AWS Whitepaper

Amazon EC2 Overview and Networking Introduction for Telecom Companies

Amazon EC2 Overview and Networking Introduction for Telecom Companies: AWS Whitepaper

Table of Contents

Amazon EC2 Overview and Networking Introduction for Telecom Companies

Publication date: **October 4, 2023** (*Document history*)

Many telecom providers are considering the AWS Cloud for their telecom workloads such as Core Networking, Operation Support System (OSS), Business Support System (BSS), Radio Access Network (RAN), Value Added Services (VAS) and Information Technology (IT). This paper describes the Amazon EC2 offerings that are available and highlights important performance considerations. Networking capabilities and connectivity options available between on-premises telecom environments and the AWS Cloud, such as Amazon Virtual Private Cloud (VPC), AWS Direct Connect (DX), AWS Transit Gateway (TGW), Virtual Private Network (VPN), AWS Network Firewall (ANF) and AWS Edge Services are also discussed.

Are you Well-Architected?

The AWS Well-Architected Framework helps you understand the pros and cons of the decisions you make when building systems in the cloud. The six pillars of the Framework allow you to learn architectural best practices for designing and operating reliable, secure, efficient, cost-effective, and sustainable systems. Using the AWS Well-Architected Tool, available at no charge in the AWS Management Console, you can review your workloads against these best practices by answering a set of questions for each pillar.

For more expert guidance and best practices for your cloud architecture—reference architecture deployments, diagrams, and whitepapers—refer to the AWS Architecture Center.

Overview

Many telecom providers are in the process of building out 5G network infrastructure, assessing their Multi-Access Edge Compute (MEC) strategy, and moving more of their IT workloads to the cloud. Equally, many telecom network functions that were once virtual machine-based solutions, are evolving into container-based offerings.

Telecommunication services are sensitive to network latency, and AWS offer various solutions to host different workloads. AWS Local Zone provides a low-latency infrastructure deployment that places compute, storage, and other select AWS services close to a telecom provider's edge. AWS offers managed infrastructures for on-premises networks called AWS Outposts in 42-unit racks, and mountable Outposts servers in 2U and 1U form factors supporting functionalities like Radio Access Network (RAN), MEC and real-time application services at far edge, enabling low-latency services to the end-users. AWS Wavelength is located within the telecom service provider's network, enabling telecom end-users to be serviced by applications hosted within their networks

With these trends, there is a need for telecom networking engineers to understand AWS elastic computing and its performance characteristics as well as AWS networking services, such as Amazon Virtual Private Cloud (Amazon VPC), AWS Transit Gateway, and AWS Direct Connect (DX). These services allow telecom providers to securely connect their on- premises environments to the cloud and achieve the high availability and performance they require. Trends in the Network Functions Virtualization Infrastructure (NFVI) for 5G workloads implemented on Amazon Elastic Compute Cloud (Amazon EC2) must now also support Kubernetes. Amazon Elastic Kubernetes Service (Amazon EKS) provides the flexible foundation for container network functions (CNFs).

In considering both Virtual Network Function (VNF) and Cloud Native Function (CNF) deployments, telecom providers have specific demands and require specific features, such as single root I/O virtualization (SR-IOV), Data Plane Development Kit (DPDK), Anti-affinity group support, Non-Uniform Memory Access (NUMA), Multus Container notes.xmlNetwork Interface (CNI) and central processing unit (CPU) pinning. Telecom providers hosts applications and services that requiring extensive packets per second (PPS) throughputs and the bandwidth requirement exceeding 100 Gbps.

AWS offers a range of services for telecom workloads and network connectivity options. Amazon VPC is a logically isolated environment in the AWS Cloud that gives telecom providers complete control over how they allocate their subnets, configure routing, and implement security through access control lists (ACLs) and security groups. AWS Transit Gateway allows inter-VPC and VPC to on-premises environments connectivity at scale.

Finally, services such as DX and VPNs allow telecom providers to connect their environments to the AWS Cloud in a secure and scalable manner, without compromising on availability. This paper also provides an example of an Operation Support System (OSS) workload running in Amazon VPC and communicating with the telecom provider's network using AWS Direct Connect.

Mapping AWS services to the NFV framework

AWS services relate to a popular framework—European Telecommunications Standards Institute (ETSI) network functions virtualization (NFV) framework. It's impossible to relate all services and the roles that they could play in building the entire stack, as this would be implementation-dependent. Instead, the roles of key services and how they map to the framework will be explained. A high-level mapping of AWS services to the ETSI NFV framework is depicted in the following figure.

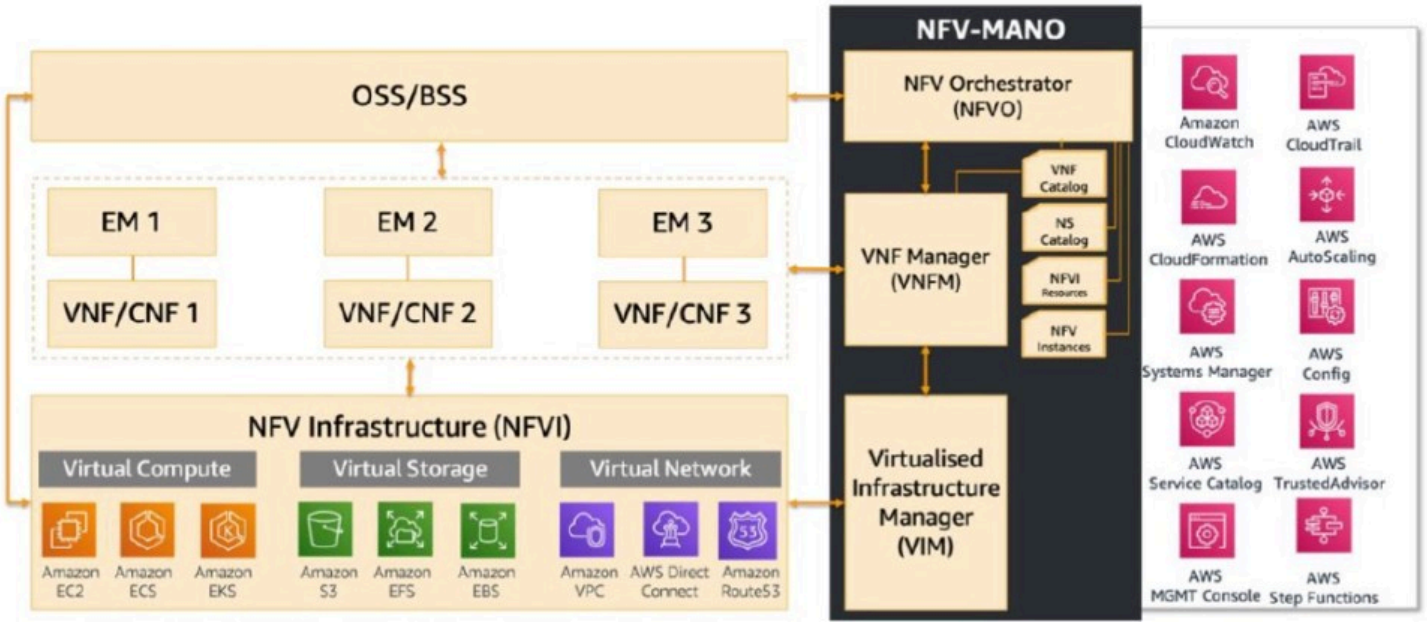


Figure 1 — AWS services mapping to the ETSI NFV framework

The NFVI layer is built using Amazon EC2, Amazon EKS, Amazon S3, Amazon EBS, instance storage, Amazon VPC, AWS Direct Connect, and AWS Transit Gateway. The Virtualized Infrastructure Manager (VIM) layer in traditional VNF implementations is typically OpenStack, however, in AWS, VIM is taken care by AWS native APIs and infrastructure as code.

VNFs can run as either Amazon EC2 instances or Amazon EKS on top of the compute and storage infrastructure. The VNF Manager function can be fulfilled by using tools, such as AWS CloudFormation, to provision the entire infrastructure stack and then leveraging Elastic Load Balancing and dynamic scaling to elastically spin-up or spin-down the compute environment. In on-premises environments, you must purchase or develop dedicated VNFM software modules. With AWS Cloud, the VNFM function is performed by AWS services such as AWS CloudFormation and Amazon EC2 Auto Scaling. Amazon CloudWatch provides alarm triggers to scale up or down the

entire environment. CloudFormation allows you to use a simple text file to model and provision, in an automated and secure manner, all the resources needed for your applications across all Regions and accounts. This file serves as the single source of truth for your cloud environment.

The NFV Orchestrator function is provided by the application vendor in partnership with AWS. State NFV-O is provided as a telecom preferred solution from a wide list of AWS Partners.

Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) provides a virtual server for running applications, which can scale up or down as your computing requirements change. EC2 instance types are grouped based on target application profiles and include the following: general purpose, compute-optimized, memory-optimized, storage-optimized (high I/O), dense storage, GPU compute, and graphics intensive. Today, there are more than 500 instance types available for a variety of virtual workloads and business needs. In addition to these broad categories, capability choices can be made based on the type of processor (for example, Intel, AMD, or AWS), memory footprint, networking, and size. If necessary, each EC2 instance can be associated with a specific choice of Amazon Elastic Block Store (Amazon EBS). There are a number of options available:

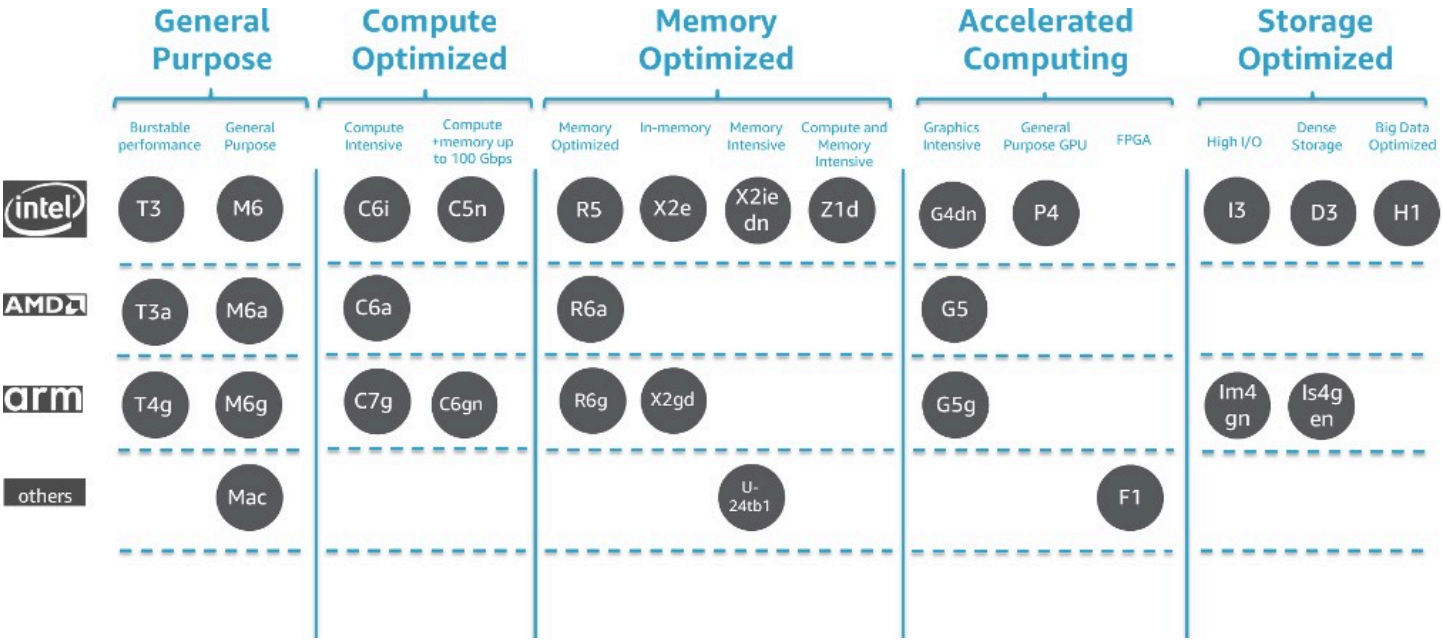


Figure 2 – Overview of Amazon EC2 instance types

Telecom providers require several performance accelerating features to be supported in their computing infrastructure, and this paper will show how AWS supports those features. First, we'll provide an overview of the different performance and optimization options available in AWS for virtualized environments. Next, we'll share a brief history of Amazon EC2 performance, followed by how that evolution has affected the different instance types. Finally, we'll offer guidance on what you can expect to achieve with the different instance families in regard to performance.

Overview of performance and optimization options

Single-Root I/O Virtualization (SR-IOV)

Single-Root I/O Virtualization (SR-IOV) is a mechanism that virtualizes a single PCIe Ethernet controller to make it appear as multiple PCIe devices. Telecom providers have been deploying SR-IOV for their virtualized 5G Packet Core VNFs and CNFs to obtain the required performance from their applications and to share a physical NIC among multiple VMs.

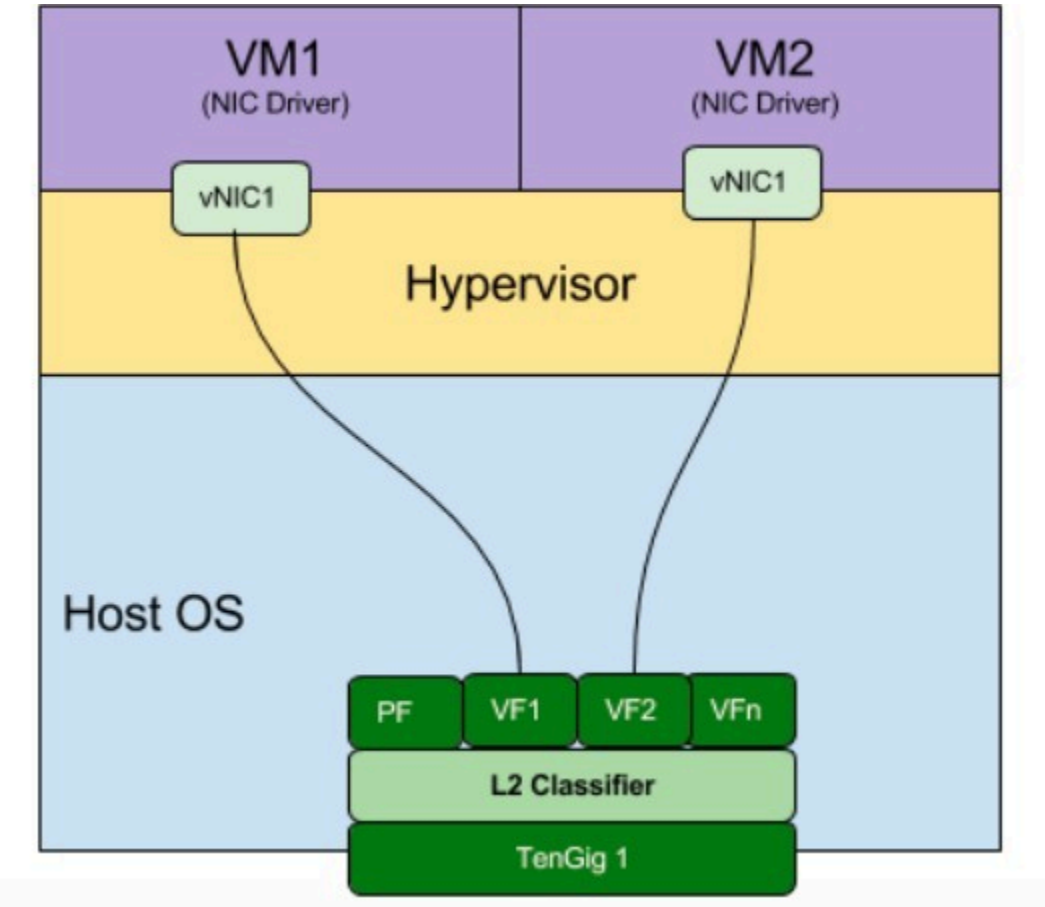


Figure 3 – Illustration of SR-IOV

AWS enhanced networking uses SR-IOV to provide high performance networking capabilities on all current EC2 instance types, except for T2 instances.

The following methods are available for enabling enhanced networking.

Elastic Network Adapter (ENA):

Supports network speeds of up to 100 Gbps for current generation instances except for C4, D2, and M4 instances smaller than m4.16xlarge.

Intel 82599 Virtual Function (VF) interface

Supports network speeds of up to 10 Gbps for instance types C3, C4, D2, I2, M4 (excluding m4.16xlarge), and R3.

Data Plane Development Kit (DPDK)

DPDK consists of a set of libraries and user-space drivers to accelerate packet processing on any CPU. Designed to run in user-space, DPDK enables applications to perform their own packet processing operations directly to and from the NIC. By enabling fast packet processing, DPDK makes it possible for the telecom providers to move performance sensitive applications, such as virtualized mobile packet core and voice, to the cloud. DPDK was also identified as a key enabling technology for network functions virtualization (NFV) by ETSI. The main benefits provided by DPDK are lower latency due to kernel and TCP stack bypass, more control of packet processing, and lower CPU overhead. The DPDK libraries provide only minimal packet operations within the application, but enable receiving and sending packets with a minimum number of CPU cycles. It does not provide any networking stack and instead helps to bypass the kernel network stack to deliver high performance.

When it comes to EC2 instance support, DPDK is supported on Enhanced Networking instances, both Intel-based ixgbevf and AWS Elastic Network Adapter (ENA). All Nitro-based instances, such as C5, M5, I3, and T3, as well as Intel-based instances, such as C4, M4, and T2, provide DPDK support. The Amazon drivers, including the DPDK driver for ENA, are available on GitHub. DPDK support for ENA has been available since version 16.04. The ENA Poll Mod Driver (PMD) is a DPDK poll-mode driver for the ENA family. The ENA driver exposes a lightweight management interface with a minimal set of memory mapped registers and an extendable command set through an admin queue.

AWS Graviton processors are Amazon-built ARM-based custom CPUs designed to deliver price performance. While Graviton2 delivered major leap in performance and capabilities over Graviton1, Graviton3 based C7g EC2 instances can deliver up to 25% better compute performance than its predecessor. C7g instances are a great fit for telecom workloads with extensive compute requirements such as data plan functions, billing, Network Data Analytics Functions (NWDAF) and

machine learning workloads. For ML workloads, Graviton3 processors deliver up to three times better performance compared to Graviton2. It also supports DDR5 memory, which increases the memory bandwidth by 50% when compared to DDR4.

