IPv4 - CIDR Cheat Sheet

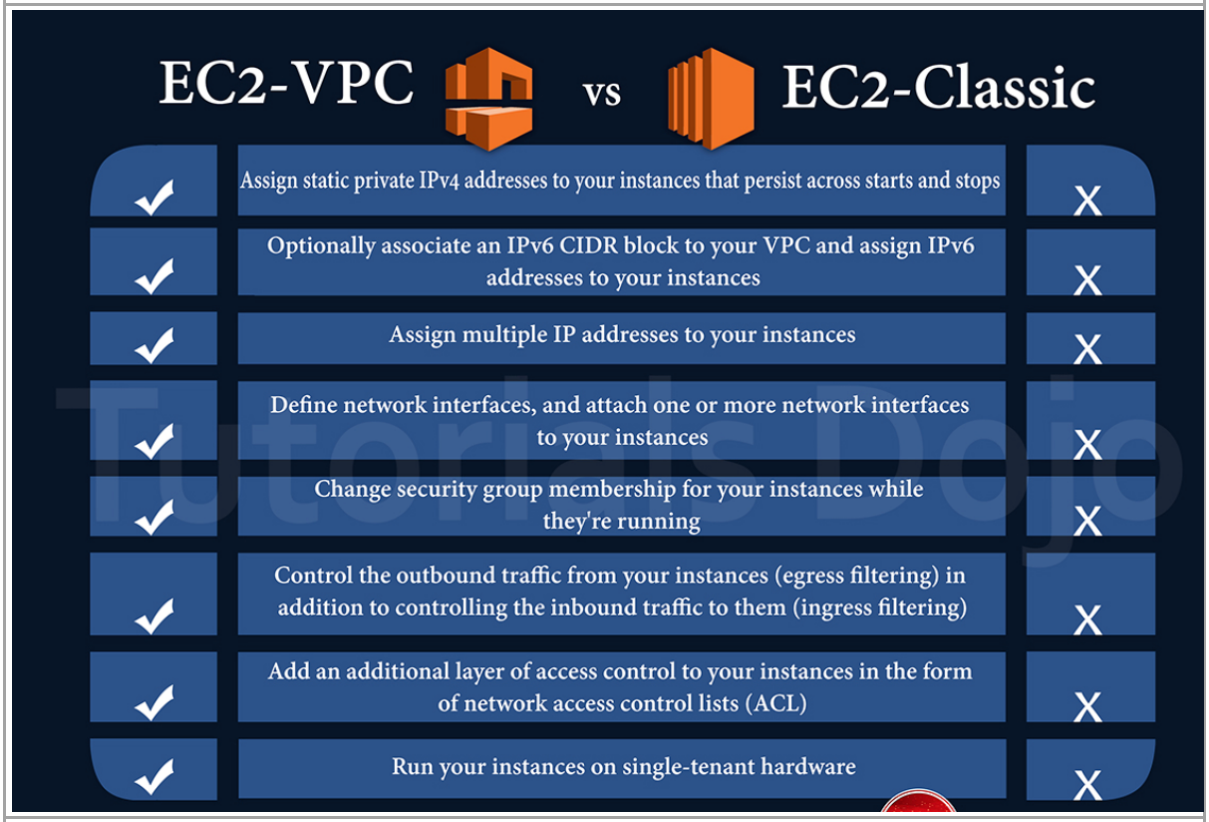
IPv4 Subnet Mask Cheat Sheet

	Addresses	Hosts	Netmask	Amount of a Class C
/30	4	2	255.255.255.252	1/64
/29	8	6	255.255.255.248	1/32
/28	16	14	255.255.255.240	1/16
/27	32	30	255.255.255.224	1/8
/26	64	62	255.255.255.192	1/4
/25	128	126	255.255.255.128	1/2
/24	256	254	255.255.255.0	1
/23	512	510	255.255.254.0	2
/22	1024	1022	255.255.252.0	4
/21	2048	2046	255.255.248.0	8
/20	4096	4094	255.255.240.0	16
/19	8192	8190	255.255.224.0	32
/18	16384	16382	255.255.192.0	64
/17	32768	32766	255.255.128.0	128
/16	65536	65534	255.255.0.0	256

EC2 - Private IPs

- In EC2-Classic, your EC2 instance receives a private IPv4 address from the EC2-Classic range each time it's started.
- In EC2-VPC on the other hand, your EC2 instance receives a static private IPv4 address from the address range of your default VPC.
- Hence, the correct answer is **launching the instances in the Amazon Virtual Private Cloud (VPC)** and not **launching the instances in EC2-Classic**.

Characteristic	EC2-Classic	Default VPC	Nondefault VPC
Public IPv4 address (from Amazon's public IP address pool)	Your instance receives a public IPv4 address from the EC2-Classic public IPv4 address pool.	Your instance launched in a default subnet receives a public IPv4 address by default, unless you specify otherwise during launch, or you modify the subnet's public IPv4 address attribute.	Your instance doesn't receive a public IPv4 address by default, unless you specify otherwise during launch, or you modify the subnet's public IPv4 address attribute.
Private IPv4 address	Your instance receives a private IPv4 address from the EC2-Classic range each time it's started.	Your instance receives a static private IPv4 address from the address range of your default VPC.	Your instance receives a static private IPv4 address from the address range of your VPC.
Multiple private IPv4 addresses	We select a single private IP address for your instance; multiple IP addresses are not supported.	You can assign multiple private IPv4 addresses to your instance.	You can assign multiple private IPv4 addresses to your instance.
Elastic IP address (IPv4)	An Elastic IP is disassociated from your instance when you stop it.	An Elastic IP remains associated with your instance when you stop it.	An Elastic IP remains associated with your instance when you stop it.



References:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-classic-platform.html#differences-ec2-classic-vpc

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-instance-addressing.html

Check out this Amazon EC2 Cheat Sheet:

https://tutorialsdojo.com/aws-cheat-sheet-amazon-elastic-compute-cloud-amazon-ec2/

VPC - Elastic IPs

Elastic Ips

An *Elastic IP address* is a static IPv4 address designed for dynamic cloud computing. An Elastic IP address is associated with your AWS account. With an Elastic IP address, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account.

An Elastic IP address is a public IPv4 address, which is reachable from the internet.
If your instance does not have a public IPv4 address, you can associate an Elastic IP address with your instance to enable communication with the internet.
For example, this allows you to connect to your instance from your local computer.

We currently do not support Elastic IP addresses for IPv6.

Elastic IP address basics

- To use an Elastic IP address, you first allocate one to your account, and then associate it with your instance or a network interface.
- When you associate an Elastic IP address with an instance, it is also associated with the instance's primary network interface. When you associate an Elastic IP address with a network interface that is attached to an instance, it is also associated with the instance.
- When you associate an Elastic IP address with an instance or its primary network interface, the instance's public IPv4 address (if it had one) is released back into Amazon's pool of public IPv4 addresses. You cannot reuse a public IPv4 address, and you cannot convert a public IPv4 address to an Elastic IP address. For more information, see Public IPv4 addresses and external DNS hostnames.
- Elastic IP Reuse: You can disassociate an Elastic IP address from a resource, and then associate it with a different resource. To avoid unexpected behavior, ensure that all active connections to the resource named in the existing association are closed before you make the change. After you have associated your Elastic IP address to a different resource, you can reopen your connections to the newly associated resource.
- A disassociated Elastic IP address remains allocated to your account until you explicitly release it.

- To ensure efficient use of Elastic IP addresses, we impose a small hourly charge if an Elastic IP address is not associated with a running instance, or if it is associated with a stopped instance or an unattached network interface. While your instance is running, you are not charged for one Elastic IP address associated with the instance, but you are charged for any additional Elastic IP addresses associated with the instance. For more information, see the section for Elastic IP Addresses on the Amazon EC2 Pricing, On-Demand Pricing page.
- An Elastic IP address is for use in a specific network border group only.
- When you associate an Elastic IP address with an instance that previously had a public IPv4 address, the public DNS host name of the instance changes to match the Elastic IP address.
- We resolve a public DNS host name to the public IPv4 address or the Elastic IP address of the instance outside the network of the instance, and to the private IPv4 address of the instance from within the network of the instance.
- When you allocate an Elastic IP address from an IP address pool that you have brought to your AWS account, it does not count toward your Elastic IP address limits. For more information, see Elastic IP address limit.
- When you allocate the Elastic IP addresses, you can associate the Elastic IP addresses with a network border group. This is the location from which we advertise the CIDR block. Setting the network border group limits the CIDR block to this group. If you do not specify the network border group, we set the border group containing all of the Availability Zones in the Region (for example, us-west-2).

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html

VPC - DNS Support

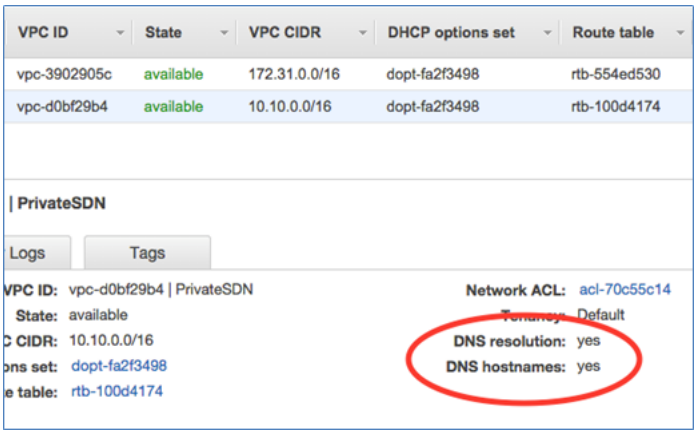
A *private hosted zone* is a container that holds information about how you want Amazon Route 53 to respond to DNS queries for a domain and its subdomains within one or more VPCs that you create with the Amazon VPC service. Your VPC has attributes that determine whether your EC2 instance receives public DNS hostnames, and whether DNS resolution through the Amazon DNS server is supported.

enableDnsHostnames - Indicates whether the instances launched in the VPC get public DNS hostnames. If this attribute is true, instances in the VPC get public DNS hostnames, but only if the enableDnsSupport attribute is also set to true.

enableDnsSupport - Indicates whether the DNS resolution is supported for the VPC. If this attribute is false, the Amazon-provided DNS server in the VPC that resolves public DNS hostnames to IP addresses is not enabled. If this attribute is true, queries to the Amazon provided DNS server at the 169.254.169.253 IP address, or the reserved IP address at the base of the VPC IPv4 network range plus two (*.*.*.2) will succeed.

Since you want your new EC2 instances to automatically have a public DNS hostname and you want to configure your VPC to resolve Amazon-provided private DNS hostnames as

well, the correct answer is ***modifying the enableDnsHostNames attribute of your VPC to true and the enableDnsSupport attribute to true.***



If both attributes are set to true, the following occurs:

- Your EC2 instance receives a public DNS hostname.
- The Amazon-provided DNS server can resolve Amazon-provided private DNS hostnames.

If either or both of the attributes is set to false, the following occurs:

- Your instance does not receive a public DNS hostname that can be viewed in the Amazon EC2 console or described by a command line tool or AWS SDK.
- The Amazon-provided DNS server cannot resolve Amazon-provided private DNS hostnames.
- Your instance receives a custom private DNS hostname if you've specified a custom domain name in your DHCP options set. If you are not using the Amazon-provided DNS server, your custom domain name servers must resolve the hostname as appropriate.

IMPORTANT NOTE

By default, both attributes are set to true in a default VPC or a VPC created by the VPC wizard. By default, only the enableDnsSupport attribute is set to true in a VPC created on the **Your VPCs** page of the VPC console or using the AWS CLI, API, or an AWS SDK.

References:

https://docs.aws.amazon.com/vpc/latest/userguide/vpc-dns.html#vpc-dns-support

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/hosted-zones-private.html

AWS - Direct Connect

To connect to AWS public endpoints such as an Amazon Elastic Compute Cloud (Amazon EC2) or Amazon Simple Storage Service (Amazon S3) with dedicated network performance, use a public virtual interface.

AWS Direct Connect provides two types of virtual interfaces: public and private.

Read the below paragraphs to help you choose which one to use to connect to different resources (public or private) in AWS.

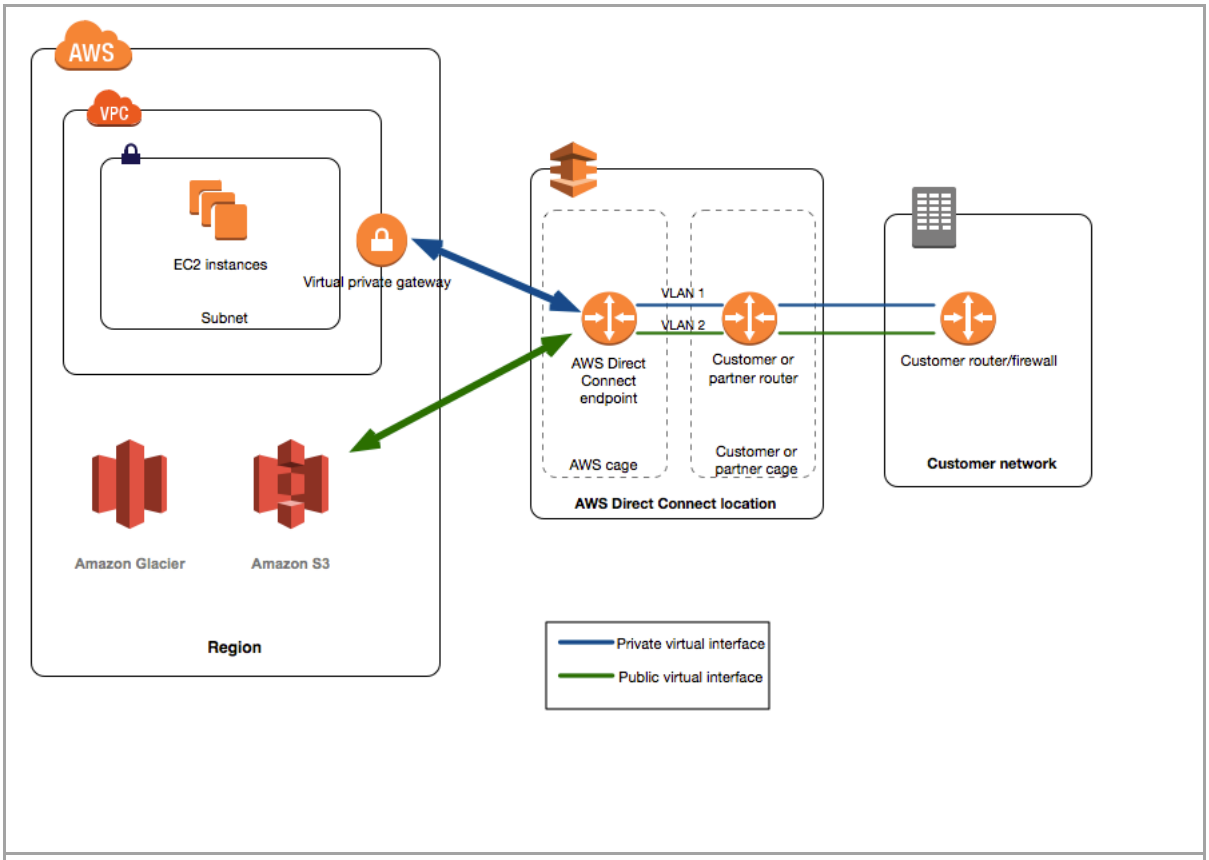
Public Virtual Interface

- To connect to AWS public endpoints, such as an Amazon Elastic Compute Cloud (Amazon EC2) or Amazon Simple Storage Service (Amazon S3), with dedicated network performance, use a public virtual interface.
- A public virtual interface allows you to connect to all AWS public IP spaces globally.
- Direct Connect customers in any Direct Connect location can create public virtual interfaces to receive Amazon's global IP routes, and they can access publicly routable Amazon services in any AWS Regions (except the AWS China Region).

Private Virtual Interface

- To connect to private services, such as an Amazon Virtual Private Cloud (Amazon VPC), with dedicated network performance, use a private virtual interface.
- A private virtual interface allows you to connect to your VPC resources (for example, EC2 instances, load balancers, RDS DB instances, etc.) on your private IP address or endpoint.
- A private virtual interface can connect to a Direct Connect gateway, which can be associated with one or more virtual private gateways in any AWS Regions (except the AWS China Region).
- A virtual private gateway is associated with a single VPC, so you can connect to multiple VPCs in any AWS Regions (except the AWS China Region) using a private virtual interface. For a private virtual interface, AWS only advertises the entire VPC CIDR over the Border Gateway Protocol (BGP) neighbor.

NOTE: ***A network device in your data center that supports Border Gateway Protocol (BGP) and BGP MD5 authentication*** is correct as a network device that supports Border Gateway Protocol (BGP) and BGP MD5 authentication is needed to establish a Direct Connect link from your data center to your VPC



AWS DirectConnect - Failover

With AWS Direct Connect plus VPN, you can combine one or more AWS Direct Connect dedicated network connections with the Amazon VPC VPN. This combination provides an IPsec-encrypted private connection that also reduces network costs, increases bandwidth throughput, and provides a more consistent network experience than internet-based VPN connections.

You can use AWS Direct Connect to establish a dedicated network connection between your network and create a logical connection to public AWS resources, such as an Amazon virtual private gateway IPsec endpoint. This solution combines the AWS managed benefits of the VPN solution with low latency, increased bandwidth, more consistent benefits of the AWS Direct Connect solution, and an end-to-end, secure IPsec connection.

