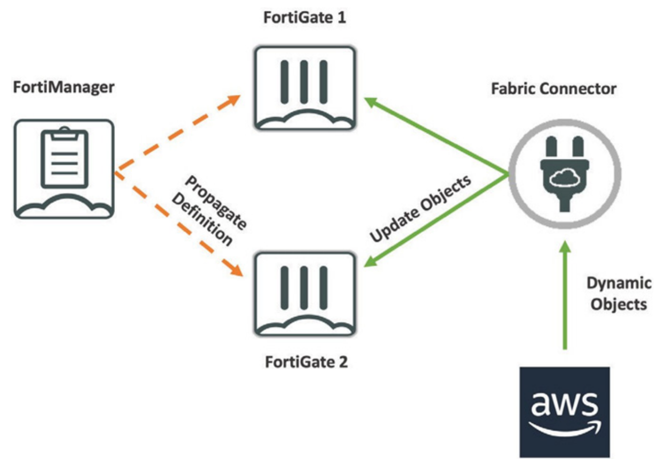


Figure 9: Fortinet Fabric Connector

To create a dynamic address object for AWS resources, an AWS connector must first be created using FortiGate CLI/API/GUI.

The next step in the process is to define an address object using the AWS connector. Customers can define address objects based on different AWS attributes such as instance ID, instance type, subnet ID, etc. Address objects can also be defined based on tags associated with the resources.

Example: Create a firewall policy using dynamic address objects

Once an address object of type AWS fabric connector is created, customers can use that address to configure a firewall policy as a source or destination. Figure 10 illustrates a dynamic address object used in a firewall policy in the FortiGate deployed in a customer's AWS account.

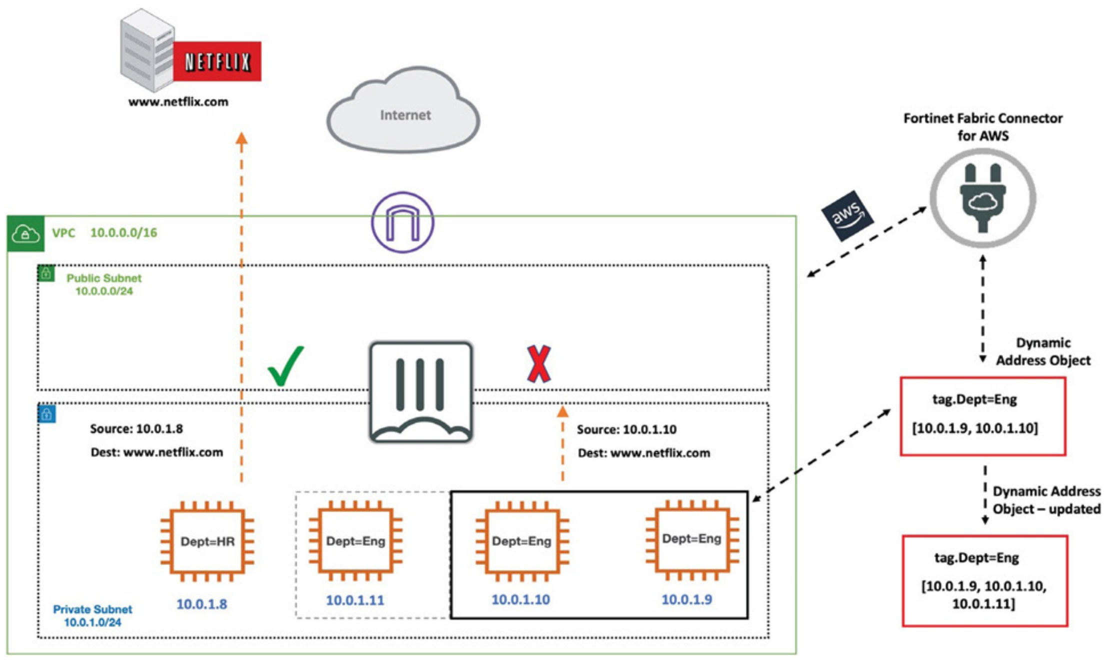


Figure 10: Creating firewall policy using dynamic address objects

In this example, the customer wants to block all engineering resources tagged with Dept=Eng from accessing audio/video streaming websites. After a dynamic address object group with "tag .Dept=Eng" filter rule is created, it initially contains two EC2 instances (10.0.1.9 and 10.0.1.10). A firewall policy with that address object as the source address is created. This policy blocks all accesses originating from the address object destined to the audio/video streaming websites (such as Netflix.com). Thus, when a new EC2 instance is launched by the engineering team, that instance will automatically join that address object if the instance is tagged appropriately. Any attempt to connect to a streaming website from the new instance will be blocked without requiring the administrator to manually configure the firewall.

Deploying a single FortiGate-VM on AWS

Customers can deploy FortiGate to secure their environment and apply deep-packet inspection to both incoming and outgoing traffic. In the next section, packet flow in each single FortiGate deployment scenario is discussed.

Inbound traffic inspection (north-south)

Customers who have adopted both AWS and web applications need to protect their applications against sophisticated cyberattacks. When deployed in a VPC, FortiGate-VM can help achieve this goal by inspecting all inbound traffic originated from the internet. A common approach in protecting multiple services behind a single FortiGate-VM is to map each service IP address/port to a port and the single elastic IP address (EIP) associated with the FortiGate NGFW's public network interface. For example, the web service in Figure 11 is hosted on port 8080 and is mapped to the FortiGate EIP/port 8080.

All traffic destined to the web server is inspected by the FortiGate-VM. The internet gateway (IGW) attached to the VPC enables internet connectivity and also performs network address translation for the private IP address of FortiGate's ENI-0 and the EIP associated with that interface.