DPDK and SR-IOV are not mutually exclusive and can be used together. An SR-IOV NIC can write data on a specific VM that hosts a virtual function. The data is then consumed by a DPDK-based application. The following figure illustrates the difference in packet flow between a non-DPDK and a DPDK-optimized application:

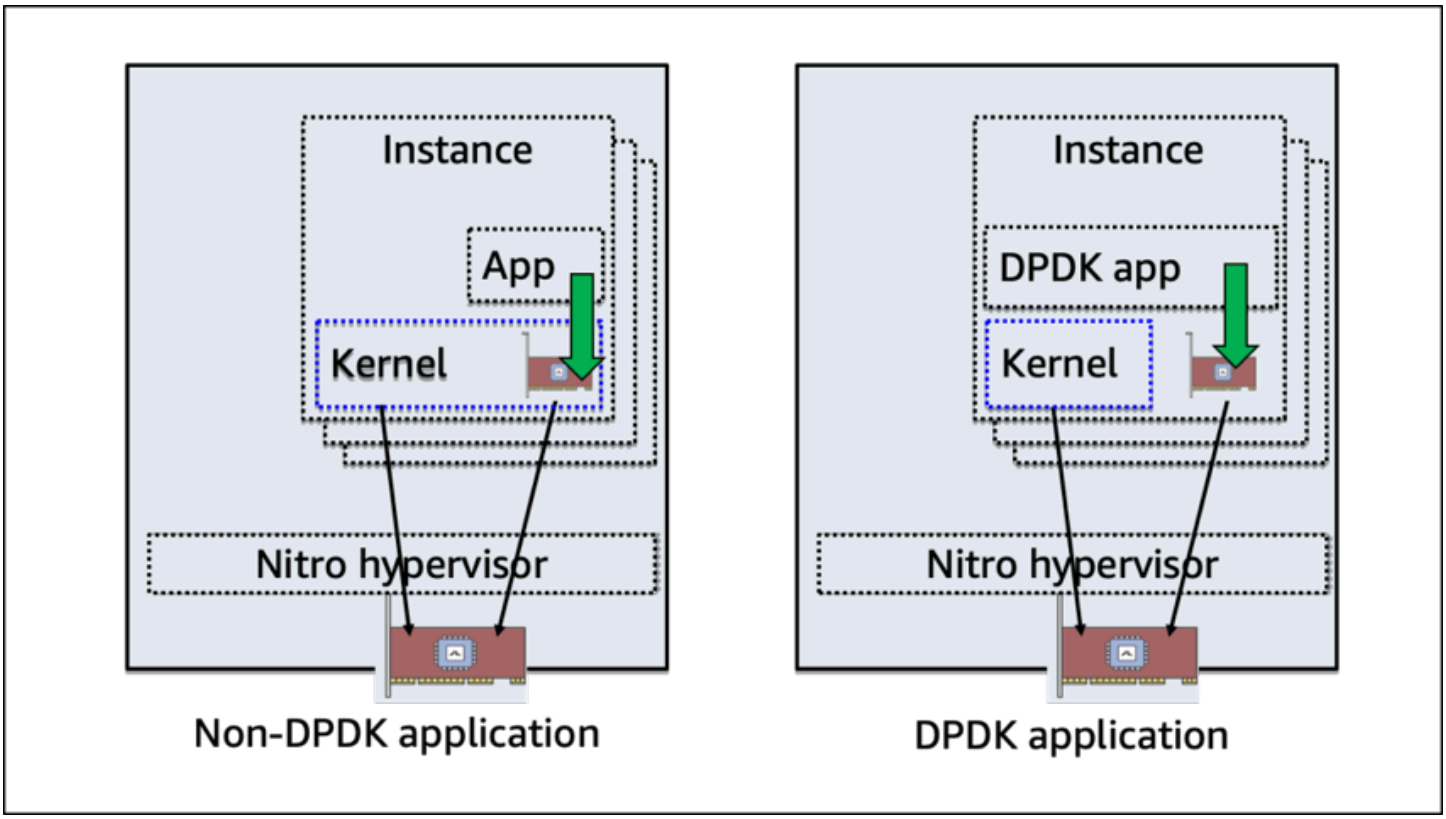


Figure 4 – Non-DPDK versus DPDK packet path

Non-Uniform Memory Access (NUMA)

There are multiple factors that can affect the performance of the VNFs and CNFs hosted on EC2 instance including CPU over utilization, memory use, the EBS volume, network statistics, or if the application isn't non-uniform memory access (NUMA) aware. In NUMA architecture, each CPU has access to its own assigned memory, known as local memory. Each CPU can also access memory allocated to other CPUs, known as foreign memory. If applications hosted on your instances aren't NUMA aware, then accessing the foreign memory incurs some additional costs and might affect performance.

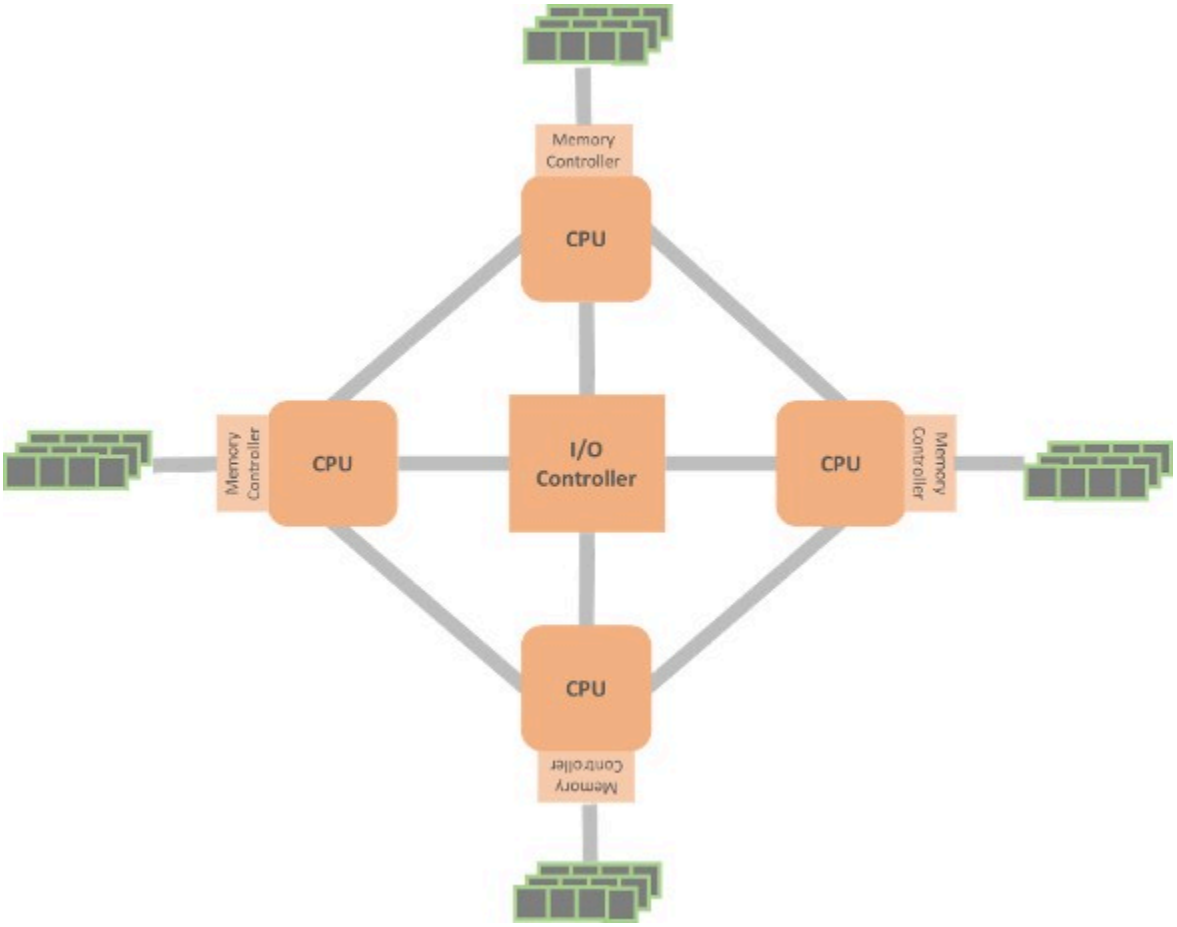


Figure 5 – NUMA architecture

The memory access time varies with the location of the data to be accessed. If the data resides in local memory, access is fast. If the data resides in remote memory, access is slower. The advantage of the NUMA architecture as a hierarchical shared memory scheme is its potential to improve average case access time through the introduction of fast, local memory. For more information, see Optimizing Applications for NUMA

All EC2 instances that support more than one CPU also support NUMA. These include `i4i.8xlarge`, `r6g.8xlarge`, `c6g.8xlarge`, `m6.8xlarge`, `m6i.8xlarge`, and above.

Running the following command on a NUMA supported instance will provide detailed information that can be used by the VNF, CNF ISV, or both.

```
sudo numactl -H
```

Huge pages

Huge pages can improve performance for workloads that execute large amounts of memory access. This feature of the Linux kernel enables processes to allocate memory pages of size 2MB/1GB (instead of 4K). Additionally, memory allocated using huge pages is pinned in physical memory and cannot be swapped out. Huge Page support is configurable on supported instance types. The important thing to note is that Huge Pages make memory access faster, however you cannot overcommit memory.

Running the following command on an EC2 instance will provide detailed information that can be used by the VNF, CNF ISV, or both.

```
sudo grep Huge /proc/meminfo
```

CPU pinning (CPU affinity)

CPU pinning is a technique that enables the binding and unbinding of a process or a thread to a CPU, or a range of CPUs, so that the process or thread will execute only on the designated CPU or CPUs rather than any CPU. This is useful when you want to dedicate vCPU to CNF and avoid sharing and dynamic rescheduling of CPUs.

Amazon Elastic Kubernetes Service (Amazon EKS) supports Kubernetes pod workloads where CPU cache affinity and scheduling latency significantly affect workload performance, by allowing alternative CPU management policies to determine some placement preferences on the Amazon EKS self-managed worker nodes.

Placement groups

Amazon EC2 placement groups allow you to influence the placement strategy of instances on the underlying hardware.

With Amazon EC2 placement groups on AWS Outposts and Local Zones, you can now use the spread and partition placement strategies to improve the resilience of workloads configured for high availability.

- A spread placement group places each instance of the group on a distinct rack to reduce correlated failures.

- A partition placement group distributes instances across logical partitions such that groups of instances in one partition do not share a rack with instances of a different partition.

Multus Container Network Interface (CNI)

Amazon Elastic Kubernetes Service (Amazon EKS) now supports the Multus Container Networking Interface (CNI) plugin, enabling pods running in EKS clusters to attach multiple network interfaces in support of advanced networking configurations.

Multus is an open source CNI plugin for Kubernetes that enables attaching multiple network interfaces to pods. Multus acts as a meta plugin, invoking additional CNI plugins that can operate on multiple network interfaces attached to pods. Use cases that commonly require pods with multiple interfaces include running 5G core and RAN networks on Kubernetes.

Amazon EKS provides a highly available managed Kubernetes service that is available in all global AWS regions, and supported in edge locations like AWS Local Zones and AWS Outposts. Using Multus with Amazon EKS enables advanced networking across these environments to run containerized network functions that deliver high quality content to end users.

Here's an example of how multi-homed pods can work on AWS. The following image shows two pods with two network interfaces, `eth0` and `net1`. In both cases, the Amazon VPC CNI manages the pod `eth0` (default Multus delegate). Interface `net1` is managed by Multus via the `ipvlan` CNI plugin for pod1, which handles the user plane (for example, voice, video) traffic separated from the Kubernetes control plane traffic. Whereas pod2 `net1` gets connected to the host elastic network interface through the host-device CNI plugin, and enables DPDK to accelerate the packet processing.

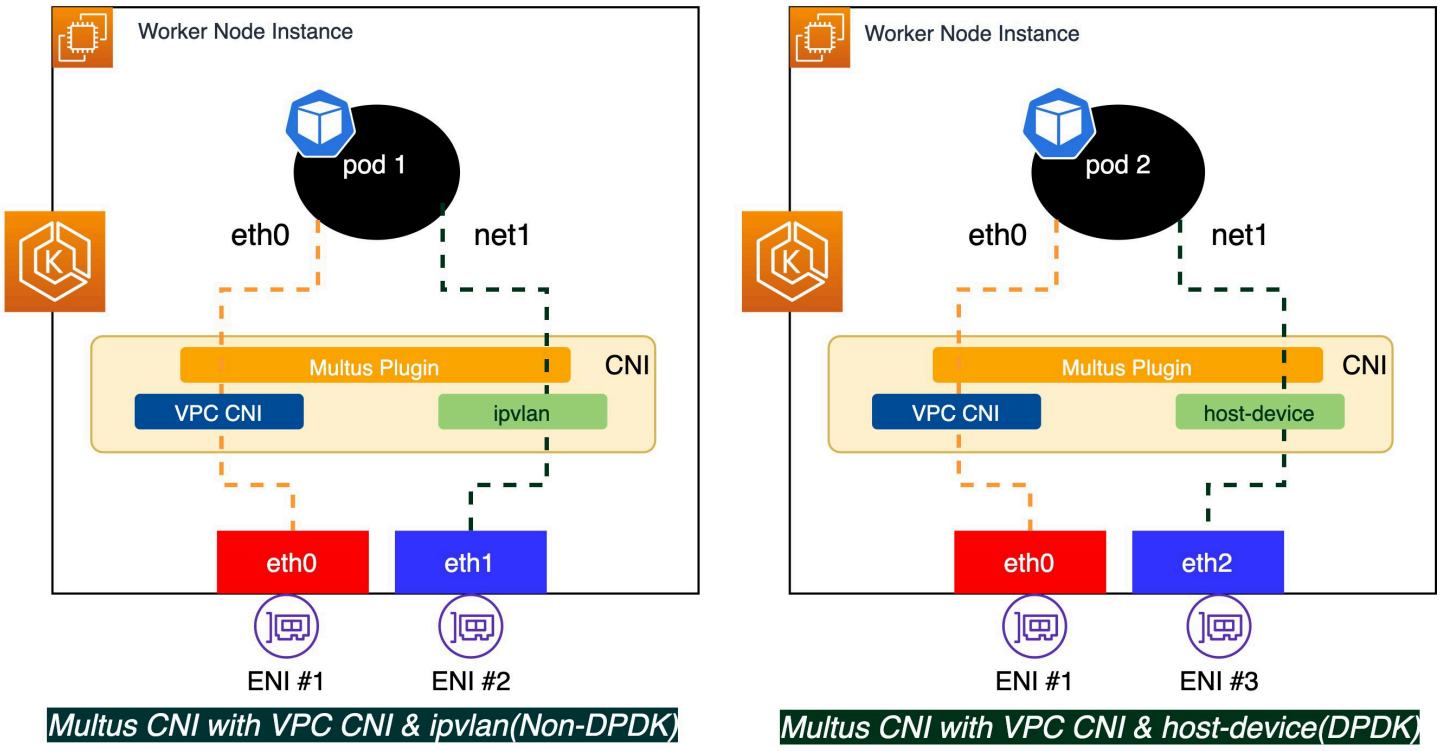


Figure 6 – Multus CNI with VPC CNI and Host Device (DPDK)

Amazon EC2 performance evolution and implementation

AWS has evolved its Amazon EC2 platform from the early days of `cc2` instances, which used the Xen hypervisor and paravirtualization with up to 10 Gbps of throughput, to the current Nitro-based family, which scale up to 400 Gbps (and millions of pps) for the largest instance types, such as `c5n`.

In order to improve performance in a virtualized environment, SR-IOV technology was used to bypass the hypervisor, resulting in the first version of enhanced networking, which provided improved performance, and lowered jitter and latency. The `C3` instance family was the first to introduce the Enhanced Networking concept, and more than halved the latency of its predecessor, `CC2`. The first release of Enhanced Networking used Intel-based chipsets (`ixgbevf`); the later release was based on an in-house based solution called Enhanced Network Adapter (ENA). This is the reason why the references are made to two variants of enhanced networking:

- Enhanced networking using Intel-based chipsets.
- Enhanced networking using AWS ENA, fully in-house developed Network Interface Card (NIC).

The C4 generation saw the introduction of the Annapurna Labs-based chipset, which replaced Intel. This instance family provides both networking and storage-optimized performance. The overall performance limit is 10 Gbps; however, workloads requiring both storage and network optimized performance were able to take advantage of this type of optimization and architecture.

AWS Nitro-powered C5 instances was another major step to improve the performance further. The AWS Nitro System delivers high-speed networking with hardware offload, high-speed EBS storage with hardware offload, NVMe local storage, hardware protection/firmware verification for bare metal instances, and all business logic required to control EC2 instances. In more simplified terms, the Nitro System is a lightweight hypervisor combined with the Nitro Security Chip, and Nitro Card for storage and networking. The switch from Intel to ENA has allowed us to deliver much better performance due to increased number of queues (8 instead of 6 with Intel-based chipsets). C5 family delivers performance of up to 25 Gbps and this limit goes to millions of pps and ~100 Gbps with the largest C5n, network optimized instances.

The AWS Graviton2 processor-based C6g family of instances deliver 40% better price performance than C5 instances. C6G provides up to 38 Gbps EBS bandwidth, which is more than two times more compared to C5n instances Graviton2 processor provides enhanced security through features like always-on 256-bit DRAM encryption and by supporting encrypted EBS storage volumes by default. It supports 50% faster per core encryption performance compared to first-generation AWS Graviton. It is important to note that chipsets are future proof to deliver performance of up to 400 Gbps.

Finally, the culmination of the performance evolution resulted in release of AWS Graviton3-powered C7 instances. C7 instances deliver the best price performance in Amazon EC2 for compute-intensive applications, and are the first in the cloud to feature DDR5 memory, which provides 50% higher memory bandwidth compared to DDR4 memory to enable high-speed access to data in memory. C7g instances deliver 20% higher enhanced networking bandwidth compared to C6g instances for network intensive applications, such as network appliances. These instances are ideal for high-compute telecom workloads like video encoding, machine learning, and distributed analytics.

AWS Nitro Enclaves enables customers to create isolated compute environments to further protect and securely process highly sensitive data such as personally identifiable information (PII), financial, and intellectual property data within their Amazon EC2 instances. Nitro Enclaves uses the same Nitro Hypervisor technology that provides CPU and memory isolation for EC2 instances. Enclaves offers an isolated, hardened, and highly constrained environment to host security-critical applications. Nitro Enclaves can help telecom customer address their confidential computing

requirement as it includes cryptographic attestation for customer software, so that customer can be sure that only authorized code is running, as well as integration with the AWS Key Management Service (AWS KMS), so that only your enclaves can access sensitive material.

Enabling enhanced networking

As covered in the previous section, enhanced networking can be based on Intel `ixgbevf` or EC2 ENA adaptor. The first step in enabling ENA is to check and verify what type of driver you have. Following commands can be run to determine driver type:

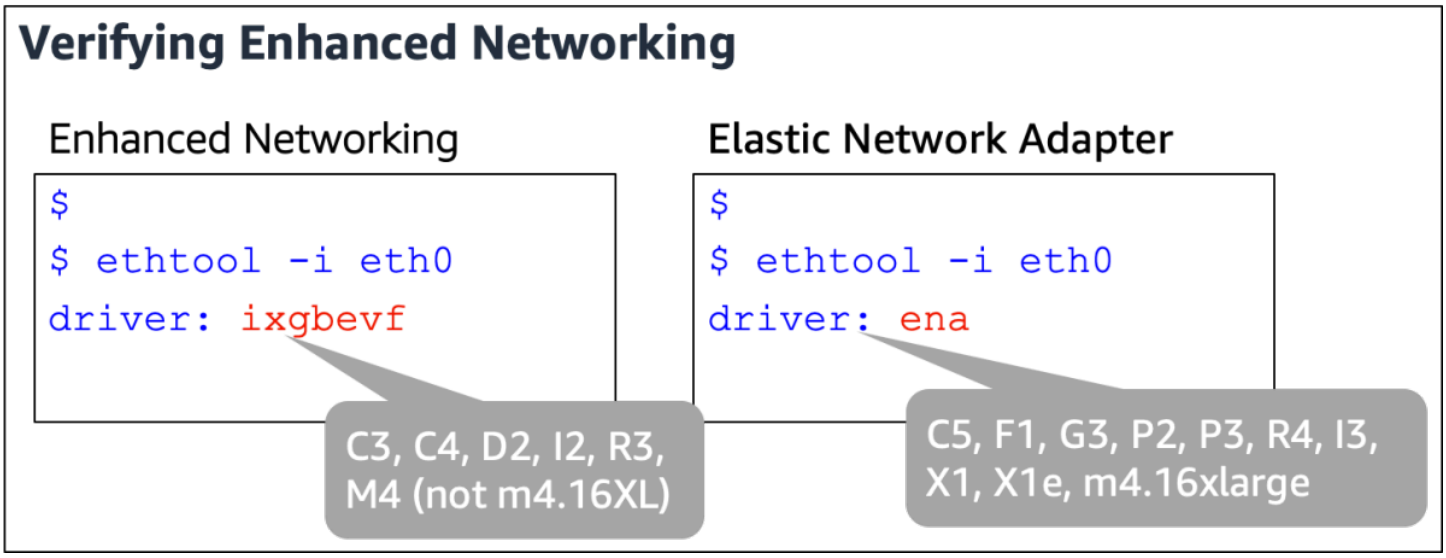


Figure 7 – Verifying enhanced networking

With the driver type determined, the following commands can be used to determine if an instance has `ixgbevf` or ENA enhanced networking enabled:

ixgbevf enhanced networking:

```
aws ec2 describe-image-attribute --image-id ami_id \
  --attribute sriovNetSupport
```

ENA enhanced networking:

```
aws ec2 describe-image-attribute --image-id ami_id \
  --attribute enaSupport
```

Figure 8 – Verifying Enhanced Networking, continued

Finally, from the following sample output, it can be seen what the output looks like with `ixgbevf` support and ENA support enabled, respectively:

```
% aws ec2 describe-instance-attribute \
  --instance-id i-07312ca8e93d69514 \
  --attribute sriovNetSupport
{
    "InstanceId": "i-07312ca8e93d69514",
     "SriovNetSupport": {
        "Value": "simple"
    }
}
```

ixgbevf Support!

Figure 9 – Verifying Enhanced Networking, continued

```
% aws ec2 describe-instances
  --instance-id i-07a94b1806d6cd309 \
  --query "Reservations[].Instances[].EnaSupport"
[
    true
]
```

ENA Support!