It costs a lot of money to establish a Direct Connect connection which you rarely use. For a more cost-effective solution, you can configure a backup VPN connection for failover with your AWS Direct Connect connection.

If you want a short-term or lower-cost solution, you might consider configuring a hardware VPN as a failover option for a Direct Connect connection. VPN connections are not designed to provide the same level of bandwidth available to most Direct Connect connections. Ensure that your use case or application can tolerate a lower bandwidth if you are configuring a VPN as a backup to a Direct Connect connection.

Hence, the correct answer is the option that says: ***Establish VPN tunnels from your on-premises data center to each of the 10 VPCs. Terminate each VPN tunnel connection at***

the virtual private gateway (VGW) of the respective VPC. Configure BGP for route management.

References:

http://docs.aws.amazon.com/directconnect/latest/UserGuide/WorkingWithVirtualInterfaces.html

https://aws.amazon.com/premiumsupport/knowledge-center/public-private-interface-dx/

https://docs.aws.amazon.com/directconnect/latest/UserGuide/Welcome.html

Check out this AWS Direct Connect Cheat Sheet:

https://tutorialsdojo.com/aws-cheat-sheet-aws-direct-connect/

References:

https://aws.amazon.com/premiumsupport/knowledge-center/configure-vpn-backup-dx/

https://docs.aws.amazon.com/whitepapers/latest/aws-vpc-connectivity-options/aws-direct-connect-plus-vpn-network-to-amazon.html

https://aws.amazon.com/answers/networking/aws-network-connectivity-over-mpls/

Direct Connect Gateway

References:

https://aws.amazon.com/blogs/aws/new-aws-direct-connect-gateway-inter-region-vpc-access/

https://docs.aws.amazon.com/directconnect/latest/UserGuide/direct-connect-gateways.html

https://aws.amazon.com/answers/networking/aws-multiple-region-multi-vpc-connectivity/

New – AWS Direct Connect Gateway – Inter-Region VPC Access

by Jeff Barr | on 01 NOV 2017 | in Amazon VPC, AWS Direct Connect, Launch, News | Permalink | Share

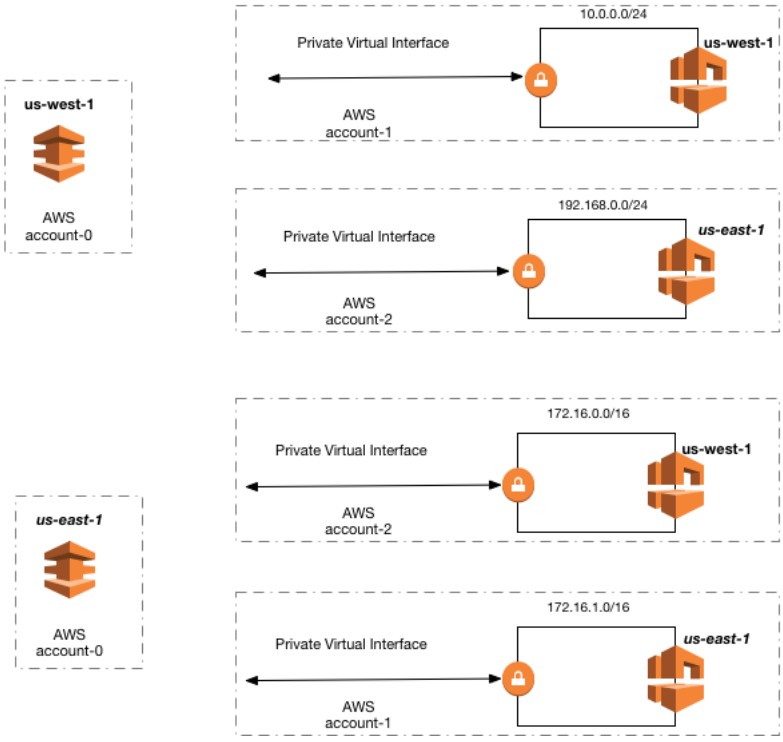
As I was preparing to write this post, I took a nostalgic look at the blog post I wrote when we launched AWS Direct Connect back in 2012. We created Direct Connect after our enterprise customers asked us to allow them to establish dedicated connections to an AWS Region in pursuit of enhanced privacy, additional data transfer bandwidth, and more predictable data transfer performance. Starting from one AWS Region and a single colo, Direct Connect is now available in every public AWS Region and accessible from dozens of colos scattered across the world (over 60 locations at last count). Our customers have taken to Direct Connect wholeheartedly and we have added features such as Link Aggregation, Amazon EFS support, CloudWatch monitoring, and HIPAA eligibility. In the past five weeks alone we have added Direct Connect locations

in Houston (Texas), Vancouver (Canada), Manchester (UK), Canberra (Australia), and Perth (Australia).

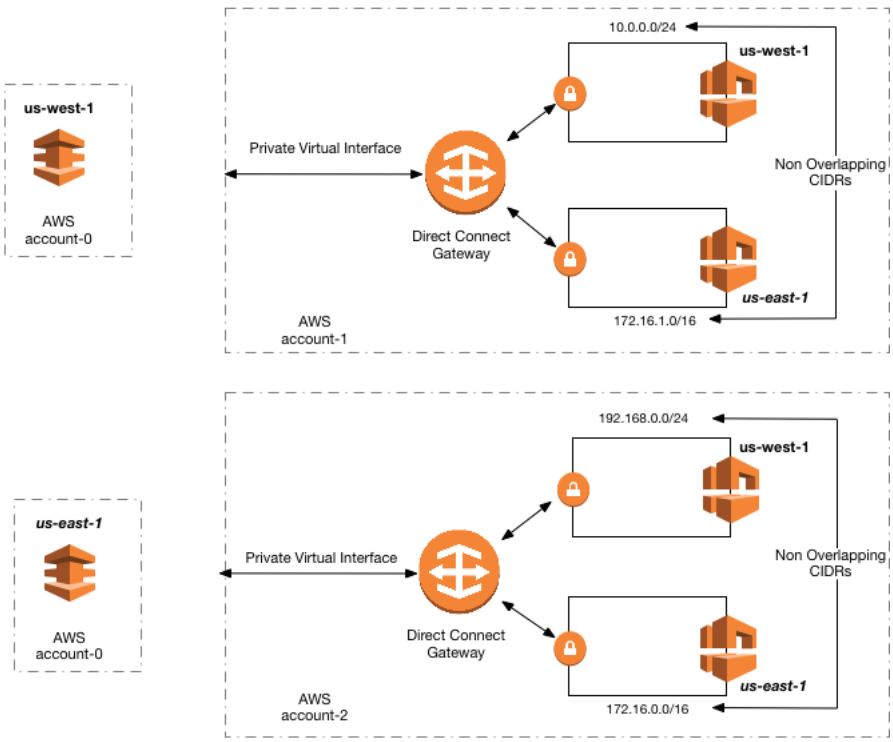
Today we are making Direct Connect simpler and more powerful with the addition of the Direct Connect Gateway. We are also giving Direct Connect customers in any Region the ability to create public virtual interfaces that receive our global IP routes and enable access to the public endpoints for our services and updating the Direct Connect pricing model.
Let's take a look at each one!

New Direct Connect Gateway

You can use the new Direct Connect Gateway to establish connectivity that spans Virtual Private Clouds (VPCs) spread across multiple AWS Regions. You no longer need to establish multiple BGP sessions for each VPC; this reduces your administrative workload as well as the load on your network devices.
This feature also allows you to connect to any of the participating VPCs from any Direct Connect location, further reducing your costs for making using AWS services on a cross-region basis.
Here is a diagram that illustrates the simplification that you can achieve with a Direct Connect Gateway (each "lock" icon represents a Virtual Private Gateway). Start with this:



And end up like this:



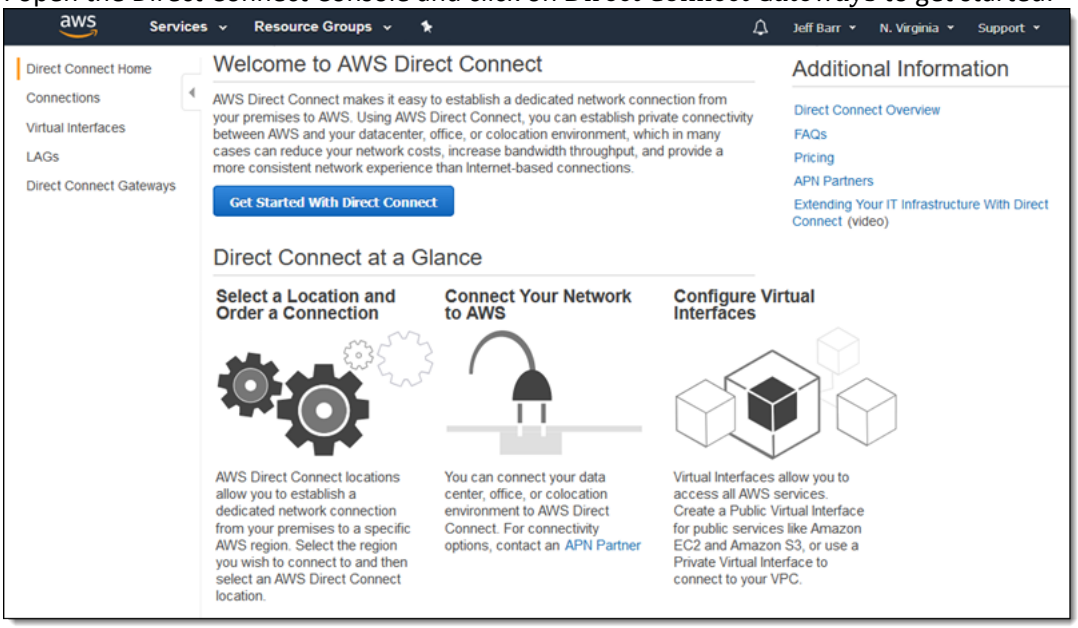
The VPCs that reference a particular Direct Connect Gateway must have IP address ranges that do not overlap. Today, the VPCs must all be in the same AWS account; we plan to make this more flexible in the future.

Each Gateway is a global object that exists across all of the public AWS Regions. All communication between the Regions via the Gateways takes place across the AWS network backbone.

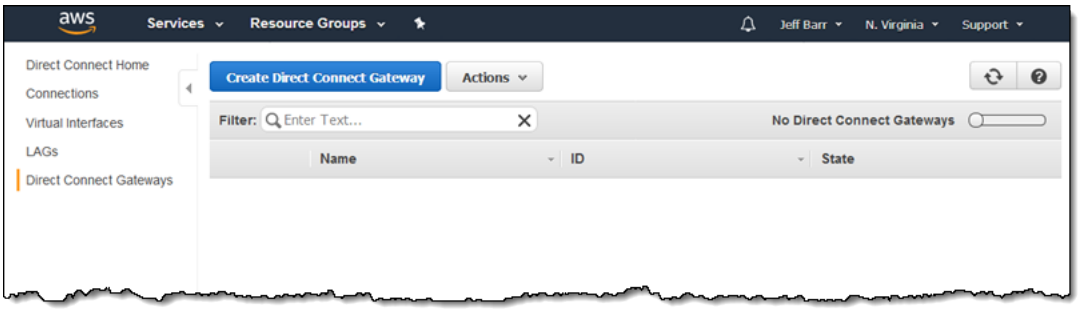
Creating a Direct Connect Gateway

You can create a Direct Connect Gateway from the Direct Connect Console or by calling the `CreateDirectConnectGateway` function from your code. I'll use the Console!

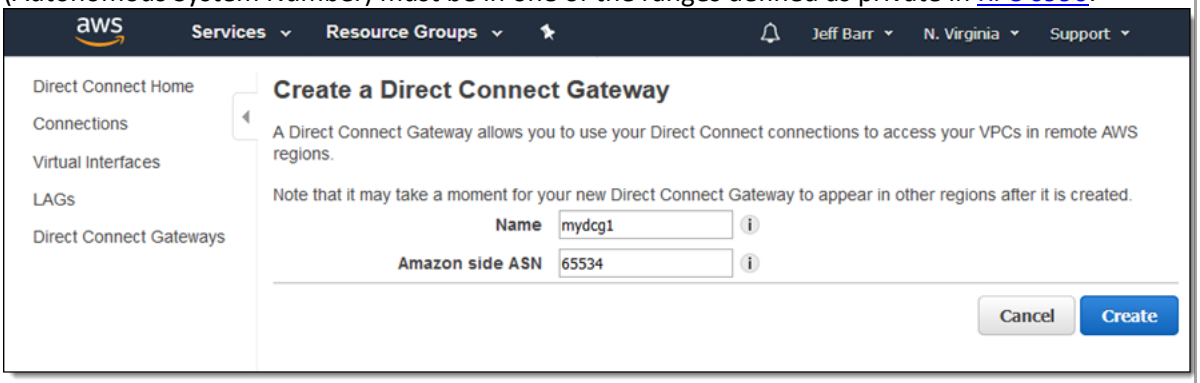
I open the Direct Connect Console and click on Direct Connect Gateways to get started:



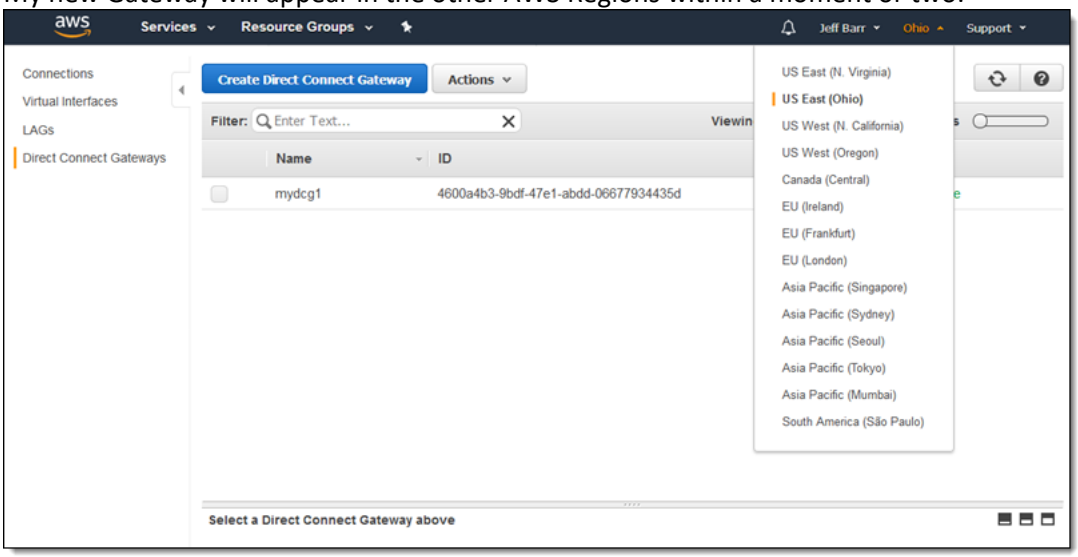
The list is empty since I don't have any Gateways yet. Click on Create Direct Connect Gateway to change that:



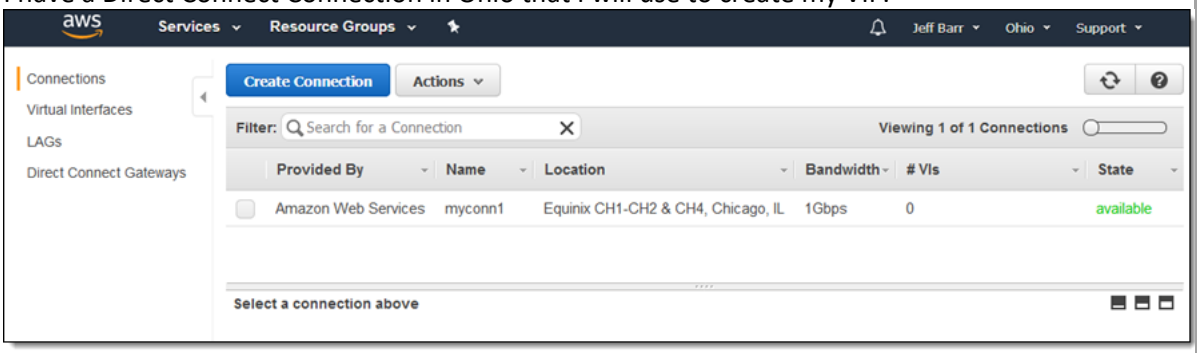
I give my Gateway a name, enter a private ASN for my network, then click on Create. The ASN (Autonomous System Number) must be in one of the ranges defined as private in RFC 6996:



My new Gateway will appear in the other AWS Regions within a moment or two:



I have a Direct Connect Connection in Ohio that I will use to create my VIF:



Now I create a private VIF that references the Gateway and the Connection:

Create a Virtual Interface

You may choose to create a private or public virtual interface. Select the appropriate option below.

- (•) Private - A private virtual interface should be used to access an Amazon VPC using private IP addresses.
- ( ) Public - A public virtual interface can access all AWS public services (including EC2, S3, and DynamoDB) using public IP addresses.

Define Your New Private Virtual Interface

Enter the name of your virtual interface. If youre creating a virtual interface for another account, youll need to provide the other AWS account ID. For more information about virtual interface ownership, see 'Hosted Virtual Interfaces' in the AWS Direct Connect Getting Started Guide.