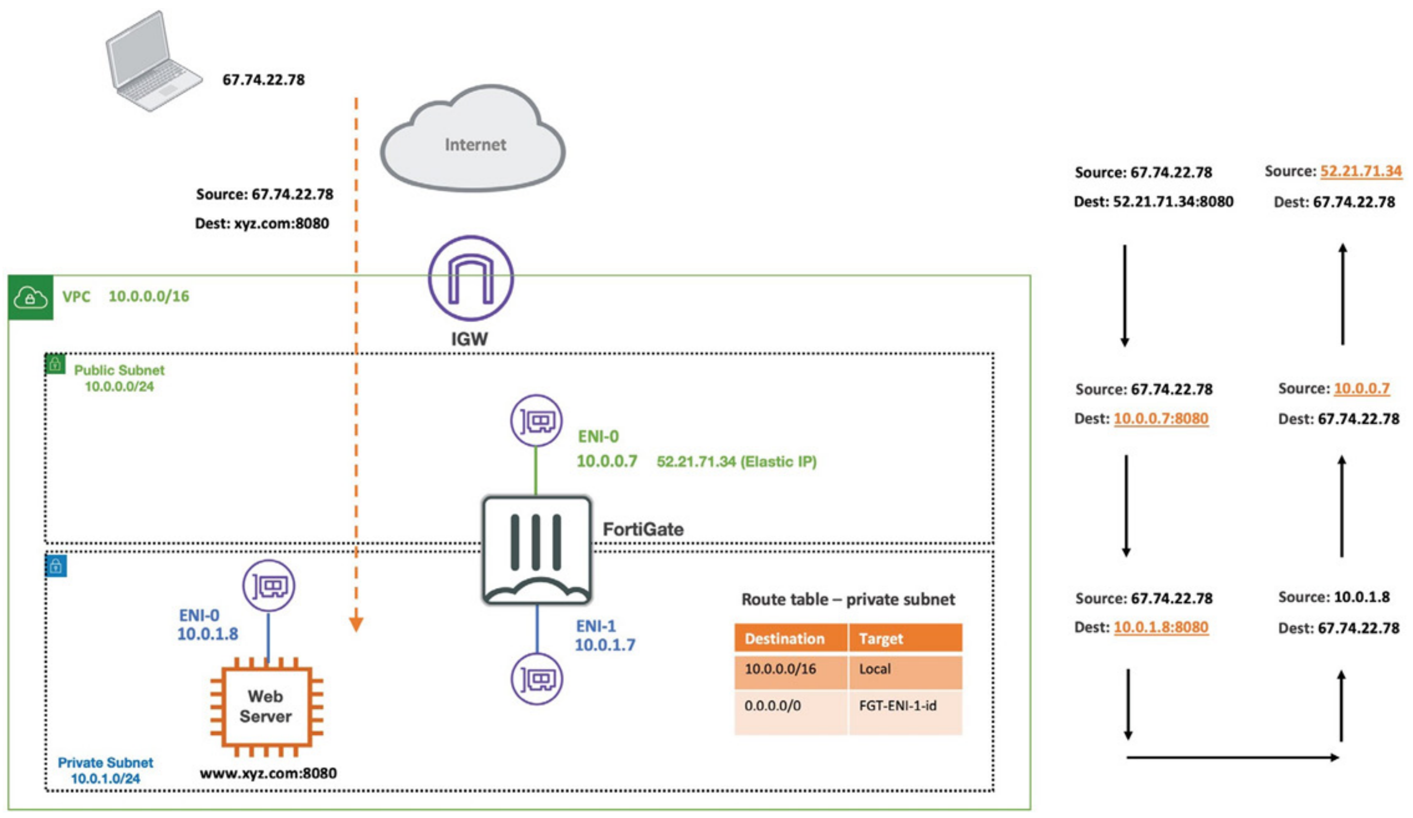


Figure 11: Inbound traffic inspection with FortiGate-VM

Figure 11 illustrates the incoming traffic packet flow as well the path for the return traffic.

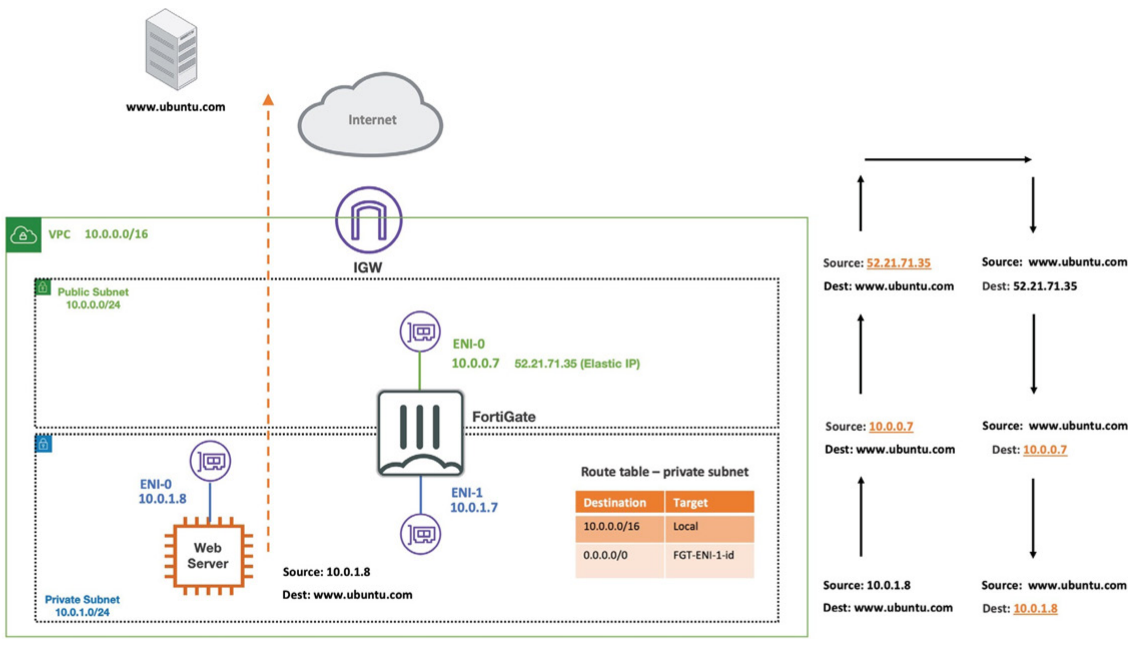
Inbound traffic. IGW translates the destination IP address (which is the EIP associated with the FortiGate's ENI-0) to the private IP address of FortiGate's ENI-1. The FortiGate NGFW will again apply destination NAT to change the destination IP address to the IP address of the back-end web server.

Return traffic. Traffic returning from the web server will follow the private subnet's route table, which includes a route entry to point all internet-bound traffic to the FortiGate. As shown in Figure 11, both FortiGate and IGW apply source NAT to the return traffic to change the source IP to a publicly routable IP address (EIP associated with the FortiGate NGFW's ENI-0).

Outbound traffic inspection

Although web application traffic is often originated from the internet, there are instances where a web application needs to originate a request to the internet. For example, a managed web service or a web server administrated by a customer may require downloading security patches or performing live updates. Customers can add to their private subnet's route table a default route that points to the FortiGate private interface as the next hop. This allows all internet-bound traffic to be inspected by FortiGate. Figure 11 shows how source NAT is applied at an internet-bound request originated by the web server as well as destination NAT applied at the return traffic.

To enable FortiGate-VM to forward traffic, it is required that the source/destination check flag is disabled at the network interface level, as explained earlier in this document.



Route table – private subnet

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	FGT-ENI-1-id

Figure 12: Outbound traffic inspection with FortiGate-VM

In the example shown in Figure 12, the web server in the private subnet needs to initiate an internet-bound request to download an OS patch. As illustrated in the figure, both FortiGate-VM and IGW apply source NAT to the outbound traffic. Also, they apply destination NAT to the response traffic to translate the EIP and the private IP address of the FortiGate's ENI-0.

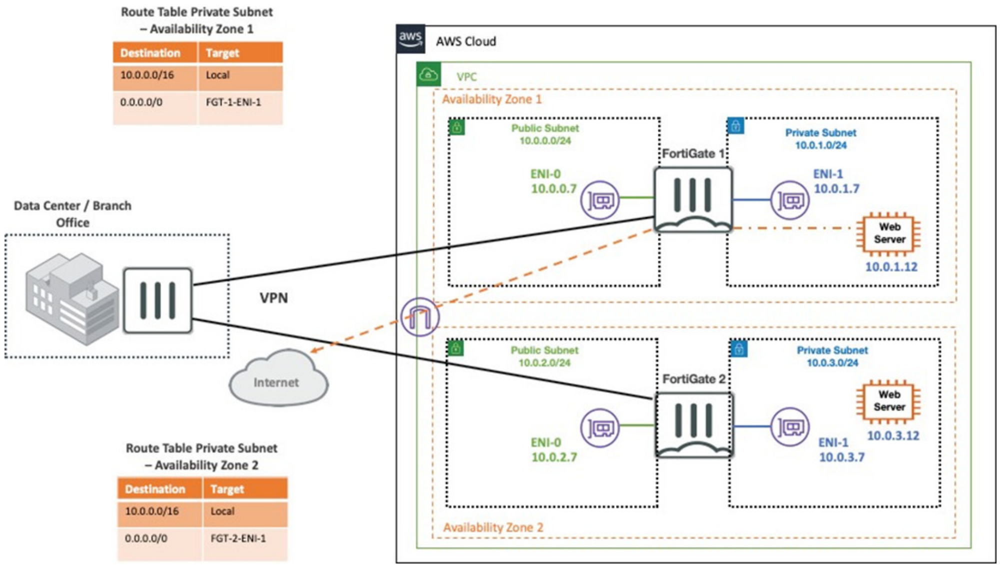
Multi-VPC design. AWS recommends segmenting networks at the VPC level. In this approach, workloads are grouped together at the VPC level instead of the subnet level. All traffic between VPCs will be inspected by network security virtual firewalls at each VPC or at a shared VPC. Design patterns such as Transit VPC or AWS Transit Gateway can be used to achieve this in an automated and scalable fashion.