Figure 10 – Verifying Enhanced Networking, continued

If an instance has been launched without enhanced networking enabled, the following process can be used to enable it:

1. Connect to the instance that does not have ENA enabled.
2. Download the driver.
3. Enable ENA support on the instance and verify that it has been enabled.

At this point, a new AMI can be built with ENA enabled so that it can be reused in the future.

4. Restart the instance to continue operating with enhanced networking support enabled.

If the instance type supports the Elastic Network Adapter for enhanced networking, the detailed procedures to enable it are outlined in Enabling Enhanced Networking with the Elastic

Network Adapter (ENA) on Linux Instances.

If the instance type supports the Intel 82599 Virtual Function interface for enhanced networking, the detailed procedures to enable it are outlined in Enabling Enhanced Networking with the Intel 82599 VF Interface on Linux Instances.

Overall instance bandwidth quotas

As a general guide, the smaller sizes of C5, M5, and R5 instance types can sustain up to 10- Gbps network performance. Larger instance sizes can sustain between 10–25 Gbps. Smaller sizes of C5N provide up to 25 Gbps with the largest C5n instances scaling up to 100 Gbps. Some examples of instance type, configuration, and network performance:

Table 1 – c5 and c5n instance family configuration and performance comparison

Model	vCPU	Mem (GiB)	Network Performance (Gbps)	Model	vCPU	Mem (GiB)	Network Performance (Gbps)
c7g.large	2	4	Up to 12.5	c6n.large	2	4	Up to 25
c7g.xlarge	4	8	Up to 12.5	c6n.xlarge	4	8	Up to 25
c7g.2xlarge	8	16	Up to 15	c6n.2xlarge	8	16	Up to 25
c7g.4xlarge	16	32	Up to 15	c6n.4xlarge	16	32	25

Model	vCPU	Mem (GiB)	Network Performance (Gbps)	Model	vCPU	Mem (GiB)	Network Performance (Gbps)
c7g.8xlarge	32	64	15	c6n.8xlarge	36	72	50
c7g.12xlarge	48	96	22.5	c6n.12xlarge	48	96	75
c7g.16xlarge	64	128	30	c6n.16xlarge	64	128	100

Aggregate bandwidth throughput for instances between Availability Zones (within a VPC) or between instances in a peered VPC scenario is 25–100 Gbps, depending on instance type (see Table 1). Similarly, aggregate bandwidth to VPC endpoints, such as Amazon S3, is 25- 100 Gbps. Single TCP flow is limited to 10 Gbps for instances in the same placement group and 5 Gbps between instances anywhere else. (TCP flow is defined as traffic going through a single TCP port.) A placement group is a logical grouping, or cluster, of instances within a single Availability Zone, that allows applications to use low latency 10-Gbps network. For more information, see Amazon EC2 instance types.

Amazon Virtual Private Cloud

Amazon Virtual Private Cloud (Amazon VPC) is the virtual data center in the AWS Cloud. A VPC closely resembles the traditional network that an organization might operate in their own data center, but with all the benefits of elastic and on-demand scaling. Like a traditional data center, VPCs can have public and/or private subnets. Private subnets do not have routes to the internet gateway, but public subnets do. You have complete control over your virtual networking environment, including the selection of your IP address range, creation of subnets, and configuration of route tables and network gateways. You can use both IPv4 and IPv6 in your VPC for secure and easy access to resources and applications.

As in traditional data centers, you can control the flow of inbound and outbound traffic by using network access control lists (NACLs). NACLs act as a firewall for associated subnets and are stateless. To control traffic flow at the instance level, use security groups. Security groups act as firewalls for associated EC2 instances and are stateful, which automatically allows return traffic without needing to define special rules.

In addition to public and private IP addresses, it's important to understand the concept of an Elastic IP address and elastic network interface (ENI). An ENI is analogous to a virtual network interface card (NIC), and you can apply multiple ENIs to an instance. You can also move an ENI to another instance in the same subnet. An Elastic IP address is a static public IP address that is applied to an ENI and it can be associated to another instance after an instance is terminated. The main reason why we have Elastic IP addresses is so that rules such as ACLs, DNS entries, and similar do not have to change if an instance fails. Multiple EIPs can be applied to an ENI. The concept of Elastic IP addresses is particularly useful when designing high availability workloads, where an Elastic IP address gets assigned as a secondary IP address of an active instance. That instance is then continuously monitored through CloudWatch tools, and that Elastic IP address can be switched through a script or API call to another instances, should failure occur.

External connectivity options for VPCs include the following:

- An **internet gateway** is a horizontally scaled, highly available VPC component that allows communication between your instances in a VPC and the internet.
- A **NAT gateway** enables instances in a private subnet to connect to the internet or other AWS services, but prevents an internet request from initiating a connection with those instances.
- A **virtual private gateway** represents the anchor of the AWS side of a VPN connection between Amazon VPC and the customer environment. In case of a VPN connection between VPC and on-

premises environment, VGW connects to the customer gateway, which can be a hardware or software appliance.

All of these building blocks have been represented in the following figure to illustrate how they relate to traditional networking constructs and connectivity.

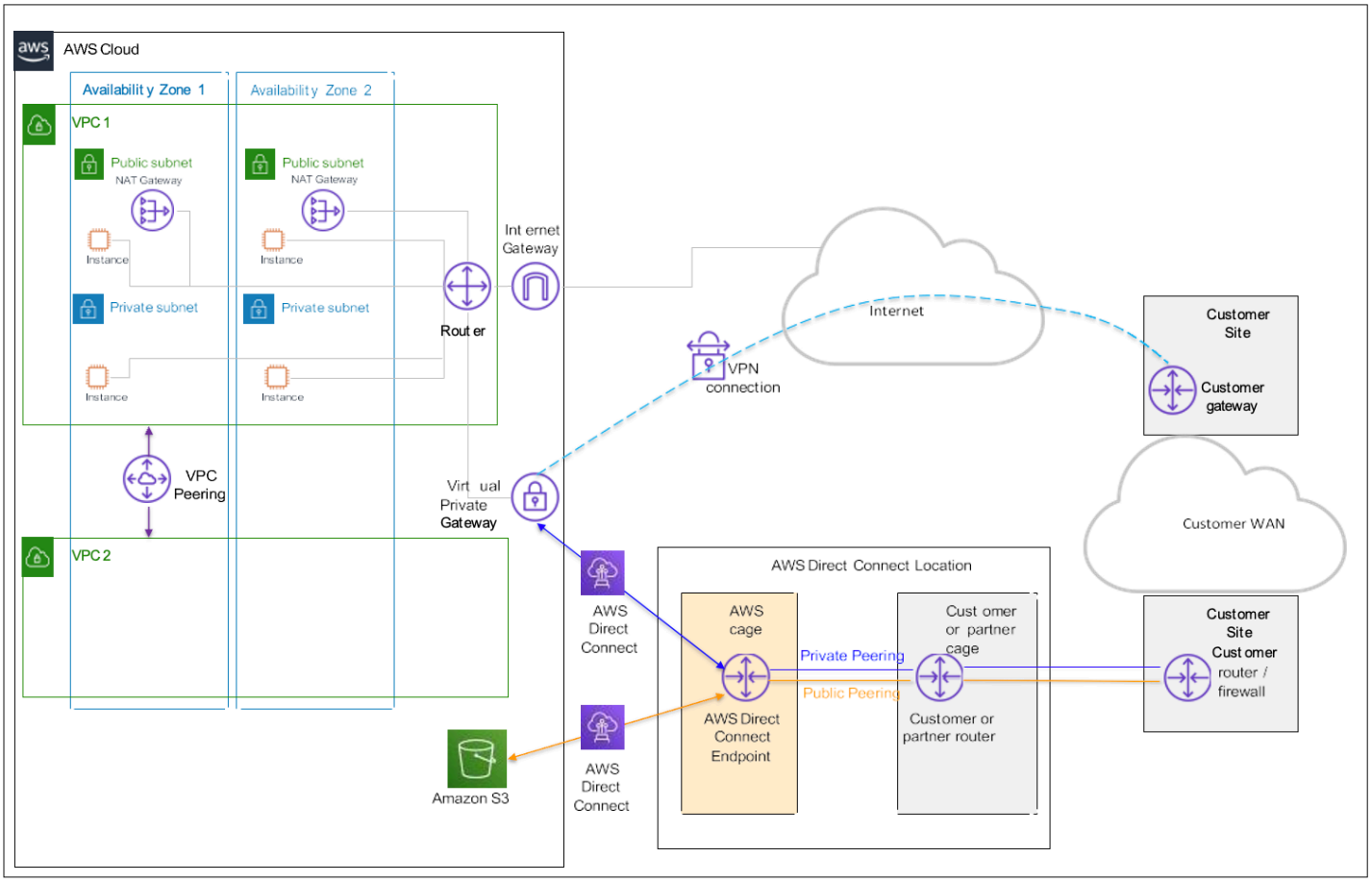


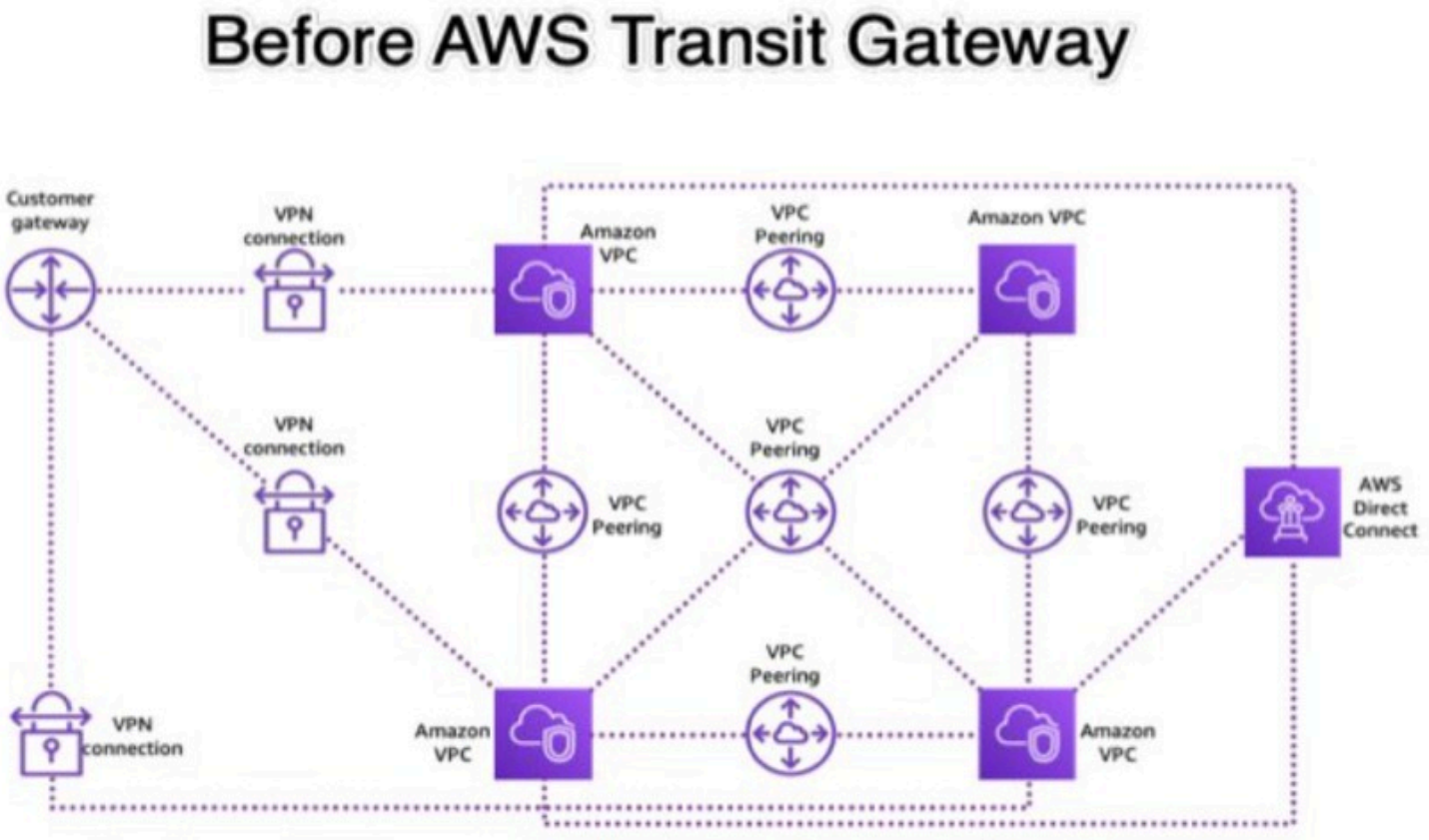
Figure 11 – Sample connectivity diagram between Amazon VPC and on-premises environment with DX and VPN connectivity

You can establish connectivity between two different VPCs by using a *VPC peering* connection. VPC peering allows instances in either VPC to communicate with each other as if they were within the same network. VPCs can be in different Regions and belong to different accounts. Since VPC peering is effectively point-to-point connectivity, it can be operationally costly and cumbersome to use without the ability to centrally manage the connectivity policies. That was the primary reason for introducing AWS Transit Gateway.

AWS Transit Gateway

As you grow the number of workloads running on AWS, you'll need to be able to scale your networks across multiple accounts and VPCs. Previously, you had to connect pairs of VPCs using VPC peering. Recently, AWS introduced AWS Transit Gateway, which provides a more scalable way for interconnecting multiple VPCs. Telecom services that has low latency requirements can be achieved via AWS Transit Gateway as it supports connectivity to the attached VPCs from the on-premises network via both AWS Site-to-Site VPN and AWS Direct Connect services using Border Gateway Protocol.

With AWS Transit Gateway, you only need to create and manage a single connection from the central gateway to each Amazon VPC, on-premises data center, or remote office across your network. AWS Transit Gateway acts as a hub that controls how traffic is routed among all the connected networks, which act like spokes. This hub-and-spoke model significantly simplifies management, and reduces operational costs because each network only has to connect to AWS Transit Gateway and not to every other network. Any new VPC is simply connected to the gateway and is then automatically available to every other network that is connected. This ease of connectivity makes it easy to scale your network as you grow. The following before-and-after diagrams illustrate the benefit of using AWS Transit Gateway:


Before AWS Transit Gateway
Customer gateway
VPN connection
Amazon VPC
VPC Peering
Amazon VPC
VPN connection
VPC Peering
VPC Peering
AWS Direct Connect
VPC Peering
VPC Peering
VPN connection
Amazon VPC
Amazon VPC

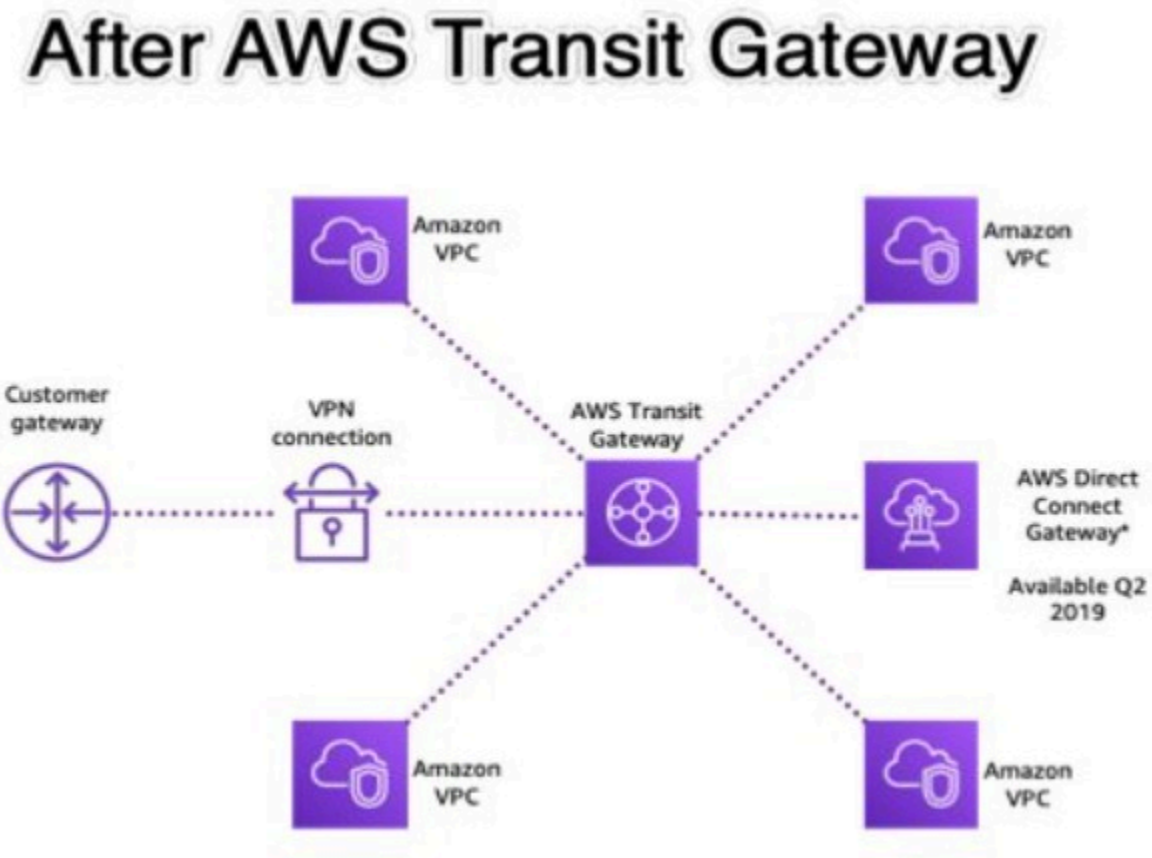

After AWS Transit Gateway
Amazon VPC
Amazon VPC
Customer gateway
VPN connection
AWS Transit Gateway
AWS Direct Connect Gateway*
Available Q2 2019
Amazon VPC
Amazon VPC

Figure 12 – Network connectivity before and after introducing AWS Transit Gateway

Transit Gateway (TGW) supports inter-regional connectivity via TGW peering, simplifying the connectivity between VPCs in different regions and on-premises networks.

