Networking & Content Delivery

# Deployment models for AWS Network Firewall

by Shakeel Ahmad and Evgeny Vaganov | on 17 NOV 2020 | in Advanced (300), AWS Transit Gateway, Networking & Content Delivery, Technical How-to | Permalink | Share

**10-Sep-2021:** With recent enhancements to VPC routing primitives and how it unlocks additional deployment models for AWS Network Firewall along with the ones listed below, read part 2 of this blog post here.

---

## Introduction

AWS services and features are built with security as a top priority. With Amazon Virtual Private Cloud (VPC), customers are able to control network security using Network Access Control Lists (NACL) and Security Groups (SG). Many customers have requirements beyond the scope of these network security controls, such as deep packet inspection (DPI), application protocol detection, domain name filtering, and intrusion prevention system (IPS).

At scale, customers require many more rules compared to what is supported in SGs and NACLs today. For these customers, we built AWS Network Firewall – a stateful, managed, network firewall and intrusion prevention service for your VPC. It is designed for scale and supports tens of thousands of rules. In AWS Network Firewall – New Managed Firewall Service in VPC (blog post) we explain the features and use cases for AWS Network Firewall. Start there if AWS Network Firewall is new to you. Keep reading this post if you're familiar with AWS Network Firewall, as we focus on deployment models for common use cases where AWS Network Firewall could be added into the traffic path. Before we look at deployment models, let's first understand how AWS Network Firewall works.

## How AWS Network Firewall works

To apply traffic-filtering logic provided by AWS Network Firewall, you must route traffic symmetrically to the AWS Network Firewall endpoint. This firewall endpoint is similar to PrivateLink VPC interface endpoint. The key difference is that it can be a route table target. AWS Network Firewall endpoint is deployed into a dedicated subnet of a VPC. We call this subnet an **AWS Network Firewall subnet** or simply **firewall subnet**. Depending on the use case and deployment model, the firewall subnet could be either public or private. For high availability (HA) and Multi-AZ deployments, allocate a subnet per Availability Zone (AZ). As a best practice, do not

use AWS Network Firewall subnet to deploy any other services since AWS Network Firewall is not able to inspect traffic from sources or destinations within firewall subnet.

Once AWS Network Firewall is deployed, you will see a firewall endpoint in each firewall subnet. As mentioned earlier, firewall endpoint is similar to interface endpoint and it shows up as **vpce-id** in your VPC route table target selection. Firewall endpoint capability is powered by AWS Gateway Load Balancer and therefore elastic network interface (ENI) of the endpoint is "gateway_load_balancer_endpoint" type. To have your network traffic inspected by AWS Network firewall, you must direct traffic to firewall endpoint using VPC route tables. In figure 1, we insert firewall endpoint in the path between a workload subnet and internet gateway (IGW) using VPC Ingress Routing feature. If you are not familiar with this feature, see documentation for more details in addition to the VPC Ingress Routing blog post. We call this workload subnet a **protected subnet** as all traffic to the internet is now inspected and protected by AWS Network Firewall.

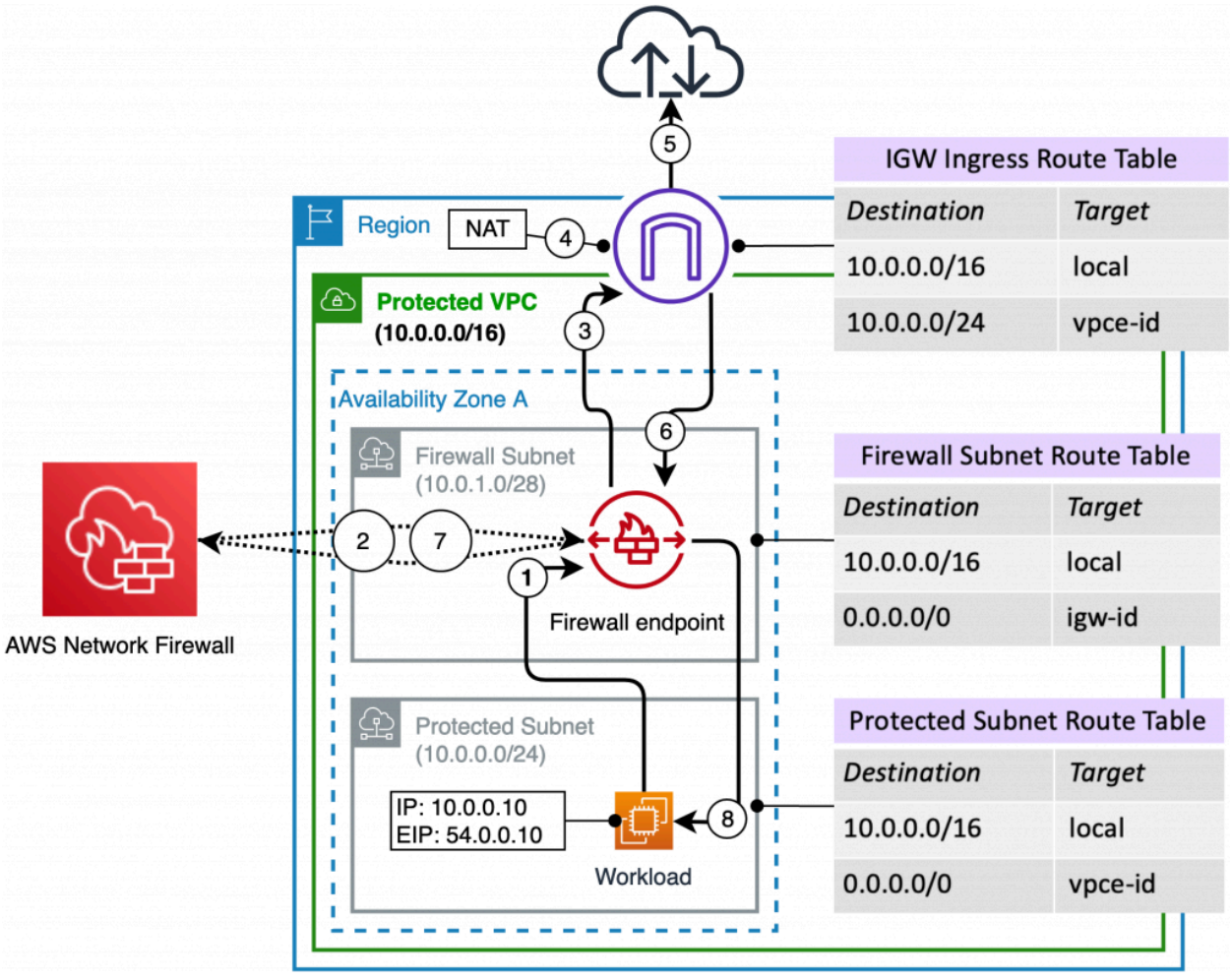

*Figure 1: AWS Network Firewall deployed in a single AZ and traffic flow for a workload in a public subnet*

AWS Network Firewall is completely transparent to the traffic flow and does not perform network address translation (NAT). It preserves source and destination IP addresses.