FortiGate High Availability and Resiliency in AWS

As covered earlier in this document, unique security challenges in the cloud warrant an innovative approach to address those challenges. Customers that deploy NGFW security solutions in their production environment often consider a highly available and resilient design to be of paramount importance. To address this concern, Fortinet supports deploying FortiGate NGFW in AWS in a number of different resilient architectures.

Resilient outbound connection to on-premises network

Although more organizations are rapidly adopting clouds and migrating their workloads, many intend to keep a sizable portion of their workloads on-premises. A FortiGate high-throughput NGFW can be deployed as a VPN gateway in an AWS VPC to connect customers' cloud workloads back to their physical data centers by establishing VPN tunnels. They can be deployed in a pair to provide resiliency for the outbound connection to the physical data center, as shown in Figure 13.


Route Table Private Subnet – Availability Zone 1

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	FGT-1-ENI-1

Route Table Private Subnet – Availability Zone 2

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	FGT-2-ENI-1

Figure 13: Resilient outbound connection to physical data center

Although AWS virtual private gateways can be used to terminate VPN tunnels in AWS VPCs, it is highly recommended to use FortiGate NGFWs instead, as they can support much higher bandwidth than AWS virtual private gateways.

Figure 13 illustrates two FortiGate-VMs deployed in two different AZ to provide resiliency. Each private subnet has a route table that contains a route that points to the ENI-1 of the FortiGate in its respective AZ. In the event of downtime in an AZ, this design ensures that workloads in the other AZ have continued outbound connectivity.

Because FortiGate NGFWs support BGP, they can learn routes dynamically from the on-premises network if the customer gateway also supports BGP. Additionally, each FortiGate can act as a default gateway for all internet-bound traffic.

Active-passive deployment with Unicast HA

Traditionally, customers have often deployed FortiGate NGFWs in a highly available manner in their data center to achieve session synchronization as well as resiliency using FortiGate Clustering Protocol (FGCP). Heartbeat traffic used in this protocol employs multicast packets, but AWS VPC networking does not allow use of multicast/broadcast packets.

In order to support a similar deployment in AWS where a pair of FortiGate instances can synchronize sessions, Fortinet has designed Unicast HA to provide an active-passive clustering solution for deployments in AWS. This solution works with two FortiGate instances configured as a collaborative pair and requires that the instances are deployed in the same subnets and same AZ within a single VPC. These FortiGate instances act as a single logical instance and share interface IP addressing. Configuration synchronization allows customers to configure a cluster in the same way as a standalone FortiGate unit. If a failover occurs, the cluster recovers quickly and automatically and can also send notifications to administrators so that the problem that caused the failure can be corrected and any failed resources restored.

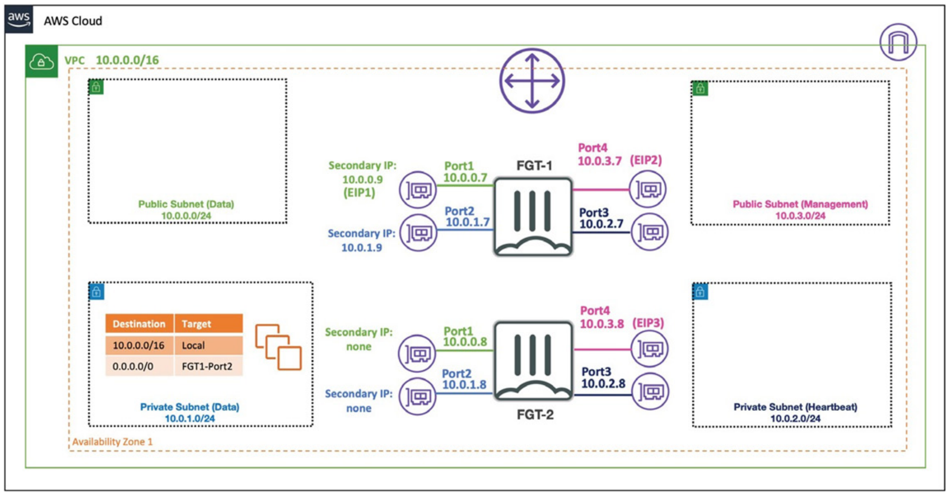


Figure 14: FortiGate FGCP deployment in AWS

Inbound failover is provided by reassigning the secondary IP addresses of ENI0\port1 from FortiGate 1's public interface to FortiGate 2's public interface. Additionally, the EIPs associated to the secondary IP addresses of ENI0\port1 are re-associated from FortiGate 1's public interface to FortiGate 2's public interface.

Outbound failover is provided by reassigning the secondary IP addresses of ENI1\port2 from FortiGate 1's private interface to FortiGate 2's private interface. Also, any route targets referencing FortiGate 1's private interface will be updated to reference FortiGate 2's private interface. Figure 14 and Figure 15 illustrate the failover process of FGCP deployment in an AWS VPC.