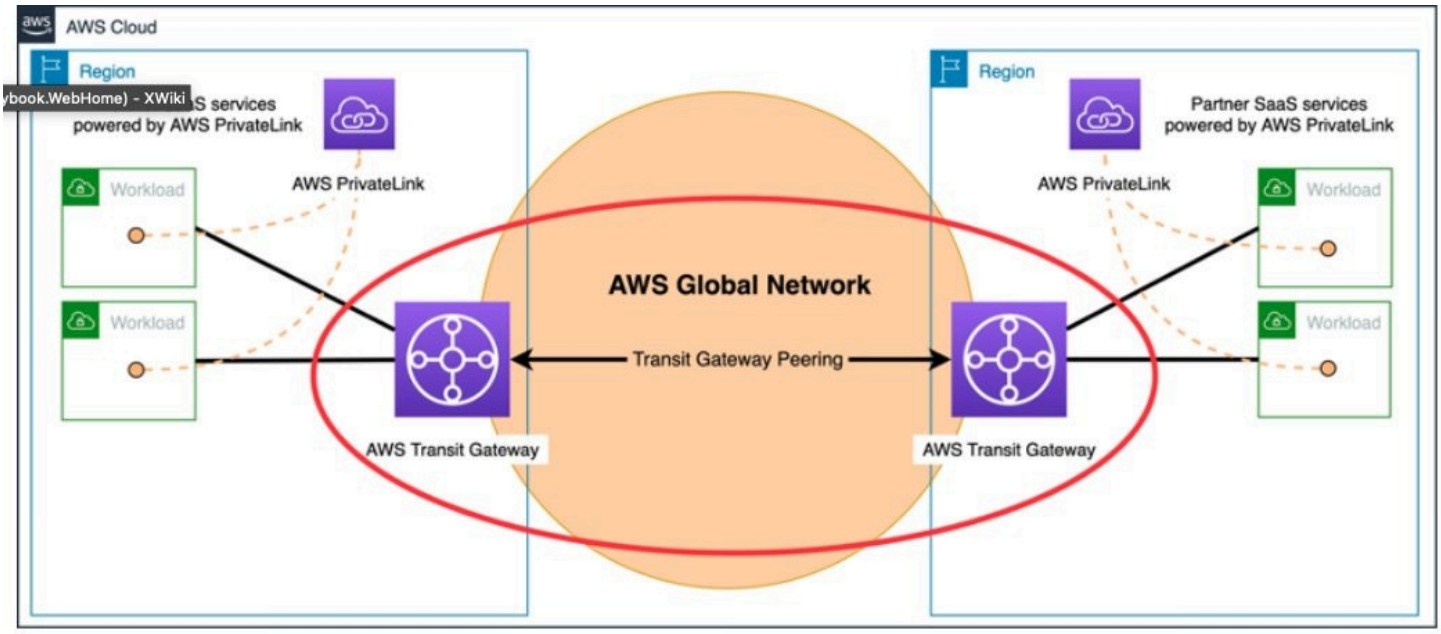


Figure 13 - TGW Inter-Region Peering

Transit Gateway supports on-premises connectivity via both Transit Vif (DX connection) and site-to-site IPsec VPN tunnels. A single VPN tunnel can achieve up to 1.25 Gbps bandwidth. TGW allows Equal Cost Multi Path (ECMP) which is used to scale VPN throughput with additional VPN tunnels associated with the TGW. More information: https://aws.amazon.com/blogs/networking-and-content-delivery/scaling-vpn- throughput-using-aws-transit-gateway/

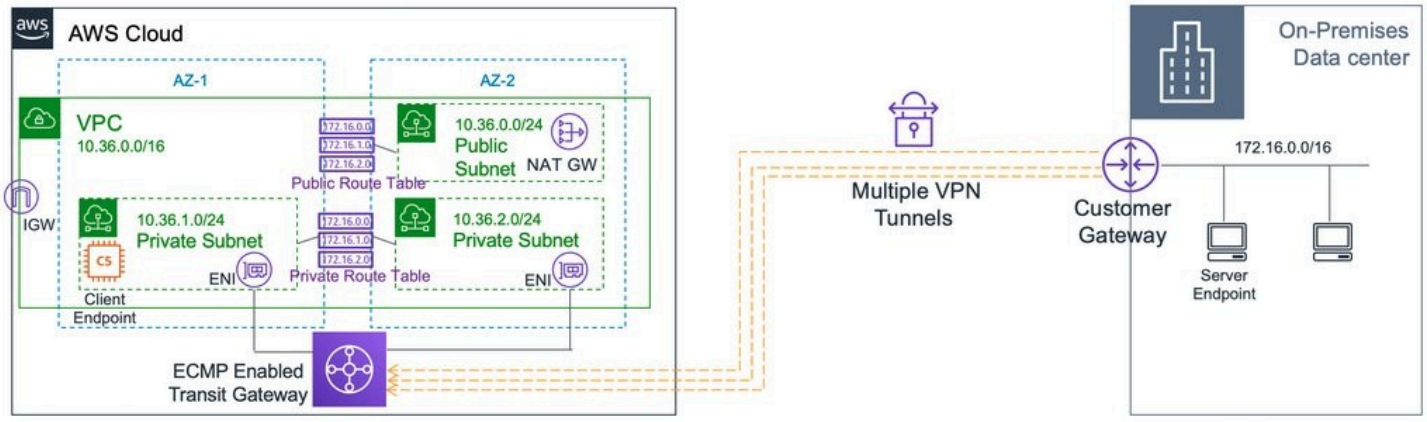


Figure 14 – TGW using ECMP with multiple VPN tunnels

Finally, Elastic Load Balancing allows incoming traffic to be equally distributed across multiple EC2 instances in a VPC and increases the availability of your application. While Elastic Load Balancing supports Application, Classic, and Network Load Balancers, typically only Network Load Balancers will be used for telecom workloads. Network Load Balancers function at Layer 4 of the OSI model, support both TCP and UDP traffic, and can handle millions of requests per second.

Network load balancer supports Elastic IPs which remains unchanged as the load balancer scales internally.

TGW also support Intra region peering with other TGWs in the customer network. With intra-region peering capability, customers no longer need to create bridge VPCs between multiple transit gateways or attach a single VPC to multiple Transit Gateways for routing traffic between different Transit Gateways in the same AWS Region. Intra-region peering simplifies

routing and interconnectivity between VPCs and on-premises networks that are serviced and managed via separate Transit Gateways. This feature allows customers the flexibility to deploy multiple Transit Gateways with separate administrative domains, while providing an easy way to interconnect these Transit Gateways in a more native manner. Using intra-region peering, you can build flexible network topologies and easily integrate your network with a third-party or partner-managed network in the same AWS Region. If you are already familiar with Transit Gateway inter-region peering, it works exactly the same way except that the peered Transit Gateways are in the same AWS Region.

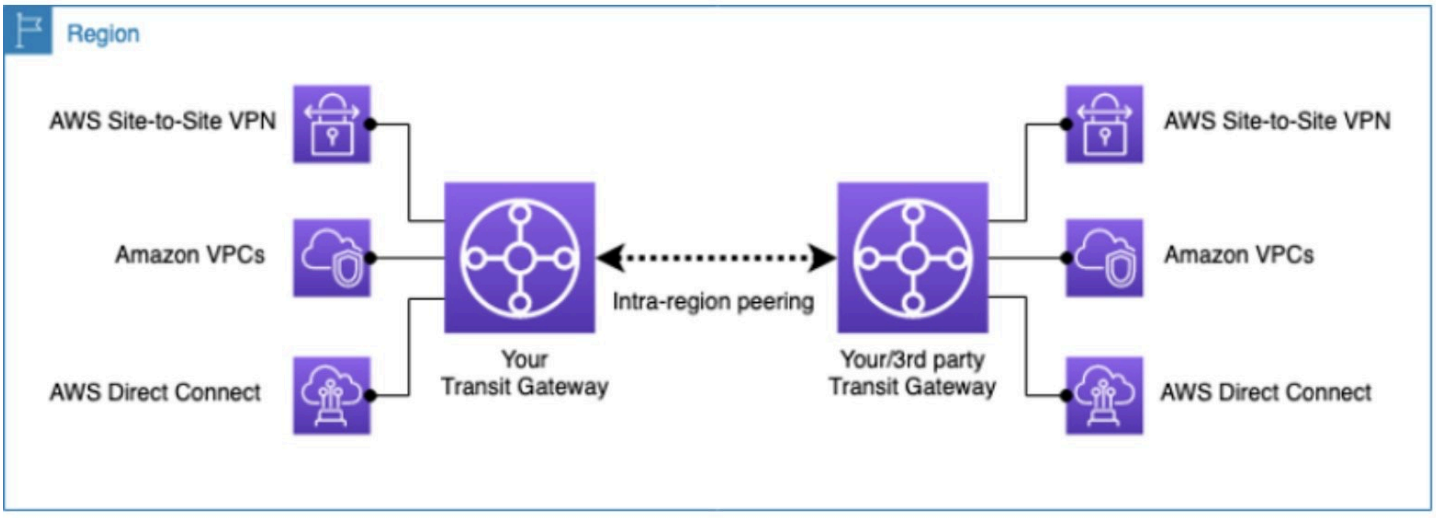


Figure 15 – TGW attachments

Customers can deploy flexible topologies to fit the use-cases, AWS recommends setting up a full mesh architecture if you have multiple Transit Gateways as shown in the following diagram.

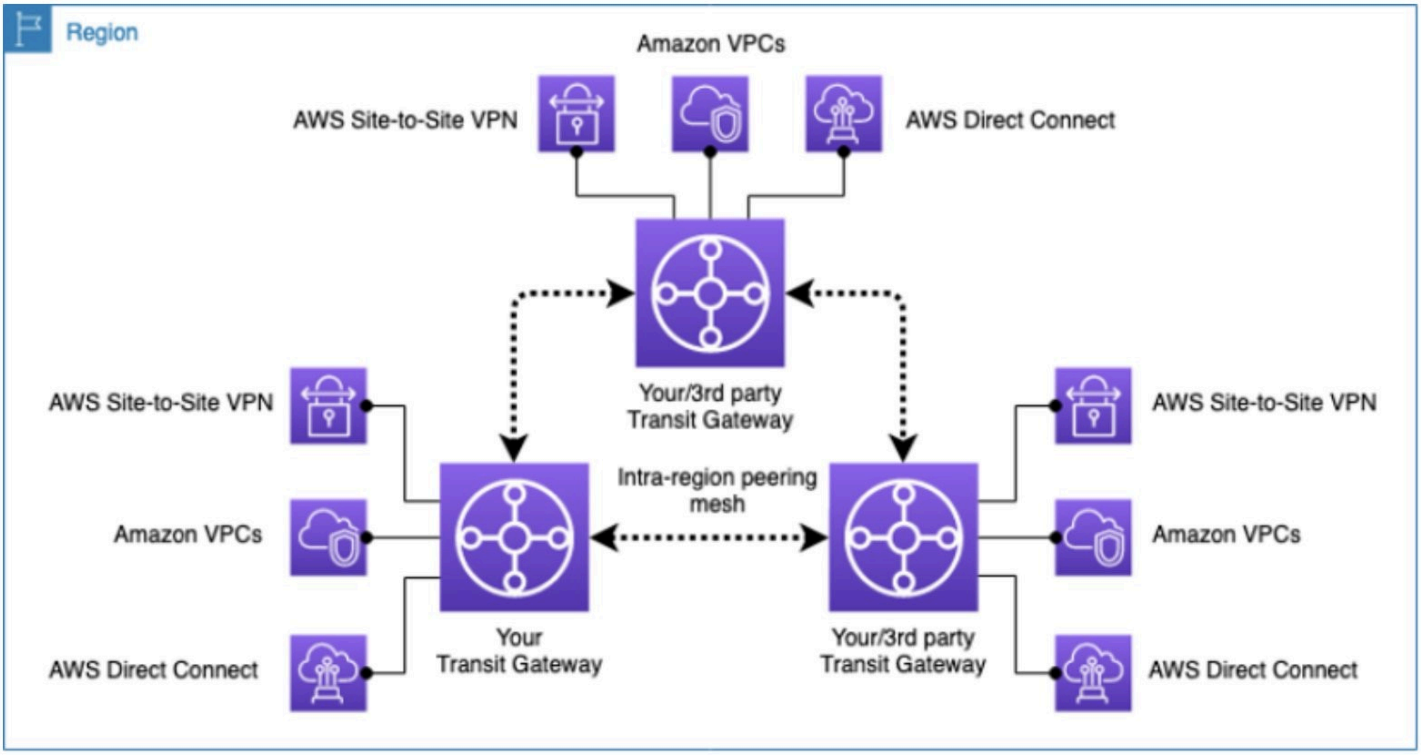


Figure 16 – TGW inter-region mesh connectivity

Centralized network traffic inspection could be facilitated by either AWS Network Firewall or an AWS Gateway Load Balancer and setting up an inspection VPC in the architecture. Using static routes in the route-table associated with the intra-region peering attachment, customers can steer traffic coming from the third-party transit gateway to the security inspection VPC. TGW has to be set with appliance mode enabled on the inspection VPC's Transit Gateway attachment to keep traffic symmetry in both directions as shown in the following diagram.

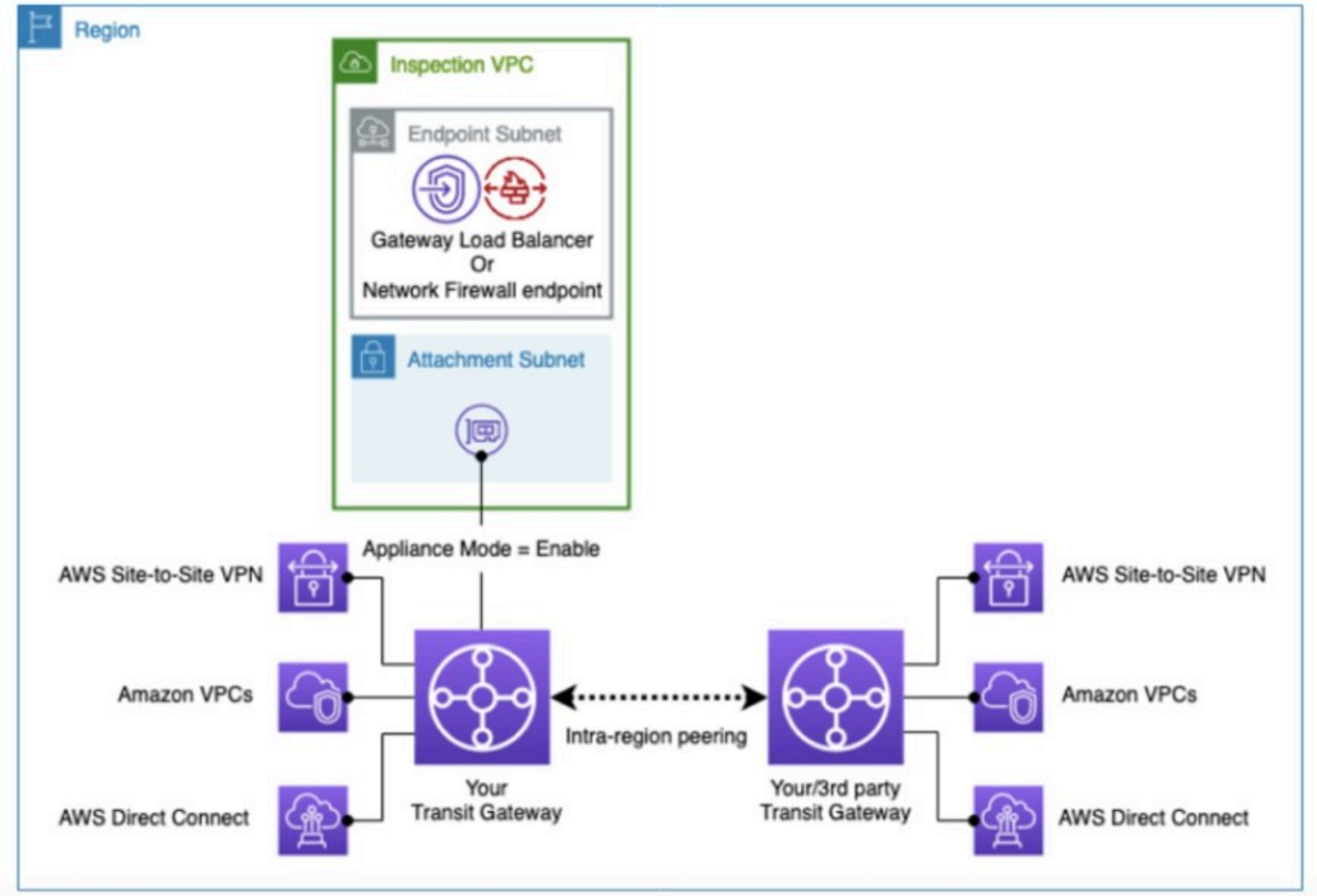


Figure 17 – TGW in appliance mode with AWS Network Firewall and Gateway load balancer

Transit Gateway Connect

TGW connect is a native support to connect SD WAN infrastructure with AWS via TGW. SD-WAN network appliances no longer require IPsec VPNs with TGW and TGW connect support Generic Routing Encapsulation for higher bandwidth performance compared to a VPN connection. TGW Connect supports Border Gateway Protocol (BGP) for dynamic routing which simplifies network design and reduces operational costs. Integration with TGW Network Manager enables increased visibility and access to performance metrics and telemetry data from both virtual appliances in AWS and the branch appliances.

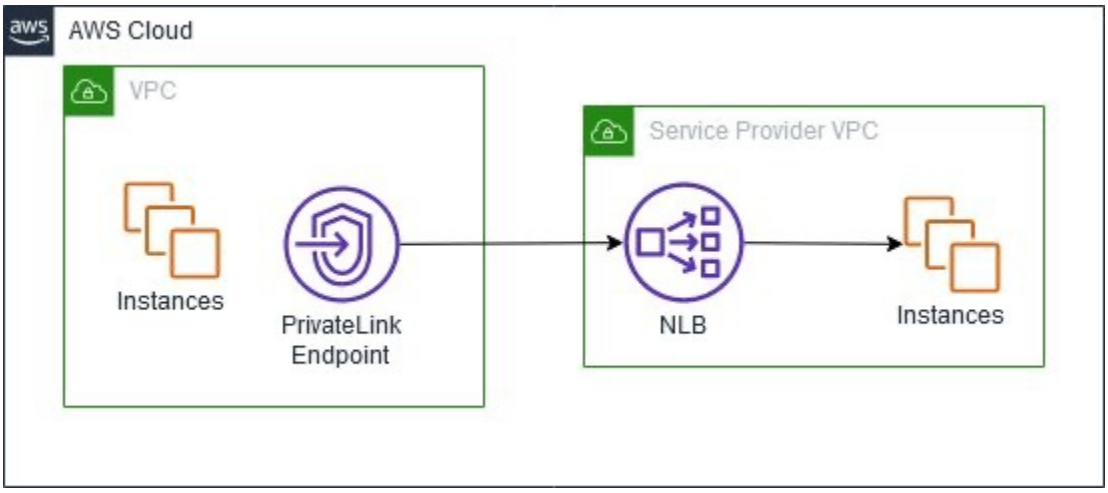
AWS PrivateLink and service endpoint

AWS PrivateLink is a highly available, scalable technology that enables you to privately connect your VPC to supported AWS services, services hosted by other AWS accounts (VPC endpoint services), and supported by AWS Marketplace Partner Services. You do not need to use an internet gateway, NAT device, public IP address, AWS Direct Connect connection, or AWS Site-to-Site VPN connection to communicate with the service. Therefore, your VPC is not exposed to the public internet. You can also create your own VPC endpoint service, powered by AWS PrivateLink, and enable other AWS customers to access your service. This is especially useful to enable Telecom use cases that provide a service for customers (for example, OSS/BSS deployments on AWS).

An AWS PrivateLink consists of a VPC endpoint (VPCE) and a corresponding Endpoint Service:

- **VPC endpoint** — A VPC endpoint enables connections between a VPC and supported services, without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. VPC endpoints are virtual devices. They are horizontally scaled, redundant, and highly available VPC components. There are different types of VPC endpoints that enable connectivity for the supported service:
  - **Gateway endpoint:** A gateway endpoint is a gateway that is a target for a route in your route table used for traffic destined to either Amazon S3 or DynamoDB. There is no charge for using gateway endpoints.
  - **Interface endpoint:** An interface endpoint is an elastic network interface with a private IP address from the IP address range of your subnet. It serves as an entry point for traffic destined to a service that is owned by AWS or owned by an AWS customer or partner. You are billed for hourly usage and data processing charges. (https://aws.amazon.com/privatelink/pricing/)
  - **Gateway Load Balancer endpoint:** A Gateway Load Balancer endpoint is an elastic network interface with a private IP address from the IP address range of your subnet. It serves as an entry point to intercept traffic, and route it to a network or security service that you've configured using a Gateway Load Balancer. You specify a Gateway Load Balancer endpoint as a target for a route in a route table. Gateway Load Balancer endpoints are supported only for endpoint services that are configured using a Gateway Load Balancer. You are billed for hourly usage and data processing charges
- **Endpoint service** — Your own application or service in your VPC. Other AWS principals (for example, accounts, and users) can create an endpoint from their VPC to your endpoint service.

To use AWS PrivateLink, create a VPC endpoint for a service in your VPC. You create the type of VPC endpoint required by the supported service. This creates an elastic network interface in your subnet with a private IP address that serves as an entry point for traffic destined to the service. The following diagram shows the basic architecture to securely connect your VPC to an AWS service that supports AWS PrivateLink.