Connection dxcon-fh0jl5fs (myconn1)

Virtual Interface Name myvif1

Virtual Interface Owner (•) My AWS Account ( ) Another AWS Account

Select the gateway for this virtual interface. You can connect to Virtual Private Gateway (VGW) or Direct Connect Gateway. Connecting with Direct Connect Gateway will enable you to associate with multiple VGWs, providing connectivity with multiple Virtual Private Clouds across multiple regions; connecting with Virtual Private Gateway will allow you to connect with one Virtual Private Cloud in the selected region.

Connection To (•) Direct Connect Gateway ( ) Virtual Private Gateway

Direct Connect Gateway mydcg1

Create Direct Connect Gateway

Enter the VLAN ID, if not already supplied by your AWS Direct Connect partner, and the IP Addresses for your router interface and the AWS Direct Connect interface.

VLAN 110

Address family (•) IPv4 ( ) IPv6

Auto-generate peer IPs [x]

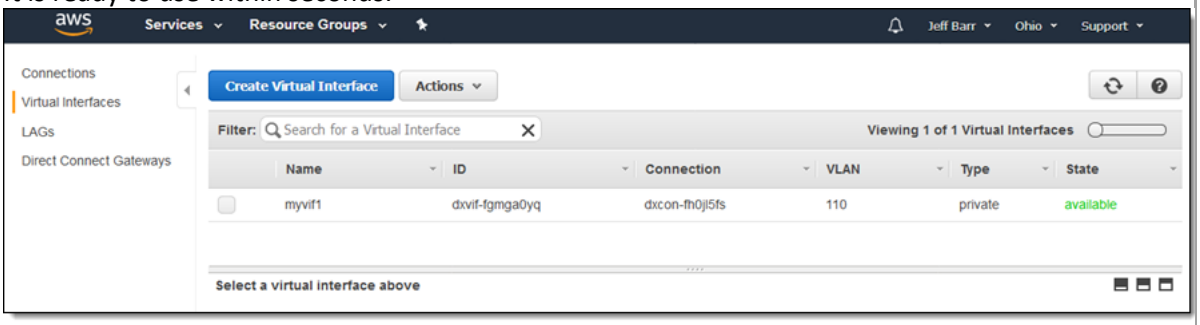
Before you can use your virtual interface, we must establish a BGP session. You must provide an ASN for your router. You will also need an MD5 key to authenticate the BGP session. We can generate one for you, or you can supply your own.

BGP ASN 65000

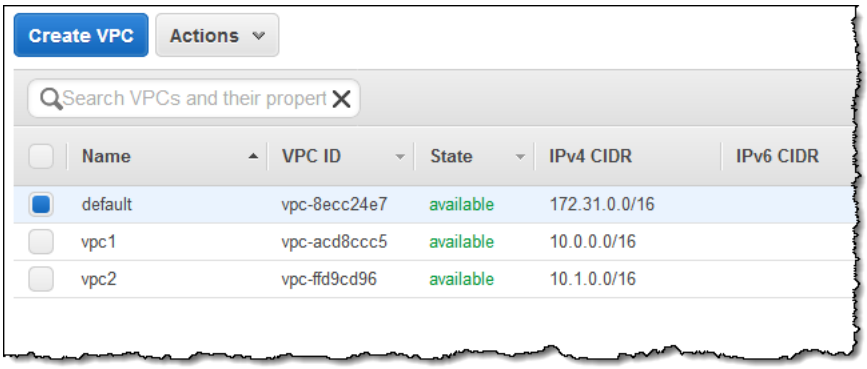
Auto-generate BGP key [x]

Cancel Continue

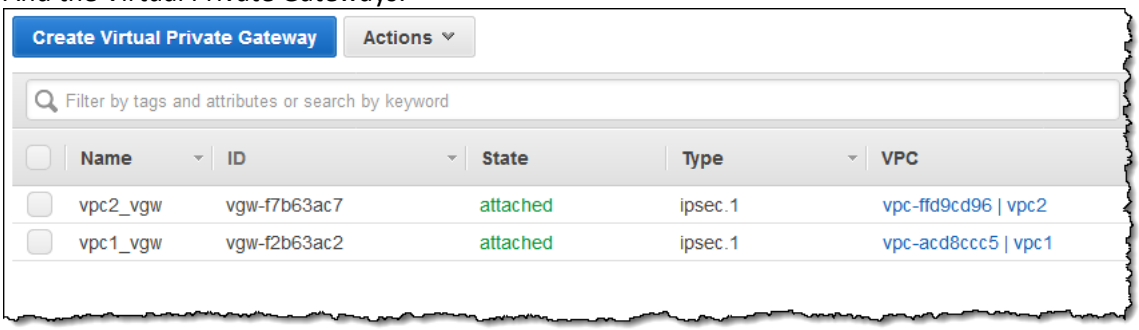
It is ready to use within seconds:



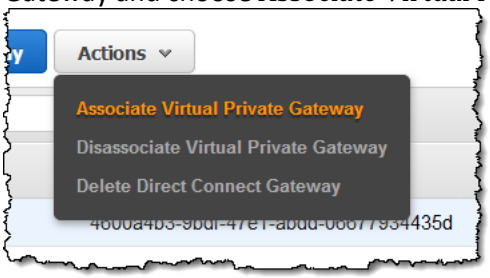
I already have a pair of VPCs with non-overlapping CIDRs, and a Virtual Private Gateway attached to each one. Here are the VPCs (since this is a demo I'll show both in the same Region for convenience):



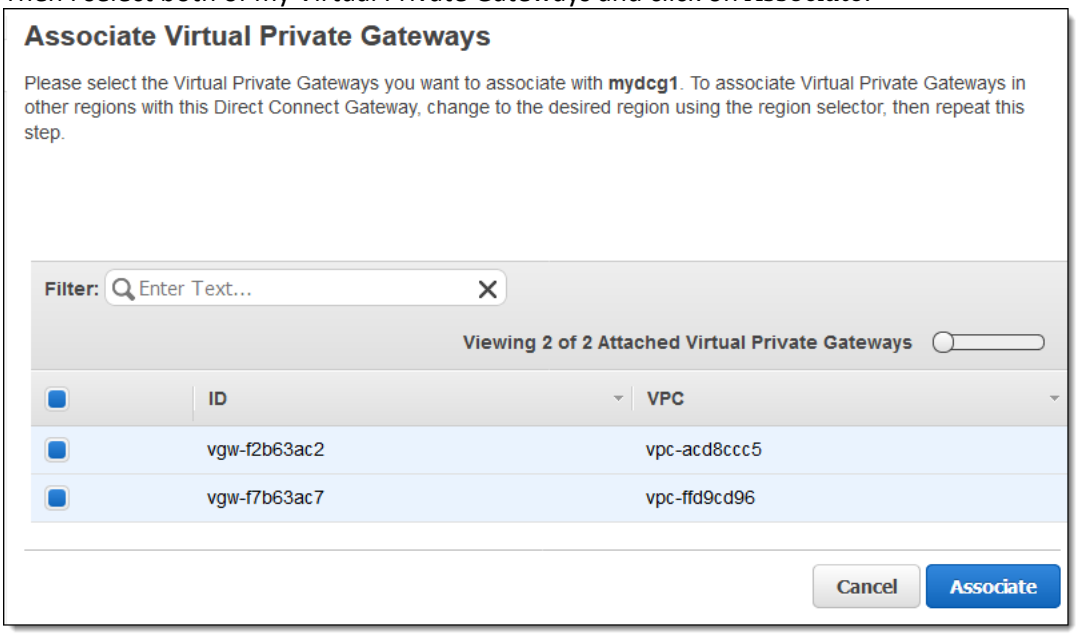
And the Virtual Private Gateways:



I return to the Direct Connect Console and navigate to the Direct Connect Gateways. I select my Gateway and choose Associate Virtual Private Gateway from the Actions menu:

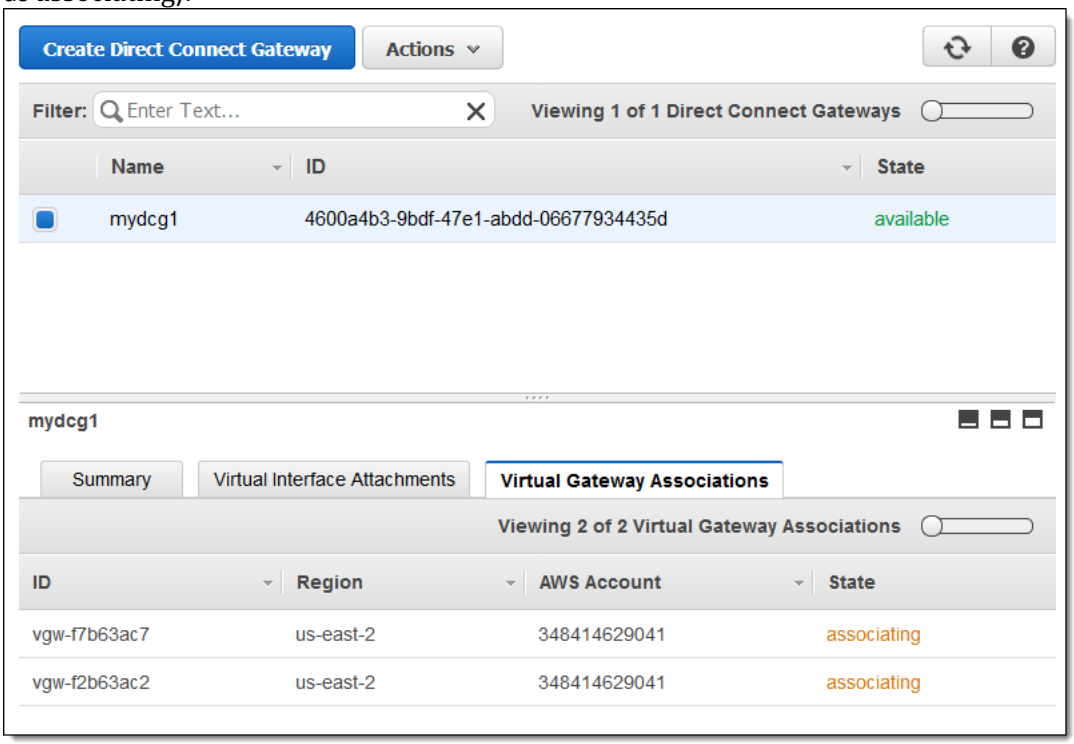


Then I select both of my Virtual Private Gateways and click on Associate:



If, as would usually be the case, my VPCs are in distinct AWS Regions, the same procedure would apply. For this blog post it was easier to show you the operations once rather than twice.

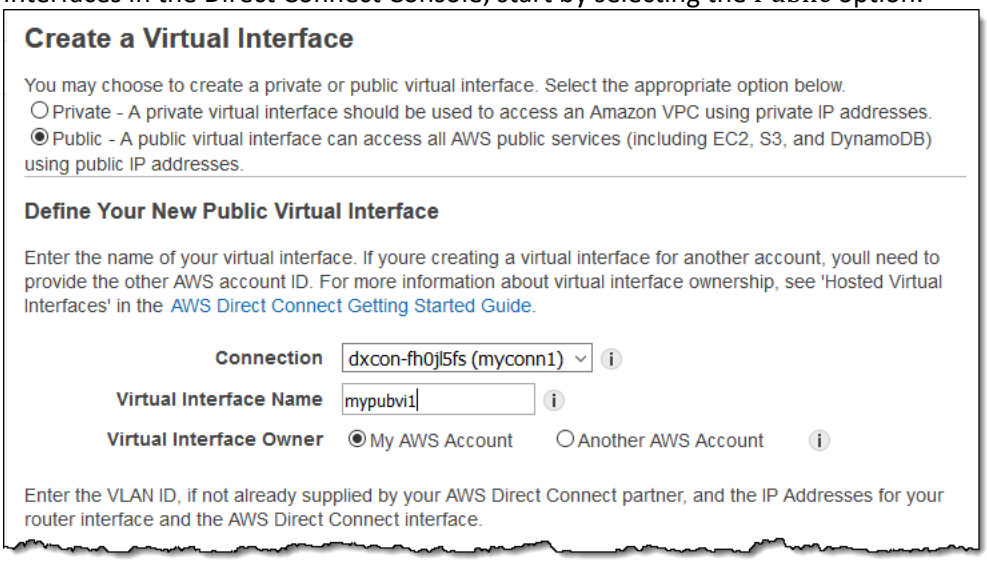
The Virtual Gateway association is complete within a minute or so (the state starts out as associating):



When the state transitions to associated, traffic can flow between your on-premises network and your VPCs, over your AWS Direct Connect connection, regardless of the AWS Regions where your VPCs reside.

Public Virtual Interfaces for Service Endpoints

You can now create Public Virtual Interfaces that will allow you to access AWS public service endpoints for AWS services running in any AWS Region (except AWS China Region) over Direct Connect. These interfaces receive (via BGP) Amazon's global IP routes. You can create these interfaces in the Direct Connect Console; start by selecting the Public option:



Updated Pricing Model

In light of the ever-expanding number of AWS Regions and AWS Direct Connect locations, data transfer pricing is now based on the location of the Direct Connect and the source AWS Region. The new pricing is simpler that the older model which was based on AWS Direct Connect locations.

Now Available
This new feature is available today and you can start to use it right now. You can create and use Direct Connect Gateways at no charge; you pay the usual Direct Connect prices for port hours and data transfer.
— Jeff;

AWS - Direct Connect Gateway

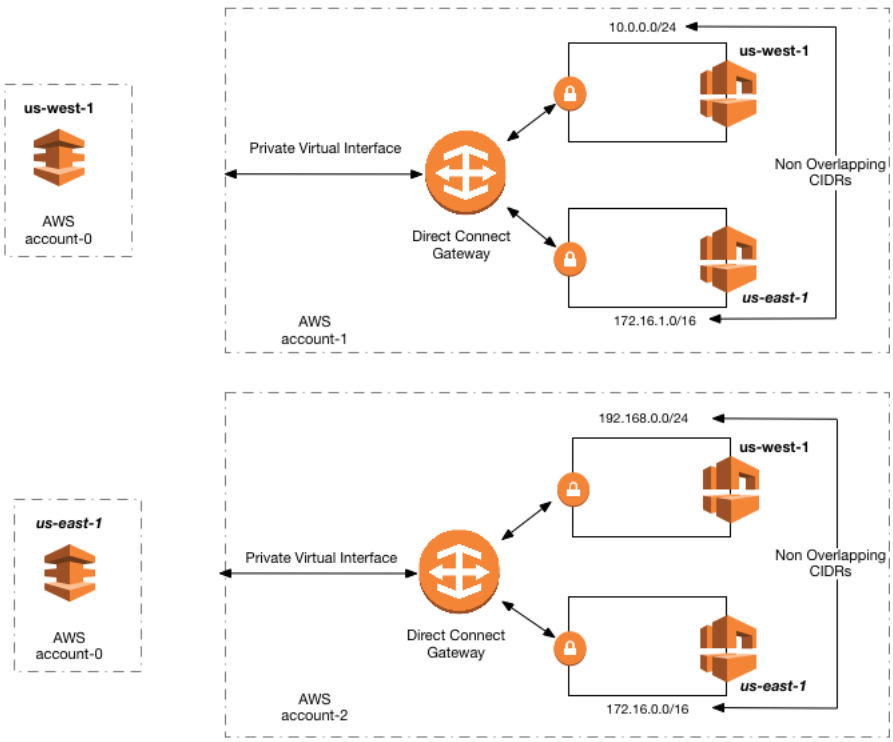
AWS Direct Connect is a cloud service solution that makes it easy to establish a dedicated network connection from your premises to AWS to achieve higher privacy benefits, additional data transfer bandwidth, and more predictable data transfer performance.

Using AWS Direct Connect, you can establish private connectivity between AWS and your datacenter, office, or colocation environment, which in many cases can reduce your network costs, increase bandwidth throughput, and provide a more consistent network experience than Internet-based connections.

Using industry standard 802.1q VLANs, this dedicated connection can be partitioned into multiple virtual interfaces. Virtual interfaces can be reconfigured at any time to meet your changing needs. You can use an ***AWS Direct Connect gateway*** to connect your AWS Direct Connect connection over a private virtual interface to one or more VPCs in your account that are located in the same or different Regions.

You associate a Direct Connect gateway with the virtual private gateway for the VPC. Then, create a private virtual interface for your AWS Direct Connect connection to the Direct Connect gateway. You can attach multiple private virtual interfaces to your Direct Connect gateway.

With Direct Connect Gateway, you no longer need to establish multiple BGP sessions for each VPC; this reduces your administrative workload as well as the load on your network devices.