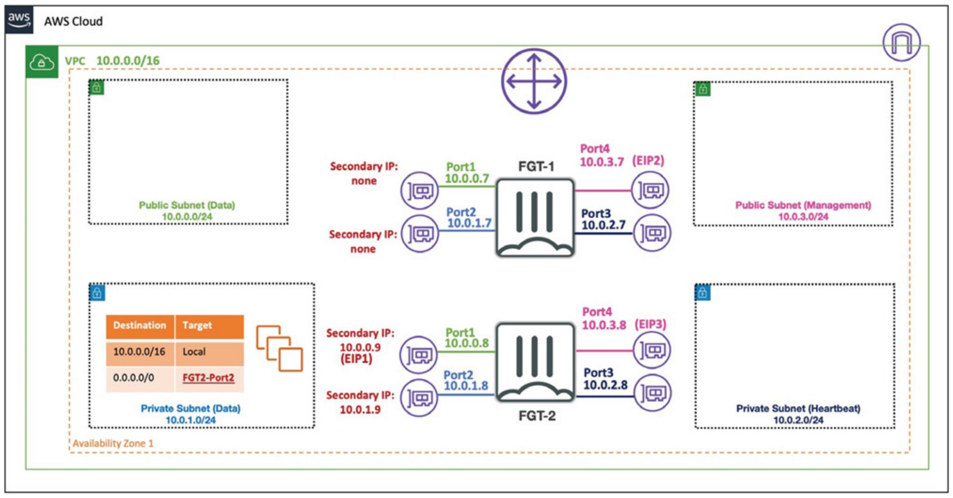


Figure 15: FortiGate FGCP failover process in AWS

Components of FGCP in AWS

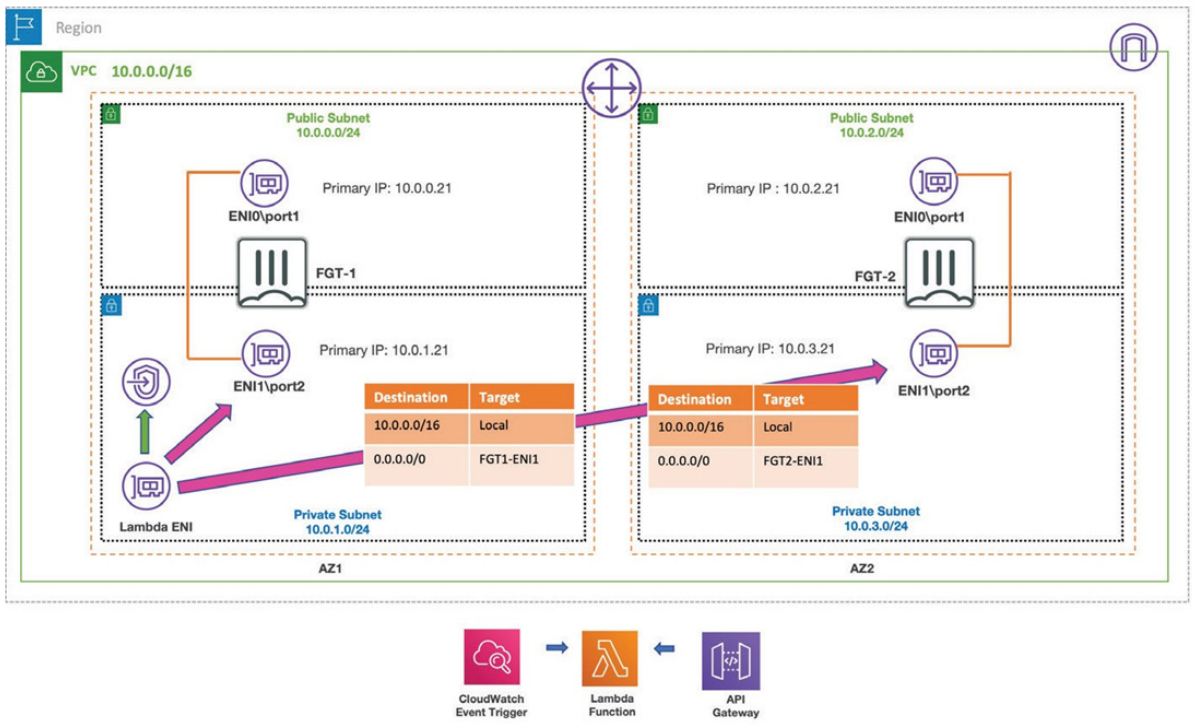
- Two FortiGate NGFWs, each with four network interfaces
- Two dedicated ENIs, one for a public interface (ENI0\port1) and another for a private interface (ENI1\port2). These ENIs will utilize secondary IP addressing to allow both FortiGate instances to share the same IP address within the actual FortiOS configuration and sync sessions natively. Note that AWS does not allow modification of an ENI's primary IP, thus secondary IP addressing must be used.
- A cluster EIP will be associated to the secondary IP of the public interface (ENI0\port1) of the current master FortiGate instance and will be reassociated to a new master FortiGate instance as well.
- A dedicated ENI (ENI2\port3) for FGCP HA communication to perform tasks such as heartbeat checks, configuration sync, and session sync
- Another dedicated ENI (ENI3\port4) for management access to each instance and also allow each instance to independently and directly communicate with the public AWS EC2 API
- AWS IAM role. By assuming this role, FortiOS will be authorized to access the AWS EC2 API to perform actions such as updating route tables, reassigning cluster EIPs, etc.

Resiliency for outbound traffic: active-active failover

While active-passive deployments provide a resilient architecture, only one of the FortiGate instances can be utilized at a given time. Active-active FortiGate failover shown in Figure 18 provides automated AWS route table updates to maintain egress traffic flow through two independent FortiGate instances in separate AZ. FortiGate instances are utilized to handle the traffic. Note that this solution only provides outbound resiliency for traffic leaving the VPC. For ingress resiliency, external resources such as AWS ELB or Route 53 services can be utilized.

Components of Active-Active Outbound Traffic

- FortiGate instances
- Lambda function
- CloudWatch event rule
- API gateway
- VPC endpoint



Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	FGT1-ENI1

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	FGT2-ENI1

Figure 18: FortiGate active-active resiliency for outbound traffic

The solution shown in Figure 18 is very similar to the Lambda-based active- passive deployment covered in the previous section. They are both composed of the same components.

The main difference, however, is that route tables in two AZ have a route entry that points to a different FortiGate private interface. This enables both FortiGate instances to be utilized based on the AZ in which the resource that originates the traffic resides. Figure 19 depicts the failover process in this design. Outbound failover is provided by updating any routes currently targeting FortiGate 1's private interface to target FortiGate 2's private interface (ENI1).

When FortiGate 1 comes back online and passes health checks, the list of routes for AZ1 will be updated to target FortiGate 1 as it is the local instance for the AZ.

The AWS SDN and tag updates are performed by the Lambda function initiating API calls (from the ENI automatically created by Lambda within the VPC) through the VPC endpoint interfaces. Reference the detailed step-by-step deployment guide for more information.

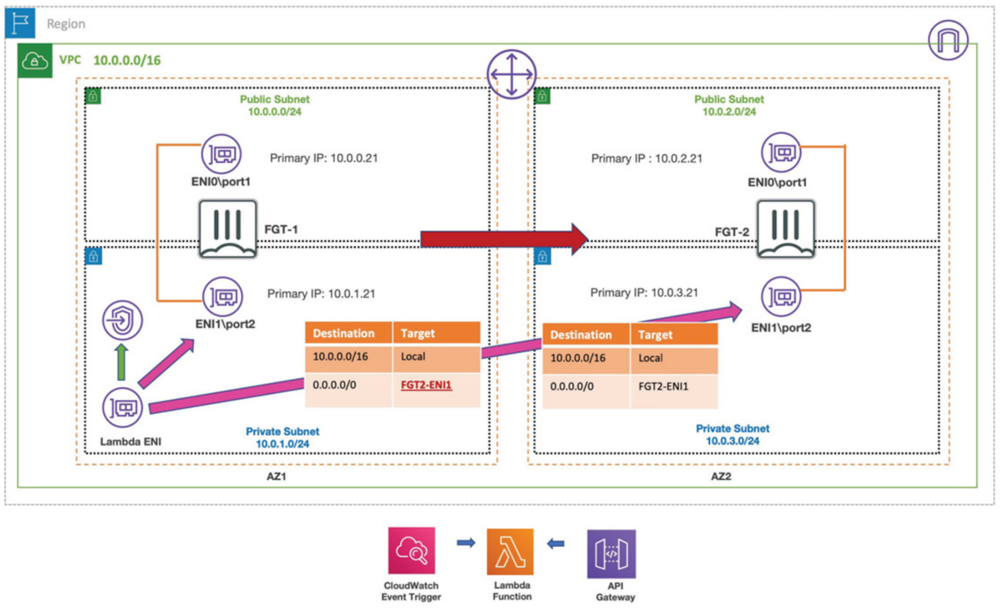


Figure 19: FortiGate active-active resiliency for outbound traffic failover process

FortiWeb high availability (HA)

FortiWeb HA supports active-passive HA and high volume active-active HA, comparable to the architectures described above for FortiGate.