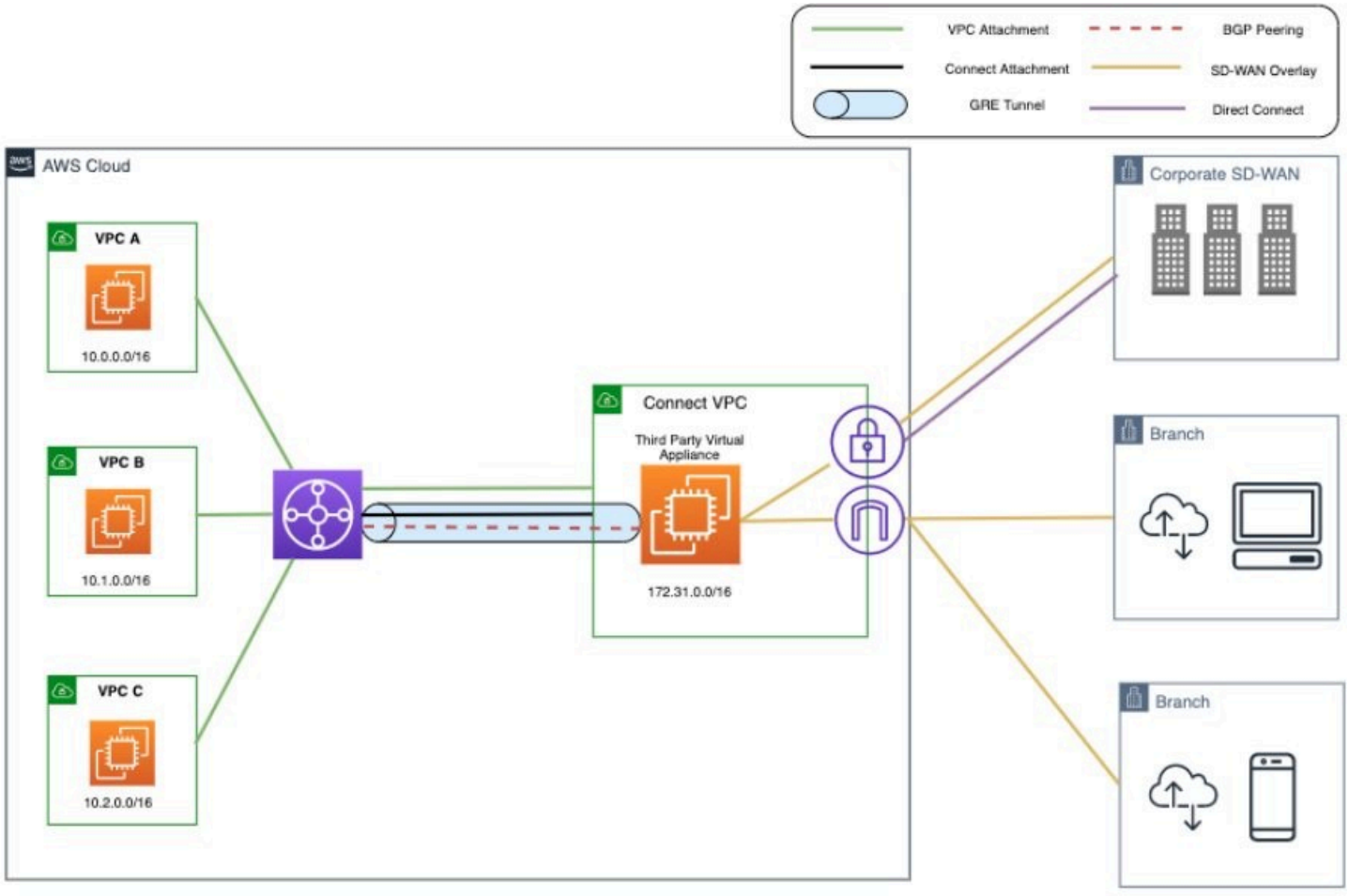


Figure 18 – AWS PrivateLink and service endpoints

Amazon CloudWatch

Amazon CloudWatch monitors your AWS resources and the applications you run on AWS in real time. You can use CloudWatch to collect and track metrics, which are variables you can measure for your resources and applications. You can also create custom dashboards to display metrics about your custom applications, and display custom collections of metrics that you choose. You can create alarms that watch metrics and send notifications or automatically make changes to the resources you are monitoring when a threshold is breached. For example, you can monitor the CPU usage and disk reads and writes of your Amazon EC2 instances and then use this data to determine whether you should launch additional instances to handle increased load. You can also use this data to stop under-used instances to save money.

AWS Cloud computing resources are housed in highly available data center facilities. To provide additional scalability and reliability, each data center facility is located in a specific geographical area, known as a Region. Each Region is designed to be completely isolated from the other Regions, to achieve the greatest possible failure isolation and stability. Metrics are stored separately in regions, but you can use CloudWatch Cross-Region functionality to aggregate statistics from different Regions.

CloudWatch also has an agent that can runs on compute instances to enable extra functionality on top of the existing offering:

- Collect internal system-level metrics from Amazon EC2 instances across operating systems. The metrics can include in-guest metrics, in addition to the metrics for EC2 instances.
- Collect system-level metrics from on-premises servers. These can include servers in a hybrid environment as well as servers not managed by AWS.
- Retrieve custom metrics from your applications or services using the StatsD (Linux, Windows) and collected (Linux) protocols.
- Collect logs from Amazon EC2 instances and on-premises servers, running either Linux or Windows Server.

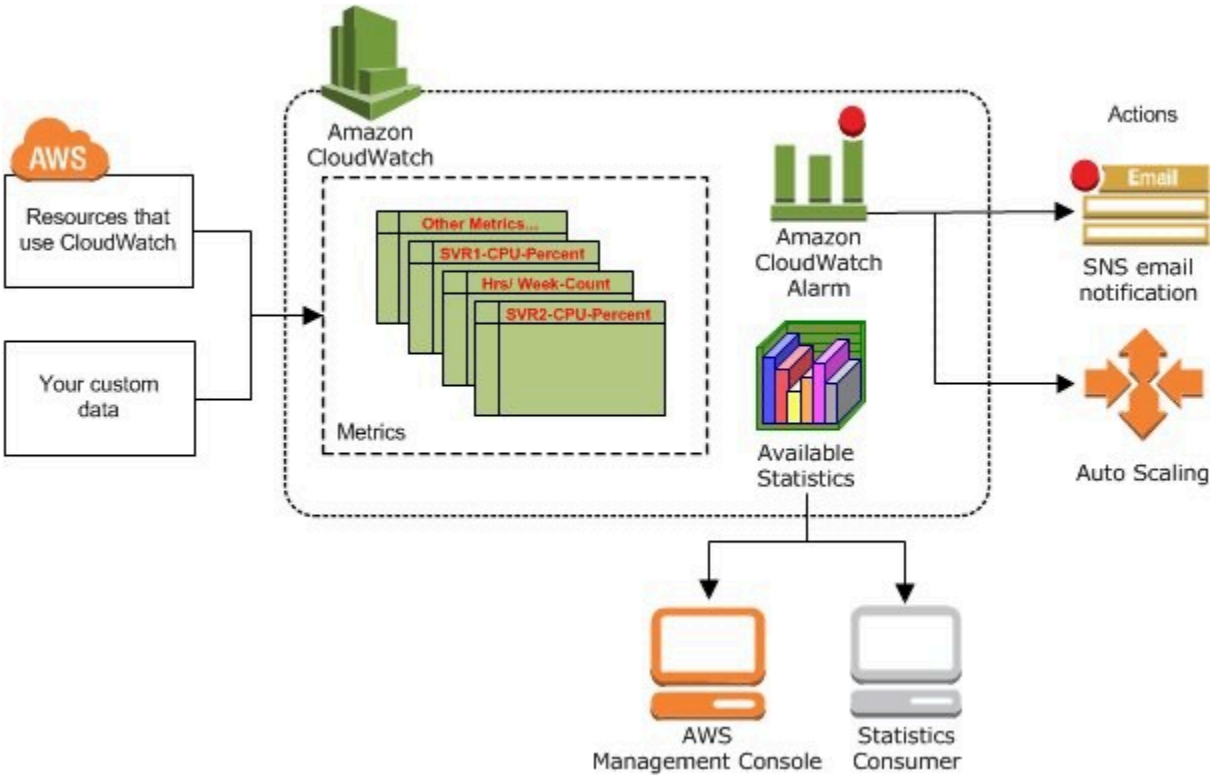


Figure 19 – CloudWatch monitoring

With CloudWatch, you gain system-wide visibility into resource utilization, application performance, and operational health.

For more information on extending CloudWatch to an on-premises network, see Build an observability solution using managed AWS services and the OpenTelemetry standard.

VPC IP Address Manager (IPAM)

IPAM enables management and auditing of IP address assignments across an organization's accounts, VPCs, and AWS Regions using a single operational dashboard. IPAM enables users to automate IP address assignment, monitor, troubleshoot, and audit network address assignments.

IPAM operates using a pool-based hierarchy. Pools are collections of CIDRs that organize IP space with an AWS account. Unused address space from top-level pools can be used to fill your regional pools. Further, applications or environments with different security needs can create additional pools. For example, developers can create different pools for `dev` (development) and `prod` (production) environments if they are subject to different connectivity requirements. The following figure demonstrates a sample IPAM pool hierarchy:

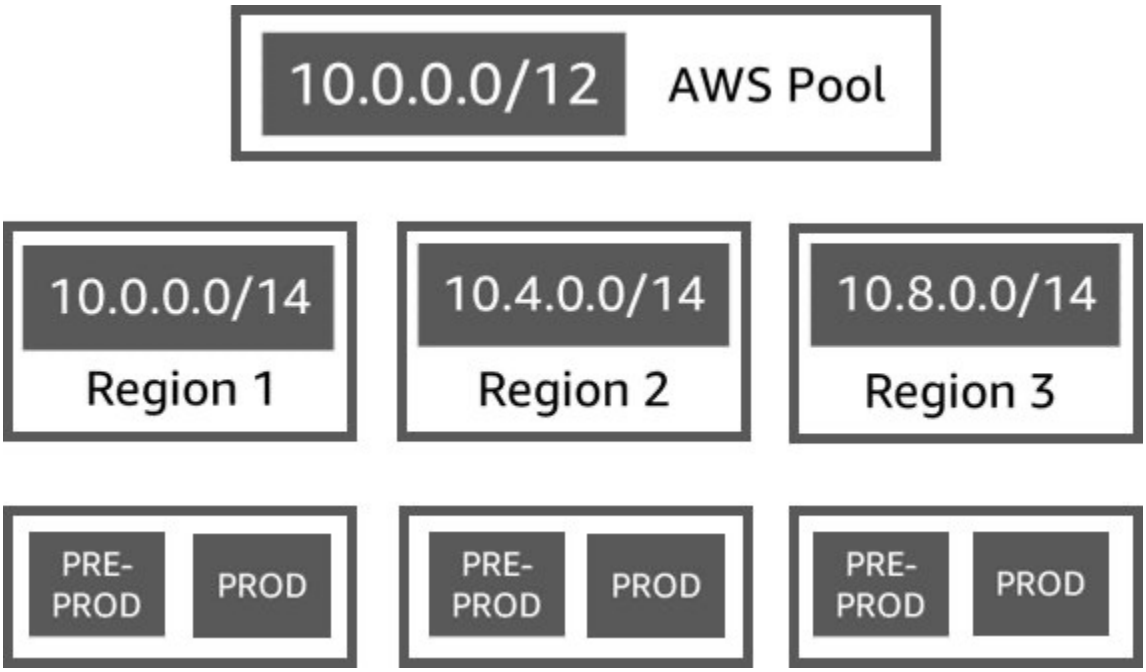


Figure 20 – IPAM service planning

After the IPAM pools have been configured, development teams and resources needing new IP address assignments are able to make use of an automated, self-service process, unblocking the developers, and eliminating errors from using manual processes that can lead to connectivity issues. With the IPAM self-service model, developers can directly create resources and receive IP addresses based on business rules in seconds, removing the delays in onboarding applications and improving the velocity of the development team

For network administrators, IPAM provides observability and auditing capabilities, helping to speed up troubleshooting, and providing oversight and monitoring of the used and unused addresses across an organization's global network address pool using a single dashboard. For each assigned address, IPAM tracks critical information such as the AWS account, the VPC, routing, and the security domain, eliminating the bookkeeping work that burdens administrators.

Having used IPAM to eliminate IP assignment errors, customers can use IPAM to monitor assigned addresses and receive alerts when potential issues are detected. This could include depleting IP addresses that can stall their network's growth or overlapping IP addresses that can result in erroneous routing. To further help troubleshoot network issues and audits of network security and routing policies, network administrators can also take advantage of the current and historical data that IPAM makes available to gain usage insights.

In summary, IPAM enables operators to:

- Organize IP address space into routing and security domains.
- Monitor IP address space that's in use and monitor resources that are using space against business rules.
- View the history of IP address assignments in your organization.
- Automatically allocate CIDRs to VPCs using specific business rules.
- Troubleshoot network connectivity issues.
- Enable cross-region and cross-account sharing of your Bring Your Own IP (BYOIP) addresses.

Network performance troubleshooting

VPC Flow Logs

VPC Flow Logs enable you to capture information about the IP traffic going to and from network interfaces in your VPC. Flow log data can be published to Amazon S3 or Amazon CloudWatch Logs. In addition to using flow logs for troubleshooting purposes, such as determining why traffic is not reaching a particular instance, they also can be used as a security tool to monitor the traffic that is reaching your instance.

VPC traffic mirroring

Traffic mirroring is an Amazon VPC feature that you can use to copy network traffic from an elastic network interface of Amazon EC2 instances. You can then send the traffic to out-of- band security and monitoring appliances for content inspection, threat monitoring, and troubleshooting. Key areas of enablement for operators include:

- **Network security:** VPC Traffic Mirroring provides access to granular network traffic that enable appliances (for example, network detection and response (NDR), intrusion detection systems (IDS), security information and event management (SIEM), next-generation firewall (NGFW), advanced threat prevention, and network forensics) to secure cloud infrastructure and workloads.
- **Network performance monitoring:** VPC Traffic Mirroring makes it possible for network performance monitoring solutions to identify performance bottlenecks and troubleshoot multi-tier applications. Agentless access to network traffic is the foundation for network observability, network performance management and diagnostics, packet capture systems, and application performance management.
- **Customer experience management:** VPC Traffic Mirroring helps provide network traffic to customer experience management systems, such as voice over IP (VoIP), and service quality analyzers.
- **Network troubleshooting:** VPC Traffic Mirroring assists with diagnosis of network issues, especially when visibility beyond what is available through VPC Flow Logs is needed. This includes situations when packet traces must be captured in a packet analyzer/packet capture appliance

VPC Traffic Mirroring works by establishing a session between a mirroring source and a mirroring target. The following demonstrates an example on how to monitor traffic that is entering/leaving a VPC from two EC2 instances in a public subnet and concurrently monitor intra-VPC traffic between two EC2 instances in a private subnet.

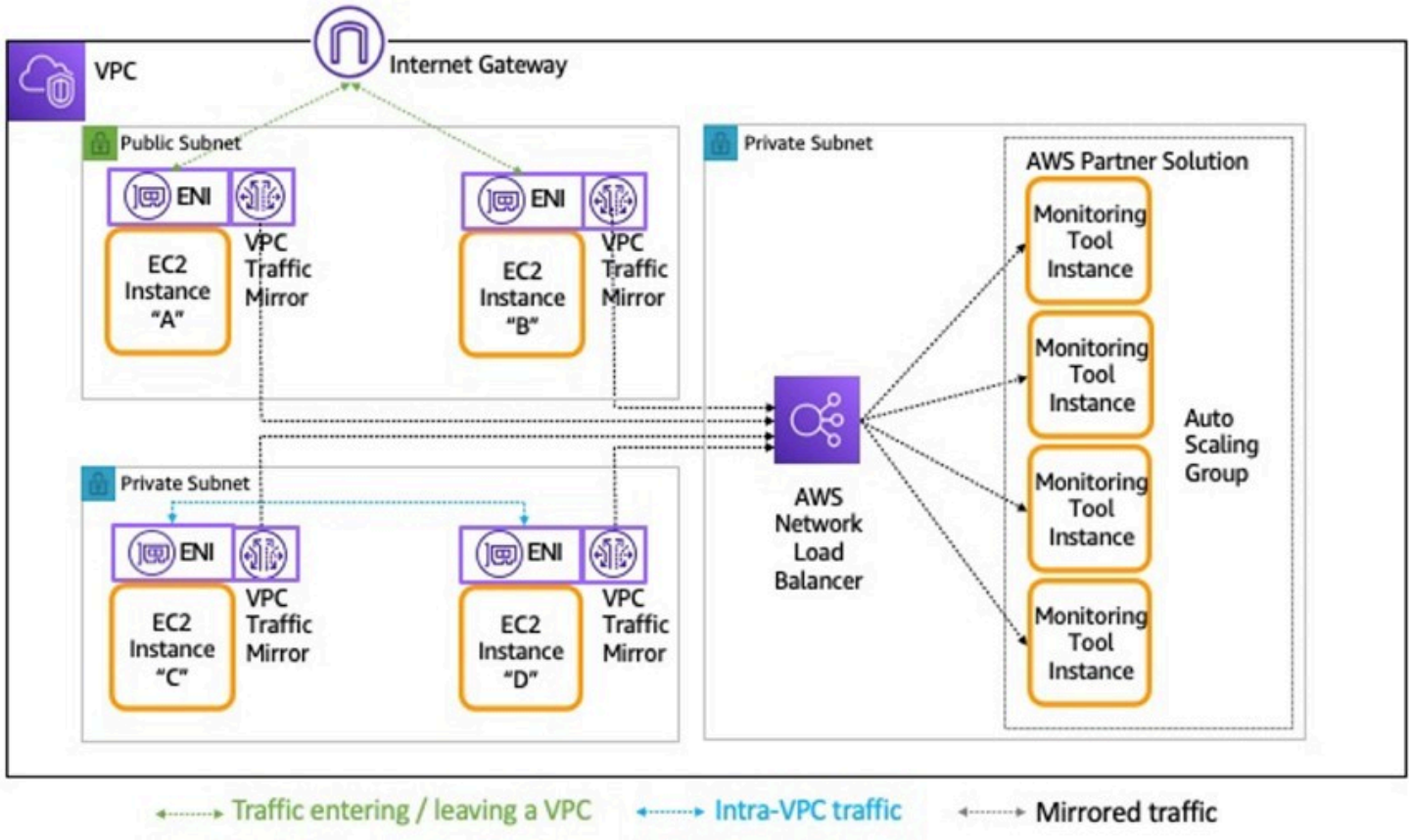


Figure 21– VPC traffic mirroring and monitoring

In this example, each of the EC2 instances are mirroring sources. The traffic mirroring target can be another ENI attached to a virtual monitoring appliance running on an EC2 instance, or a network load balancer (NLB) that balances traffic across multiple instances of a virtual monitoring appliance. These virtual monitoring appliance instances are front-ended by an NLB and deployed as part of an auto-scaling group, allowing the virtual monitoring appliance to scale-out or scale-in based on load. The monitoring appliance instances and the NLB can be placed in the same or different VPC. VPC traffic mirroring allows the user to configure filters to extract only the traffic that they are interested in. This minimizes load on the network. Note that mirrored traffic is counted as part of the instance bandwidth, so you must factor this into the sizing of the source instances.

AWS Direct Connect and VPNs

AWS Direct Connect (DX) provides a dedicated connection from your on-premises network to one or more Amazon VPCs. It's possible to create a single sub-1 Gbps connection or use a link aggregation group (LAG) to aggregate multiple 1 Gbps or 10- Gbps connections into a single managed connection. DX supports 100 Gbps dedicated connections at select locations. For upgrading your existing DC connections to 100 Gbps please follow this blog post: Upgrading AWS Direct Connect to 100 Gbps in 5 steps

DX uses VLANs to access Amazon EC2 instances running within the VPC. DX supports both static and dynamic routing through BGP. One of the following virtual interfaces (VIFs) must be created in order to use a DX connection:

- **Private virtual interface –** used to access VPC resources using private IP addresses
- **Public virtual interface –** used to access all AWS public services using public IP addresses
- **Transit virtual interface –** used to access one or more AWS Transit Gateways associated with DX gateways.

