VPC & Direct Connect

...

AWS Certified Solutions Architect– Professional (SAP-C01) - Study notes - Sep'2019

AWS VPC

/28 is minimum supports 16 IP addresses, /16 supports 65536

Concepts

Amazon VPC is the networking layer for EC2

A VPC is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS Cloud. You can launch your AWS resources, such as EC2 instances, into your VPC. You can specify an IP address range for the VPC, add subnets, associate security groups, and configure route tables.

A subnet is a range of IP addresses in your VPC. You can launch AWS resources into a specified subnet. Use a public subnet for resources that must be connected to the internet, and a private subnet for resources that won't be connected to the internet. Security groups control inbound and outbound traffic for your instances, and network ACLs control inbound and outbound traffic for your subnets.

Benefits of VPC as against EC2-classic:

- Assign static private IPv4 addresses to your instances that persist across starts and stops
- Optionally associate an IPv6 CIDR block to your VPC and assign IPv6 addresses to your instances
- Assign multiple IP addresses to your instances
- Define network interfaces, and attach one or more network interfaces to your instances
- Change security group membership for your instances while they're running
- Control the outbound traffic from your instances (egress filtering) in addition to controlling the inbound traffic to them (ingress filtering)
- Add an additional layer of access control to your instances in the form of network access control lists (ACL)
- Run your instances on single-tenant hardware

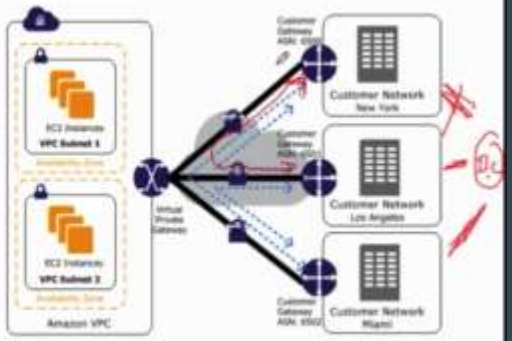
Route propagation b/w AWS subnets and On-prem subnets

- You need to update your vpn-only subnets' route table(s) to point at the VGW for subnets that are on the other side of the VPN connection

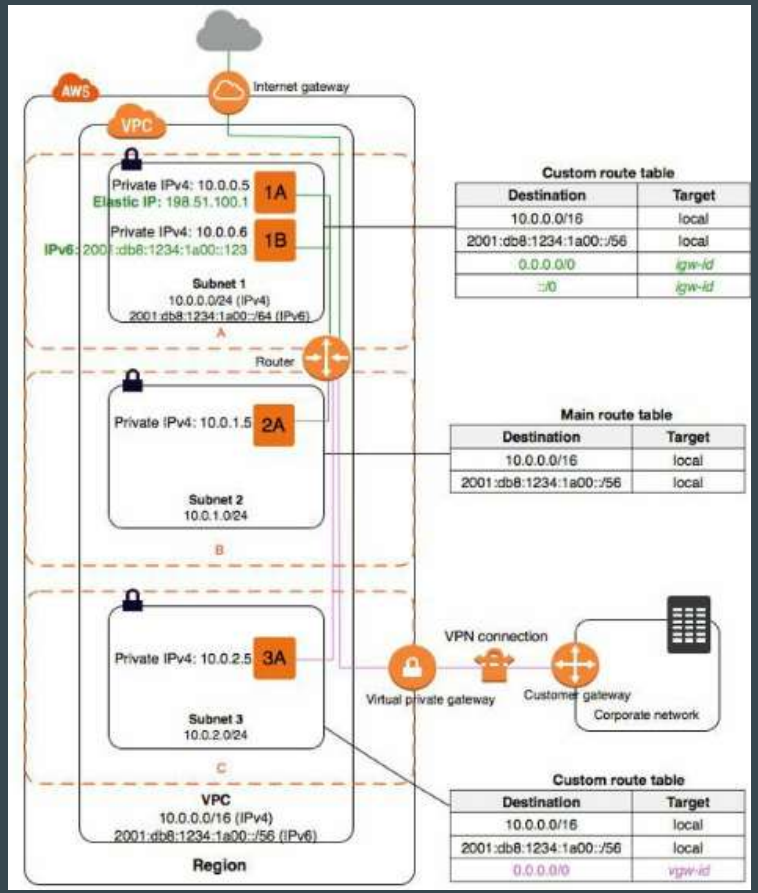
So if in a scenario like this everything was fine between Customer Gateway and VGW what could be missing is enabling route propagation, or manually configuring routes on the Private Subnet's Route

BGP is dynamic route propagation

- VPN cloud hub can also be used to allow the branches to communicate with one another
- This can be a redundant connectivity for the branch offices to the main office or data center too
- IPSec VPN tunnels + BGP need to be used
- You are charged hourly VPN rates plus data transfer rate for data sent to your spokes



VPCs and Subnets - Concepts



➔A VPC spans all the Availability Zones in the region.