VPC - End Points and Private Link

VPC Endpoints

Use Case: Control S3 bucket access to VPC Network range

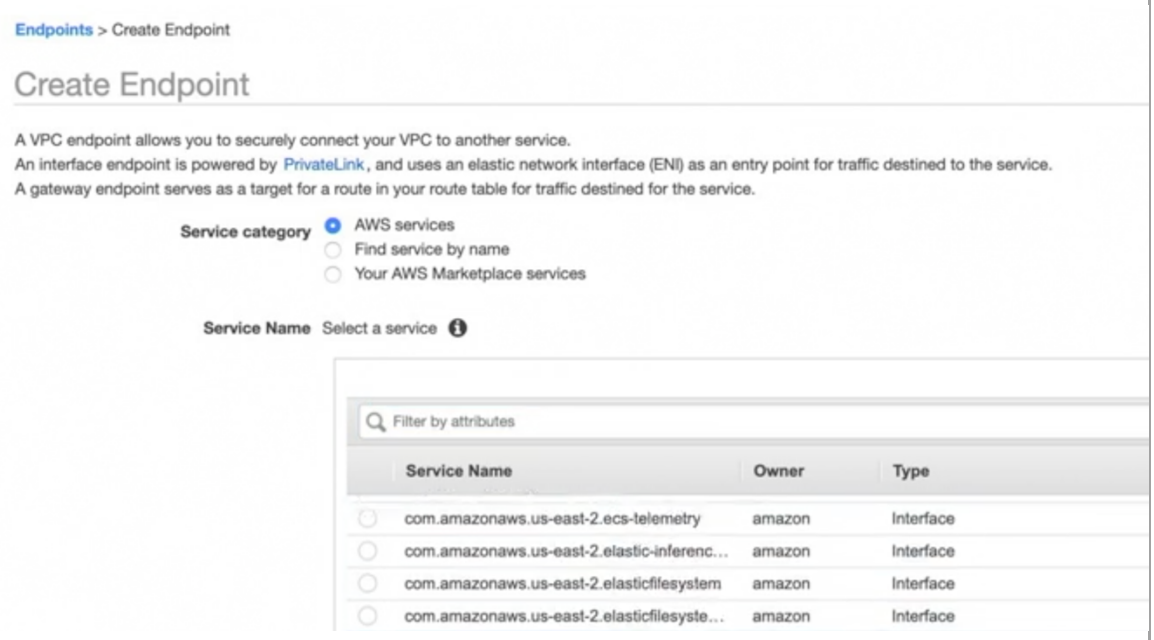
- VPC Gateway End-Point
  - For connecting to S3 and DynamoDB
- VPC Interface End-Point
  - For connecting to other AWS services

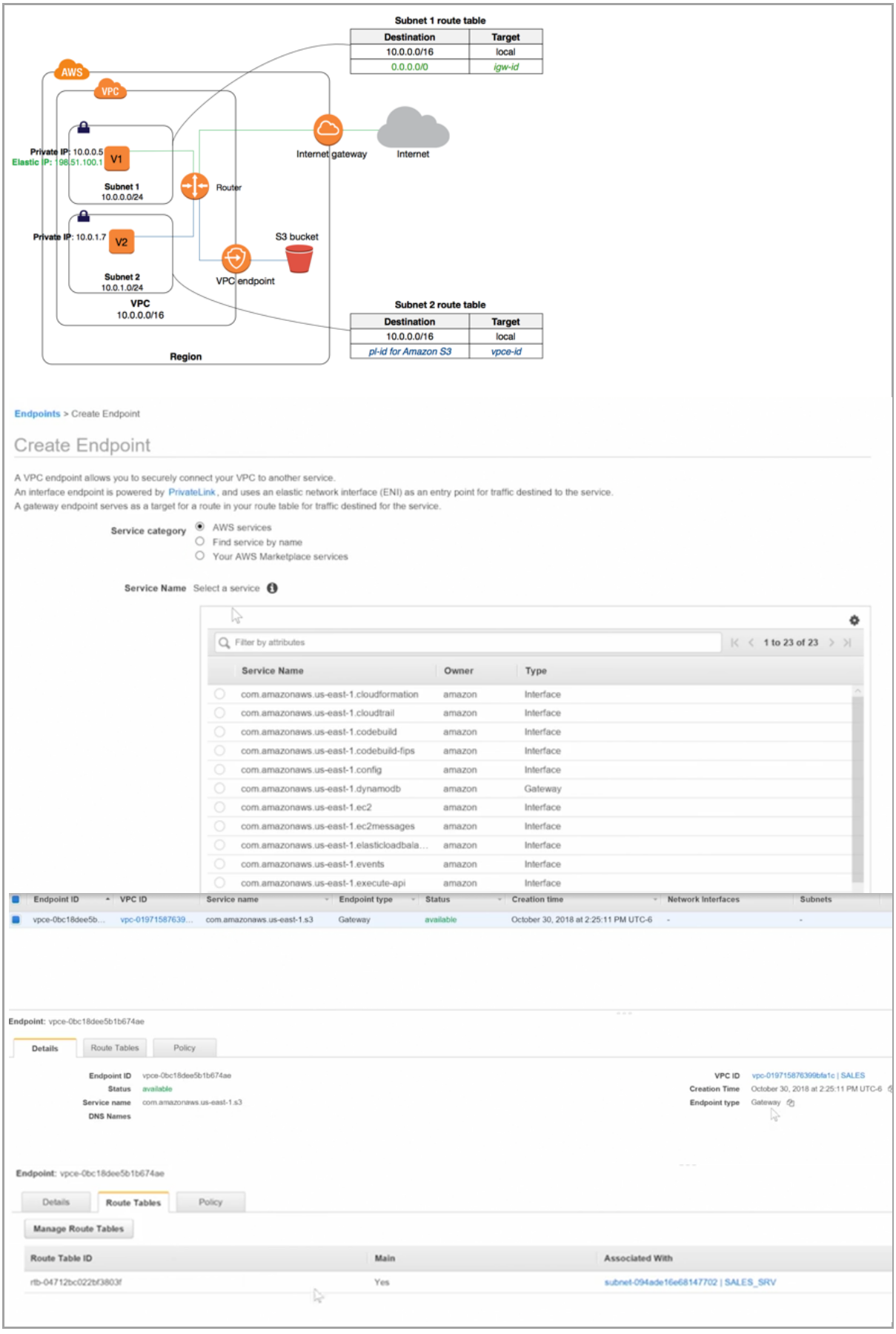
VPC Endpoints

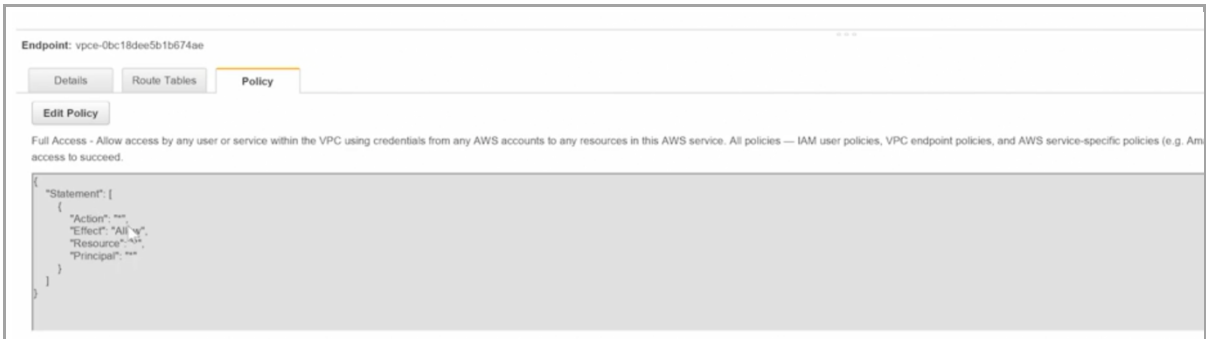
- Take note that your VPC lives within a larger AWS network and the services, such as S3, DynamoDB, RDS and many others, are located outside of your VPC, but still within the AWS network.
- By default, the connection that your VPC uses to connect to your S3 bucket or any other service traverses the public Internet via your Internet Gateway.
- A VPC endpoint enables you to privately connect your VPC to supported AWS services and VPC endpoint services powered by PrivateLink without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC do not require public IP addresses to communicate with resources in the service.

- Traffic between your VPC and the other service does not leave the Amazon network.

- There are two types of VPC endpoints:
  - interface endpoints and
  - gateway endpoints.
- You have to create the type of VPC endpoint required by the supported service.
- An interface endpoint is an elastic network interface with a private IP address that serves as an entry point for traffic destined to a supported service.
- A gateway endpoint is a gateway that is a target for a specified route in your route table, used for traffic destined to a supported AWS service.
  - It is important to note that for Amazon S3 and DynamoDB service, you have to create a gateway endpoint and then use an interface endpoint for other services




Subnet 1 route table
Destination
Target
10.0.0.0/16
local
0.0.0.0/0
igw-id
AWS
VPC
Private IP: 10.0.0.5
Elastic IP: 198.51.100.1
V1
Subnet 1
10.0.0.0/24
Router
Internet gateway
Internet
Private IP: 10.0.1.7
V2
S3 bucket
Subnet 2
10.0.1.0/24
VPC endpoint
VPC
10.0.0.0/16
Region
Subnet 2 route table
Destination
Target
10.0.0.0/16
local
pl-id for Amazon S3
vpce-id
Endpoints > Create Endpoint
Create Endpoint
A VPC endpoint allows you to securely connect your VPC to another service.
An interface endpoint is powered by PrivateLink, and uses an elastic network interface (ENI) as an entry point for traffic destined to the service.
A gateway endpoint serves as a target for a route in your route table for traffic destined for the service.
Service category
AWS services
Find service by name
Your AWS Marketplace services
Service Name
Select a service
Filter by attributes
1 to 23 of 23
Service Name
Owner
Type
com.amazonaws.us-east-1.cloudformation amazon Interface
com.amazonaws.us-east-1.cloudtrail amazon Interface
com.amazonaws.us-east-1.codebuild amazon Interface
com.amazonaws.us-east-1.codebuild-fips amazon Interface
com.amazonaws.us-east-1.config amazon Interface
com.amazonaws.us-east-1.dynamodb amazon Gateway
com.amazonaws.us-east-1.ec2 amazon Interface
com.amazonaws.us-east-1.ec2messages amazon Interface
com.amazonaws.us-east-1.elasticloadbala... amazon Interface
com.amazonaws.us-east-1.events amazon Interface
com.amazonaws.us-east-1.execute-api amazon Interface
Endpoint ID
VPC ID
Service name
Endpoint type
Status
Creation time
Network Interfaces
Subnets
vpce-0bc18dee5b...
vpc-01971587639...
com.amazonaws.us-east-1.s3
Gateway
available
October 30, 2018 at 2:25:11 PM UTC-6
-
-
Endpoint: vpce-0bc18dee5b1b674ae
Details
Route Tables
Policy
Endpoint ID vpce-0bc18dee5b1b674ae
Status available
Service name com.amazonaws.us-east-1.s3
DNS Names
VPC ID vpc-019715876399bfa1c | SALES
Creation Time October 30, 2018 at 2:25:11 PM UTC-6
Endpoint type Gateway
Endpoint: vpce-0bc18dee5b1b674ae
Details
Route Tables
Policy
Manage Route Tables
Route Table ID
Main
Associated With
rtb-04712bc022bf3803f
Yes
subnet-094ade16e68147702 | SALES_SRV

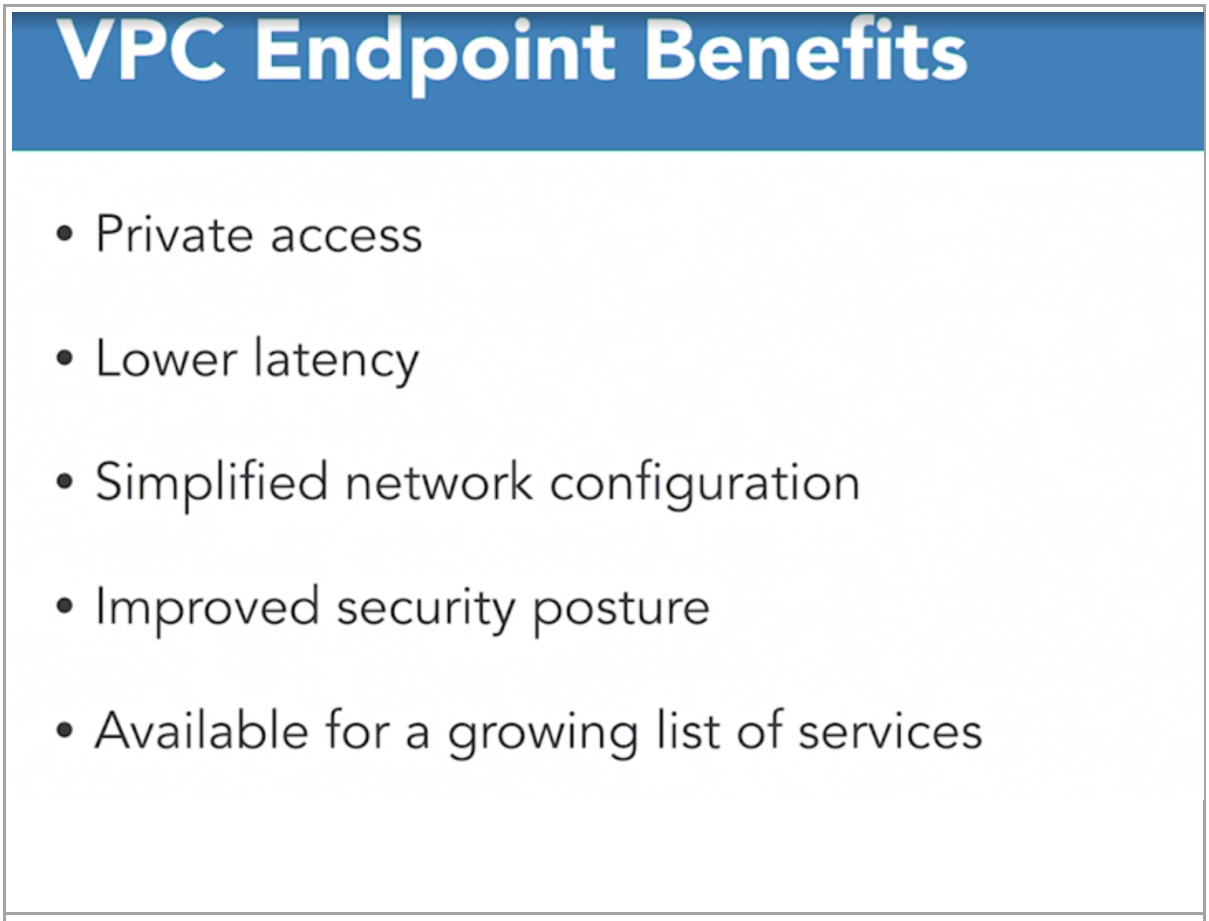


PrivateLink

- Enables private access to AWS services
- Enables private access to partner services
- Incurs hourly charges
- Incurs per-GB charges

https://aws.amazon.com/privatelink/pricing

VPC - End Points



References:

https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-endpoints.html

https://docs.aws.amazon.com/vpc/latest/userguide/vpce-gateway.html

Check out this Amazon VPC Cheat Sheet:

https://tutorialsdojo.com/aws-cheat-sheet-amazon-vpc/

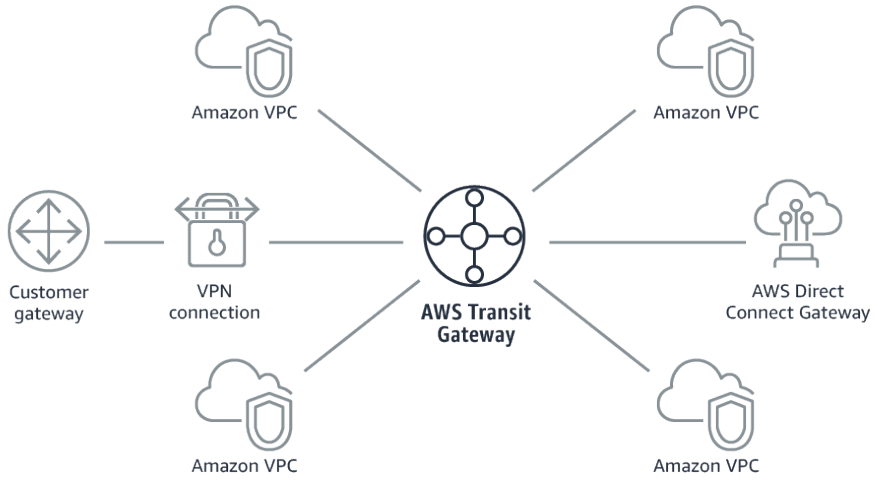
https://aws.amazon.com/transit-gateway/

AWS - Transit Gateways

Transit Gateway

- AWS Transit Gateway is a service that enables customers to connect their Amazon Virtual Private Clouds (VPCs) and their on-premises networks to a single gateway.
- As you grow the number of workloads running on AWS, you need to be able to scale your networks across multiple accounts and Amazon VPCs to keep up with the growth.
- Today, you can connect pairs of Amazon VPCs using peering.

- However, managing point-to-point connectivity across many Amazon VPCs, without the ability to centrally manage the connectivity policies, can be operationally costly and cumbersome.
- For on-premises connectivity, you need to attach your AWS VPN to each individual Amazon VPC. This solution can be time consuming to build and hard to manage when the number of VPCs grows into the hundreds.
- With AWS Transit Gateway, you only have to create and manage a single connection from the central gateway in to each Amazon VPC, on-premises data center, or remote office across your network.
- 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 the Transit Gateway and not to every other network.
- Any new VPC is simply connected to the Transit Gateway and is then automatically available to every other network that is connected to the Transit Gateway.
- This ease of connectivity makes it easy to scale your network as you grow.



- It acts as a Regional virtual router for traffic flowing between your virtual private clouds (VPC) and VPN connections.
- A transit gateway scales elastically based on the volume of network traffic. Routing through a transit gateway operates at layer 3, where the packets are sent to a specific next-hop attachment, based on their destination IP addresses.