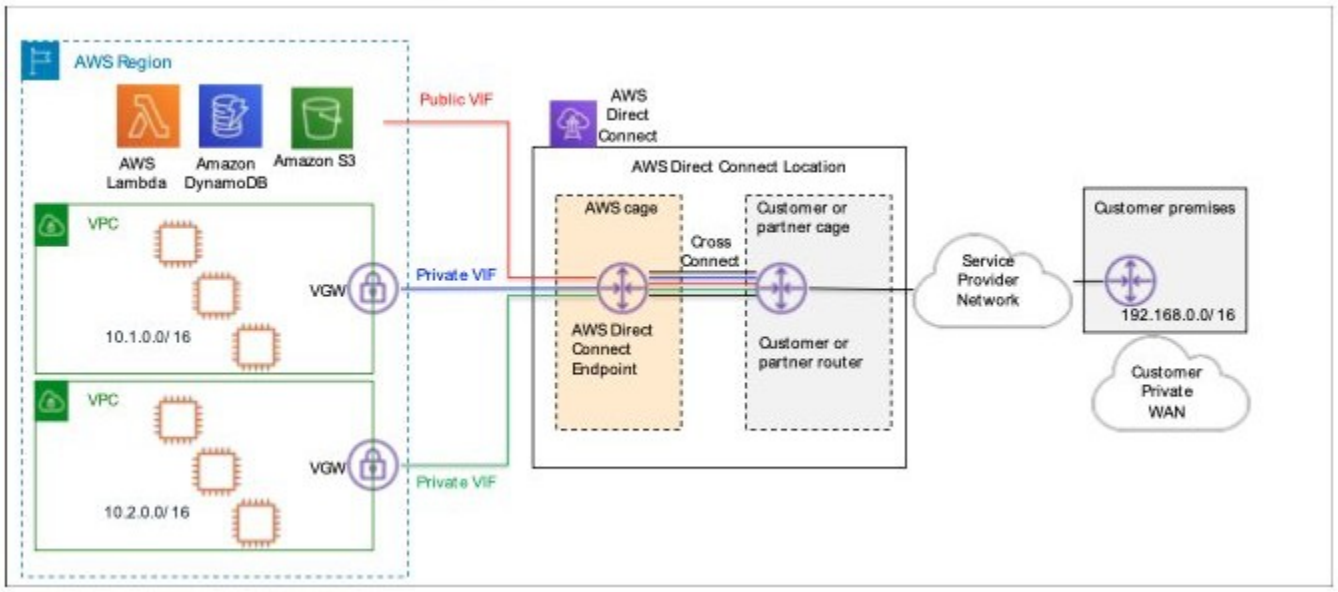


Figure 22 – AWS Direct Connect Virtual Interface Types

Direct connect makes use of BGP communities to influence the customer traffic. Local preference BGP communities can be used to influence the return traffic from the AWS to the on-premises

network for private and transit virtual interfaces. In public virtual interfaces, customers can use the scope BGP communities to define how far their public prefixes will be propagated into AWS network. The options to propagate the public prefixes are local AWS region, all AWS regions for a continent and global (all public AWS Regions. The AWS public routes are advertised with BGP community tags to the customer's device and the customer can filter the routes they receive from AWS. The options available are routes originating from the same region as the DX, routes originating from the same continent as the DX and global.

Direct Connect Gateway (DXGW) is a global resource, supporting multi-region and multi- accounts, that allows the customers to connect multiple VPCs to the on-premises network. A single DXGW supports connectivity up to 10 virtual gateways (VGW) and up to three transit gateways.

DX provides resiliency toolkit to help the customer architecture the network based on the workload type and required model of resiliency. For more information, see Explore the AWS Direct Connect Resiliency Toolkit.

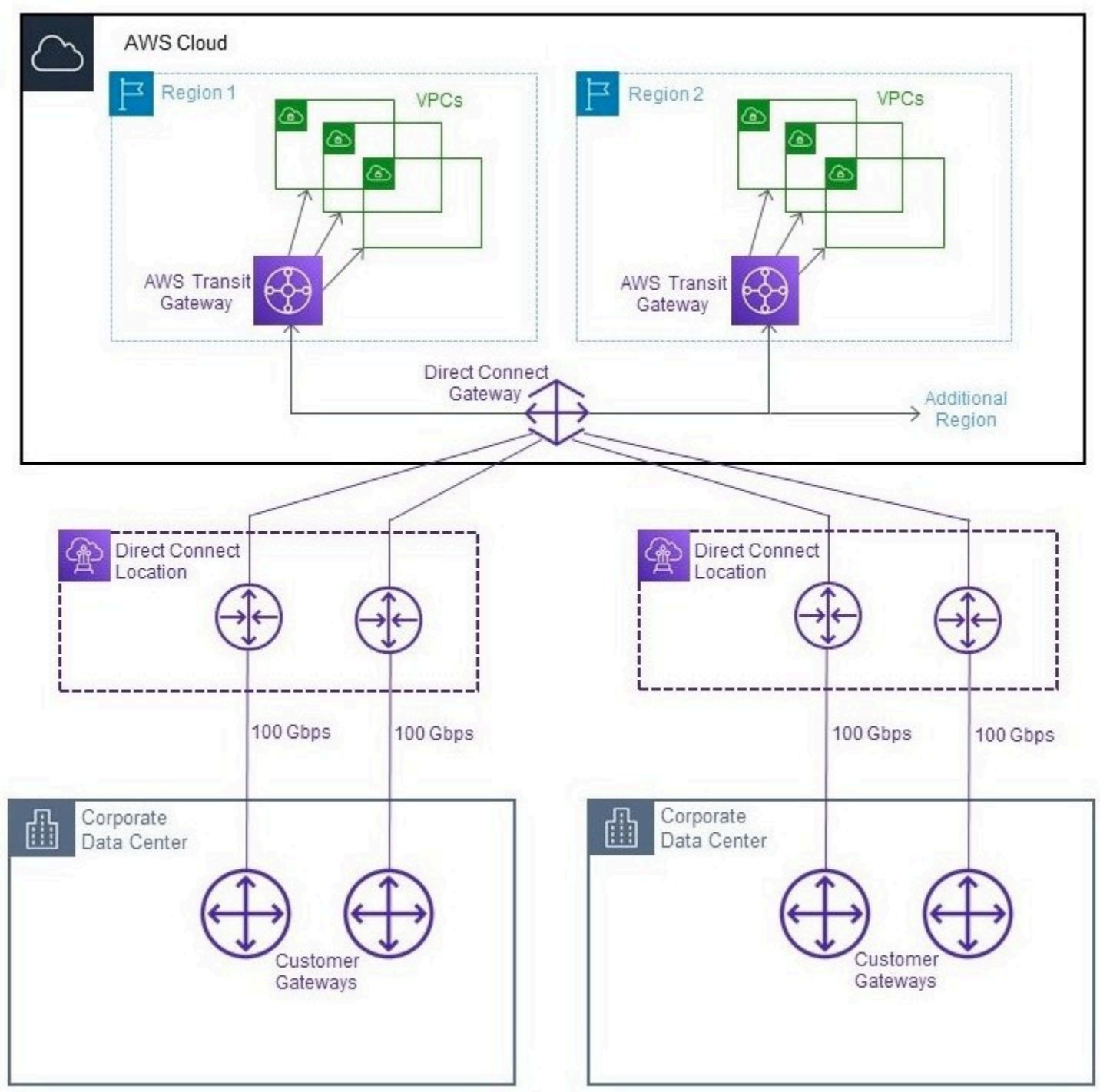


Figure 23 – AWS Direct Connect Gateway

Typically, DX is used for critical, latency sensitive workloads given the dedicated nature of the connectivity. AWS also offers a Service Level Agreement for AWS Direct Connect as per the following policy: https://aws.amazon.com/directconnect/sla/.

MAC Security

Direct Connect support MACsec to encrypt data from the on-premises network to AWS Direct Connect location. MACsec is an IEEE 802.1 Layer 2 standard, providing data confidentiality, data integrity and data origin authenticity. The feature is available on 10 Gbps and 100 Gbps dedicated connections and selected Regions.

More information:

1. https://aws.amazon.com/directconnect/locations/
2. https://aws.amazon.com/blogs/networking-and-content-delivery/adding-macsec-security-to-aws-direct-connect-connections/

MACsec benefits customers who want to exchange data with AWS securely and at the highest bandwidth available. This includes customers in regulated industries, such as financial services or healthcare, and customers with high-bandwidth workloads that have strict security requirements, such as media production and autonomous vehicle development. We strongly recommend using connections in more than one AWS Direct Connect location to help ensure resilience against device or colocation failure.

If the workload does not require the dedicated nature of DX, using an AWS managed VPN provides the option of creating an IPsec VPN connection over the internet between your on- premises environment and Amazon VPC. With an AWS managed VPN, you can take advantage of automated multi-data center redundancy and failover, which is built into the AWS side of VPN. Basically, a virtual private gateway will terminate two distinct VPN endpoints in two separate data centers. The redundancy can be further improved by also implementing redundancy at your side of connection and terminating VPN endpoints on two separate customer gateways at the on-premises environment. Finally, both dynamic and static routing options are supported to give you flexibility in setting your routing configuration. Dynamic routing uses BGP peering to exchange routing information between AWS and your on-premises environment. With dynamic routing, you can also specify routing priorities, policies, and weights (metrics) in your BGP advertisements and influence the network path taken between your networks and AWS.

The potential drawbacks of using an AWS managed VPN are that availability is dependent on the internet conditions, and the VPN adds complexity to implementing redundancy and failover (if necessary) at your end. DX, on the other hand, provides dedicated connectivity and minimal latency. It also requires new network circuits to be provisioned through your hosting provider, unless you are the hosting provider.

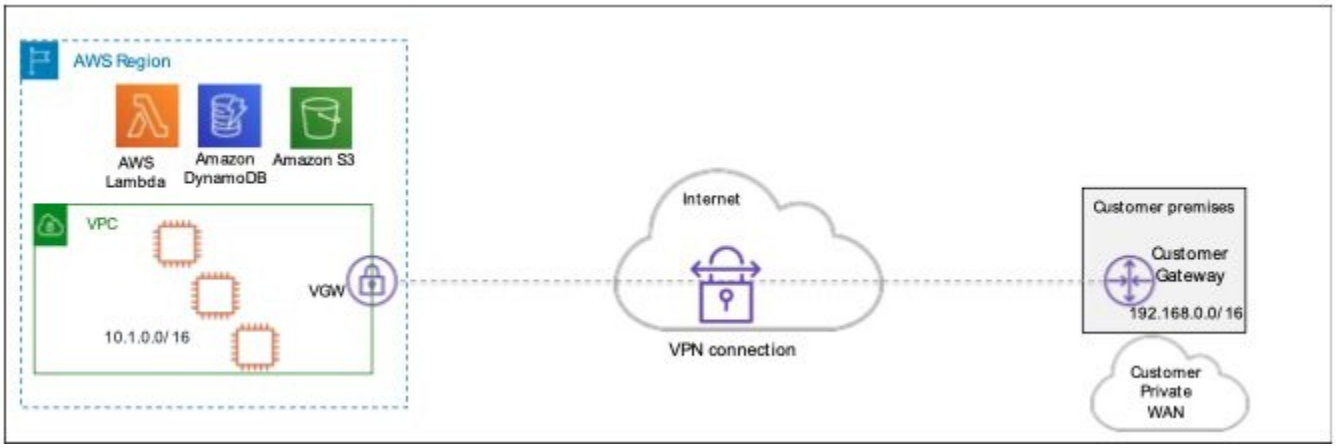


Figure 24 – AWS VPN connection

Multiple VPN endpoints from a VGW can support a cloud hub model. The AWS VPN Cloud Hub operates on a simple hub-and-spoke model that you can use with or without a VPC. Use this approach if you have multiple branch offices and existing internet connections and would like to implement a convenient, potentially low-cost hub-and-spoke model for primary or backup connectivity between these remote offices. The remote network prefixes for each spoke must have unique ASNs, and the sites must not have overlapping IP ranges.

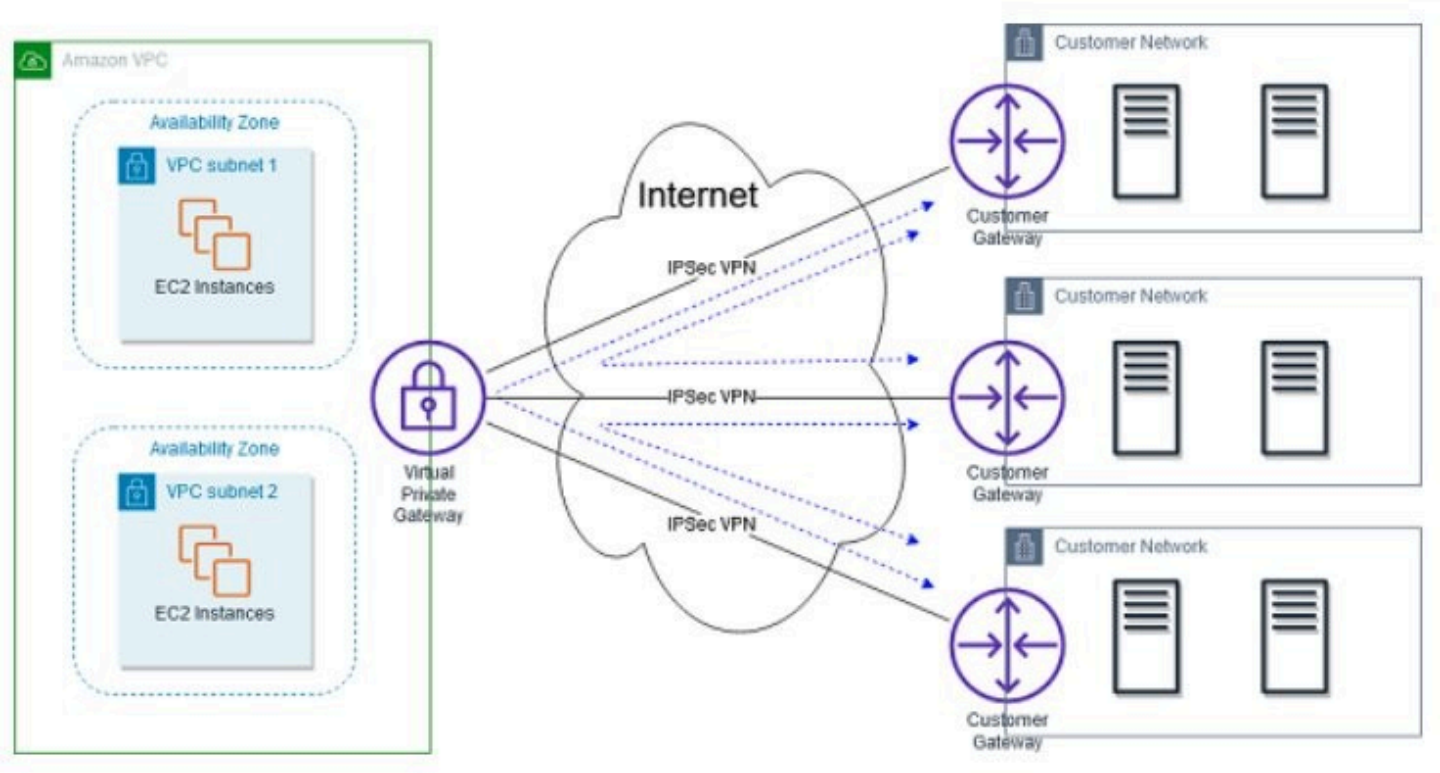


Figure 25 – Multiple VPN – CloudHub model

Additionally, IPsec VPN over DX public virtual interface is supported for a low latency encrypted tunnel carrying the traffic between the on-premises network and AWS.

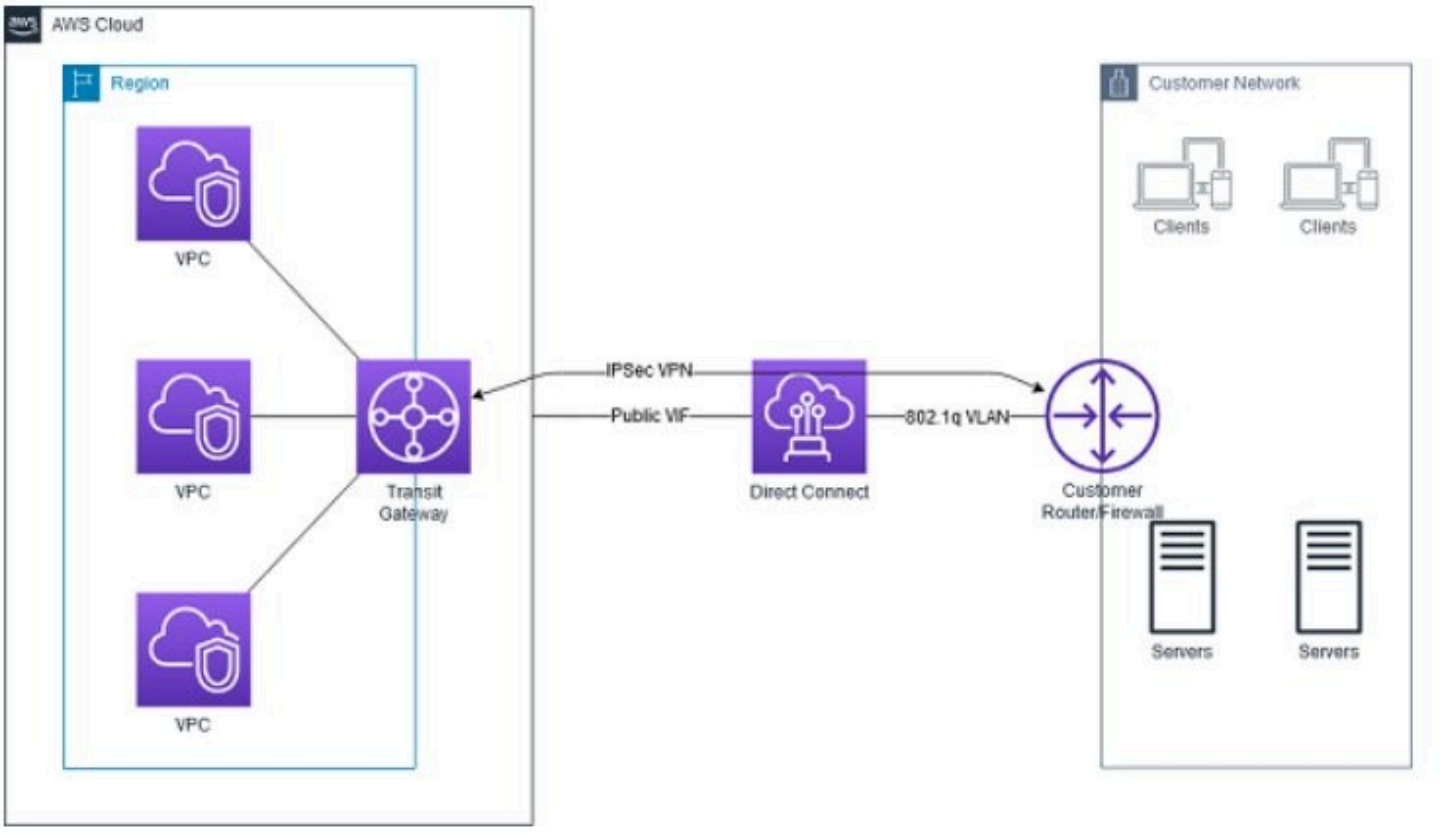


Figure 26 – IPSec VPN over DX

Direct Connect SiteLink

SiteLink is a feature of AWS Direct Connect that makes it possible to send data from one Direct Connect location to another, bypassing AWS Regions. Customers can create global, reliable, and pay-as-you-go connections between the offices and data centers in your global network by sending data over the fastest path between AWS Direct Connect locations.