➔Each subnet must reside entirely within one Availability Zone and cannot span zones. By launching instances in separate Availability Zones, you can protect your applications from the failure of a single location.

➔By default, all VPCs and subnets must have IPv4 CIDR blocks—you can't change this behavior. You can optionally associate an IPv6 CIDR block with your VPC.

➔A public subnet is a subnet that's associated with a route table that has a route to an Internet gateway.

➔If a subnet doesn't have a route to the internet gateway, the subnet is known as a private subnet.

➔If a subnet doesn't have a route to the IGW, but has its traffic routed to a virtual private gateway for a Site-to-Site VPN connection, the subnet is known as a VPN-only subnet.

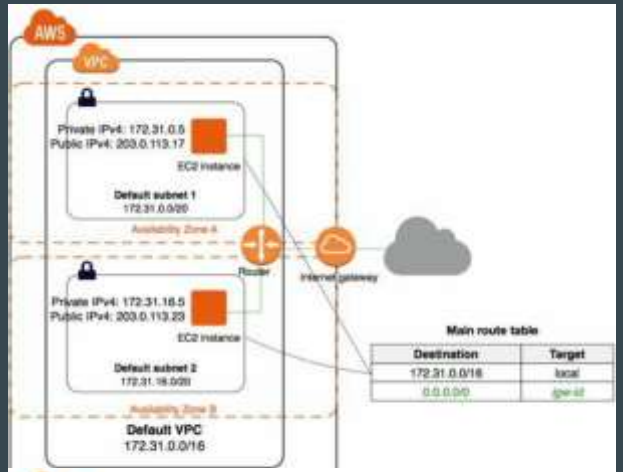
➔The first four IP addresses and the last IP address in each subnet CIDR block are not available for you to use, and cannot be assigned to an instance. e.g with CIDR block 10.0.0.0/24, the foll. five IP addresses are reserved: 10.0.0.0/1/2/3/255

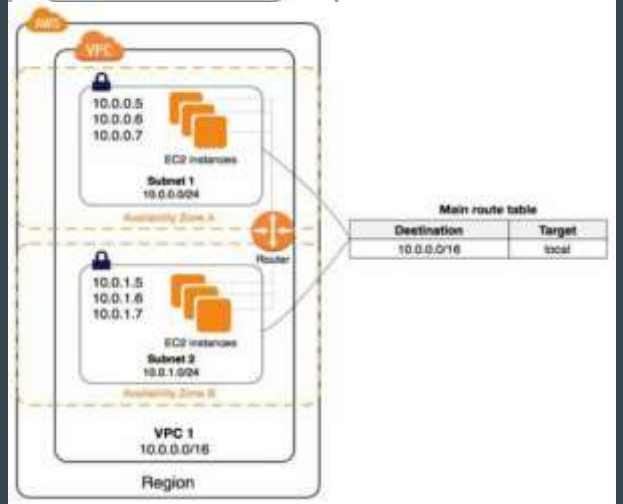
➔You can associate secondary IPv4 CIDR blocks with your VPC.

➔You cannot create a VPC peering connection between VPCs that have overlapping CIDR blocks.

➔By design, each subnet must be associated with a NACL. Every subnet that you create is automatically associated with the VPC's default NACL

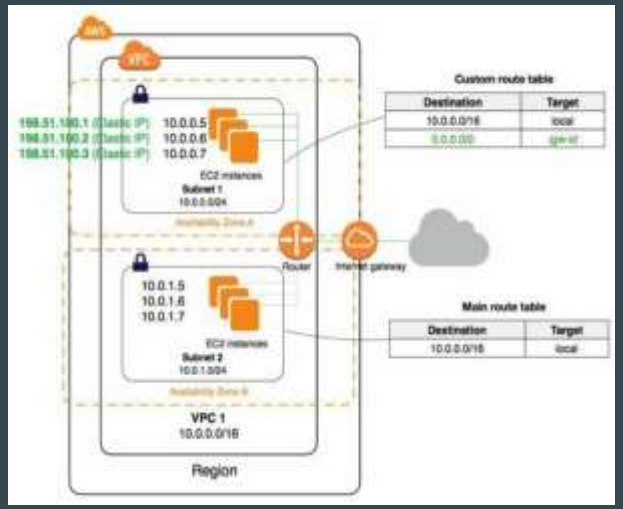
Concepts contd. - Accessing the Internet





Default VPC setup - by default public subnet with an IGW and private and public Ips

Non-Default VPC setup - by default private with a private IP only, can assign public IP if needed, no IGW, instances can talk to each other but no access to internet



Non-Default VPC w/ IGW setup - Attach an IGW for Internet Access and assign EIP to instances.. A better soln. to prevent unsolicited inbound connections from the internet, use a NAT device for IPv4 traffic.