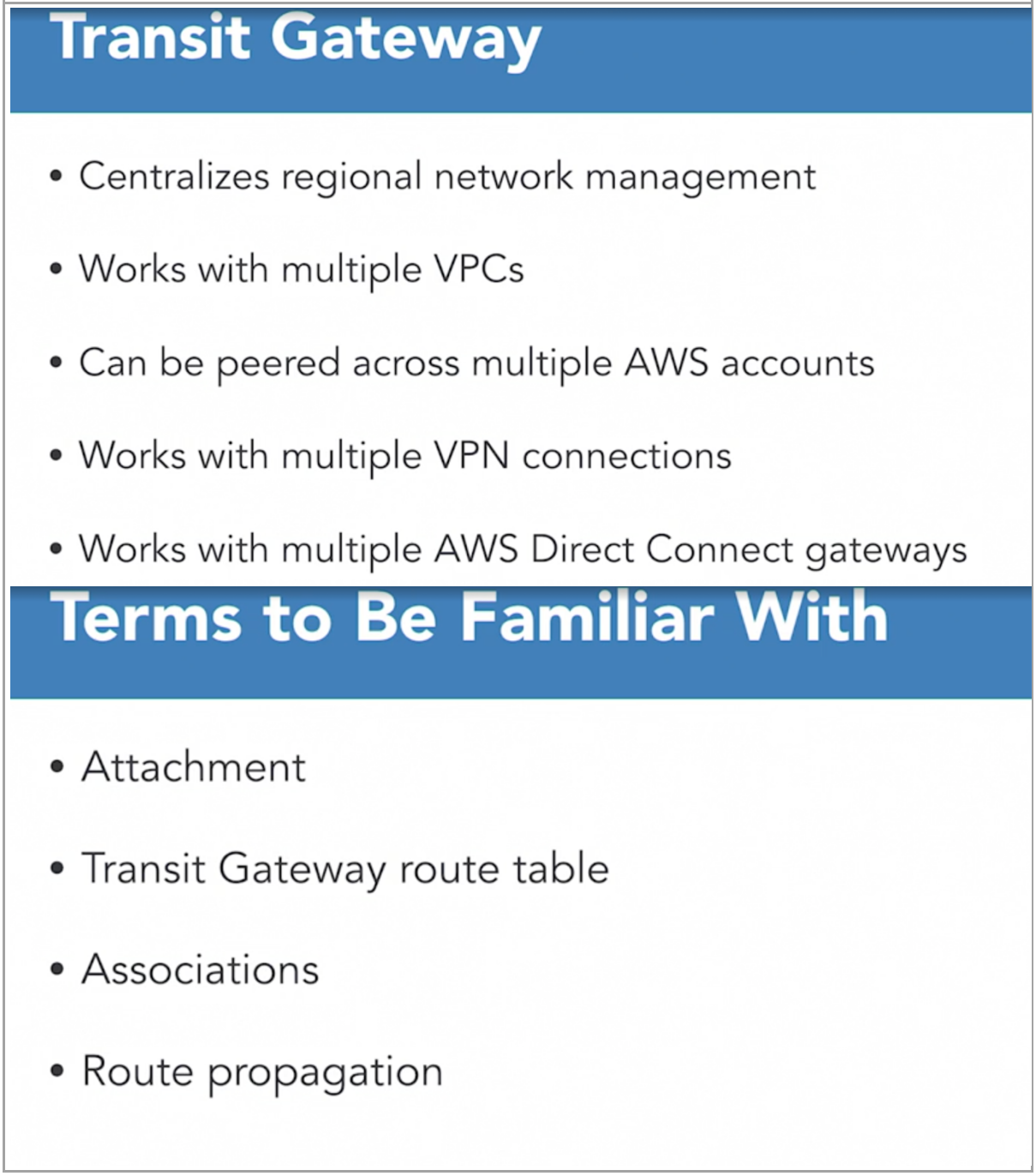
- A transit gateway attachment is both a source and a destination of packets. You can attach the following resources to your transit gateway:
- - One or more VPCs

- - One or more VPN connections
- - One or more AWS Direct Connect gateways
- - One or more transit gateway peering connections
- If you attach a transit gateway peering connection, the transit gateway must be in a different Region.

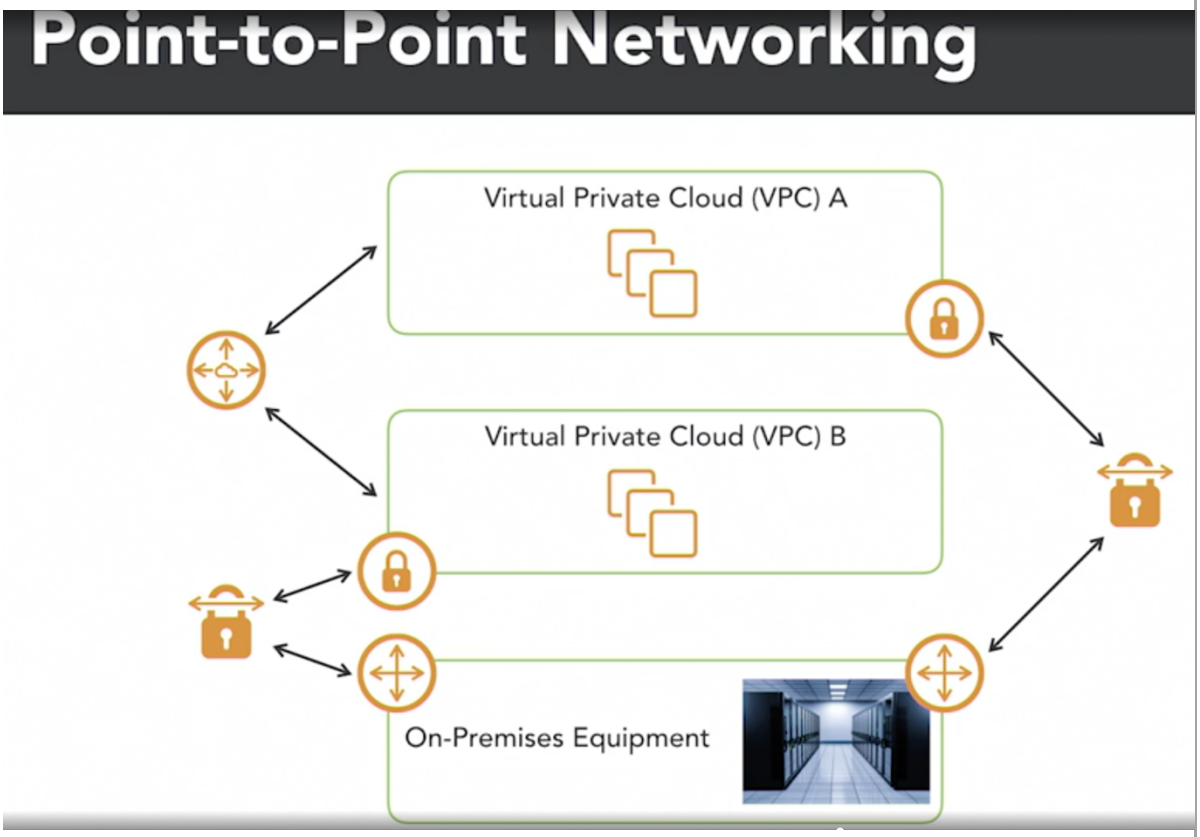
References:

https://aws.amazon.com/transit-gateway/

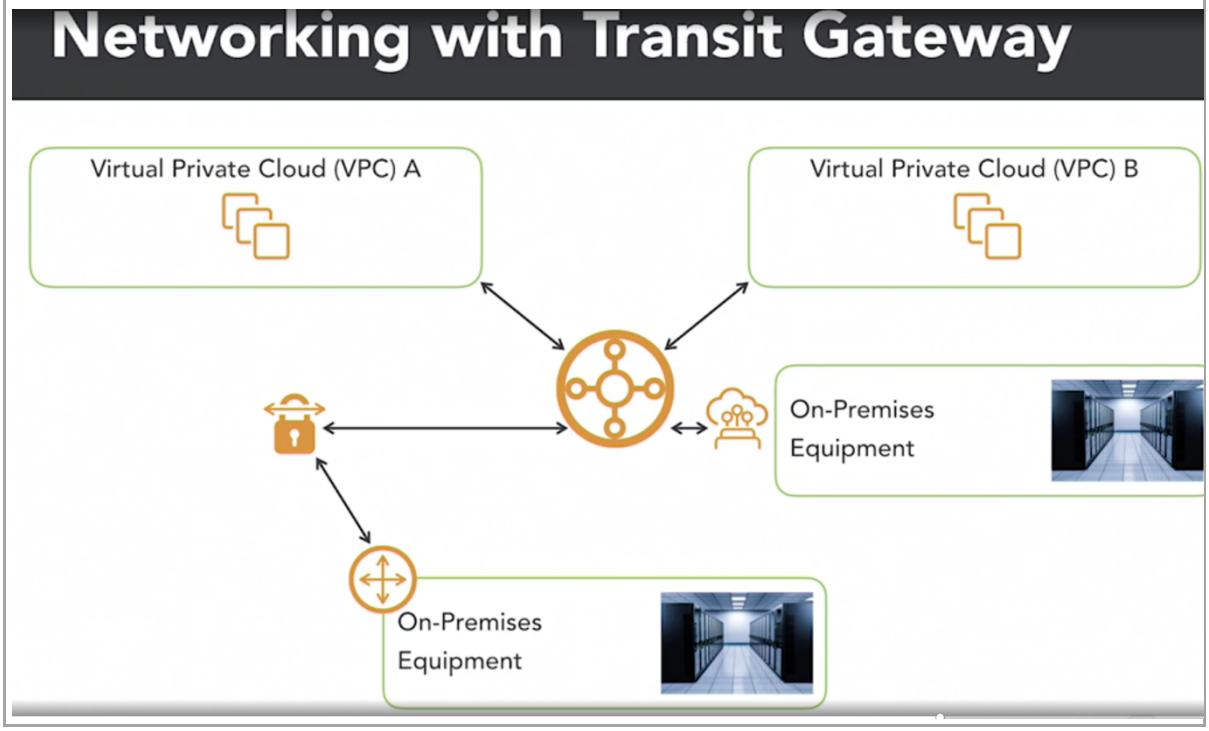
https://docs.aws.amazon.com/vpc/latest/tgw/how-transit-gateways-work.html



TRADITIONAL - point2point connection



Transit Gateway - HUB and SPOC model

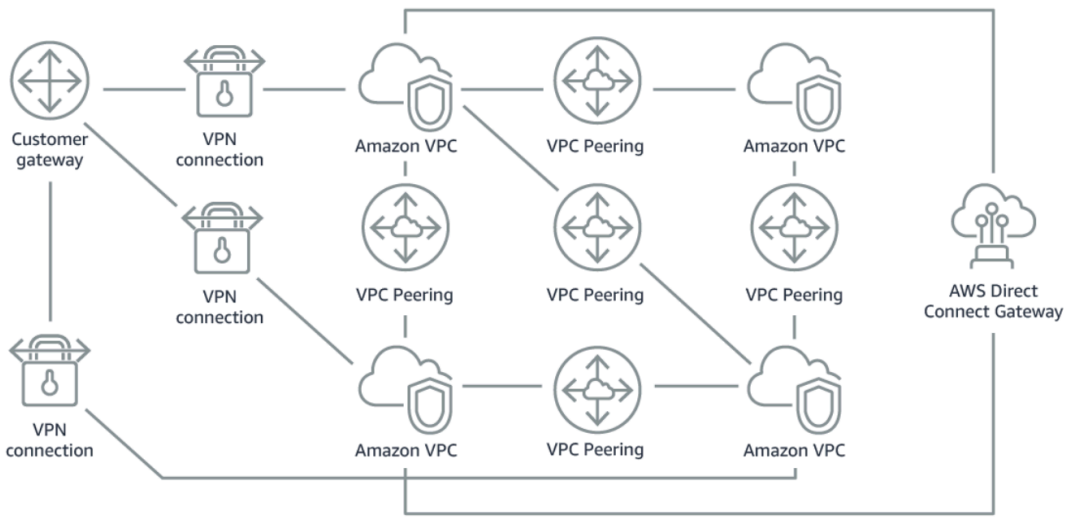


Transit Gateway Benefits

- Simplified connection management
- Improved security posture
- Highly available
- Reasonable hourly billing

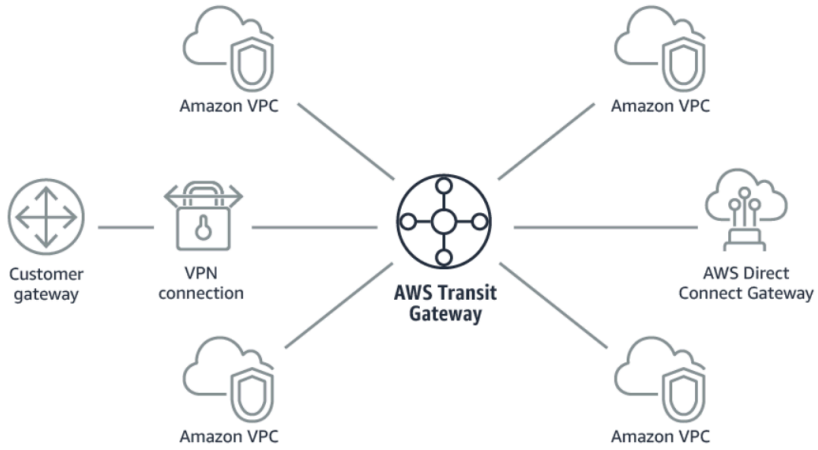
ttps://aws.amazon.com/transit-gateway/pricing

Without AWS Transit Gateway



Complexity increases with scale. You must maintain routing tables within each VPC and connect to each onsite location using separate network gateways.

With AWS Transit Gateway



Your network is streamlined and scalable. AWS Transit Gateway routes all traffic to and from each VPC or VPN, and you have one place to manage and monitor it all.

AWS - CLI to create transit gateway

Examples

Example 1: To associate a Transit Gateway with a VPC

The following `create-transit-gateway-vpc-attachment` example creates a transit gateway attachment to the specified VPC.

```
aws ec2 create-transit-gateway-vpc-attachment \
    --transit-gateway-id tgw-0262a0e521EXAMPLE \
    --vpc-id vpc-07e8ffd50f49335df \
    --subnet-id subnet-0752213d59EXAMPLE
```

Output:

```
{
    "TransitGatewayVpcAttachment": {
```

```
aws ec2 create-route --route-table-id rtb-0ff352a1de46530e1 \
  --destination-cidr-block 172.16.0.0/16 \
  --transit-gateway-id tgw-035c136f200828d30 \
  --profile snijim.ireland.admin

# Add route from vpc2 (ireland) to the admin VPC,
# specifying the transit gateway
aws ec2 create-route --route-table-id rtb-076d8f2a0f00bae6c \
  --destination-cidr-block 172.16.0.0/16 \
  --transit-gateway-id tgw-035c136f200828d30 \
  --profile snijim.ireland.admin

# Add route from vpc3 (ireland) to the admin VPC,
# specifying the transit gateway
aws ec2 create-route --route-table-id rtb-0dfdf8bbb180fe287 \
  --destination-cidr-block 172.16.0.0/16 \
  --transit-gateway-id tgw-035c136f200828d30 \
```

AWS - VPN CloudHub

- If you have multiple VPN connections, you can provide secure communication between sites using the ***AWS VPN CloudHub***.
- This enables your remote sites to communicate with each other, and not just with the VPC.
- The VPN CloudHub operates on a simple hub-and-spoke model that you can use with or without a VPC.
- This design is suitable for customers with multiple branch offices and existing internet connections who'd like to implement a convenient, **potentially low-cost hub-and-spoke model** for primary or backup connectivity between these remote offices.

References:

https://docs.aws.amazon.com/vpc/latest/userguide/VPN_CloudHub.html

https://docs.aws.amazon.com/vpc/latest/userguide/vpn-connections.html

Check out this Amazon VPC Cheat Sheet:

https://tutorialsdojo.com/amazon-vpc/

Direct Connect Gateway Vs Transit Gateway Vs VPG

https://www.megaport.com/blog/aws-vgw-vs-dgw-vs-tgw/#:~:text=Transit%20Gateway%20provides%20enhanced%20routing,to%201.25Gbps%20of%20

throughput.&text=A%20list%20of%20supported%20regions%20is%20available%20via%20the%20AWS%20FAQs.

Exploring the evolution of the AWS network gateway and choosing the best option for your business.

AWS launched the newest version of their native network routing service, Transit Gateway (TGW), in November 2018. The cloud-based network gateway, that allows customers to connect Virtual Private Clouds (VPCs) across different accounts in a hub and spoke topology, is the third evolution in this feature set. The release was preceded by Direct Connect Gateway (DGW) which was announced in 2017, and prior to that came Virtual Private Gateway (VGW).

Navigating these options and figuring out which fits your use case can be tricky. We're demystifying each service so you can more easily determine which solution is right for your business. Firstly, it's best to consider the requirements of your workloads as each service offers certain features and doesn't offer others. Take a look at the table below for a quick overview.