The first step is to connect the on-premises networks to AWS at any of more than 100 Direct Connect locations worldwide. Next, create Virtual Interfaces (VIFs) on those connections and enable SiteLink. Once all VIFs are attached to the same Direct Connect gateway (DXGW), which is a global and highly available AWS resource, traffic can be sent between them. The data sent, follows the shortest path between AWS Direct Connect locations to its destination, using the fast, secure, and reliable AWS global network. More information on setting up DX SiteLink, see https://aws.amazon.com/blogs/networking-and-content-delivery/introducing-aws-direct-connect-sitelink/

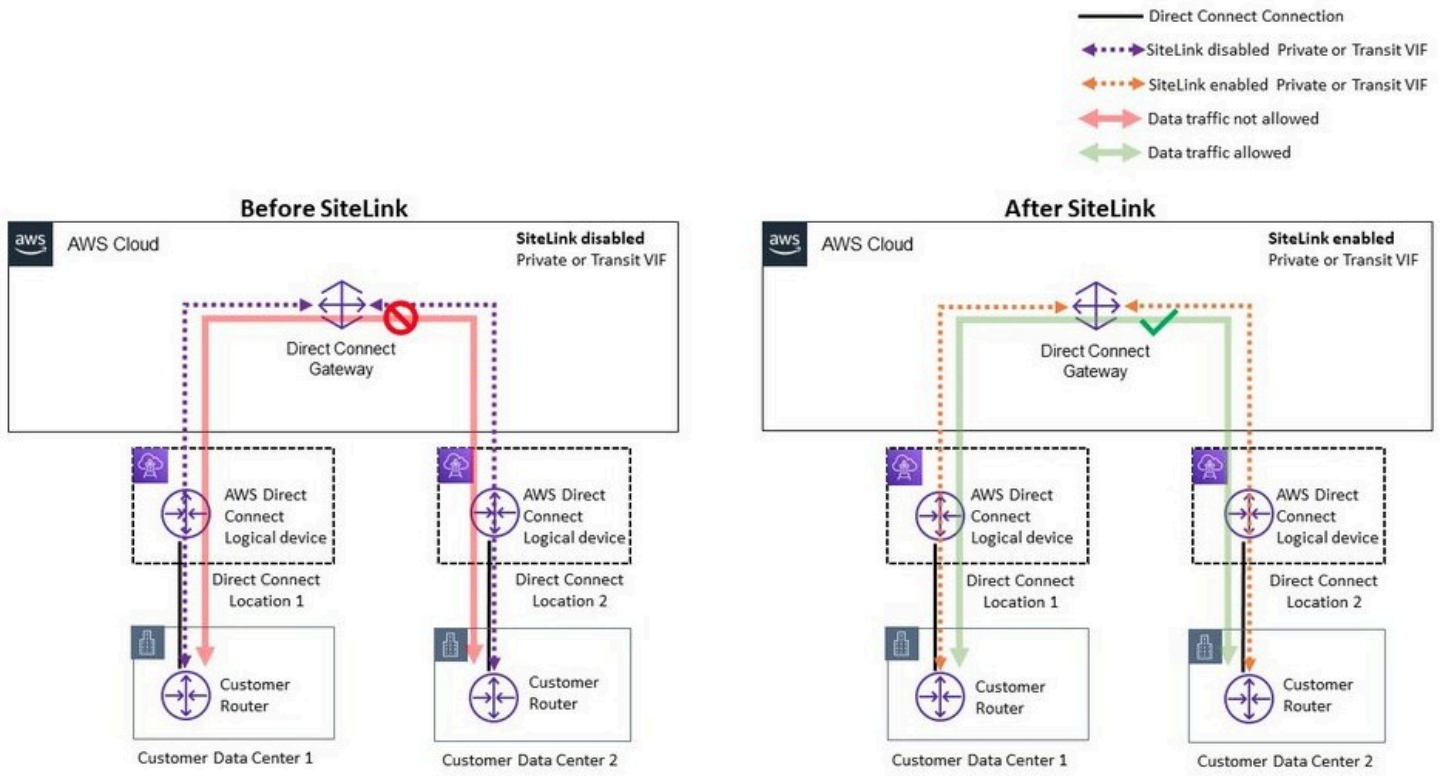


Figure 27 – AWS Direct Connect SiteLink

AWS Network Firewall

AWS Network Firewall (ANF) provides the customer to deep packet inspection (DPI), application protocol detection, domain name filtering, and intrusion prevention system (IPS). ANF provides both stateless and stateful rule engines for traffic at the customer's VPC level with north-south and east-west traffic inspection supporting tens of thousands of rules. Customers can point the incoming traffic to the ANF endpoint with ANF located at a dedicated subnet within the VPC. ANF is powered by AWS Gateway load balancer and uses VPC inbound routing for traffic inspection. The deployment models supported are:

- **Distributed AWS Network Firewall deployment model:** AWS Network Firewall is deployed into each individual VPC.
- **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.
- **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.

The architecture is a combined AWS network firewall deployment model supporting traffic inspection for VPCs with Internet Gateway (IGW) with dedicated ANF endpoint and traffic inspection with a centralized inspection VPC for East - West traffic and an egress VPC for outbound traffic to the internet. The spoke VPC B has an ANF endpoint inspecting incoming traffic and a dedicated Inspection VPC for inspecting VPC to VPC traffic. The ANF can have set of groups with policies to inspect the traffic matching the source and destination prefixes.

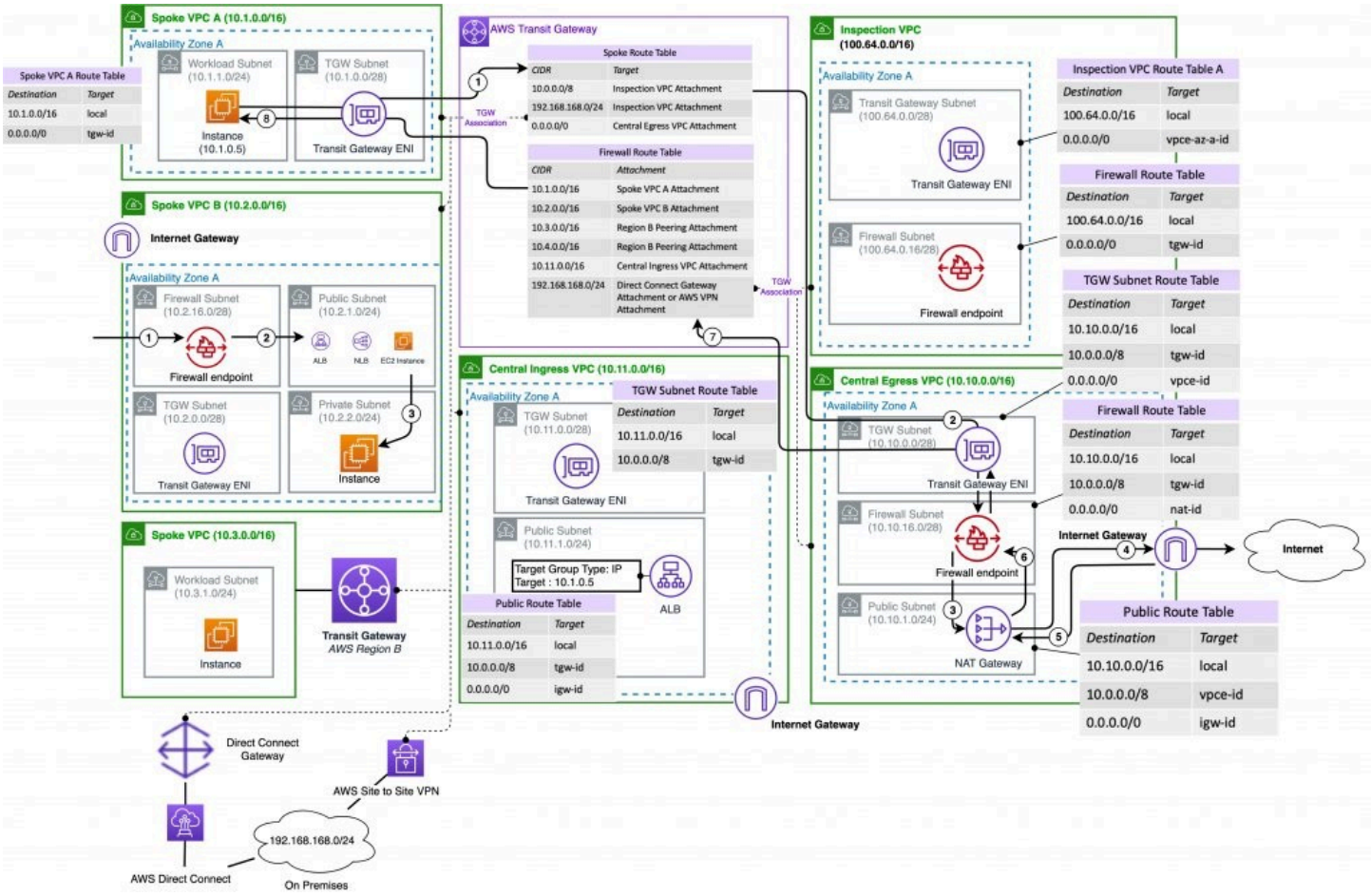


Figure 28 – AWS Network Firewall – Deployment Models

For more information, see Deployment models for AWS Network Firewall.

Edge services

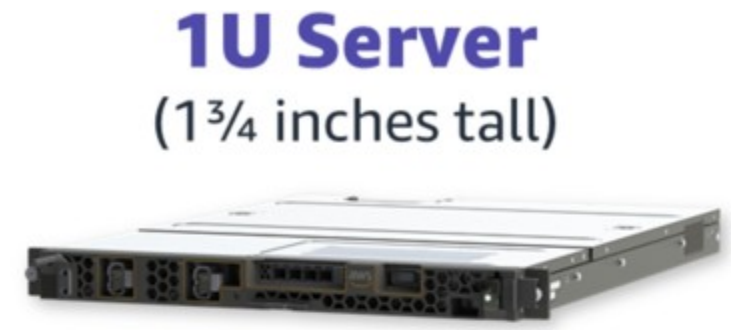
Topics

AWS Outposts

AWS Outposts is a family of fully managed solutions delivering AWS infrastructure and AWS services to virtually any on-premises or edge location for a consistent hybrid experience. Outposts allow you to extend and run AWS services on premises, and are available in a number of form factors, including multiple rack deployments:

- 1U Outposts server
- 2U Outposts server
- 42U Outposts rack

AWS Outposts extends the Amazon VPC in the on-premises network with the ability to connect to a broader range of AWS services in a nearby AWS Region (referred as the AWS home Region). Customers can use AWS APIs, tools, and security controls to run, manage, and secure applications. AWS maintains the hardware infrastructure and software patching as of regional operations.


1U Server
(1¾ inches tall)


2U Server
(3½ inches tall)

Figure 29 – AWS Outposts form factors (1U, 2U, and 42U)

Outposts resources can be shared with other AWS accounts or organizational units (OUs) within the same AWS organization. The following services are available on Outposts:

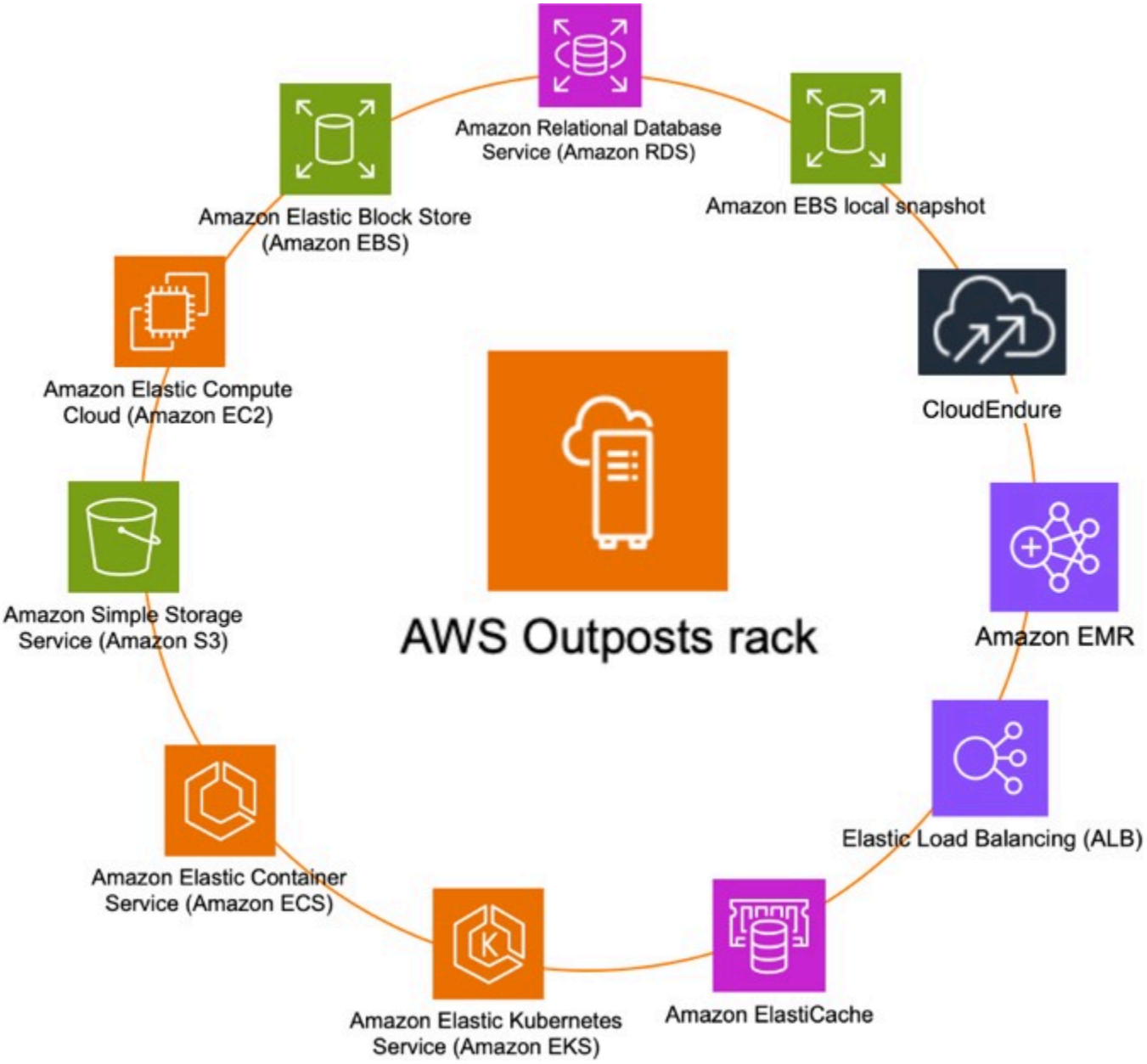


Figure 30 – AWS Outposts – Some of the supported services

Outposts 42U rack to AWS home Region connectivity

Network connectivity of Outposts 42U rack with the AWS Region can be achieved either via private WAN access (using AWS Direct Connect – Private Virtual Interface) or public WAN access (public internet connectivity via local internet service provider or AWS Direct Connect – Public Virtual Interface)

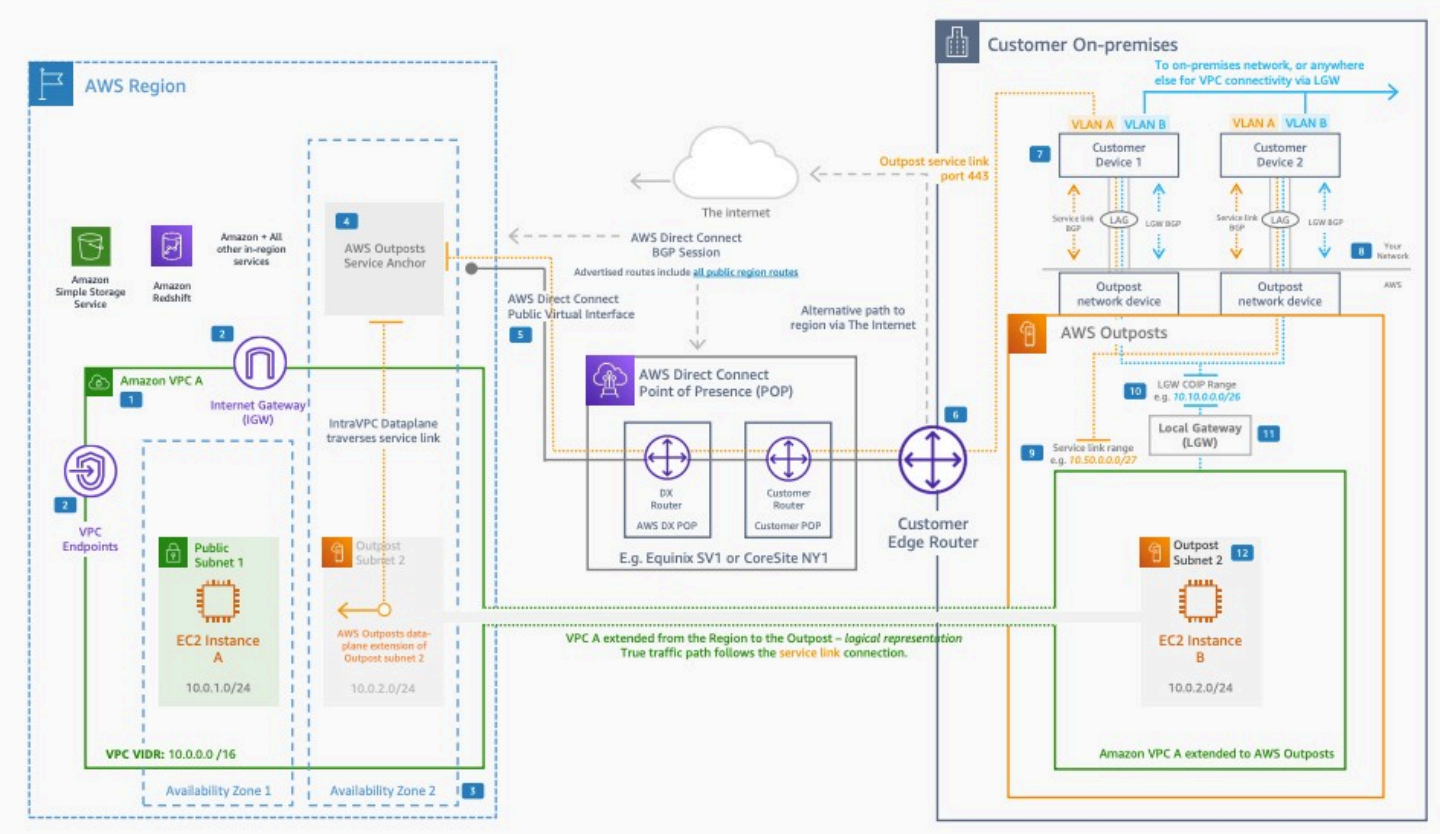


Figure 31 – AWS Outposts – Networking Architecture

Outposts 42U rack to on-premises network connectivity

Outposts 42U racks come with top-of-rack (TOR) switches, supporting link aggregation control protocol (LACP) to bundle multiple links from these TOR switches to the customer's on-premises devices. Outposts TOR switches comes preconfigured with BGP configurations and links are established when the customer configures BGP in the local devices.

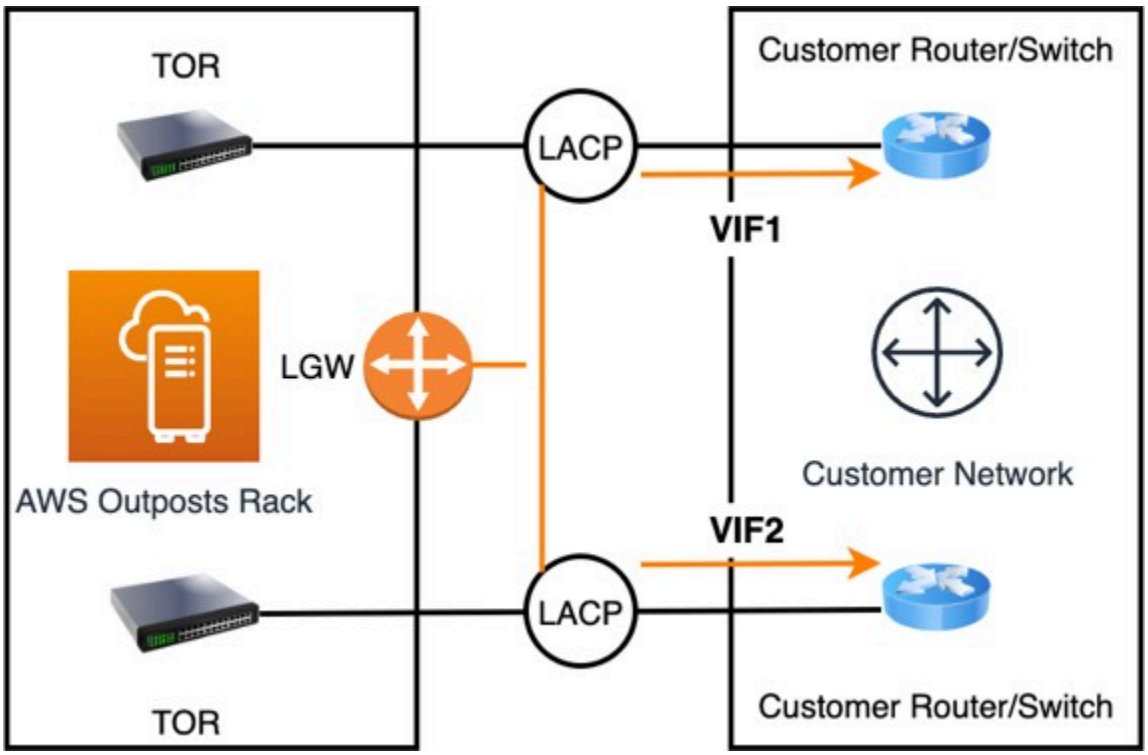


Figure 32 – AWS Outposts – Local Gateway connectivity

Outposts servers (1U and 2U) are rack mountable servers ideal for edge locations with limited space or low-capacity requirements. Outposts servers are ideal for customers with low-latency or local data processing needs for on-premises locations like retail stores, branch offices, healthcare provider locations, or factory floors.

AWS delivers Outposts servers directly to the customer, and the customer's technical team or a preferred third-party vendor can install them. After the Outposts servers are connected to the on-premises network, AWS will remotely provision compute and storage resources so that you can start launching applications that require low-latency—bringing local data processing closer to end users and on-premises systems.