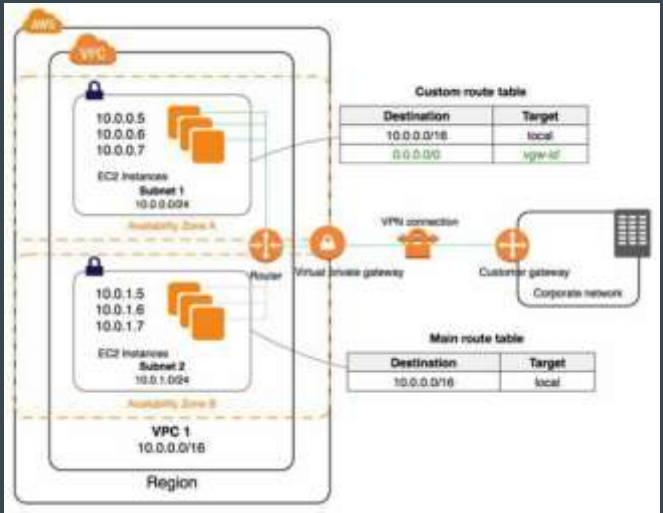
NAT maps multiple private IPv4 addresses to a single public IPv4 address. A NAT device has an Elastic IP address and is connected to the internet through an internet gateway. You can connect an instance in a private subnet to the internet through the NAT device, which routes traffic from the instance to the internet gateway, and routes any responses to the instance.

You can optionally associate an Amazon-provided IPv6 CIDR block with your VPC and assign IPv6 addresses to your instances. Instances can connect to the internet over IPv6 through an IGW. Alternatively, instances can initiate outbound connections to the internet over IPv6 using an egress-only internet gateway. IPv6 traffic is separate from IPv4 traffic; your route tables must include separate routes for IPv6 traffic.

Integrating your VPC w/

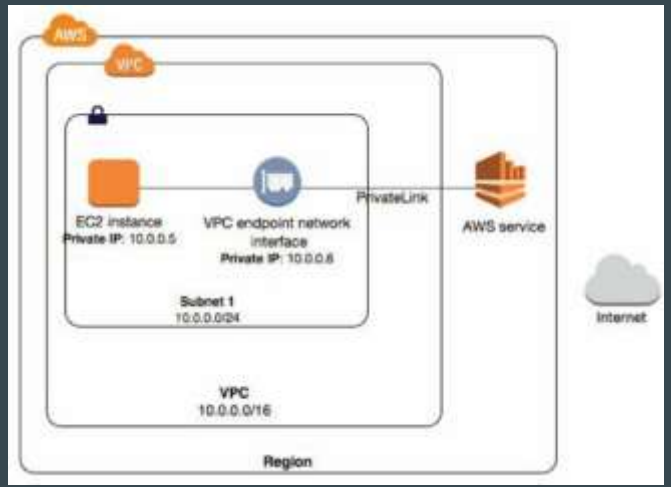
Accessing a Corporate Network

Connect your VPC to your corporate data center using an IPsec AWS Site-to-Site VPN connection, making the AWS Cloud an extension of your data center. A S2S VPN connection consists of a virtual private gateway attached to your VPC and a CG located in your data center.



Accessing Services Through AWS PrivateLink

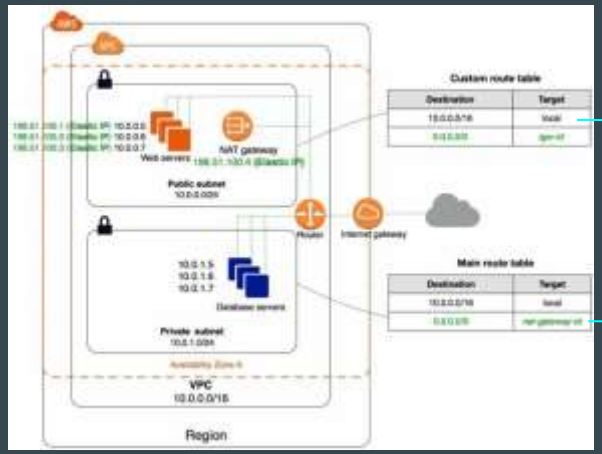
AWS PrivateLink is a highly available, scalable technology that enables you to privately connect your VPC to supported AWS services, services hosted by other AWS accounts (VPC endpoint services), and supported AWS Marketplace partner services. You do not require an internet gateway, NAT device, public IP address, AWS Direct Connect connection, or AWS Site-to-Site VPN connection to communicate with the service. Traffic between your VPC and the service does not leave the Amazon network.



Scenario: VPC with Public and Private Subnets (NAT)

Recommended for public-facing web application. A common example is a multi-tier website, with the web servers in a public subnet and the database servers in a private subnet.