In active-passive HA mode, one of the member instances will be selected as the master node, while the others are slaves. If the master node fails, the slave takes over as the master. The FortiWeb-VMs run heartbeats between dedicated UDP-tunnel and synchronize the master node's configuration to all the members in the HA group. Only the master instance processes traffic. We associate an EIP (Elastic IP address) to the master instance. When the master node fails, the slave immediately takes the master role and processes traffic. The EIP is switched to the new master.

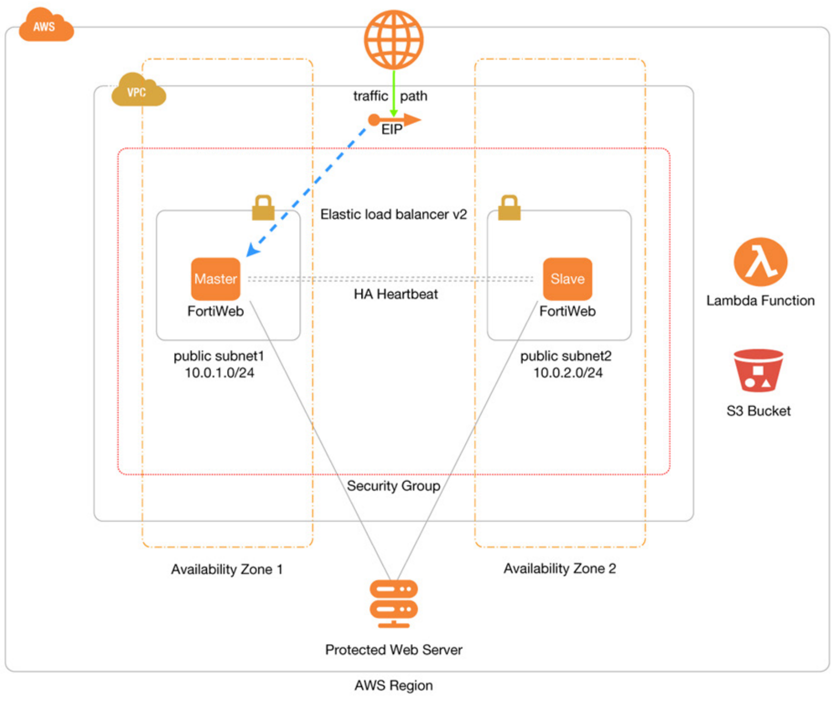


Figure 20: FortiWeb active-passive HA for immediate failover

In high-volume active-active HA mode, all of the instances in the HA group process traffic. We use an external Elastic Load Balancer (ELB) to distribute traffic to all of the HA members. If an instance is down, it will be ignored by the load balancer for traffic distribution. If the failed instances are the master node, one of the slave instances immediately takes its role to become the new master.

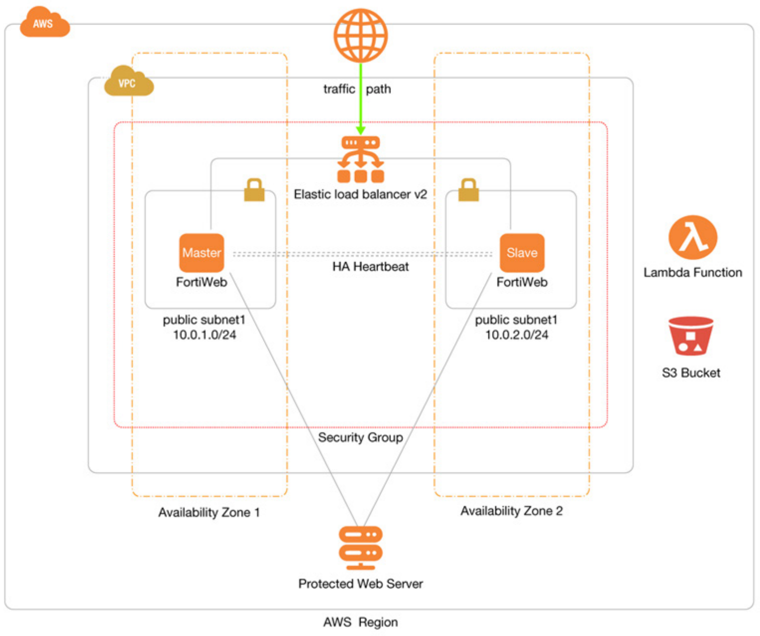


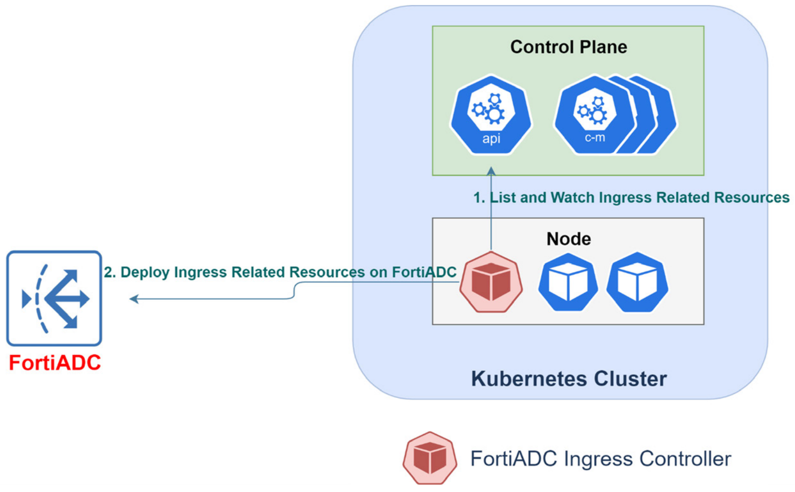
Figure 21: FortiWeb in active-active high-volume mode

Other Architectures

FortiADC Ingress Controller

The FortiADC Ingress Controller fulfills the Kubernetes Ingress resources and allows you to manage FortiADC objects from Kubernetes. It is deployed in a container of a pod in an EKS Kubernetes cluster. The list below outlines the major functionalities of the FortiADC Ingress Controller:

- To list and watch Ingress-related resources, such as Ingress, Service, Node, and Secret
- To convert Ingress-related resources to FortiADC objects, such as virtual server, content routing, real server pool, and more
- To handle Add/Update/Delete events for watched Ingress resources and automatically implement corresponding actions on FortiADC


Control Plane
api
c-m
1. List and Watch Ingress Related Resources
Node
2. Deploy Ingress Related Resources on FortiADC
FortiADC
Kubernetes Cluster
FortiADC Ingress Controller

For more details, see our deployment template.