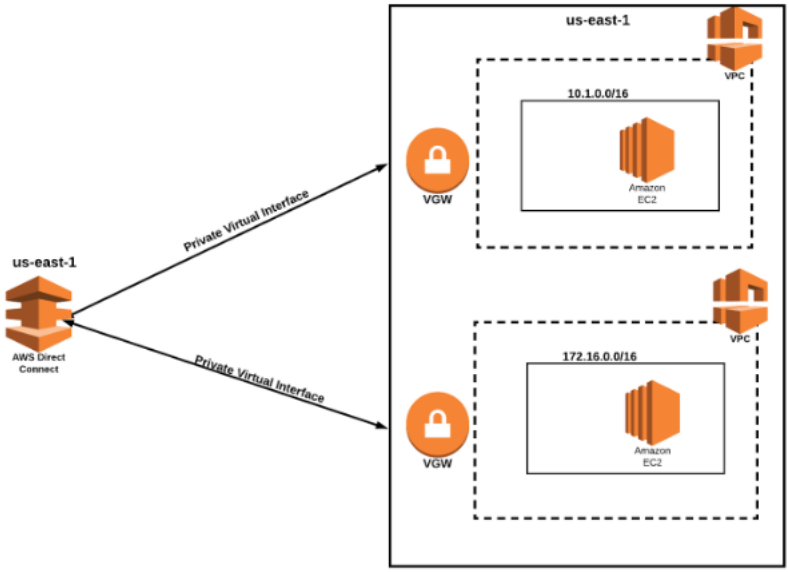
	Multiple Region	Multiple Account	S2S VPN	Direct Connect	Transitive Routing	Globally Available	Route Segmentation
VGW	❌	❌	✅	✅	❌	✅	❌
DGW	✅	✅	❌	✅	❌	✅	❌
TGW	❌	✅	✅	✅	✅	❌	✅

Virtual Private Gateway – VGW

The introduction of the VGW introduced the ability to let multiple VPCs, in the same region, on the same account, share a Direct Connect. Prior to this, you'd need a Direct Connect Private Virtual Interface (VIF) for each VPC, establishing a 1:1 correlation, which didn't scale well both in terms of cost and administrative overhead. VGW became a solution that reduced the expense of requiring new Direct Connect circuits for each VPC as long as both VPCs were in the same region, on the same account. This construct can be used with either Direct Connect or the Site-to-Site VPN.

Use Case:

Multiple VPCs in the same region sharing the same Direct Connect.

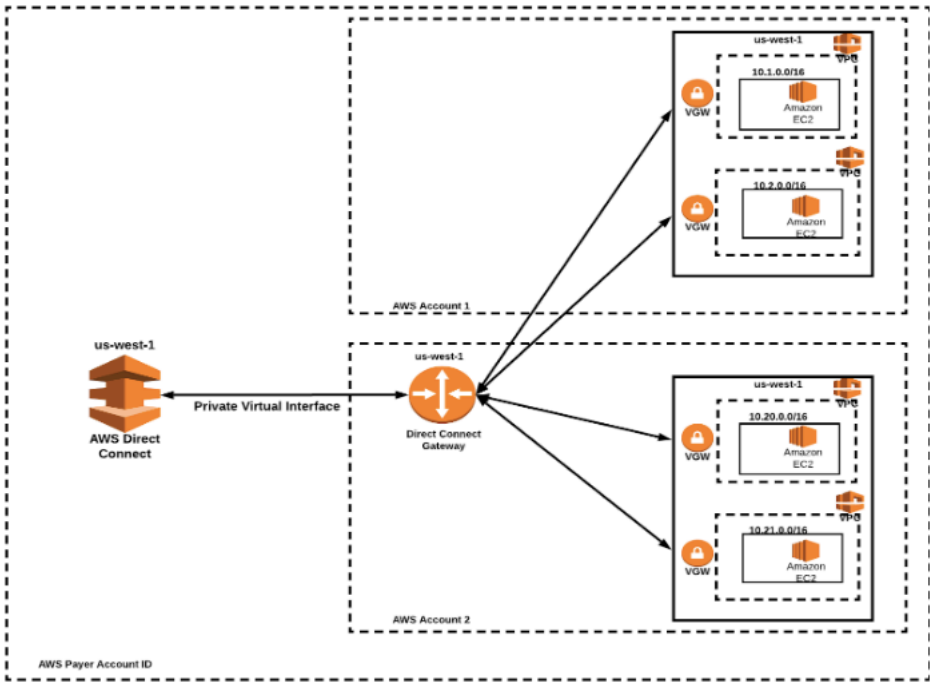


Direct Connect Gateway – DGW

DGW builds upon VGW capabilities adding the ability to connect VPCs in one region to a Direct Connect in another region. CIDR addresses cannot overlap. In addition, traffic will not route from VPC-A to the Direct Connect Gateway and to VPC-B. Traffic will have to route from the VPC-A —> Direct Connect —-> Data Centre Router —-> Direct Connect —> VPC-B.

Use Case:

Multiple VPCs spread across multiple regions sharing the same Direct Connect.



	Multiple Region	Multiple Account	S2S VPN	Direct Connect	Transitive Routing	Globally Available	Route Segmentation
DGW	✅	✅	❌	✅	❌	✅	❌

Transit Gateway – TGW

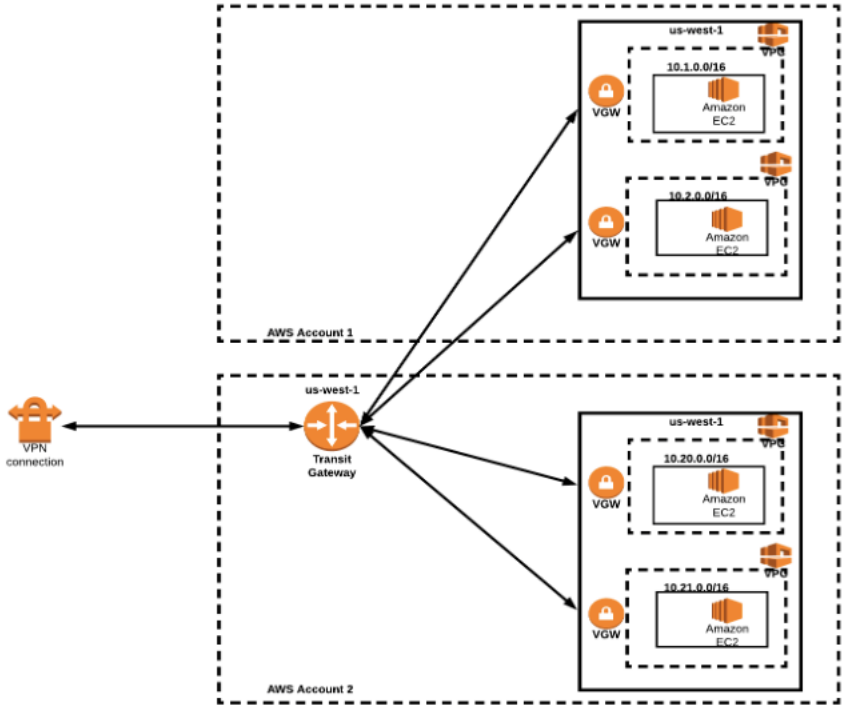
Transit Gateway provides enhanced routing services over the previous offerings from AWS. The initial launch of Transit Gateway doesn't support Direct Connect and requires Site-to-Site VPN. Each VPN session is limited to 1.25Gbps of throughput. If you want to scale beyond this, you'll need to add multiple VPN connections to reach your desired aggregate bandwidth and then leverage ECMP to multipath traffic across all VPN connections. Even with ECMP, a single flow would be limited to 1.25Gbps.

TGW coupled with AWS Resource Access Manager will allow you to use a single Transit Gateway across multiple AWS accounts, however, it's still limited to a single region. In addition, CIDR overlap is permitted with the addition of multiple route tables. Being able to leverage multiple route tables on TGW delivers a VRF type of capability that allows you to isolate routing domains to enforce traffic segmentation. A significant advantage of the TGW is you can route between VPCs without

your data having to hairpin over the VPN to your on-premises router and back in to AWS as observed with VGW and DGW. A list of supported regions is available via the AWS FAQs.

Use Case:

Multiple VPCs in the same region, spread across different AWS accounts using the same Direct Connect.

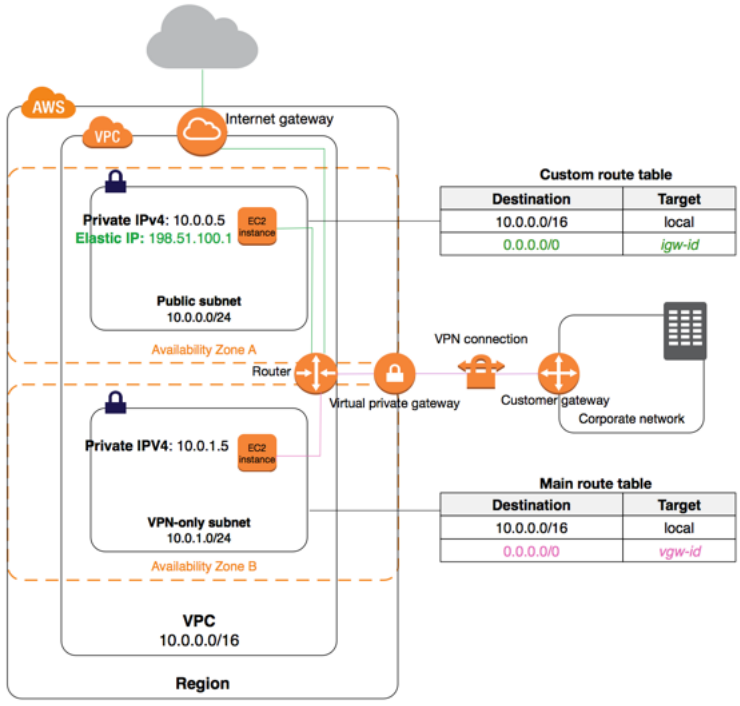


	Multiple Region	Multiple Account	S2S VPN	Direct Connect	Transitive Routing	Globally Available	Route Segmentation
TGW	❌	✅	✅	✅	✅	❌	✅

For more information on connecting to AWS services via Megaport's Software Defined Network and designing the right network architecture for your business, get in touch with our team here.

Route Tables

- Your VPC has an implicit router, and you use route tables to control where network traffic is directed. Each subnet in your VPC must be associated with a route table, which controls the routing for the subnet (subnet route table).
- You can explicitly associate a subnet with a particular route table. Otherwise, the subnet is implicitly associated with the main route table.
- A subnet can only be associated with one route table at a time, but you can associate multiple subnets with the same subnet route table.
- You can optionally associate a route table with an internet gateway or a virtual private gateway (gateway route table).
- This enables you to specify routing rules for inbound traffic that enters your VPC through the gateway
- Be sure that the subnet route table also has a route entry to the internet gateway.
- If this entry doesn't exist, the instance is in a private subnet and is inaccessible from the internet.
- In cases where your EC2 instance cannot be accessed from the Internet (or vice versa), you usually have to check two things:
- - Does it have an EIP or public IP address?
- - Is the route table properly configured?



Route Table Key Concepts

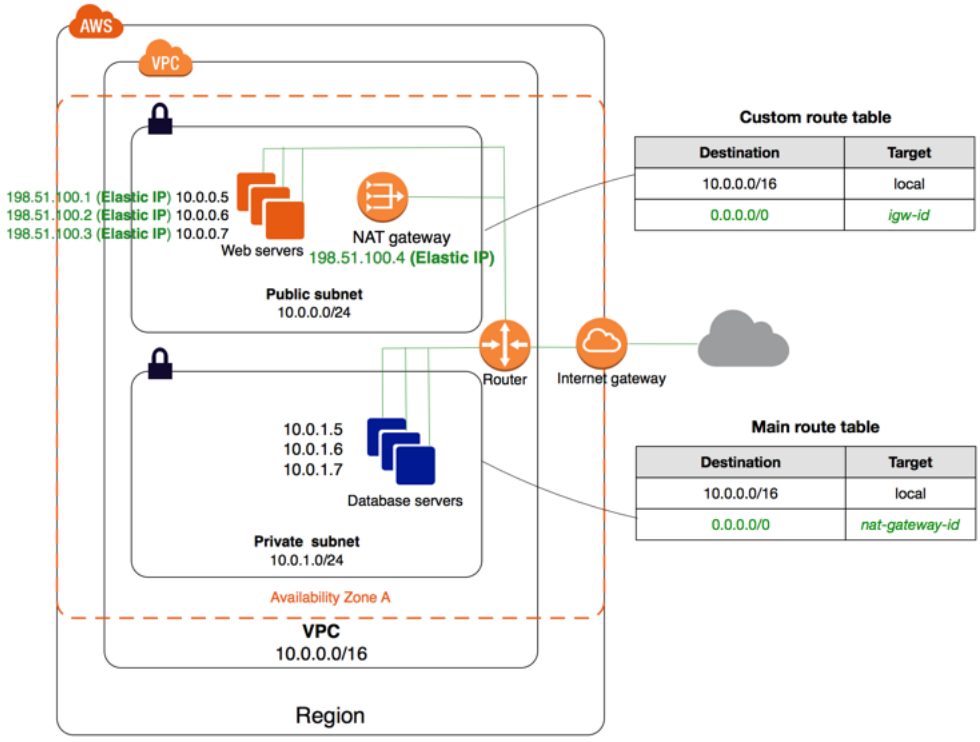
- Main route table can be modified, but not deleted
- Control subnet routing via implicit router
- Each subnet must be associated with a route table
- Main route table is catch-all for unassigned subnets

```
/Users/sbn
-> aws ec2 create-route-table --vpc-id vpc-0e041836fa0f65c66 --profile snijim.dev.admin
{
    "RouteTable": {
        "Associations": [],
        "RouteTableId": "rtb-0c087647d347dcd12",
        "VpcId": "vpc-0e041836fa0f65c66",
        "PropagatingVgws": [],
        "Tags": [],
        "Routes": [
            {
                "GatewayId": "local",
                "DestinationCidrBlock": "192.168.0.0/22",
                "State": "active",
                "Origin": "CreateRouteTable"
            }
        ],
        "OwnerId": "828594214613"
    }
}
```

NAT - Gateway (Network Address Translation - gateway)

- A **NAT Gateway** is a highly available, managed Network Address Translation (NAT) service for your resources in a private subnet to access the Internet.
- NAT gateway is created in a specific Availability Zone and implemented with redundancy in that zone.

- You must create a NAT gateway on a public subnet to enable instances in a private subnet to connect to the Internet or other AWS services, but prevent the Internet from initiating a connection with those instances.



- If you have resources in multiple Availability Zones and they share one NAT gateway, and if the NAT gateway's Availability Zone is down, resources in the other Availability Zones lose Internet access.
- To create an Availability Zone-independent architecture, create a NAT gateway in each Availability Zone and configure your routing to ensure that resources use the NAT gateway in the same Availability Zone.

IPv4 traffic

- AWS offers two kinds of NAT devices — a NAT gateway or a NAT instance. It is recommended to use NAT gateways, as they provide better availability and bandwidth over NAT instances. The NAT Gateway service is also a managed service that does not require your administration efforts. A NAT instance is launched from a NAT AMI.
- Just like a NAT instance, you can use a network address translation (NAT) gateway to enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those instances.

Here is a diagram showing the differences between NAT gateway and NAT instance:

Tutorials Dojo

Attribute	NAT gateway	NAT instance
Availability	Highly available. NAT gateways in each Availability Zone are implemented with redundancy. Create a NAT gateway in each Availability Zone to ensure zone-independent architecture.	Use a script to manage failover between instances
Bandwidth	Can scale up to 45 Gbps.	Depends on the bandwidth of the instance type
Maintenance	Manage by AWS	Manage by you.
Performance	Software is optimized for handling NAT traffic	A generic Amazon Linux AMI that's configured to perform NAT
Cost	Charged depending on the number of NAT gateways you use, duration of usage, and amount of data that you send through the NAT gateways.	Charged depending on the number of NAT instances that you use, duration of usage, and instance type and size.
Type and size	Uniform offering; you don't need to decide on the type or size.	Choose a suitable instance type and size, according to your predicted workload
Public IP addresses	Choose the Elastic IP address to associate with a NAT gateway at creation.	Use an elastic IP address or a public IP address with a NAT instance. You can change the public IP address at any time by associating a new elastic IP address with the instance.
Private IP addresses	Automatically selected from the subnet's IP address range when you create the gateway.	Assign a specific private IP address from the subnet's IP address range when you launch the instance.
Security groups	Cannot be associated with a NAT gateway	Associate with your NAT instance and the resources behind your NAT instance to control inbound and outbound traffic.
Network ACLs	Use a network ACL to control the traffic to and from the subnet in which your NAT gateway resides.	Use a network ACL to control the traffic to and from the subnet in which your NAT instance resides.
Flow logs	Use flow logs to capture the traffic.	Use flow logs to capture the traffic.
Post Forwarding	Not supported.	Manually customize the configuration to support port forwarding.
Bastion Servers	Not supported.	Use as a bastion server.
Traffic Metrics	Monitor your NAT gateway using cloudwatch	View Cloudwatch metrics for the instance.
Timeout Behavior	When a connection times out, a NAT gateway returns an RST packet to any resources behind the NAT gateway that attempt to continue the connection (it does not send a FIN packet).	When a connection times out, a NAT instance sends a FIN packet to resources behind the NAT instance to close the connection.
IP Fragmentation	Supports forwarding of IP fragmented packets for the UDP protocol. Does not support fragmentation for the TCP and ICMP protocols. Fragmented packets for these protocols will get dropped.	Supports reassembly of IP fragmented packets for the UDP, TCP, and ICMP protocols.