When a network packet arrives to AWS Network Firewall, it enters rules engine and gets inspected. To learn more about the rules engine and what rules are available, see documentation.

## Deployment models

There are multiple deployment models available with AWS Network Firewall. The right model depends on the use case and requirements. The following models are most common:

1. **Distributed AWS Network Firewall deployment model:** AWS Network Firewall is deployed into each individual VPC.
2. **Centralized AWS Network Firewall deployment model:** AWS Network Firewall is deployed into centralized VPC for East-West (VPC-to-VPC) and/or North-South (internet egress and ingress, on-premises) traffic. We refer to this VPC as inspection VPC throughout this blog post.
3. **Combined AWS Network Firewall deployment model:** AWS Network Firewall is deployed into centralized inspection VPC for East-West (VPC-to-VPC) and subset of North-South (On Premises/Egress) traffic. Internet ingress is distributed to VPCs which require dedicated inbound access from the internet and AWS Network Firewall is deployed accordingly.

For each deployment model, you can have AWS Network Firewall chained together with other services (service chaining). For example, you can chain AWS Network Firewall and NAT gateway.

## Distributed AWS Network Firewall deployment model

For the distributed deployment model, we deploy AWS Network Firewall into each VPC which requires protection. Each VPC is protected individually and blast radius is reduced through VPC isolation. Each VPC does not require connectivity to any other VPC or AWS Transit Gateway. Each AWS Network Firewall can have its own firewall policy or share a policy through common rule groups (reusable collections of rules) across multiple firewalls. This allows each AWS Network Firewall to be managed independently, which reduces the possibility of misconfiguration and limits the scope of impact.

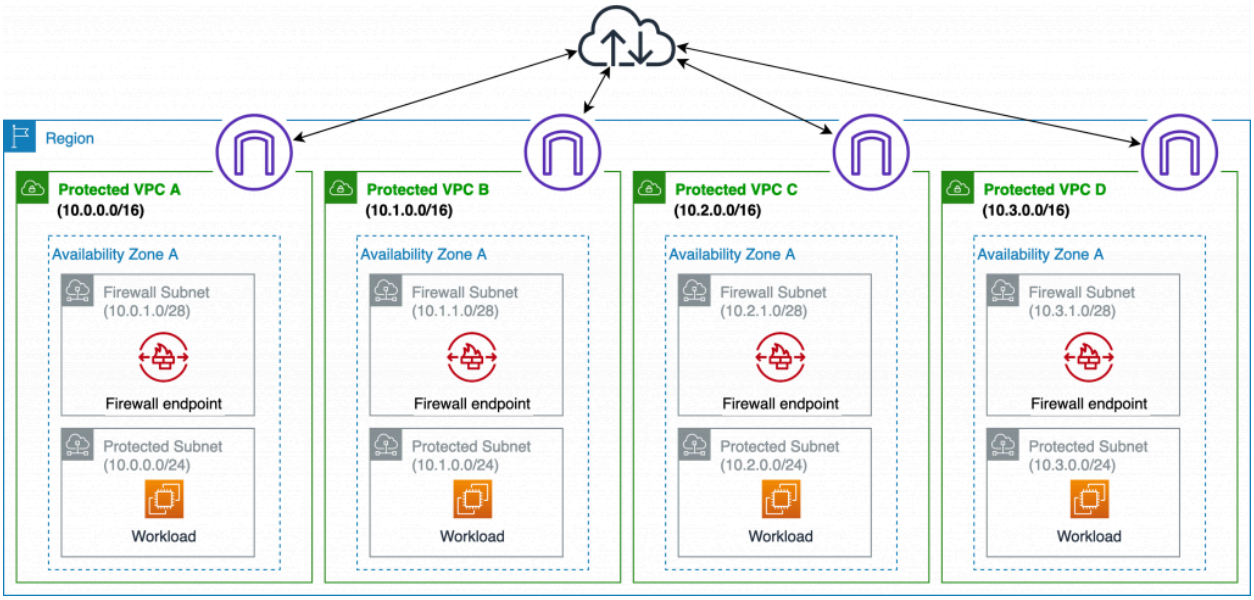


*Figure 2: AWS Network Firewall deployed in each protected VPC*

Depending on the workload and traffic pattern, there are a number of AWS Network Firewall deployment models to consider. Below are those models.

## 1) AWS Network Firewall is deployed to protect traffic between a workload public subnet and IGW

With this deployment model, AWS Network Firewall is used to protect any internet-bound traffic. The workload subnet has the default route to the firewall endpoint in the corresponding AZ. The firewall subnet has default route via IGW. Since AWS Network Firewall doesn't perform NAT, ingress and egress to the internet depends on public IPs or EIPs associated to individual elastic network interfaces (ENIs) in the workload subnet. Finally, IGW has ingress route table associated to it. Route table has entry for each protected subnet directing traffic to firewall endpoint in the corresponding AZ. This ensures that traffic is symmetrically returned to the right firewall endpoint to maintain stateful traffic inspection.