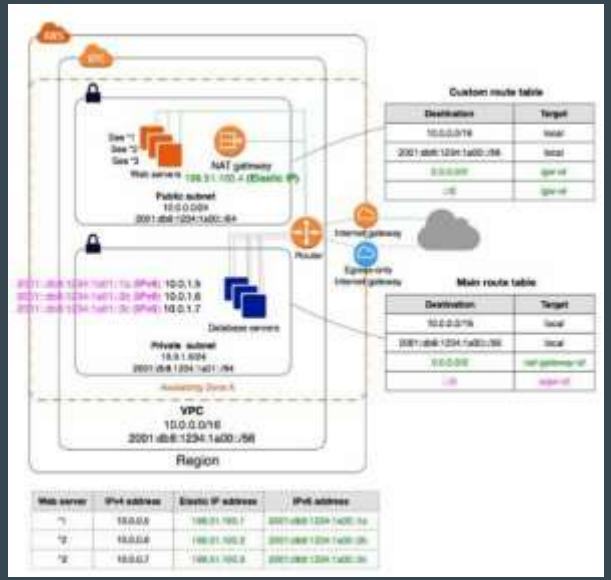
Overview (IPv4)



this entry enables the instances in the VPC to communicate with each other.

sends all other subnet traffic to the NAT gateway

Overview (IPv6)



Configuration details:

- A VPC with a size /16 IPv4 CIDR block (65,536 private IPs).
- A public subnet with a size /24 IPv4 CIDR block (256 private IPs)
- A private subnet with a size /24 IPv4 CIDR block
- An IGW. This connects the VPC to the Internet and to other AWS
- Instances with private IPv4 addresses in the subnet range
- Instances in the public subnet with Elastic IPv4 (public) addresses
- A NAT gateway with its own Elastic IPv4 address.
- A custom route table associated with the public subnet.
- The main route table associated with the private subnet.

Configuration details: In addition to the components listed above

- A size /56 IPv6 CIDR block associated with the VPC. Amazon automatically assigns the CIDR; you cannot choose the range yourself.
- A size /64 IPv6 CIDR block associated with the public subnet . You can choose the range for your subnet from the range allocated to the VPC. You cannot choose the size of the VPC IPv6 CIDR block.
- A size /64 IPv6 CIDR block associated with the private subnet.
- IPv6 addresses assigned to the instances from the subnet range
- An egress-only Internet gateway.
- A custom route table associated with the public subnet.
- The main route table associated with the private subnet.

Scenario: VPC with Public and Private Subnets (NAT) - Security groups IPv4

WebServerSG

Inbound			
Source	**Protocol**	**Port Range**	**Comments**
0.0.0.0/0	TCP	80	Allow inbound HTTP access to the web servers from any IPv4 address.
0.0.0.0/0	TCP	443	Allow inbound HTTPS access to the web servers from any IPv4 address.
Your client network's range	TCP	22	Allow inbound SSH access to Linux instances
Your client network's range	TCP	3389	Allow inbound RDP access to Windows instances
Outbound			
The ID of your DBServerSG security group	TCP	1433	Allow outbound Microsoft SQL Server access to the database servers assigned to the DBServerSG security group.
The ID of your DBServerSG security group	TCP	3306	Allow outbound MySQL access to the database servers assigned to the DBServerSG security group.
0.0.0.0/0	TCP	80	Allow outbound HTTP access to any IPv4 address.
0.0.0.0/0	TCP	443	Allow outbound HTTPS access to any IPv4 address.

DBServerSG

Inbound			
Source	**Protocol**	**Port Range**	**Comments**
ID of WebServerSG security group	TCP	1433	Allow inbound Microsoft SQL Server access from the web servers
ID of WebServerSG security group	TCP	3306	Allow inbound MySQL Server access from the web servers
Outbound			
Destination	**Protocol**	**Port Range**	**Comments**
0.0.0.0/0	TCP	80	Allow outbound HTTP access to the Internet over IPv4 e.g. for software updates
0.0.0.0/0	TCP	443	Allow outbound HTTPS access to the Internet over IPv4 e.g. for software updates

Scenario: VPC with Public and Private Subnets (NAT) - Security groups IPv6

If you associate an IPv6 CIDR block with your VPC and subnets, you must add separate rules to WebServerSG and DBServerSG security groups to control inbound and outbound IPv6 traffic for your instances.

Inbound Following are IPv6-specific rules for the WebServerSG security group (which are in addition to the rules listed above)

WebServerSG

Source	Protocol	Port Range	Comments
::/0	TCP	80	Allow inbound HTTP access to the web servers from any IPv6 address.
::/0	TCP	443	Allow inbound HTTPS access to the web servers from any IPv6 address.
IPv6 address range of your network	TCP	22	(Linux instances) Allow inbound SSH access over IPv6 from your network.
IPv6 address range of your network	TCP	3389	(Windows instances) Allow inbound RDP access over IPv6 from your network

Outbound

Destination	Protocol	Port Range	Comments
::/0	TCP	HTTP	Allow outbound HTTP access to any IPv6 address.
::/0	TCP	HTTPS	Allow outbound HTTPS access to any IPv6 address.