Securing Your SAP S/4 HANA

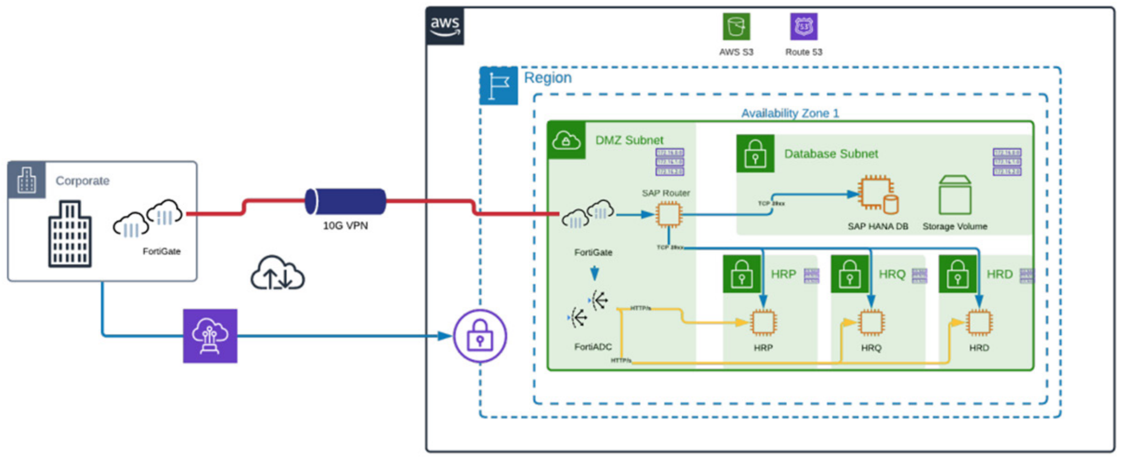


Figure 22: Fortinet reference architecture for SAP S/4HANA

FortiGate delivers high-performance, low-latency SAP security through the deep-packet and content inspection specific to SAP services. The FortiGate, combined with FortiGuard threat intelligence, delivers validated industry-leading IPS technology. FortiGuard Labs provides SAP threat intelligence to the FortiGate's IPS engine to protect from well-known and emerging threats.

FortiADC is an advanced application delivery controller that enhances SAP applications' security, scalability, and performance. FortiADC provides WAF, IPS, SSL inspection, link load balancing, and user authentication in one solution, whether SAP applications are hosted on-premises or in the cloud. FortiADC secures SAP both with SAP connector and by integrating application delivery into the Fortinet Security Fabric. The SAP connector gets changes from the SAP Message Server. All SAP web traffic to the SAP Application Servers is protected with end-to-end encryption using the FortiADC. An intuitive user interface streamlines the configuration of CLI and APIs. Automated configuration gathers information from the SAP ICM configuration (HTTP/HTTPs ports, virtual hosts, etc.) and additional application server instances. The SAP connector provides a topology view of the SAP landscape within the network for easier management and unified visibility for multi-cloud and on-premises SAP deployments.

Design Patterns

In the previous sections of this document, AWS key concepts as well as FortiGate deployment models were discussed in detail. This section focuses on how different AWS services and design patterns can be utilized to deliver scalable and automated security solutions for customers who have decided to migrate their physical data center, partially or entirely, to an AWS environment. When designing security for your infrastructure, take the following into consideration:

Scale. Depending on the number of VPCs in the customers' environment, a design pattern such as Transit VPC, transit gateway, or a single VPC design may be the best option. Also, routing requirements can influence the design decision.

Automation. Due to the distributed and dynamic nature of the cloud (including AWS), utilizing Infrastructure-as-a-Code (IaaC) tools such as Terraform and CloudFormation is of paramount importance. Additionally, native AWS services such as AWS Auto Scaling can help customers efficiently and automatically take advantage of AWS elasticity.

High availability and resiliency. Customers who have workloads in production environments ideally require resilient and highly available security solutions to protect their data and applications. FortiGate resilient deployment options and AWS managed services can help achieve this goal.

Resilient FortiGate deployment integrated with AWS Transit Gateway

The AWS Transit Gateway provides a highly scalable, distributed service that allows connectivity at scale. Because it supports equal-cost multi-path (ECMP) routing, traffic can be equally distributed over two or more VPN connections that propagate the same IP prefix. This allows for significantly more flexibility in the network. And because it is part of the AWS suite, native services such as CloudFormation, CloudWatch, and VPC flow logs can be used to manage and monitor the AWS Transit Gateway.

The AWS Transit Gateway can help route traffic in all directions, including north-south and east-west. The Fortinet Cloud Services Hub leverages the AWS Transit Gateway service to enable and improve upon several critical use cases.

Hybrid cloud. In a typical hybrid-cloud deployment, a large volume of data is continuously transferred between multiple remote branches, the corporate data center, as well as application VPCs. The Fortinet Cloud Services Hub creates a central hub VPC in the cloud to facilitate interconnectivity and traffic inspection. While application VPCs attach directly to the Transit Gateway, physical data center and remote branch locations can connect to the FortiGate NGFW in the Fortinet Cloud Services Hub using ECMP in a scalable fashion.

Inbound application traffic with firewall resiliency. It's often preferable to deploy applications in private subnets in VPCs that do not require any public IP addresses. Yet, the need to protect applications from outside attacks is still present. The Fortinet Cloud Services Hub integrates with the AWS Transit Gateway, allowing customers to conveniently deploy web applications in a private VPC while resilient FortiGate NGFWs are provisioned in a public VPC to protect their applications. Figure 24 depicts a scenario where two FortiGate NGFWs are fronted by an internet-facing AWS load balancer. Back-end services are deployed in two spoke VPCs that connect to a Transit Gateway.

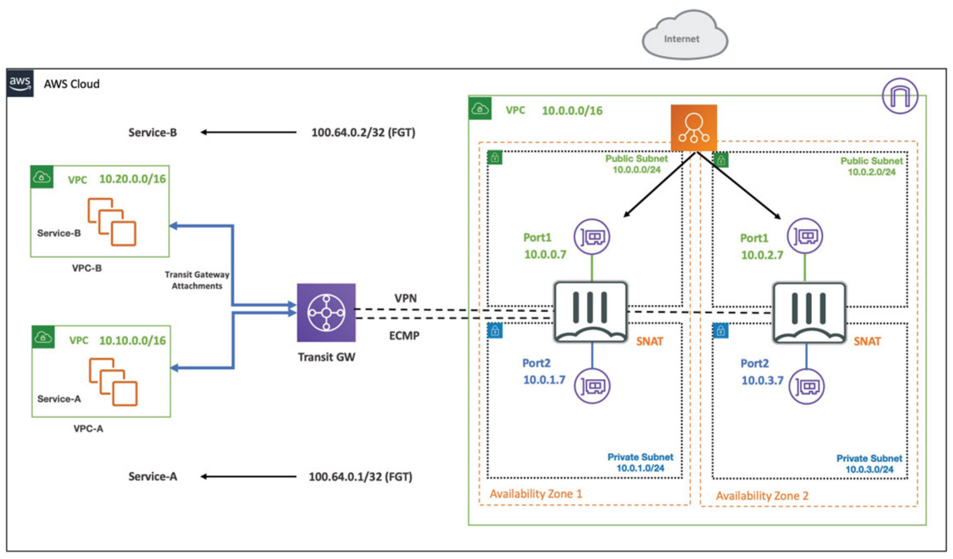


Figure 23: Transit Gateway with firewall resiliency for inbound traffic

Key Components of This Design

- A VPC configured with public and private subnets according to AWS best practices. This deployment spans two availability zones to achieve resiliency.
- In the public subnets, a FortiGate host in an auto-scaling group complements AWS security groups to provide Layer 7 security features such as intrusion protection, web filtering, and threat detection.
- In the public subnets, FortiGate NGFWs that act as NAT gateways, allowing outbound internet access for resources in the private subnets. This also ensures that outbound traffic is inspected by the FortiGate NGFW instance.
- An externally facing NLB. An internally facing NLB is optional.
- Amazon API Gateway, which acts as a front door by providing a callback URL for the FortiGate auto-scaling group.
- Lambda functions are used to handle auto scaling, failover management, and configuration for other related components.
- An Amazon DynamoDB database that uses Fortinet-provided scripts to store information about auto-scaling condition states.