AWS Local Zones

AWS infrastructure deployment that places AWS compute, storage, database, and other select services closer to large population, industry, and IT centers where no AWS Region exists today. Local Zones are data centers and co-location buildings that AWS manages.

Local Zones support customers to run location-sensitive portions of applications (for low latency, or data residency needs) close to end-users in a specific geographical location.

Customers can access Amazon EC2, Amazon EBS, Amazon VPC, Amazon FSx, Elastic Load Balancing, and AWS Direct Connect with Local Zones. Local Zones deliver single-digit millisecond latency.

AWS Local Zones are connected to the parent AWS Region via Amazon's private network that provide fast, secure, and seamless access to other AWS services. This works for customers who do not intend to install AWS Outposts within their on-premises networks.

AWS Wavelength

AWS Wavelength enables customers to use AWS services at the edge of the 5G network. End-users can reach application servers running in Wavelength Zone, which are AWS infrastructures within the telecommunications service provider's data centers at the edge of the 5G networks. The end-user requests are served from the Wavelength Zone with minimal the latency compared to requests leaving the service provider's network to reach the application at the customer's on-premises network or AWS regions. Wavelength delivers a consistent developer experience across multiple 5G networks around the world, allowing you to build the next generation of ultra-low latency applications using familiar AWS services, APIs, and tools. Customers can extend the Amazon Virtual Private Cloud (Amazon VPC) to one or more Wavelength Zones.

Customers can connect using a dedicated high bandwidth connection or via Transit or Peering point.

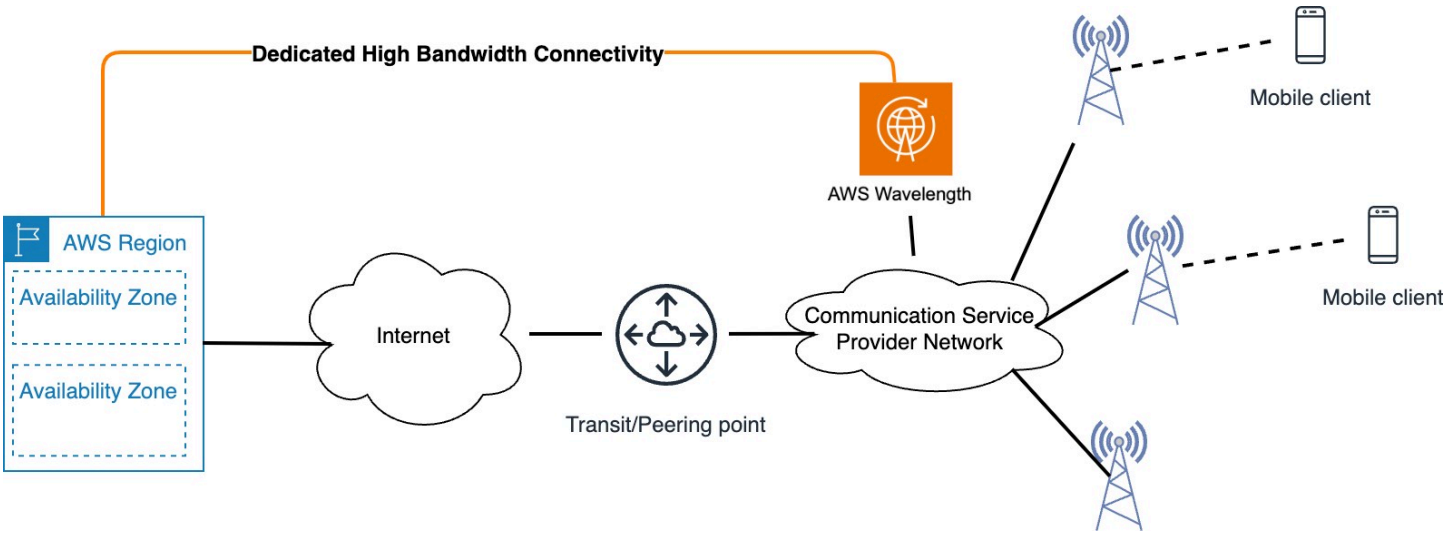


Figure 33 – AWS Wavelength – network connectivity

The following architecture simplifies the use case of Outposts 42U rack, Outposts 1U and 2U servers, AWS Wavelength, AWS Local Zones, and AWS Snowball Edge with supported network connectivity.

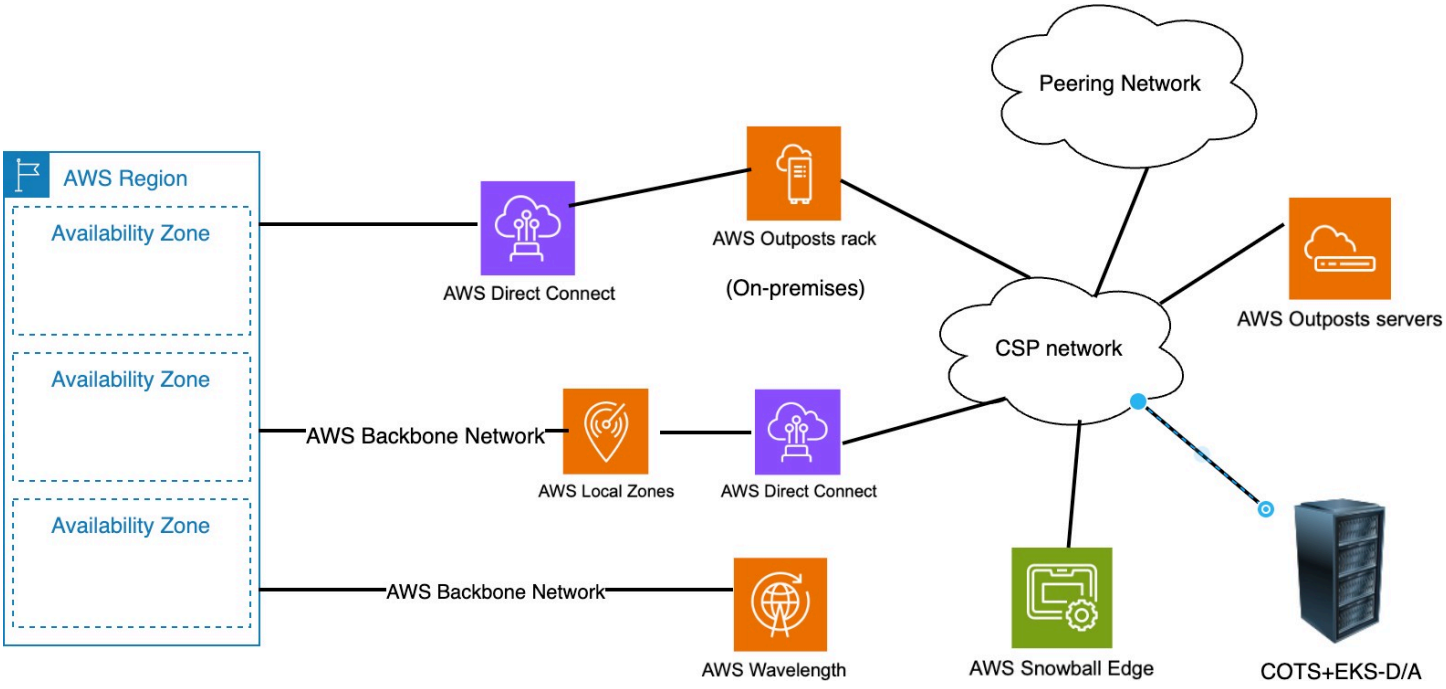


Figure 34 – AWS Edge services overview

AWS Snowball

AWS Snowball uses physical storage devices to transfer large amounts of data between Amazon Simple Storage Service (Amazon S3) and your onsite data storage location at faster-than-internet speeds. AWS Snowball Edge is a type of Snowball device with on-board storage and compute power for select AWS capabilities.

Snowball Edge can do local processing and edge-computing workloads in addition to transferring data between your local environment and the AWS Cloud. It comes pre-configured and does not have to be connected to the internet, so processing and data collection can take place within isolated operating environments. It can operate in remote locations or harsh operating environments, such as factory floors, oil and gas rigs, mining sites, hospitals, and on moving vehicles. Snowball Edge allows you to run the same software at the edge and access select AWS capabilities as you would with full connectivity to AWS.

Snowball Edge is an effective mechanism for extending cloud workloads to on-premises edge in lieu of more heavyweight options (such as AWS Outposts). The following diagram shows how you

can make use of Snowball Edge to run OpenRAN Centralized Units (CU) or Distributed Units (DU) at Cell/Radio Sites and how this may extend to a broader deployment on AWS.

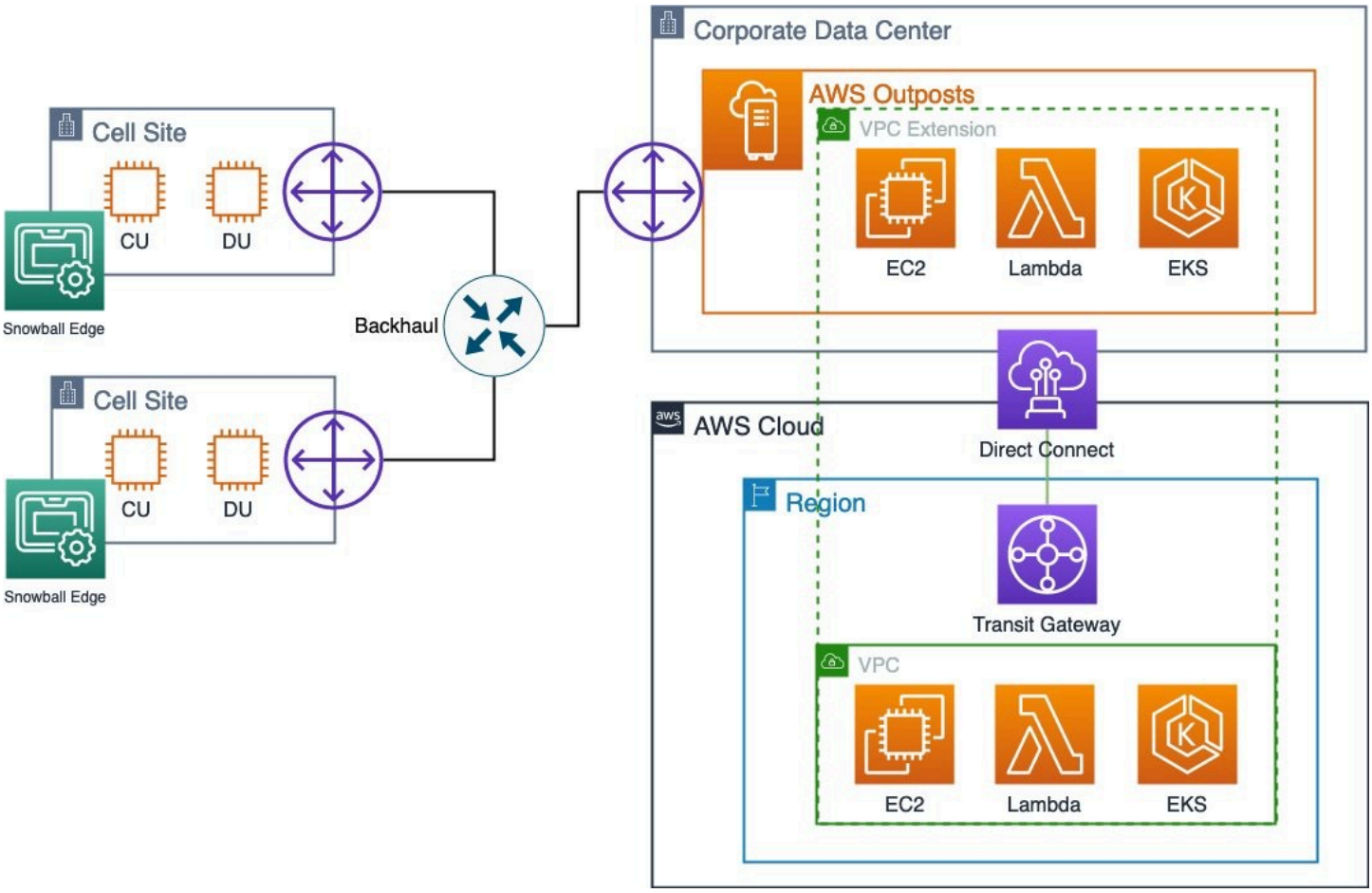


Figure 35 – AWS Snowball Edge

VPC design example with telecom OSS workload

This section provides an example of an OSS workload running in the AWS Cloud and communicating with a telecom's network via DX link. At a high level, the application is gathering performance data from a variety of network elements and this data is being correlated and presented through an OSS application running in Amazon VPC. In this example, the application is provided as a SaaS offering, and is managed by the SaaS provider in a dedicated VPC. The VPC is connected with both the telecom network and the operations network.

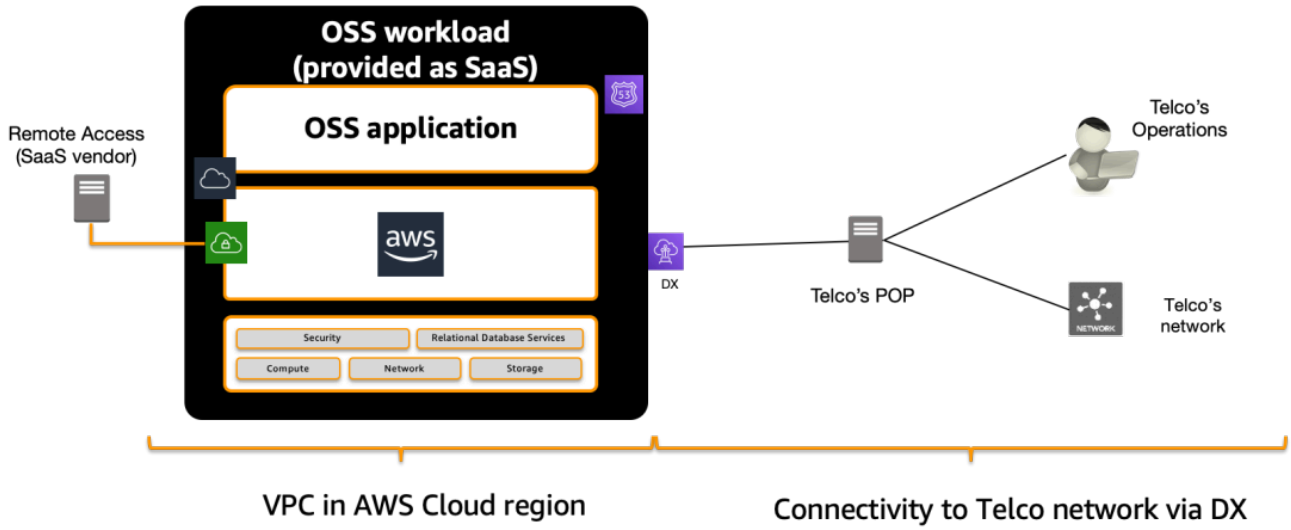


Figure 36 – OSS workload in Amazon VPC

Workloads can run as virtual machines (VMs) or containers. In this example, the OSS application is implemented as container workloads on Red Hat OpenShift, using a Multi-AZ deployment for high availability purposes. Amazon EBS, Amazon Elastic File System (Amazon EFS) and Amazon Relational Database Service (Amazon RDS) are used in the overall design.

The advantage of this cloud-based implementation for telecom providers is:

- Elastic scaling of the entire application using Elastic Load Balancing and automatic scaling.
- Secure data handling as incoming data into the VPC is encrypted, data leaving the VPC is encrypted, and data held within the VPC is encrypted at both the storage and database level.
- Secure access through ACLs, security groups, and multi-factor authentication (MFA).

- High availability implementation spanning three Availability Zones, with private and public subnet in each AZ. Internet gateway provides internet access to each subnet.
- AWS CloudFormation is used to deploy the entire infrastructure without the need for manual installation and stand-up.

The following diagram provides a logical representation of the key building blocks in the VPC and their connectivity to the telecom network through a DX connection:

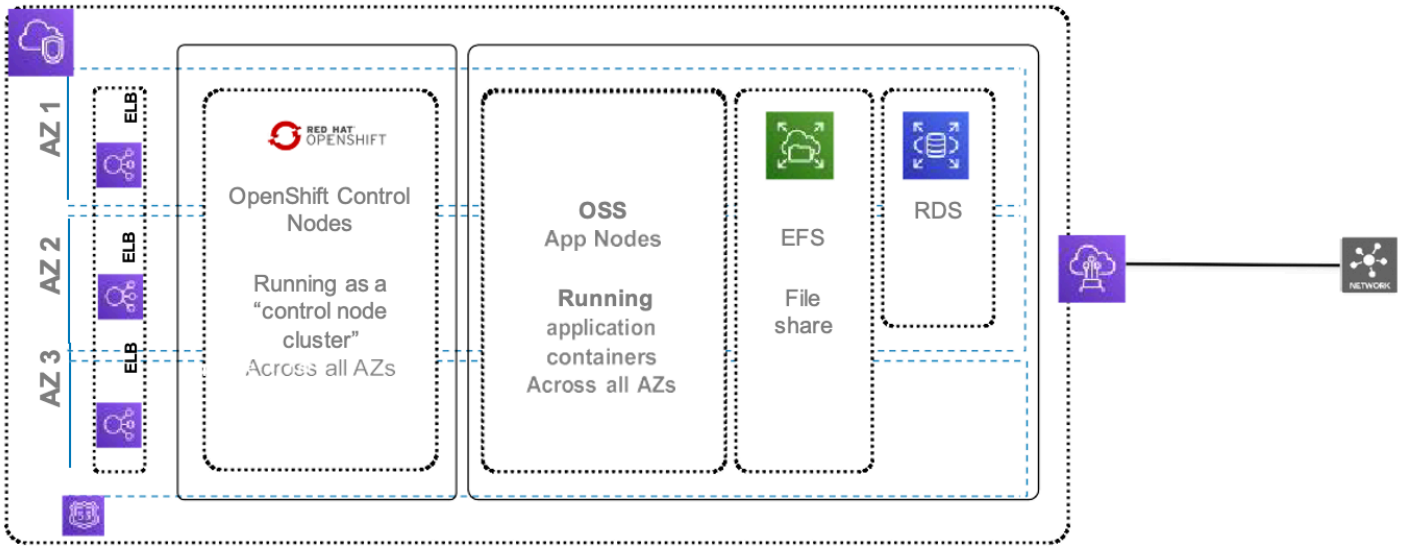


Figure 37 – Logical representation of the OSS workload architecture in the cloud

Conclusion

The AWS Cloud offers a wide variety of elastic compute choices and provides a strong, cloud native alternative to traditional NFVI platforms, such as OpenStack. With 100 Gbps capable, compute-optimized instances, and the support of performance- enhancing tools, such as DPDK, CPU affinity, NUMA, and Huge pages, AWS provides a suitable environment for running mission critical telecom workloads. AWS networking services give telecom providers the ability to extend their on-premises networking to the cloud in a secure and reliable manner by using AWS Direct Connect, Amazon VPC, VPNs, AWS Transit Gateway, and Elastic Load Balancing.

Contributors

Contributors to this version of the document include:

- Ishtiaq Islam, Senior Cloud Support Engineer, AWS Support
- Moiz Alam, Senior Solutions Architect, Strategic Accounts
- Amanveer Singh, Senior Solution Architect, AWS Telecom Business
- George Oaks, Senior Solutions Architect, WWSO Networking

Contributors to earlier versions:

- Rada Stanic, Principal Solutions Architect, APAC
- Dr. Young Jung, Senior Partner Solutions Architect, AWS Telecom Business Unit
- Tipu Qureshi, Principal Cloud Support Engineer, AWS Support

Additional resources

For additional information, see:

- Exploring NUMA on Amazon Cloud Instances
- Enabling Enhanced Networking with the Elastic Network Adapter (ENA) on Windows Instances
- Amazon VPC for On-Premises Network Engineers – Part 1
- Amazon Virtual Private Cloud Connectivity Options whitepaper
- NFV reference architecture for deployment of mobile networks (PDF)

Document history

To be notified about updates to this whitepaper, subscribe to the RSS feed.

Change	Description	Date
Whitepaper updated	Guidance updated with latest best practices and editorial changes throughout.	October 4, 2023
Whitepaper updated	Guidance updated with latest best practices.	October 1, 2022
Initial publication	Whitepaper published.	September 9, 2019

AWS Glossary

For the latest AWS terminology, see the AWS glossary in the *AWS Glossary Reference*.