Following are the IPv6-specific rules for the DBServerSG security group (which are in addition to the rules listed above).

Outbound

DBServerSG

Destination	Protocol	Port Range	Comments
::/0	TCP	80	Allow outbound HTTP access to any IPv6 address.
::/0	TCP	443	Allow outbound HTTPS access to any IPv6 address.

Security Group & Network ACL

Security Groups - securing EC2 instances
Network ACL - securing subnets

Security Groups

You can specify allow rules, but not deny rules.

When you create a security group, it has no inbound rules.

By default, a security group includes an outbound rule that allows all outbound traffic.

Security groups are stateful — if you send a request from your instance, the response traffic for that request is allowed to flow in regardless of inbound security group rules. Responses to allowed inbound traffic are allowed to flow out, regardless of outbound rules.

Instances associated with a security group can't talk to each other unless you add rules allowing it (exception: the default security group has these rules by default).

Security groups are associated with network interfaces.

Network ACL

Your VPC automatically comes with a modifiable default NACL. By default, it allows all inbound and outbound IPv4 and if applicable, IPv6 traffic

You can create a custom NACL and associate it with a subnet. By default, each custom NACL denies all inbound and outbound traffic until you add rules.

Each subnet in your VPC must be associated with a NACL. If you don't, the subnet is automatically associated with the default NACL.

You can associate a NACL with multiple subnets; however, a subnet can be associated with only one NACL at a time.

A NACL contains a numbered list of rules that we evaluate in order

A NACL has separate inbound and outbound rules, and each rule can either allow or deny traffic.

NACLs are stateless; responses to allowed inbound traffic are subject to the rules for outbound traffic (and vice versa).

VPC Networking components - Elastic Network Interfaces

An elastic network interface is a virtual network interface that can include the following attributes:

1. a primary private IPv4 address
2. one or more secondary private IPv4 addresses
3. one Elastic IP address per private IPv4 address
4. one public IPv4 address, which can be auto-assigned to the network interface for eth0 when you launch an instance
5. one or more IPv6 addresses
6. one or more security groups
7. a MAC address
8. a source/destination check flag
9. a description

You can create a network interface, attach it to an instance, detach it from an instance, and attach it to another instance.
Each instance in your VPC has a default network interface (the primary network interface) that is assigned a private IPv4 address from the IPv4 address range of your VPC. You cannot detach a primary network interface from an instance. You can create and attach an additional network interface to any instance in your VPC.

What is an Elastic IP Address?

An EIP address is a public static IPv4 address which is reachable from the internet.

To use an EIP, you first allocate one to your account, and then associate it with your instance or a network interface.

When you associate an EIP with an instance or its primary network interface, the instance's public IPv4 address (if it had one) is released back into AMZ's pool of public IPv4 addresses.

When you associate an EIP with an instance that previously had a public IPv4 address, the public DNS hostname of the instance changes to match the EIP.

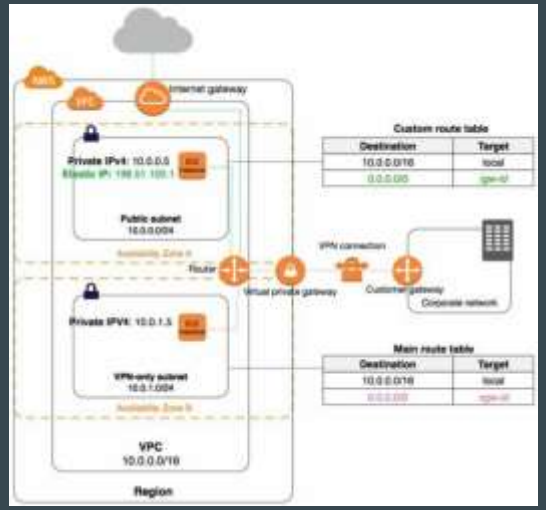
AWS resolves a public DNS hostname to the public IPv4 address or the EIP outside the network of the instance, and to the private IPv4 address of the instance from within the network of the instance.

To ensure efficient use, we impose a small hourly charge when they aren't associated with a running instance, or with a stopped or unattached ENI

An EIP is accessed through the IGW of a VPC. If you have set up an AWS S2S VPN connection between your VPC and your network, the VPN traffic traverses a virtual private gateway, not an IGW, and therefore cannot access the Elastic IP address.

VPC Networking components - Route Tables

- Your VPC has an implicit router.
- Your VPC comes with a main route table that you can modify.
- You can create add. custom route tables for your VPC.
- Each subnet must be associated with a route table(default main unless associated with a custom route table)
- You cannot delete the main table, but you can replace the main route table with a custom table that you've created
- Each route specifies a destination CIDR and a target.
- CIDR blocks for IPv4 and IPv6 are treated separately.
- Every route table contains a local route for communication within the VPC over IPv4.
- When you add an IGW, an egress-only Internet gateway, a virtual private gateway, a NAT device, a peering connection, or a VPC endpoint in your VPC, you must update the route table for any subnet that uses these gateways or connections.
- There is a limit on the number of route tables you can create per VPC, and the number of routes you can add per route table.