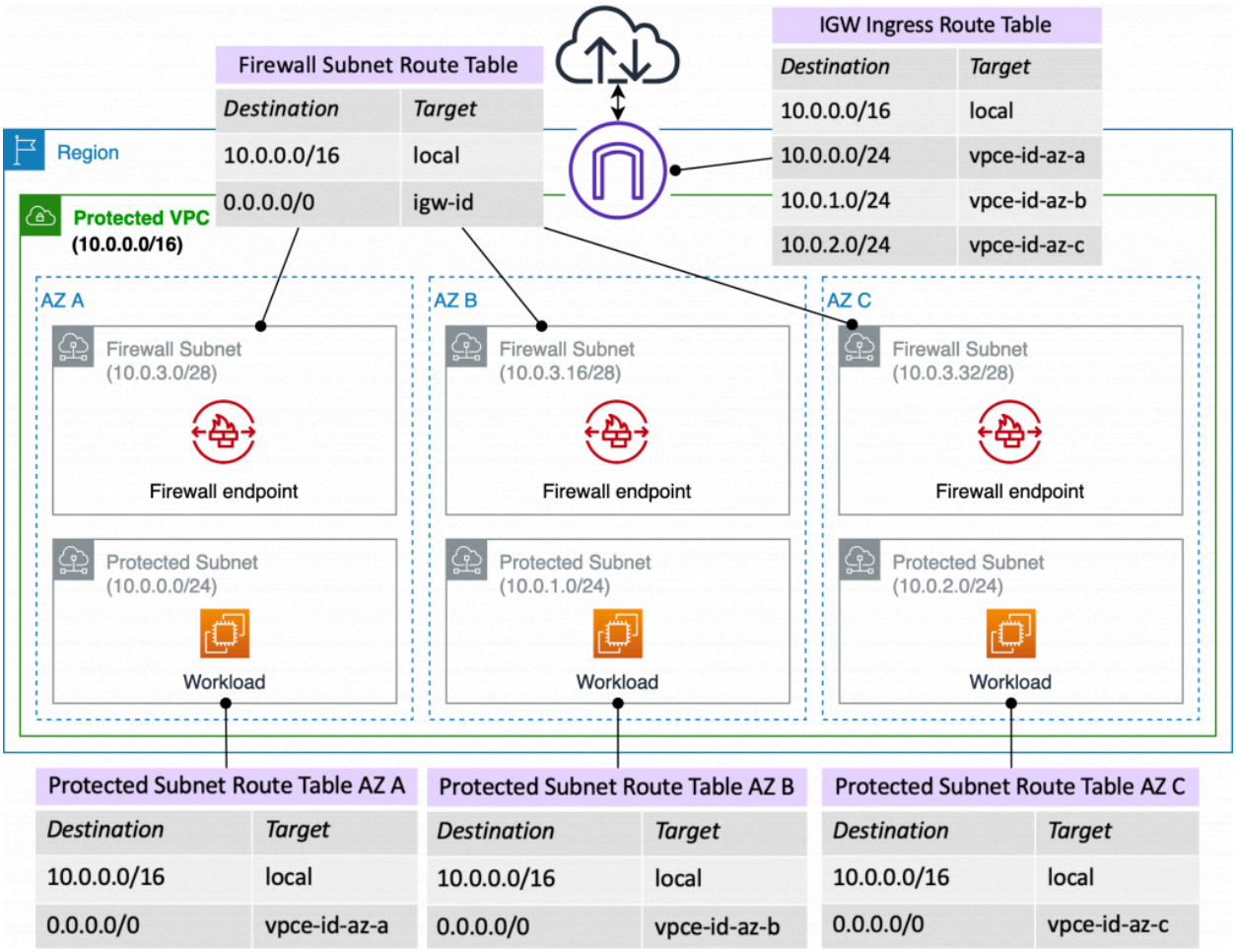


*Figure 3: AWS Network Firewall deployed in Multi-AZ configuration protecting public subnets*

## 2) AWS Network Firewall is deployed to protect traffic between an AWS service in a public subnet and IGW

AWS Network Firewall can also be deployed to protect AWS services such Application Load Balancer (ALB) and NAT gateway. With ALB, backend targets could be deployed within private subnets. Any traffic between ALB and the internet is inspected by AWS Network Firewall before delivery to backend targets. Similarly, NAT gateway could be placed in the protected public subnet. NAT gateway is a default route table target in the private subnet route table. Traffic between NAT gateway and the internet is inspected. To maintain Multi-AZ, NAT gateway is also deployed in each AZ.

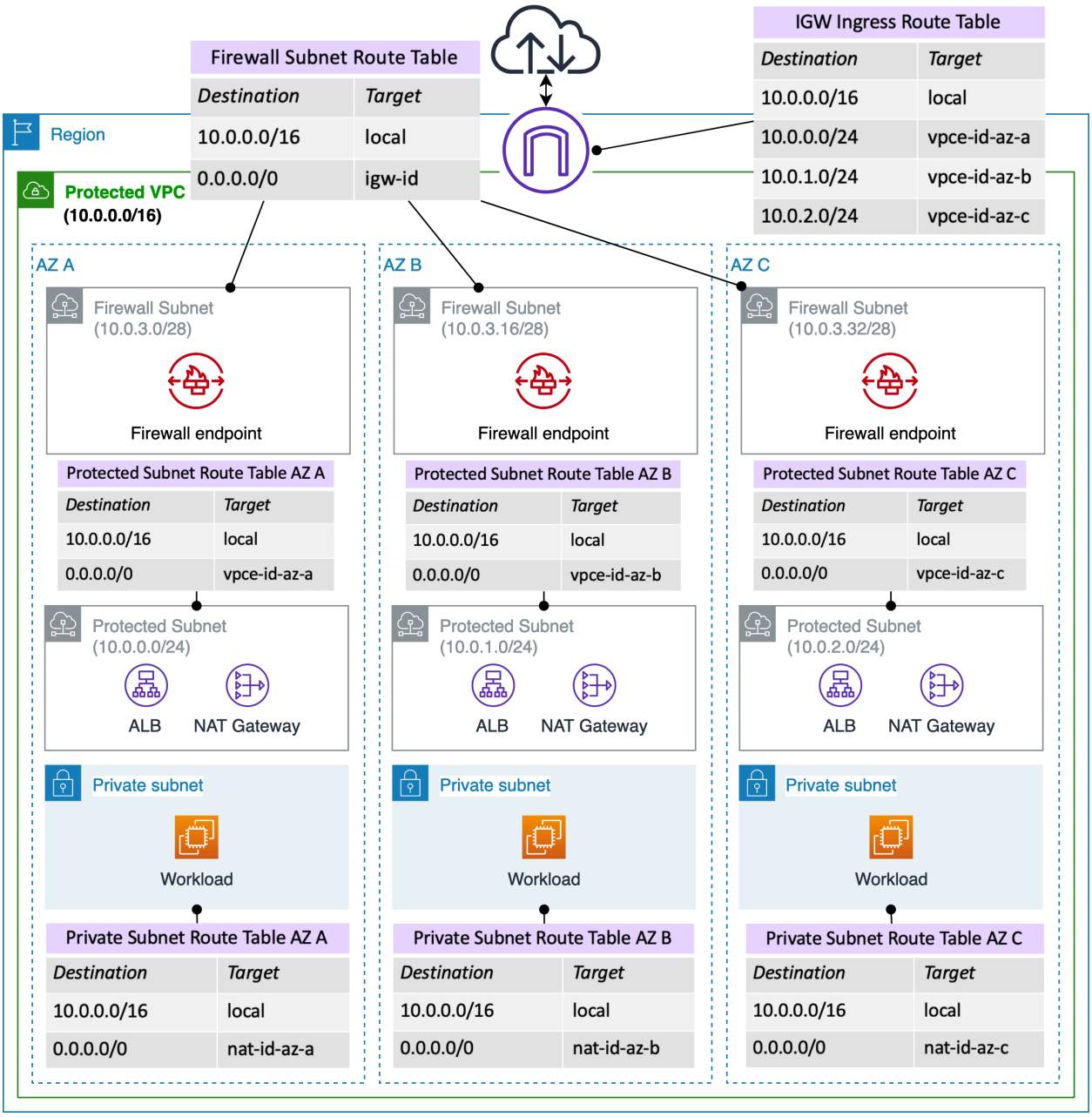


*Figure 4: AWS Network Firewall is deployed to inspect traffic between the internet and ALB/NAT gateway*

Note that, with either ALB or NAT Gateway in Figure 4, the source IP of workload ENI is not preserved. AWS Network Firewall sees source IP of NAT gateway or ALB. If workload source IP visibility is required, use the centralized internet egress and ingress deployment model discussed later in this blog post.