ONLY for IPv6. - Egress only INTERNET gateway

- **Egress-Only Internet Gateway** is incorrect because this is primarily used for VPCs that use IPv6 to enable instances in a private subnet to connect to the Internet or other AWS services, but prevent the Internet from initiating a connection with those instances, just like what NAT Instance and NAT Gateway do.
- The scenario explicitly says that the EC2 instances are using IPv4 addresses which is why Egress-only Internet gateway is invalid, even though it can provide the required high availability.

References:

https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-nat-gateway.html

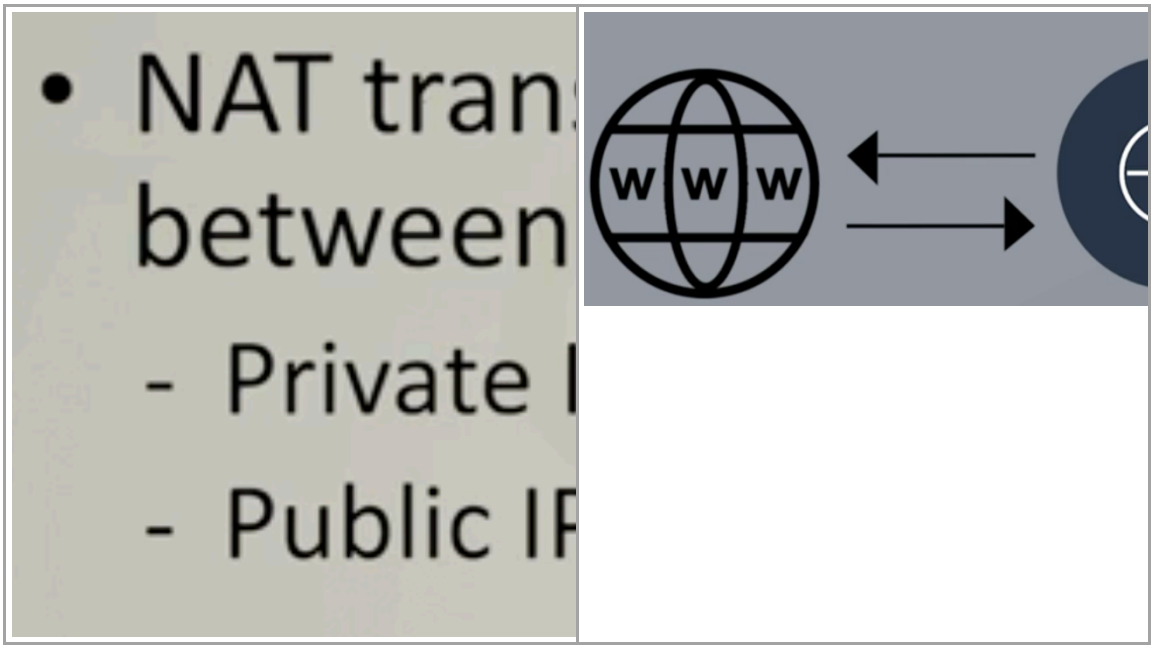
https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-comparison.html

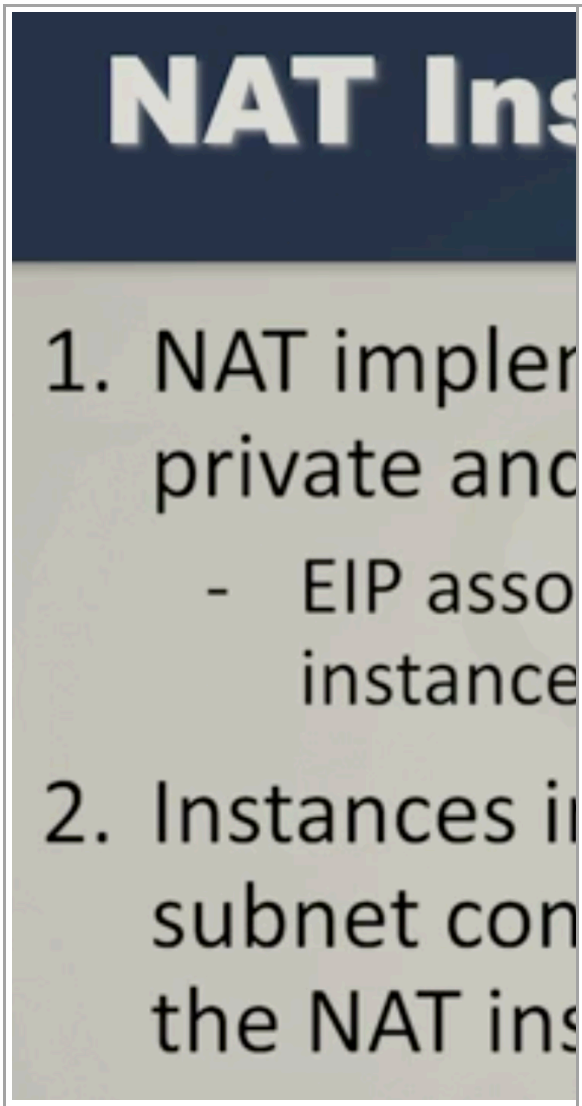
https://docs.aws.amazon.com/vpc/latest/userguide/egress-only-internet-gateway.html

Check out this Amazon VPC Cheat Sheet:

https://tutorialsdojo.com/aws-cheat-sheet-amazon-vpc/

- Map - Multiple private hosts single routable network IP address





NAT Options in

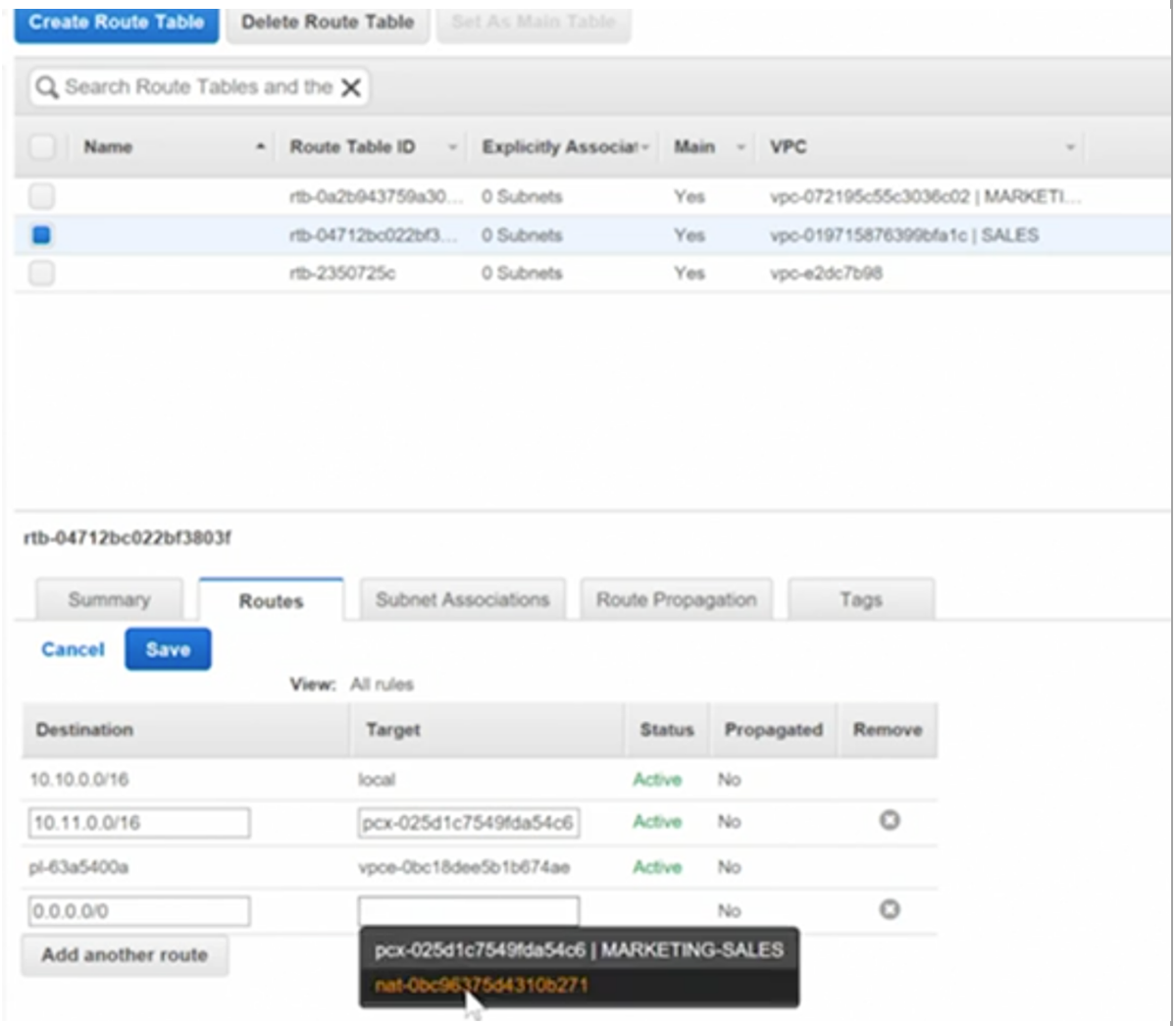
NAT Instance

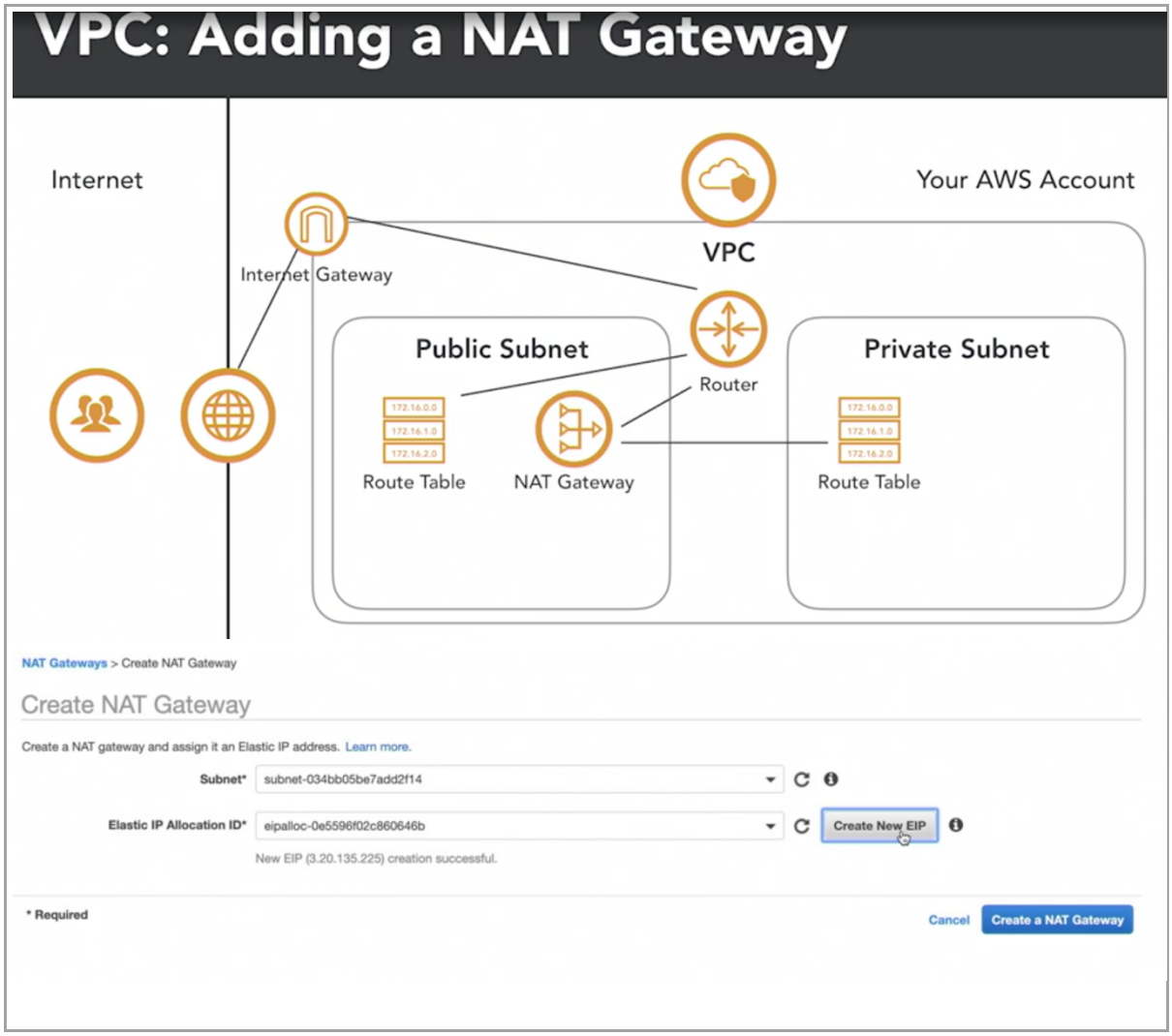
- Managed by you
- Bandwidth limited by instance
- Able to specify private IP
- Supports port forwarding
- Acts as a bastion host

- Network Address Translation (NAT) is used to interconnec Point private networks and public networks
- An Elastic IP (EIP) is associatd with the NAT instance for the public-facing side
- Instances in the private subnet of the VPC use the NAT to connect to the Internet.
- NAT can be implemented using a dedicated NAT instance or using an AWS NAT Gateway

Step#1 : Create a NAT gateway to Private Subnet and attach Elastic IP

Step#2 : Create a ROUTE to Private VPC route pointing ROUTES to NAT instance gateway

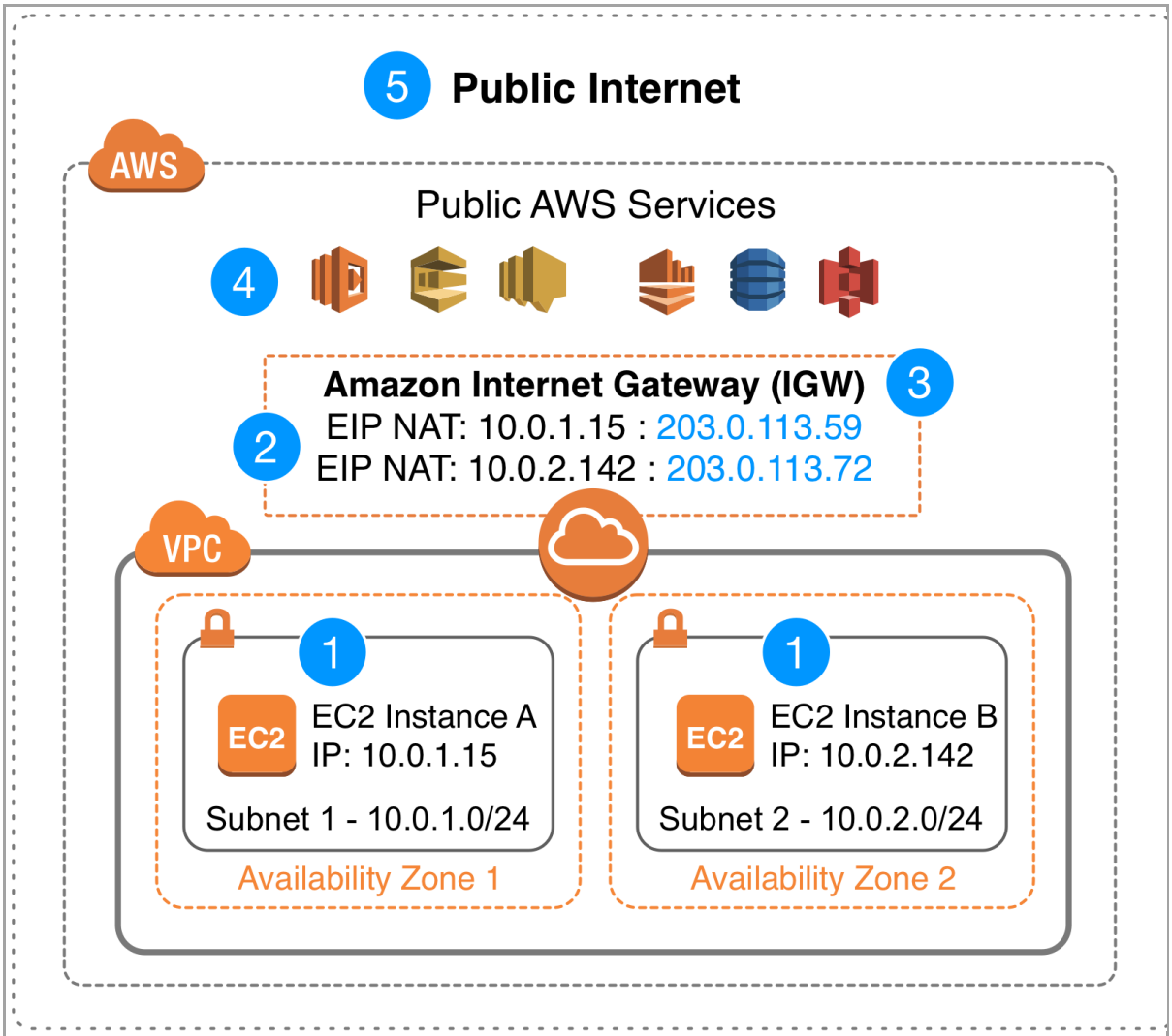




Elastic IPs

You can bring part or all of your public IPv4 address range from your on-premises network to your AWS account. You continue to own the address range, but AWS advertises it on the Internet. After you bring the address range to AWS, it appears in your account as an address pool. You can create an Elastic IP address from your address pool and use it with your AWS resources, such as EC2 instances, NAT gateways, and Network Load Balancers. This is also called "Bring Your Own IP Addresses (BYOIP)".