Custom Route tables: e.g.The main route table came with the VPC, and it also has a route for the VPN-only subnet. A custom route table is associated with the public subnet. If you create a new subnet, it's automatically associated with the main route table, which routes its traffic to the virtual private gateway.

Subnets can be implicitly or explicitly associated with the main route table.

VPC Networking components - Route Tables contd.

Route Tables for an Internet Gateway:
Add a route with a destination of 0.0.0.0/0 for IPv4 traffic or ::/0 for IPv6 traffic, and a target of the Internet gateway ID (igw-xxxxxxxx).

Route Tables for a NAT Device:
Add a route for the private subnet that routes IPv4 Internet traffic (0.0.0.0/0) to the NAT device.

Route Tables for a Virtual Private Gateway:
Add a route with the destination of your network and a target of the virtual private gateway (vgw-xxxxxxxx). You can then create and configure your Site-to-Site VPN connection.

Route Tables for a VPC Peering Connection:
Add a route to one or more of your VPC route tables that points to the VPC peering connection to access all or part of the CIDR block of the other VPC in the peering connection. Similarly, the owner of the other VPC must add a route to their VPC route table to route traffic back to your VPC.

VPC A →

Destination	Target
10.0.0.0/16	Local
172.31.0.0/16	pcx-1a2b1a2b
2001:db8:5678:2b00::/56	**pcx-1a2b1a2b**

VPC B →

Destination	Target
172.31.0.0/16	Local
10.0.0.0/16	pcx-1a2b1a2b
2001:db8:1234:1a00::/56	**pcx-1a2b1a2b**

Route Tables for a VPC Endpoint:
When you create an endpoint, you specify the route tables in your VPC that are used by the endpoint. A route is automatically added to each of the route tables with a destination that specifies the prefix list ID of the service (pl-xxxxxxxx), and a target with the endpoint ID (vpce-xxxxxxxx).

Route Tables for an Egress-Only IGW:
Add a route for the private subnet that routes IPv6 Internet traffic (::/0) to the egress-only IGW

Route Tables for Transit Gateways:
When you attach a VPC to a *tgw*, you need to add a route for traffic to route through the *tgw*

What is longest prefix match?

Longest prefix match (also called **Maximum prefix length match**) refers to an algorithm used by routers in Internet Protocol (IP) networking to select an entry from a forwarding table. [1]

Because each entry in a forwarding table may specify a sub-network, one destination address may match more than one forwarding table entry. The most specific of the matching table entries — the one with the longest subnet mask — is called the longest prefix match. It is called this because it is also the entry where the largest number of leading address bits of the destination address match those in the table entry. [2]

For example, consider this IPv4 forwarding table (CIDR notation is used):

192.168.20.16/28
192.168.0.0/16

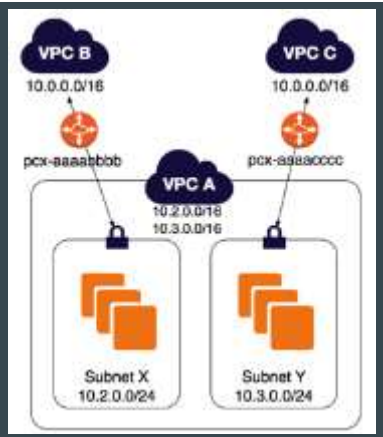
When the address 192.168.20.19 needs to be looked up, both entries in the forwarding table "match". That is, both entries contain the looked up address. In this case, the longest prefix of the candidate routes is 192.168.20.16/28, since its subnet mask (/28) is longer than the other entry's mask (/16), making the route more specific.

VPC Networking components - Configurations with Specific Routes

https://docs.aws.amazon.com/vpc/latest/peering/peering-configurations-partial-access.html#one-to-two-vpcs-lpm

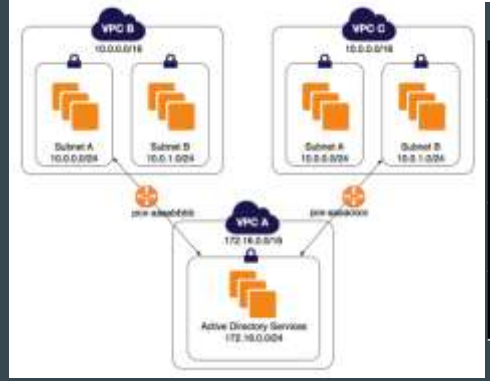
In these examples, a central VPC is peered to two or more VPCs that have overlapping CIDR blocks.