## Centralized deployment model

For centralized deployment model, AWS Transit Gateway is a prerequisite. AWS Transit Gateway acts as a network hub and simplifies the connectivity between VPCs as well as on-premises networks. AWS Transit Gateway also provides inter-region peering capabilities to other Transit Gateways to establish a global network using AWS backbone.

Another key characteristic of the centralized deployment is a dedicated inspection VPC. Inspection VPC consists of two subnets in each AZs. One subnet is a dedicated firewall endpoint subnet and second is dedicated to AWS Transit Gateway attachment. Figure 5 depicts an example in which an AWS Region with three AZs has six subnets in total for Inspection VPCs. This is a Multi-AZ configuration.

Each Transit Gateway subnet requires a dedicated VPC route table to ensure the traffic is forwarded to firewall endpoint within the same AZ. These route tables have a default route (0.0.0.0/0) pointing towards firewall endpoint in the same AZ.


*Figure 5: Overview of Inspection VPC*

For the return traffic from firewall endpoint, a single VPC route table is configured. The route table contains a default route towards AWS Transit Gateway. Traffic is returned to AWS Transit Gateway in the same AZ after it has been inspected by AWS Network Firewall. Figure 6 shows traffic flow of inspection VPC.

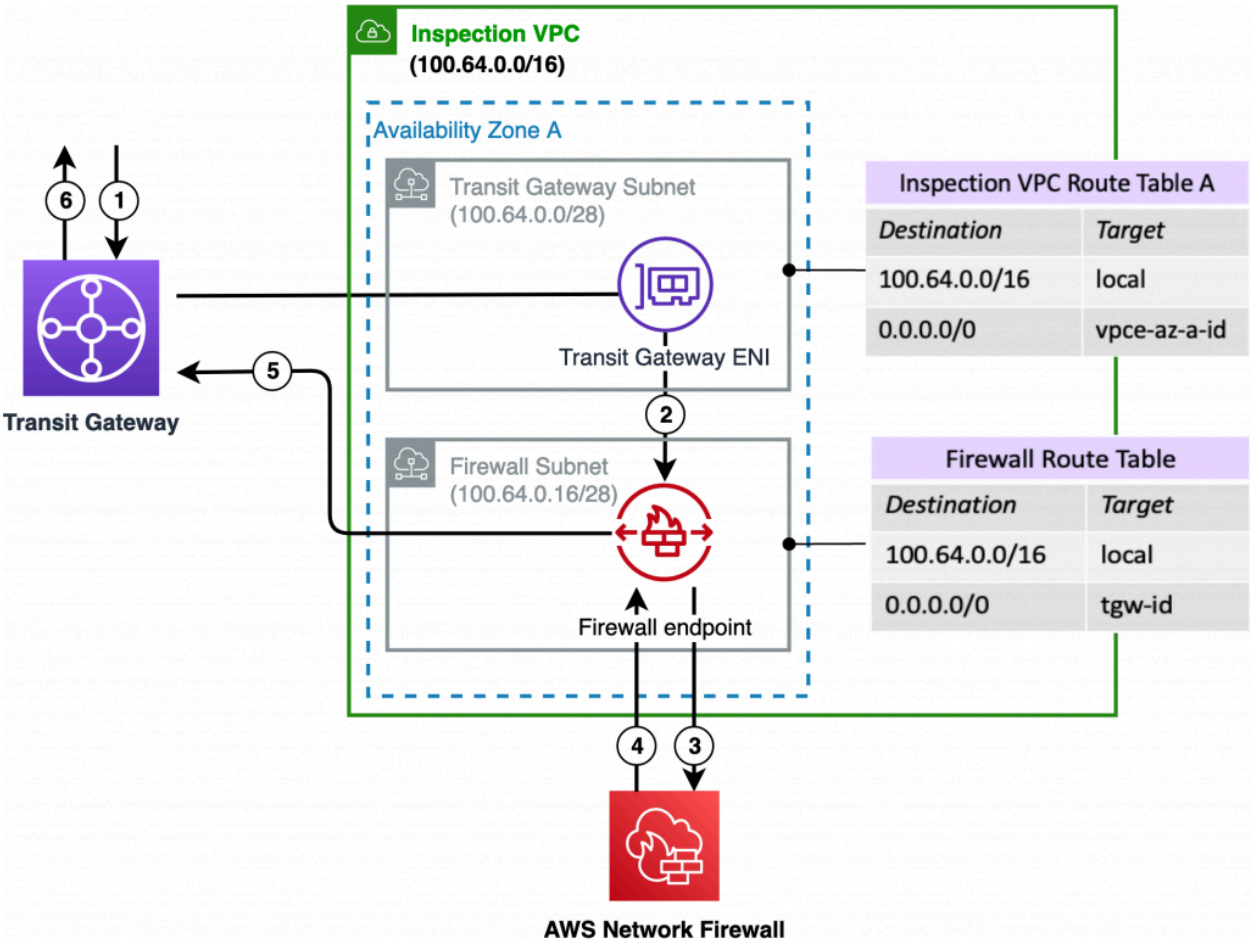


*Figure 6: High level logical traffic flow from AWS Transit Gateway to inspection VPC*

Inspection VPC CIDR doesn't need to be routable since AWS Network Firewall is completely transparent to network traffic. In our example, we used CGNAT range (100.64.0.0/16) to preserve IP addresses. Avoid using inspection VPC for any other workload. Inside inspection VPC, AWS Network Firewall maintains stateful traffic inspection capability. This is possible due to a new feature: AWS Transit Gateway appliance mode. This feature ensures that return traffic is processed by the same AZ. Make sure to enable it by following this documentation.

A dedicated inspection VPC provides a simplified and central approach to manage inspection between VPCs (same or different region), internet, and on-premises networks. Be careful to ensure that firewall policies and rules groups have enough Rule Group Capacity available and allow for rules growth. Inspection VPC architecture gives AWS Network Firewall source and destination IP visibility. Both source and destination IPs are preserved. Now, let's explore how to use inspection VPC for various centralized inspection models:

**Note:** HOME_NET rule group variable is used to define source IP range eligible for processing in Stateful Domain list and optionally Suricata compatible IPS Rule Groups. By default, it is set to the VPC CIDR where firewall endpoints are deployed. With centralized deployment model, this variable must be expanded to include all CIDR ranges of your VPCs and on-premises networks to make them eligible for processing. See documentation for more details.