To ensure that only you can bring your address range to your AWS account, you must authorize Amazon to advertise the address range and provide proof that you own the address range.



A **Route Origin Authorization (ROA)** is a document that you can create through your Regional internet registry (RIR), such as the American Registry for Internet Numbers (ARIN) or Réseaux IP Européens Network Coordination Centre (RIPE). It contains the address range, the ASNs that are allowed to advertise the address range, and an expiration date.

The ROA authorizes Amazon to advertise an address range under a specific AS number. However, it does not authorize your AWS account to bring the address range to AWS. To authorize your AWS account to bring an address range to AWS, you must publish a self-signed X509 certificate in the RDAP remarks for the address range. The certificate contains a public key, which AWS uses to verify the authorization-context signature that you provide. You should keep your private key secure and use it to sign the authorization-context message.

Hence, **creating Elastic IP addresses from your Bring Your Own IP (BYOIP) address prefix and using them with AWS resources such as EC2 instances, Network Load Balancers, and NAT Gateways** is the correct answer.

Below is In-correct

Creating a Route Origin Authorization (ROA) document through your Domain Name Registrar and afterwards, provisioning and advertising your whitelisted IP address

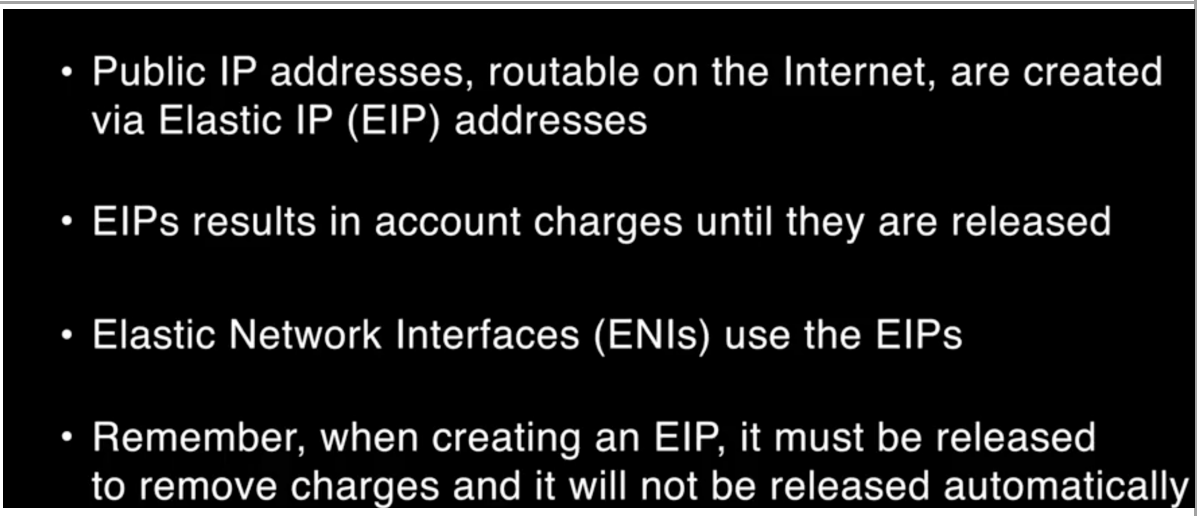
range to your AWS account is incorrect because although creating a Route Origin Authorization (ROA) document is part of the solution, you have to create it through your Regional Internet Registry (RIR) and not via your Domain Name Registrar.

References:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-byoip.html

https://aws.amazon.com/vpc/faqs/#Bring_Your_Own_IP

https://aws.amazon.com/blogs/networking-and-content-delivery/introducing-bring-your-own-ip-byoip-for-amazon-vpc/



VPN - Virtual Private Network

- Amazon VPC offers you the flexibility to fully manage both sides of your Amazon VPC connectivity by creating a VPN connection between your remote network and a software VPN appliance running in your Amazon VPC network.
- This option is recommended if you must manage both ends of the VPN connection either for compliance purposes or for leveraging gateway devices that are not currently supported by Amazon VPC's VPN solution.
- You can create an IPsec VPN connection between your VPC and your remote network. On the AWS side of the VPN connection, a *virtual private gateway* provides two VPN endpoints (tunnels) for automatic failover.
- You configure your *customer gateway* on the remote side of the VPN connection.
- If you have more than one remote network (for example, multiple branch offices), you can create multiple AWS managed VPN connections via your virtual private gateway to enable communication between these networks.

- With AWS Site-to-Site VPN, you can connect to an Amazon VPC in the cloud the same way you connect to your branches.

- AWS Site-to-Site VPN establishes secure and private sessions with IP Security (IPSec) and Transport Layer Security (TLS) tunnels.

What is AWS Site-to-Site VPN?

- By default, instances that you launch into an Amazon VPC can't communicate with your own (remote) network.
- You can enable access to your remote network from your VPC by creating an AWS Site-to-Site VPN (Site-to-Site VPN) connection, and configuring routing to pass traffic through the connection.
- Although the term *VPN connection* is a general term, in this documentation, a VPN connection refers to the connection between your VPC and your own on-premises network.
- Site-to-Site VPN supports Internet Protocol security (IPsec) VPN connections.
- Your Site-to-Site VPN connection is either an AWS Classic VPN or an AWS VPN.
  For more information, see Site-to-Site VPN categories.

Your Site-to-Site VPN connection is either an AWS Classic VPN connection or an AWS VPN connection.
Any new Site-to-Site VPN connection that you create is an AWS VPN connection.

The following features are supported on AWS VPN connections only:

- Internet Key Exchange version 2 (IKEv2)
- NAT traversal
- 4-byte ASN (in addition to 2-byte ASN)
- CloudWatch metrics
- Reusable IP addresses for your customer gateways
- Additional encryption options; including AES 256-bit encryption, SHA-2 hashing, and additional Diffie-Hellman groups
- Configurable tunnel options
- Custom private ASN for the Amazon side of a BGP session
- Private Certificate from a subordinate CA from AWS Certificate Manager Private Certificate Authority
- Support for IPv6 traffic
  For information about identifying and migrating your connection, see Identifying a Site-to-Site VPN connection and Migrating from AWS Classic VPN to AWS VPN.

References:

http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpn-connections.html

https://docs.aws.amazon.com/whitepapers/latest/aws-vpc-connectivity-options/software-vpn-network-to-amazon.html

Check out this Amazon VPC Cheat Sheet:

https://tutorialsdojo.com/aws-cheat-sheet-amazon-vpc/

Setting Up VPN connection

https://docs.aws.amazon.com/vpc/latest/userguide/SetUpVPNConnections.html

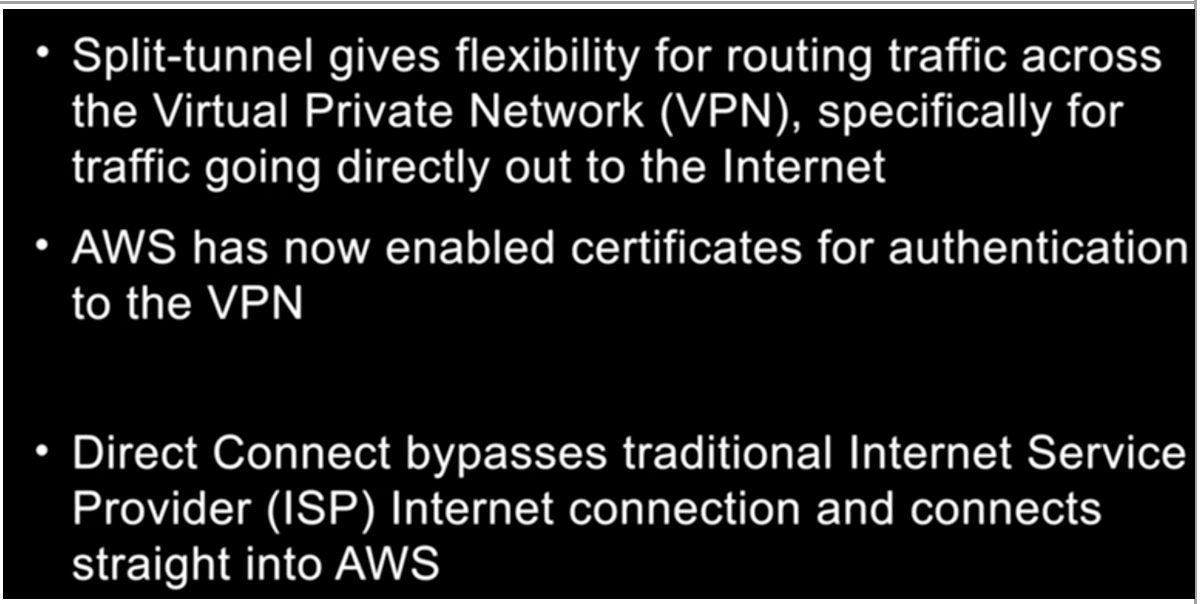
Create VPN with VPC
https://www.youtube.com/watch?v=pf0J6oPCIbE

http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpn-connections.html

https://docs.aws.amazon.com/whitepapers/latest/aws-vpc-connectivity-options/software-vpn-network-to-amazon.html

Connect on-premise data-center to AWS VPC

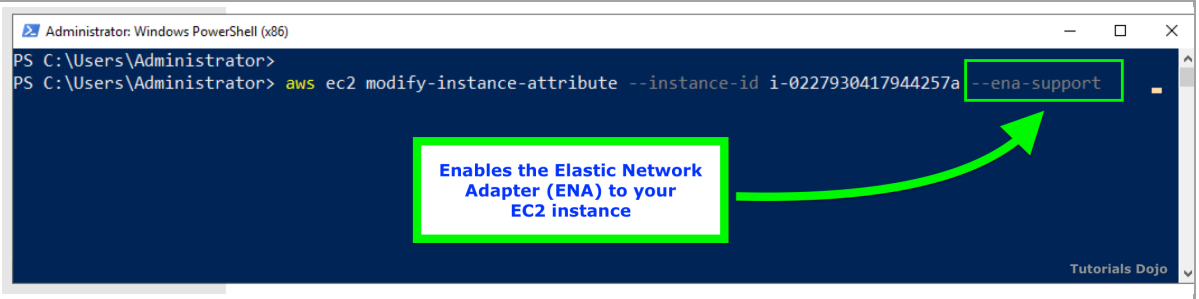
https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_VPN.html



Elastic Network Adaptor (ENA)

- Enhanced networking uses single root I/O virtualization (SR-IOV) to provide high-performance networking capabilities on supported instance types.
- SR-IOV is a method of device virtualization that provides higher I/O performance and lower CPU utilization when compared to traditional virtualized network interfaces.

- Enhanced networking provides higher bandwidth, higher packet per second (PPS) performance, and consistently lower inter-instance latencies.
- There is no additional charge for using enhanced networking.



- Amazon EC2 provides enhanced networking capabilities through the Elastic Network Adapter (ENA).
- It supports network speeds of up to 100 Gbps for supported instance types.
- Elastic Network Adapters (ENAs) provide traditional IP networking features that are required to support VPC networking.

- An Elastic Fabric Adapter (EFA) is simply an Elastic Network Adapter (ENA) with added capabilities.
- It provides all of the functionality of an ENA, with additional OS-bypass functionality.
- OS-bypass is an access model that allows HPC and machine learning applications to communicate directly with the network interface hardware to provide low-latency, reliable transport functionality.

The OS-bypass capabilities of EFAs are not supported on Windows instances.
If you attach an EFA to a Windows instance, the instance functions as an Elastic Network Adapter, without the added EFA capabilities.

References:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/enhanced-networking.html

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/efa.html

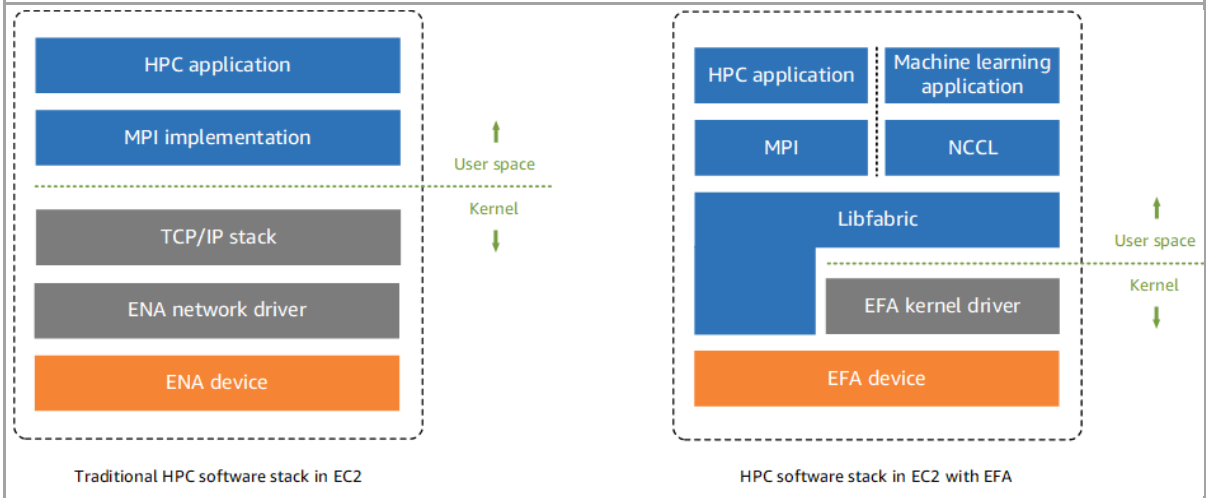
Elastic Fabric Adaptor (EFA)

- **An Elastic Fabric Adapter (EFA)** is a network device that you can attach to your Amazon EC2 instance to accelerate High Performance Computing (HPC) and machine learning applications.
- EFA enables you to achieve the application performance of an on-premises HPC cluster, with the scalability, flexibility, and elasticity provided by the AWS Cloud.

- EFA provides lower and more consistent latency and higher throughput than the TCP transport traditionally used in cloud-based HPC systems.
- It enhances the performance of inter-instance communication that is critical for scaling HPC and machine learning applications.
- It is optimized to work on the existing AWS network infrastructure and it can scale depending on application requirements.

- EFA integrates with

- Libfabric 1.9.0 and it supports Open MPI 4.0.2 and Intel MPI 2019 Update 6 for HPC applications,
- and Nvidia Collective Communications Library (NCCL) for machine learning applications.
- MPI = Message Parsing Interaface



- The OS-bypass capabilities of **EFAs are not supported on Windows instances.**
- If you attach an EFA to a Windows instance, the instance functions as an Elastic Network Adapter, without the added EFA capabilities.

KEY Limitations

- One EFA per instance,
- EFA OS-Bypass is limited to single subnet., So traffic from EFA can't be send from subnet to another subnet.
- EFA must be member of security group that allows all inbound and outbound traffic to and from the security group itself

- **Elastic Network Adapters (ENAs)**
  - provide traditional IP networking features that are required to support VPC networking.
- **EFAs provide all of the same traditional IP networking features as ENAs**,
  - and they also support OS-bypass capabilities.
  - OS-bypass enables HPC and machine learning applications to bypass the operating system kernel and to communicate directly with the EFA device.

References:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/efa.html

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/enhanced-networking-ena

Check out this Elastic Fabric Adapter (EFA) Cheat Sheet:

https://tutorialsdojo.com/aws-cheat-sheet-elastic-fabric-adapter-efa/

Multicast Networking - Support

- **Multicast is a network capability that allows one-to-many distribution of data.**
- **With multicasting, one or more sources can transmit network packets to subscribers that typically reside within a multicast group.**
- **However, take note that Amazon VPC does not support multicast or broadcast networking.**
- **You can use an overlay multicast in order to migrate the legacy application.**
- **An overlay multicast is a method of building IP level multicast across a network fabric supporting unicast IP routing, such as Amazon Virtual Private Cloud (Amazon VPC).**

Reference:

https://aws.amazon.com/articles/overlay-multicast-in-amazon-virtual-private-cloud

Check out this Amazon VPC Cheat Sheet:

https://tutorialsdojo.com/aws-cheat-sheet-amazon-vpc/

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

https://tutorialsdojo.com/aws-cheat-sheet-aws-certified-solutions-architect-associate/

Forward Proxy (Squid server) - Setup

A forward proxy server acts as an intermediary for requests from internal users and servers, often caching content to speed up subsequent requests.
Companies usually implement proxy solutions to provide URL and web content filtering, IDS/IPS, data loss prevention, monitoring, and advanced threat protection.
AWS customers often use a VPN or AWS Direct Connect connection to leverage existing corporate proxy server infrastructure, or build a forward proxy farm on AWS using software such as Squid proxy servers with internal Elastic Load Balancing (ELB).

References:

https://docs.aws.amazon.com/cli/latest/userguide/cli-http-proxy.html

https://www.ibm.com/support/knowledgecenter/en/SSAW57_8.5.5/com.ibm.websphere.nd.multiplatform.doc/ae/tjpx_setupps.html