Two VPCs Peered to Two Subnets in One VPC with multiple CIDR blocks



Rt Table	Destination	Target
Subnet X	10.2.0.0/16	Local
	10.3.0.0/16	Local
	10.0.0.0/16	pcx-aaaabbbb
Subnet Y	10.2.0.0/16	Local
	10.3.0.0/16	Local
	10.0.0.0/16	pcx-aaaacccc
VPC B	10.0.0.0/16	Local
	10.2.0.0/24	pcx-aaaabbbb
VPC C	10.0.0.0/16	Local
	10.3.0.0/24	pcx-aaaacccc

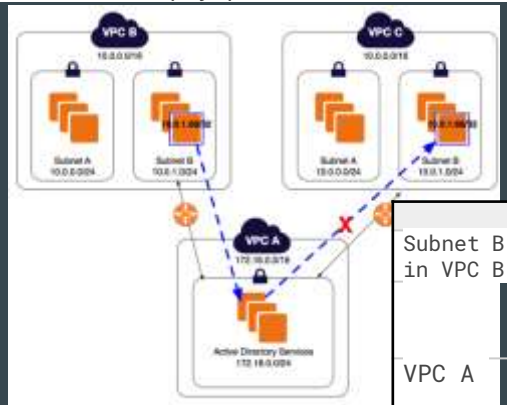
One VPC Peered to Specific Subnets in Two VPCs



Route Table	Destination	Target
VPC A	172.16.0.0/16	Local
	10.0.0.0/24	pcx-aaaabbbb
	10.0.1.0/24	pcx-aaaacccc
Subnet A	10.0.0.0/16	Local
	172.16.0.0/24	pcx-aaaabbbb
Subnet B	10.0.0.0/16	Local
	172.16.0.0/24	pcx-aaaacccc

Routing for Response Traffic

If you have a VPC peered with multiple VPCs that have overlapping or matching CIDR blocks, ensure that your route tables are configured to avoid sending response traffic from your VPC to the incorrect VPC. AWS currently does not support unicast reverse path forwarding in VPC peering connections that checks the source IP of packets and routes reply packets back to the source.



	Destn	Target
Subnet B in VPC B	10.0.0.0/16	Local
	172.16.0.0/24	pcx-aaaabbbb
VPC A	10.0.0.0/16	pcx-aaaacccc

You can create a route to a specific IP address in 'cc : A to ensure that traffic routed back to the correct server (the route table uses longest prefix match to prioritize the routes): following either soln will work

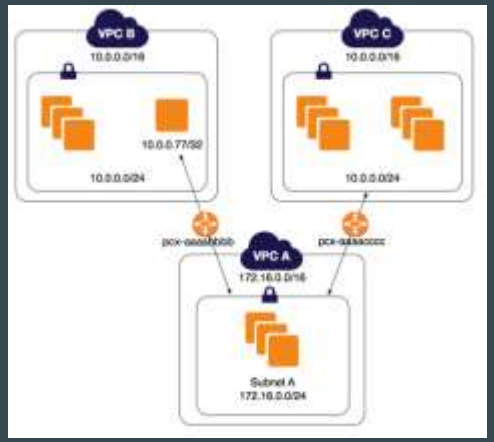
Destination	Target
172.16.0.0/16	Local
10.0.1.0/24	pcx-aaaabbbb
10.0.0.0/24	pcx-aaaacccc

Destn	Target
172.16.0.0/16	Local
10.0.1.66/32	pcx-aaaabbbb
10.0.0.0/16	pcx-aaaacccc

VPC Networking components - Configurations with Specific Routes

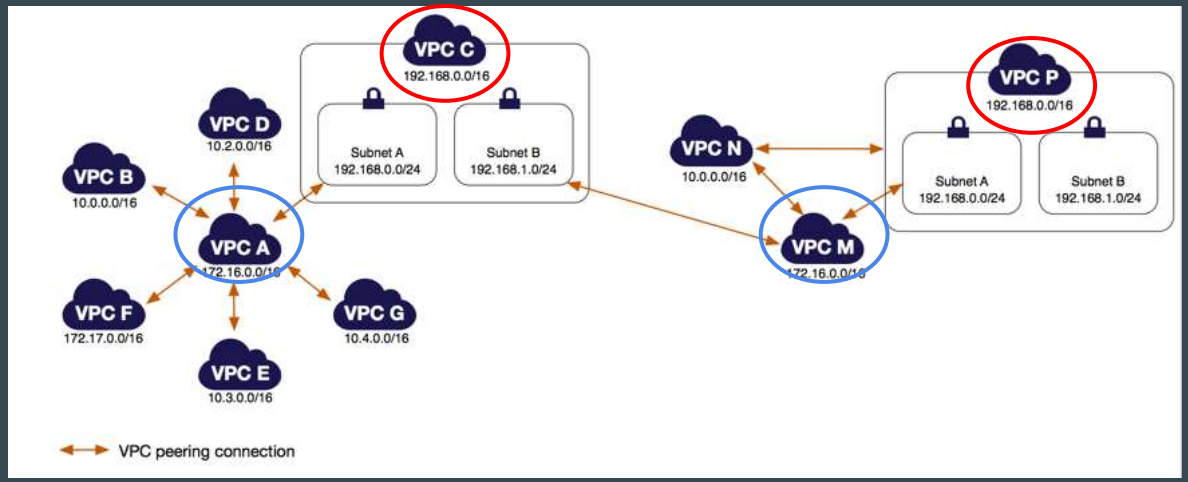
https://docs.aws.amazon.com/vpc/latest/peering/peering-configurations-partial-access.html#one-to-two-vpcs-lpm

In these examples, a central VPC is peered to two or more VPCs that have overlapping CIDR blocks.