## 1) East-West Traffic Inspection Model:

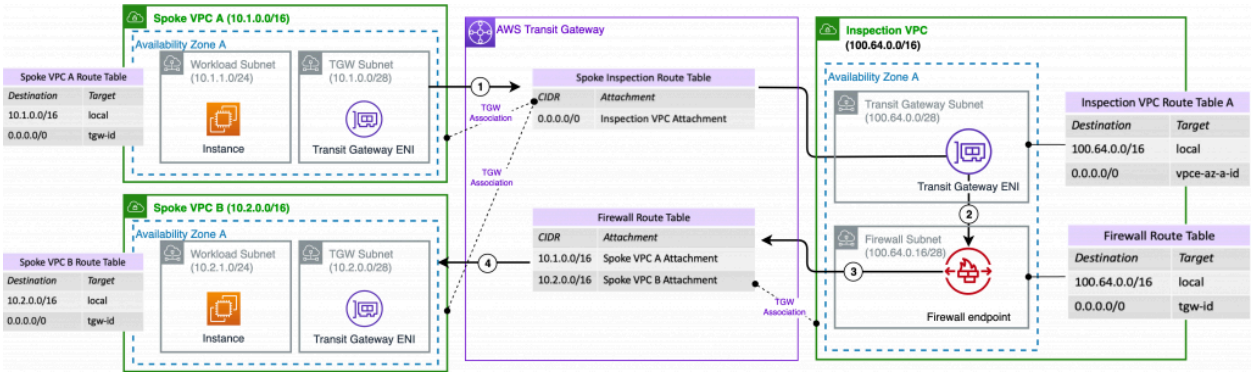


*Figure 7: Traffic between two VPCs protected by centrally deployed AWS Network Firewall*

This model covers requirements where there's a need to inspect East-West traffic. For example, VPC to VPC in the same or different AWS Accounts using AWS Transit Gateway.

In this model, we use two AWS Transit Gateway route tables – As shown in figure 7 as Firewall Route Table & Spoke Route Table – within the same Transit Gateway to ensure that all traffic between VPCs is passing through inspection VPC. All the spoke VPCs are associated to the spoke route table which has a default static route towards inspection VPC attachment. All spoke VPCs routes are propagated into Transit Gateway firewall route table which ensures the packets have a return path to spoke VPCs.

You can use the same model for inspection of traffic to other AWS Regions using AWS Transit Gateway Inter-Region Peering feature as shown in Figure 8. Remote AWS Regions are treated as spokes.

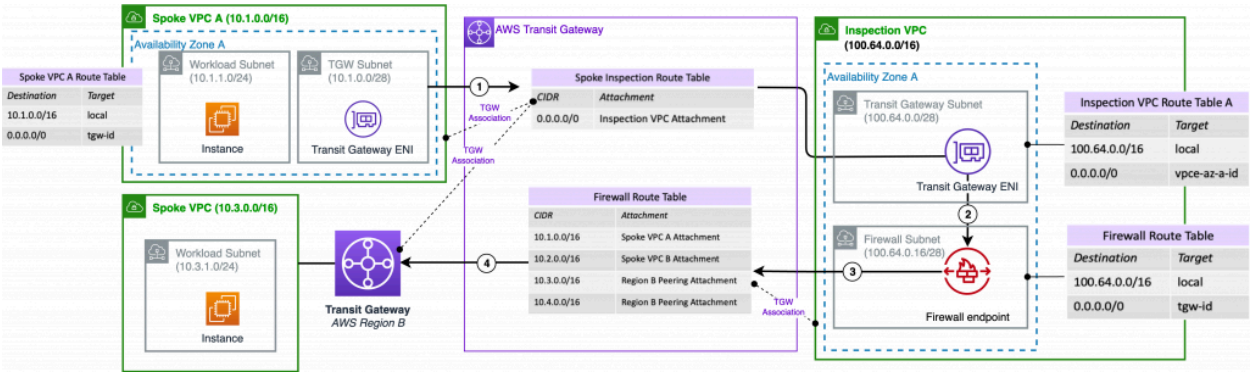


*Figure 8: Traffic between VPC in AWS Region A and VPC in AWS Region B protected by centrally deployed AWS Network Firewall*

## 2) North-South: Centralized on-premises egress & ingress via Transit Gateway and Transit VIF/Direct Connect gateway/AWS Site-to-Site VPN

Let's expand the previous model and add inspection for North-South traffic between AWS VPC and on-premises via AWS Transit Gateway. AWS Transit Gateway can connect to your on-premises via AWS Direct Connect or via AWS Site-to-Site VPN.

A key requirement for this model is to connect AWS Direct Connect using Transit VIF to AWS Transit Gateway. In case of VPN to on-premises, AWS Site-to-Site VPN can also be used and must be established to AWS Transit Gateway as per Figure 9.

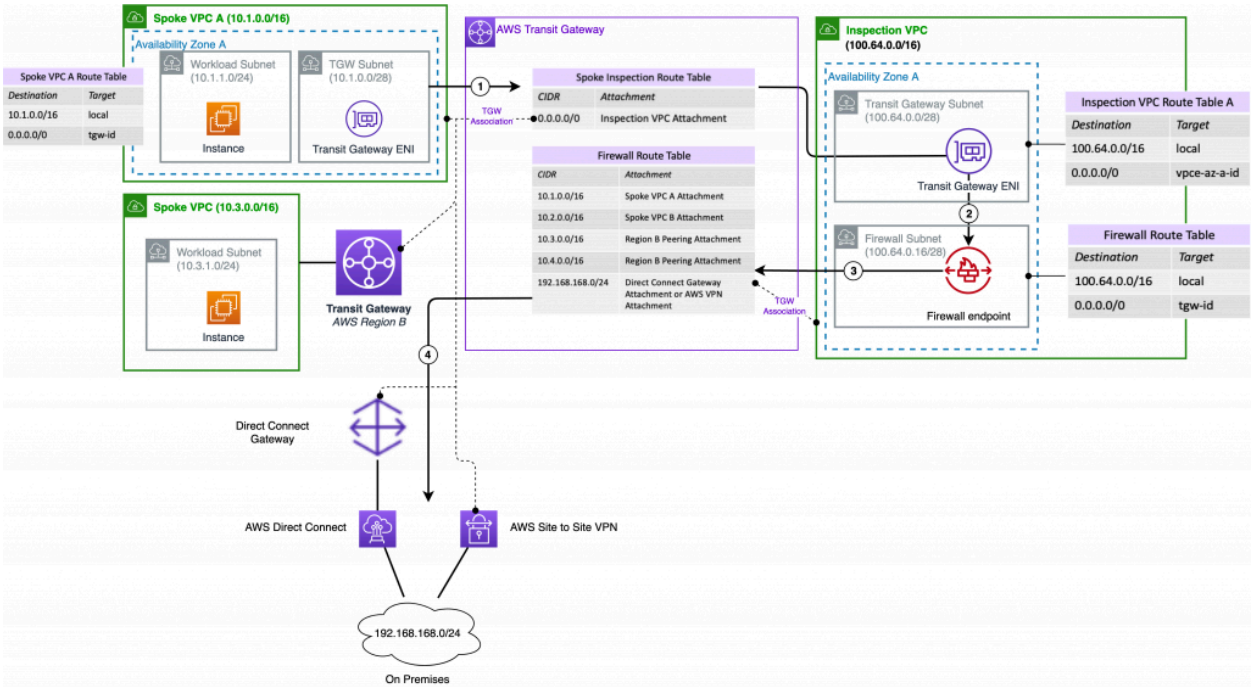


*Figure 9: Traffic between VPC and on-premises protected by centrally deployed AWS Network Firewall*

Using AWS Transit Gateway Routing Tables functionality, AWS Direct Connect/VPNs are attached to spoke route table. In other words, they are treated as spokes again. For return traffic from inspection VPC to work, propagate the routes learned from Direct Connect gateway and/or VPN attachments to the firewall route table as shown in Figure 9.