East-west traffic inspection with transit gateway. As zero-trust security deployment strategies are adopted by large and small enterprises, the ability to inspect all traffic is critical.

The Fortinet Cloud Services Hub supports a deployment model where all traffic is inspected to stop the lateral propagation of threats. One way to achieve this when utilizing a transit gateway service is to create multiple route tables in the AWS Transit Gateway. This approach allows all or a subset of inter-VPC traffic to be inspected by the advanced security capabilities in FortiGate (and, by extension, potentially other security solutions across the Security Fabric). All traffic that needs to be inspected is sent to the FortiGate solutions deployed in the Fortinet Cloud Services Hub.

As shown in Figure 24, two Transit Gateway route domains are used in this deployment: one route table that attaches to all spoke VPCs and another one that attaches to the Cloud Services Hub VPC. To ensure flow affinity, source NAT must be applied at each FortiGate NGFW. Additionally, each spoke VPC route table must have a route back to the Cloud Services Hub.

Note that the deployment in Figure 24 assumes active-active FortiGate deployment. Alternatively, customers can create transit gateway attachments between the transit gateway and Cloud Services Hub where FortiGates will be deployed in an active-passive fashion. Customers who prefer to preserve source IP addresses in east-west traffic inspection might choose this deployment as source NAT is not required. Similarly, auto-scaling FortiGates can be positioned in a similar fashion.

Fortinet Transit VPC Key Components

- Two FortiGate instances deployed in the Transit VPC in active-active mode
- Transit VPC (hub) and application VPCs (spoke)
- AWS VPN gateway deployed at each spoke VPC
- Two Lambda functions: virtual private gateway poller and Fortinet VPN configurator
- AWS CloudWatch event rule
- Optional physical data center (customer gateway)

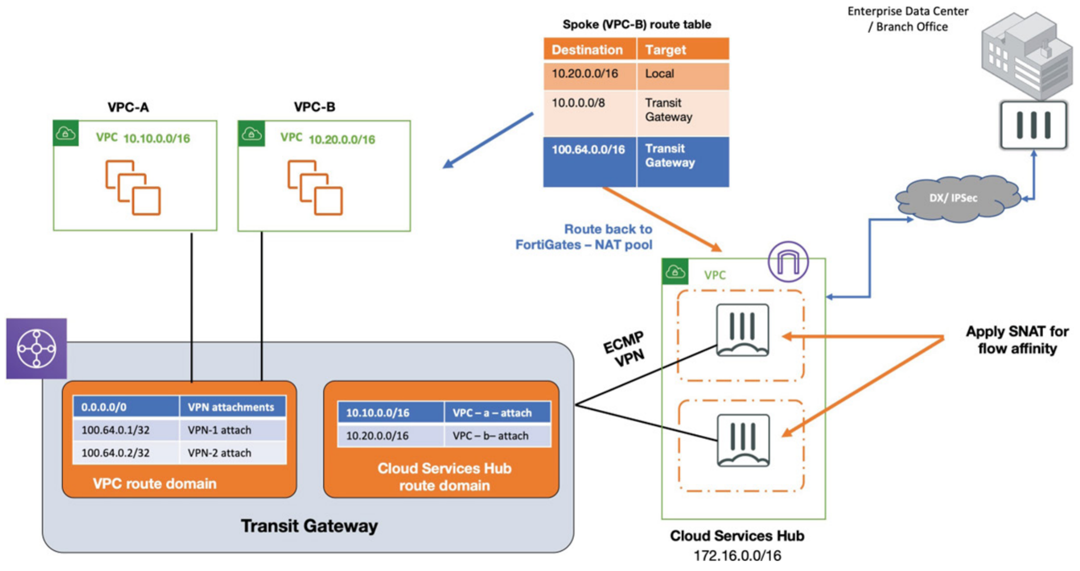


Figure 24: VPC-to-VPC traffic inspection with FortiGate NGFW and AWS Transit Gateway

Elastic Load Balancing with FortiGate auto scaling

AWS Elastic Load Balancing (ELB) is a load-balancing service for AWS deployments. ELB automatically distributes incoming application traffic and scales resources to meet traffic demands. Customers with high-throughput NGFW requirements can rely on FortiGate instances integrated with AWS ELB and AWS Auto Scaling to meet their changing traffic volume. FortiGate NGFW instances, which are placed in an auto-scaling group, are fronted by an external AWS Network Load Balancer (NLB). A second internal NLB can optionally be used to distribute the traffic to the back-end servers located in private subnets. This deployment is commonly referred to as an ELB Sandwich, as FortiGate instances are sandwiched between the two NLBs.

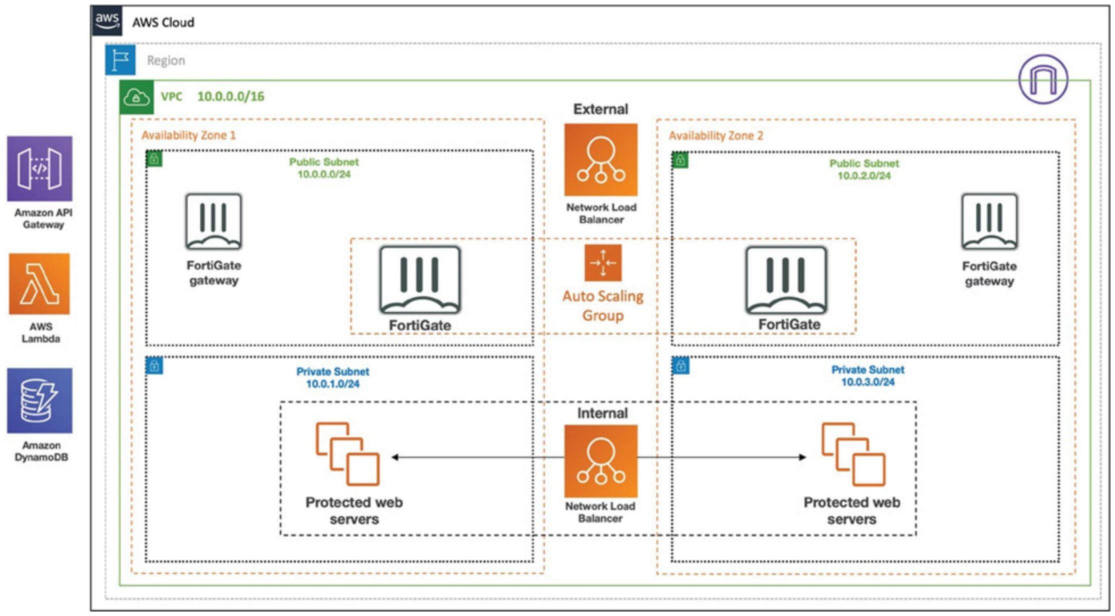


Figure 25: FortiGate auto scaling in ELB Sandwich deployment

FortiGates use API gateways to send API calls and to process FortiGate Config- Sync tasks to synchronize operating system configuration across multiple FortiGate instances at the time of the auto-scaling event. This is currently only for internal use in the VPC. There is no public access available.