While we don't specifically discuss the AWS Client VPN use case here, you can think about AWS Client VPN as a workload associated with one of your spoke VPCs. AWS Client VPN creates an ENI per subnet in the associated VPC and traffic from remote clients is NAT'ed to the IP of the ENI. This allows traffic to be routed and inspected by AWS Network Firewall just like any other workload in your spoke VPCs.

## 3) North-South: Centralized internet egress (VPC to internet via Transit Gateway) and NAT gateway

This model covers the inspection for North-South internet bound traffic from AWS Transit Gateway attachments. For this model, we also have a dedicated, central egress VPC which has NAT gateway configured in a public subnet with access to IGW.

Traffic originating from spoke VPCs is forwarded to inspection VPC for processing. It is then forwarded to central egress VPC using a default route in Transit Gateway firewall route table. Default route is set to target central egress VPC Attachment as shown in Figure 10.

You can use NAT gateway to enable workloads in private subnets from spoke VPCs to connect to internet or AWS services in public IP space. You can also use NAT instance or a partner solution from AWS Marketplace instead of NAT gateway. You can find more information about different partners under the AWS Network Competency Program.

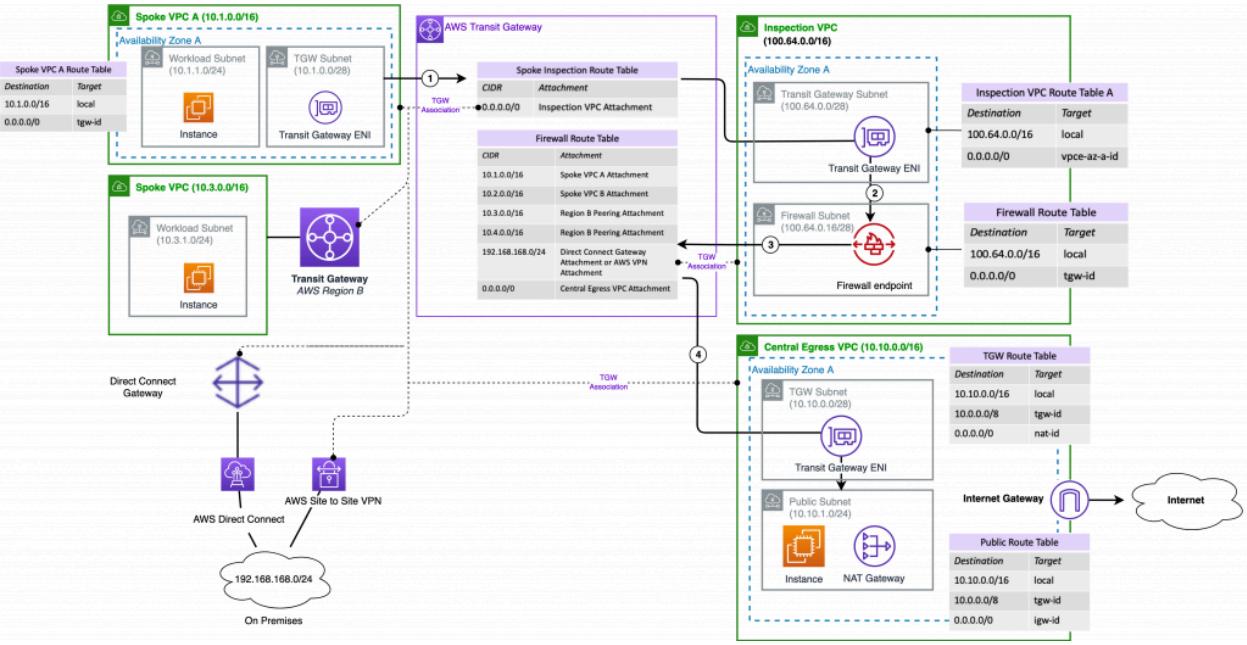


*Figure 10: Traffic between VPC and internet via centralized egress VPC protected by AWS Network Firewall*

Note that it is also possible to deploy AWS Network Firewall inside the central egress VPC which is covered in the combined deployment model.

## 4) North-South: Centralized Internet Ingress via Transit Gateway and NLB/ALB or reverse proxy

For traffic originating outside of your AWS environment inbound to your workloads, it's possible to centralize ingress. In this scenario, a marketplace network appliance such as a Web Application Firewall (WAF) or a third-party Load Balancer (LB) is deployed into a centralized Ingress VPC. The appliance terminates the connection and establishes a new connections to the backend instance (also known as reverse proxy). This approach allows introduction of application logic between

client and a server and also could be used for security purposes. It is possible to deploy AWS services such as ALB and Network Load Balancer (NLB) in this configuration. The major difference with this deployment model is that the source IP is not preserved and ALB/NLB use IPs as targets (as opposed to EC2 instance IDs). Figure 11 shows the possible architecture of ingress VPC combined with AWS Network Firewall in centralized deployment model.