Incoming requests to the web servers in the private subnets present in your existing VPC will go through a connection that flows through the internet gateway, NLB, and the FortiGate auto-scaling group before reaching the web server. The web server returns the response using the same connection.

Outgoing requests from the web servers go through the individual FortiGate NAT gateway and the internet gateway to the external network. The external network returns the response using the same path.

This solution is fully automated and can be deployed to existing or new VPCs. It is available as an AWS Quick Start Guide.

Highly Scalable Security Inspection for Customer VPCs with GWLB

The newest in network services from AWS includes the GWLB. While transit gateway was used for north-south and east-west inspection, the consolidated egress and ingress came with limitations around total inbound applications and routing complexities for multi-account and many-VPC deployments. GWLB allows for inspection, without the routing complexities, and allows for individual customers to own their routing/network, while still giving visibility and inspection capabilities to their security teams.

GWLB is a fully managed service from Amazon, with support to include in your IaC deployments, along with the native visibility of VPC flow logs and the full suite of AWS supporting services.

Use case 1: North-South Inspection

In typical AWS deployments, most of the application instances in a VPC reside in a private subnet and are blocked from accessing resources outside the local network. But some application instances need to be accessible to users over the internet, and in some other cases applications or servers need to access other services, such as automatic software updates. In these cases, the traffic to and from the internet must be inspected to prevent attacks and reduce the risk of breaches. For these reasons, customers can deploy FortiGate-VM with the GWLB service to protect their application instances.

The first option is to use AWS Gateway Load Balancer Endpoints (GWLBE) from the customers' VPCs. GWLBE makes it easy for users to secure their internet-bound traffic without the hassle of having to set up and manage virtual firewalls and policies.

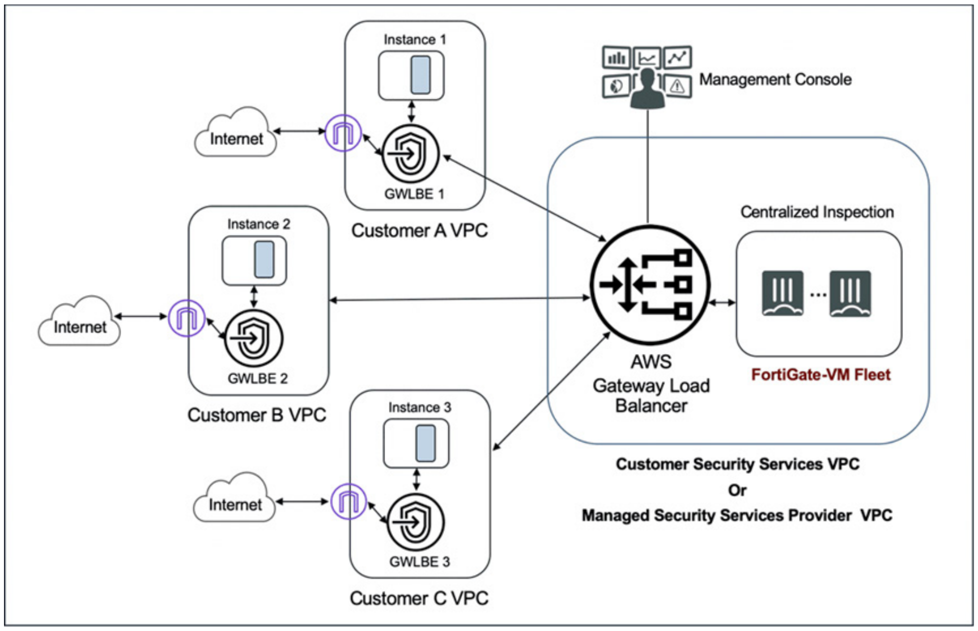


Figure 26: North-South Traffic Inspection with AWS Gateway Load Balancer

As shown in the diagram, all traffic in the VPC is destined for the GWLBE before it leaves the VPC. While the diagram shows an internet gateway, the destination could just as easily be another VPC via transit gateway attachment, or over VPN to a destination on-premises.

Use Case 2: East-West Inspection

With the rapid adoption of AWS, organizations are quickly evolving from single VPC deployments to having many VPCs. In many cases, these VPCs are managed by application development teams without a security background. Applications are often built using open-source code that may have vulnerabilities that malicious actors can leverage to launch attacks between VPCs. To avoid these scenarios, the VPC-to-VPC traffic flows must be inspected. Organizations can deploy FortiGate-VM with GWLB service to provide this protection.

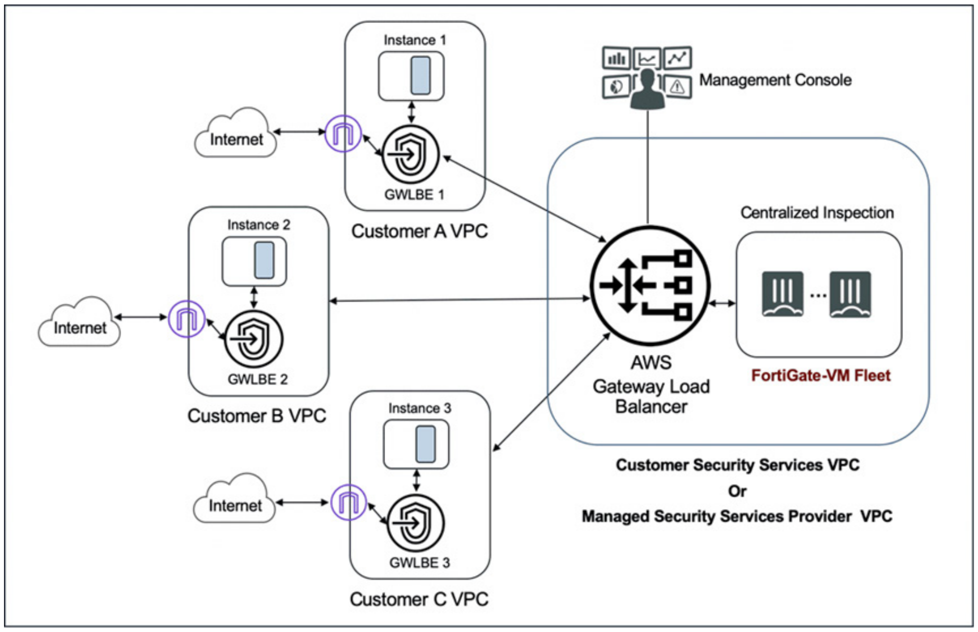


Figure 27: East-West VPC-to-VPC Traffic Inspection with AWS Gateway Load Balancer

This diagram highlights how leveraging GWLB with existing TGW deployments can allow for easy east-west inspection across accounts and even in multi-tenant deployments.

Conclusion

As organizations adopt cloud deployments, it is of paramount importance to design a security architecture that fits their requirements early in their cloud journey. Fortinet FortiGate-VM on AWS, FortiWeb-VM on AWS, FortiWeb Cloud WAF-as-a-Service, FortiADC-VM on AWS, and FortiCNP support various deployments and design models to satisfy the scale, availability, and resiliency requirements of AWS customers. Additionally, the integration with native AWS services and automation frameworks allows Fortinet Security Services to scale dynamically and automatically to meet changing traffic volumes. Finally, Fortinet Security Fabric Connectors natively integrate with AWS to create dynamic address objects that help network and security administrators automatically manage and enforce security policies across your cloud footprint.

www.fortinet.com

October 27, 2022 4:21 PM

381696-B-0-EN