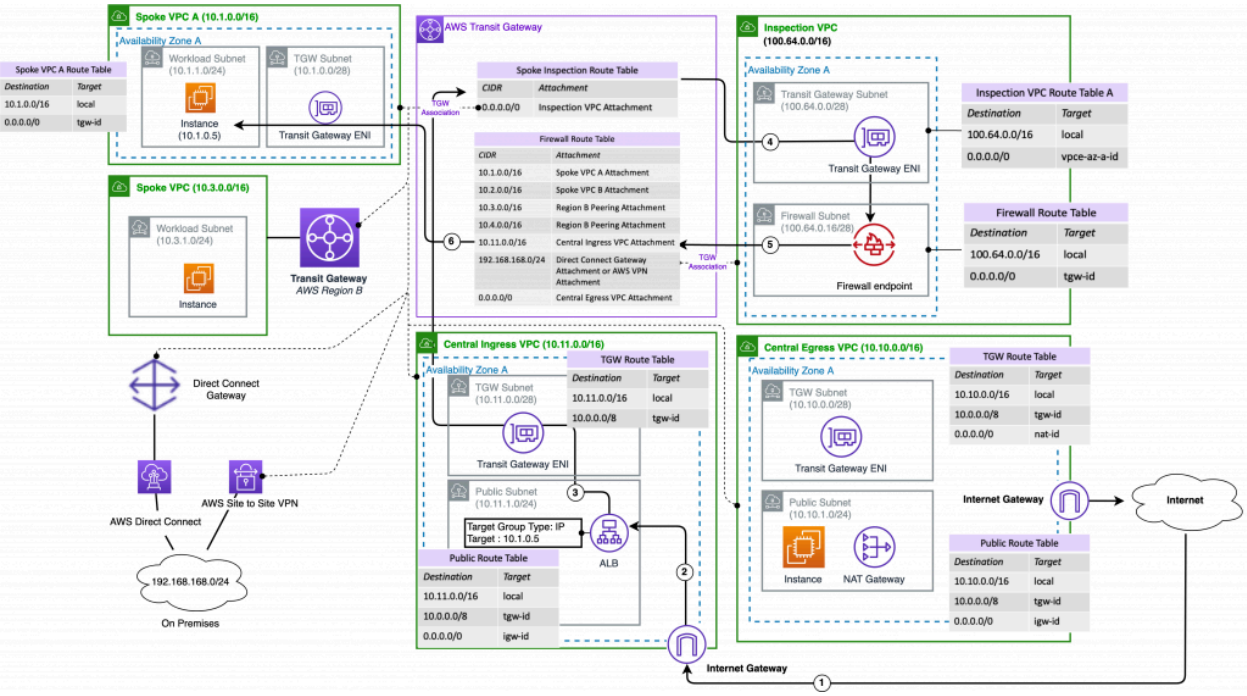


*Figure 11: Centralized ingress with centrally deployed AWS Network Firewall, traffic inspected by AWS Network Firewall as it sent to backend workload from ALB*

Note that it is also possible to deploy AWS Network Firewall inside the central ingress VPC which is covered in the combined deployment model.

## Combined centralized and distributed deployment model

For combined centralized and distributed deployment model, we can deploy
AWS Network Firewall in the central inspection VPC and also in each VPC which requires local internet ingress and/or egress. In this model, workloads can use local IGW and connect to internal networks (VPCs, on-premises) through the centralized inspection VPC. Private subnets in such VPC retain an option to use centralized internet egress VPC.

### 1) Some VPCs optionally have their own IGW for internet ingress/egress and traffic is protected by dedicated AWS Network Firewall

Public subnets can be protected by AWS Network Firewall as previously discussed. With this architecture, load balancers such as ALB and NLB can target containers and applications in EC2 Auto Scale groups without any additional configuration to track target IPs. We can also have a local internet egress for private subnet with NAT gateway where required, but consider deploying it when convenience of having centralized egress VPC is outweighed by cost factors. In the Figure 12, spoke VPC B has local IGW and firewall endpoint to protect public subnets.

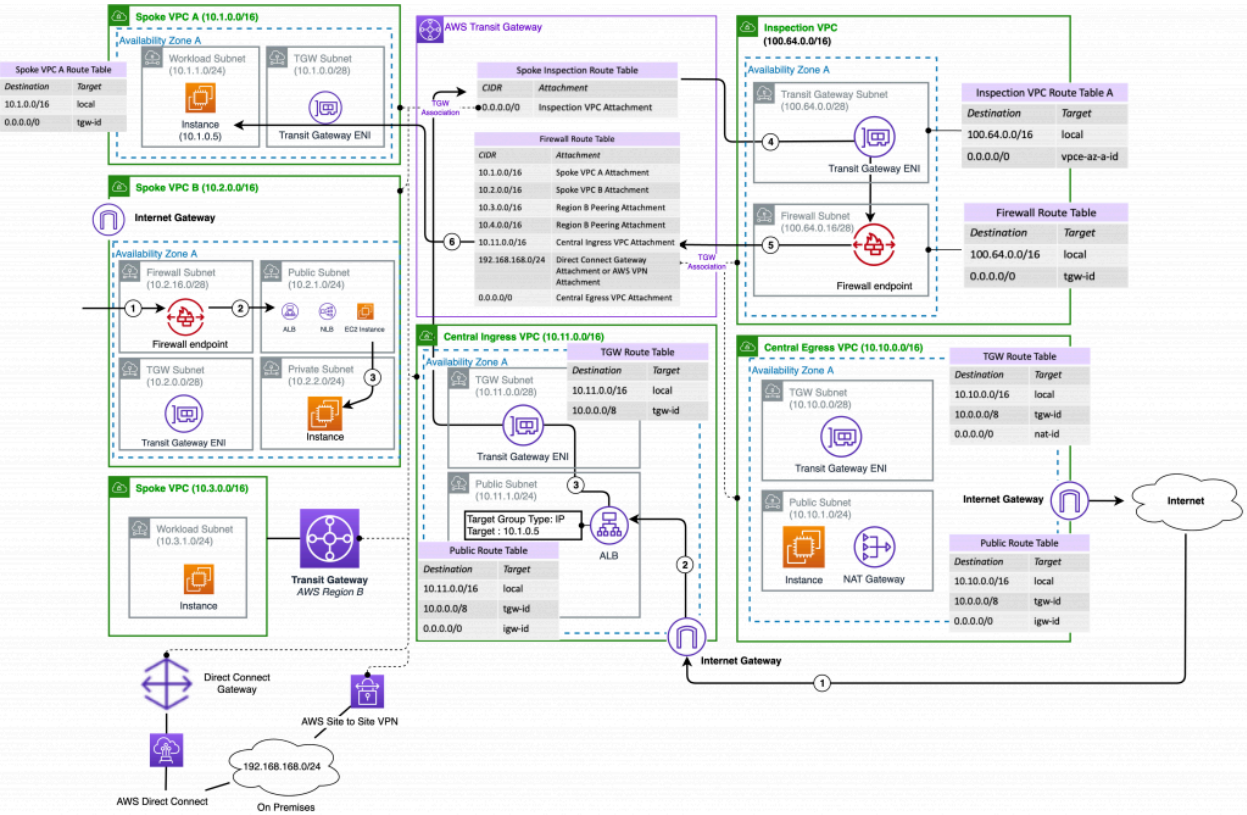


*Figure 12: Spoke VPC B with local IGW protected by dedicated AWS Network Firewall*

## 2) Inspection VPC only for East-West traffic and egress VPC with inspection for internet

Some customers may wish to have centralized internet egress protected by AWS Network Firewall and a separate instance of the AWS Network Firewall for East-West traffic. In this case, there is a cost saving with this architecture in terms of data processing cost reduction as traffic goes directly from a source VPC to centralized egress VPC without using inspection VPC.

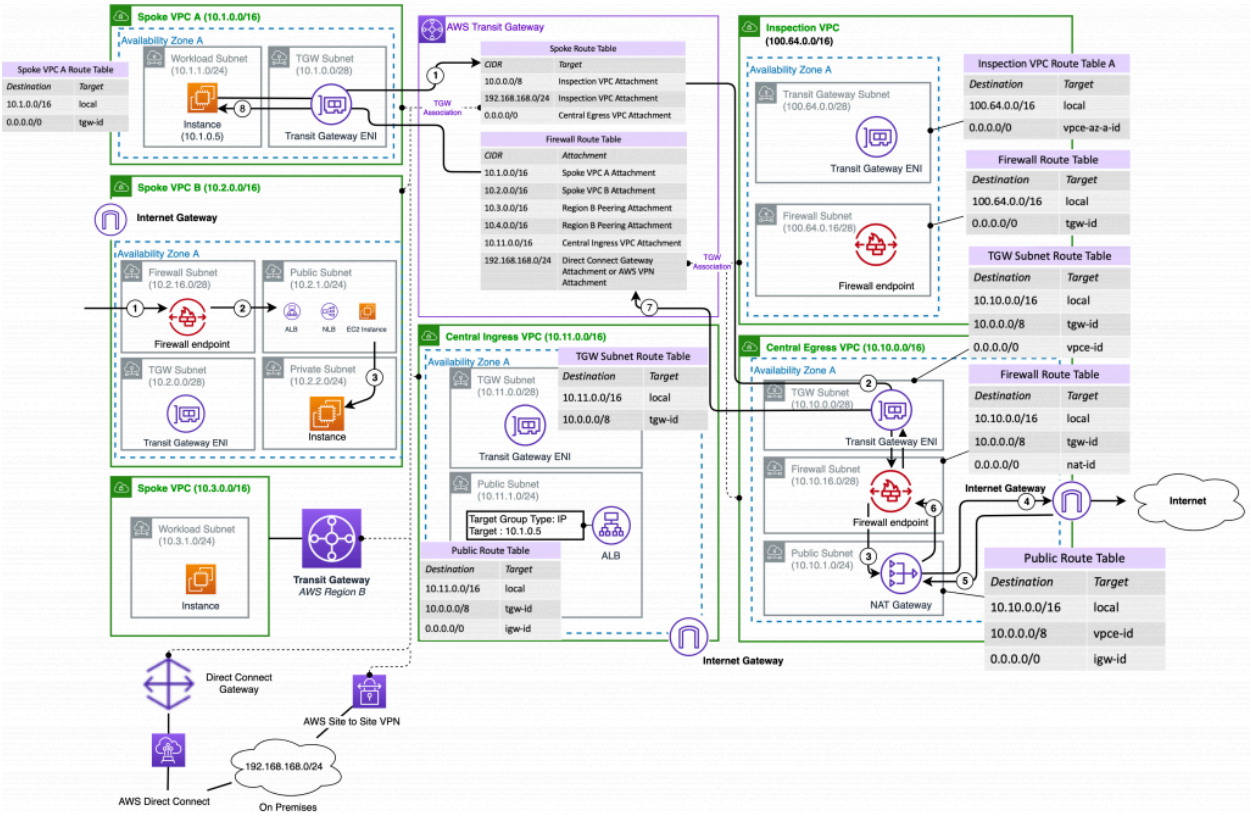


*Figure 13: Centralized Inspection VPC for East-West traffic and dedicated AWS Network Firewall for centralized internet egress VPC*

## Deployment model comparison

The following table can help you to decide which deployment model is most applicable for your use case.

| | **Distributed AWS Network Firewall deployment model** | **Centralized AWS Network Firewall deployment model** | **Combined AWS Network Firewall deployment model** |
|---|---|---|---|
| **East-West: VPC to VPC traffic flow** | Not supported | Supported | Supported |

| | | | |
|---|---|---|---|
| **North-South: VPC to Internet traffic flow** | Supported | Supported | Supported |
| **North-South: VPC to on-prem via VPN or DX traffic flow** | Not supported | Supported | Supported |
| **Prerequisites** | AWS Network Firewall subnet | Inspection VPC and AWS Transit Gateway | AWS Network Firewall subnets in each protected VPC; Inspection VPC and AWS Transit Gateway |
| **Centralized management** | Through AWS Firewall Manager | Through a single instance of AWS Network Firewall | Through AWS Firewall Manager |
| **Source IP visibility** | Configuration dependent | Yes | Configuration dependent |
| **Misconfiguration risk and potential blast radius** | Lowest | Medium | Low |
| **Cost** | Per AWS Network Firewall endpoint | Per AWS Transit Gateway attachments & AWS Network Firewall endpoints; AWS Transit Gateway data processing | Per AWS Transit Gateway attachments & AWS Network Firewall endpoints (including any additional endpoints per protected VPC); AWS Transit Gateway data processing |

## Considerations

Key considerations with any AWS Network Firewall deployment model are:

- Although AWS Network Firewall is represented by a single firewall endpoint per subnet/AZ, it is highly available and based on AWS Hyperplane technology. No single component failure of AWS Network Firewall would cause an AZ failure.

- Ensure that firewall endpoint is deployed in all AZs used by your workloads so traffic is inspected within the same AZ.
- AWS Transit Gateway has data processing and attachment cost. AWS Transit Gateway is required for centralized and combined deployment models.
- There is a cost per AWS Network Firewall endpoint. Consider using combined deployment model if you have a large number of Amazon VPCs. Deploy AWS Network Firewall endpoints only where required.
- AWS Network Firewall cannot be deployed to inspect traffic between VPCs that are peered together; or on premises networks that are connected directly to a VPC using Virtual Private Gateway. It also cannot be deployed between workload subnet and TGW attachment where both are within the same VPC. It can be deployed to inspect traffic with Transit Gateway when using an inspection VPC. Note that the inspection VPC cannot be traffic source or destination.
- AWS Network Firewall can be used to inspect traffic between branch offices as it transits through AWS. Deployment model is similar to centralized where each branch is a unique AWS Transit Gateway attachment.

## Conclusion

AWS Network Firewall provides the ability to transparently inspect traffic at scale in a variety of use cases. The right deployment model depends on the desired outcome. AWS Network Firewall can be deployed to a single VPC or multi-VPCs architectures and it scales to traffic requirements with both distributed and centralized deployment models. This is achieved through uniquely designed software defined network services like Amazon VPC and AWS Hyperplane combined together with AWS Network Firewall. The centralized management and traffic inspection could benefit large organizations that require additional network security capabilities.


### Shakeel Ahmad

Shakeel Ahmad is a Senior Solutions Architect based out of Melbourne, Australia specializing in Networking & Cloud Infrastructure. He has a BS in Computer Science and a Master of Science in Network Systems. He's passionate about technology and helping customers architect and build solutions to adopt the art of the possible on AWS.

## Evgeny Vaganov

Evgeny Vaganov is a Senior Specialist Solutions Architect – Networking, at AWS in Asia Pacific Japan (APJ) region. Prior to this role, Evgeny supported customers across Australia and New Zealand adopting Cloud. Passionate about learning and experimenting, he has a goal of making Cloud networking simpler for everyone.

TAGS: AWS Network Firewall, Networking, transit gateway, VPC