Rt Table	Destination	Target
VPC A	172.16.0.0/16	Local
	10.0.0.77/32	pcx-aaaabbbb
	10.0.0.0/16	pcx-aaaacccc
VPC B	10.0.0.0/16	Local
	172.16.0.0/16	pcx-aaaabbbb
VPC C	10.0.0.0/16	Local
	172.16.0.0/16	pcx-aaaacccc

Important
If an instance other than 10.0.0.77/32 in VPC B sends traffic to VPC A, the response traffic may be routed to VPC C instead of VPC B.



VPC A and VPC M have overlapping CIDR blocks. This means that peering traffic between VPC A and VPC C is limited to a specific subnet (subnet A) in VPC C. This is to ensure that if VPC C receives a request from VPC A or VPC M, it sends the response traffic to the correct VPC. AWS currently does not support unicast reverse path forwarding in VPC peering connections that checks the source IP of packets and routes reply packets back to the source.

VPC Networking components - NAT

Use a NAT device to enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating connections with the instances. 2 kinds – a NAT gateway or a NAT instance. AMZ recommends NGW, as they provide better availability and bandwidth over instances. NGW is also a managed service

NAT Gateways

- NAT gateway hr. usage and data rates apply. EC2 charges for data transfer also apply.
- NGWs are not supported for IPv6 traffic—use an egress-only internet gateway instead.
- To create a NAT gateway, you must specify the public subnet in which the NGW should reside.
- Also specify an EIP to associate with the NGW
- To create an Availability Zone-independent architecture, create a Ngw in each AZ and configure your routing to ensure that resources use the NGW in the same AZ

Best Practice When Sending Traffic to Amazon S3 or DynamoDB in the Same Region:

To avoid data processing charges for NAT gateways when accessing Amazon S3 and DynamoDB that are in the same Region, set up a gateway endpoint and route the traffic through the gateway endpoint instead of the NAT gateway. There are no charges for using a gateway endpoint.

Attribute	NAT gateway	NAT instance
Availability	Highly available. NAT gateways in each AZ are implemented with redundancy. Create a NAT gateway in each AZ 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	Managed by AWS. You do not need to perform any maintenance.	Managed 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 processed	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
Public IP addresses	Choose the Elastic IP address to associate with a NAT gateway at creation.	Use an EIP or a public IP with a NAT instance. You can change the public IP address at any time by associating a new EIP
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 NGW. Associate security groups with your resources behind the NGW to control i/ and o/ traffic.	Associate with your NAT instance and the resources behind your NAT instance to control inbound and outbound traffic.
Network ACLs	Use a NACL to control the traffic to and from the subnet in which your NGW 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.
Port forwarding	Not supported.	Manually customize config to support port forwarding.
Bastion servers	Not supported.	Use as a bastion server.
Traffic metrics	View CloudWatch metrics for the NAT gateway.	View CloudWatch metrics for the instance.
Timeout behavior	When a connection times out, a NAT gateway returns an RST packet ,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 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.

VPC Networking components - DHCP Options Sets

DHCP provides a standard for passing configuration information to hosts on a TCP/IP network. The options field of a DHCP message contains the configuration parameters. Some of those parameters are the domain name, domain name server, and the netbios-node-type.

DHCP Option Name	Description
domain-name-servers	The IP addresses of up to four domain name servers, or AmazonProvidedDNS. The default DHCP option set specifies AmazonProvidedDNS. ***If you want your instance to receive a custom DNS hostname as specified in domain-name, you must set domain-name-servers to a custom DNS server.***
domain-name	If using AmazonProvidedDNS e.g. ***ip-private-ipv4-address.region.compute.internal or* ec2-public-ipv4-address.region.compute.amazonaws.com.** Otherwise, specify a domain name (for example, example.com). This value is used to complete unqualified DNS hostnames.
ntp-servers	The IP addresses of up to four Network Time Protocol (NTP) servers.
netbios-name-servers	The IP addresses of up to four NetBIOS name servers.
netbios-node-type	The NetBIOS node type (1, 2, 4, or 8). We recommend that you specify 2 (point-to-point, or P-node).

VPC Networking components - DNS

Attribute	Description
enableDnsHostnames	Indicates whether instances with public IP addresses get corresponding public DNS hostnames.
enableDnsSupport	Indicates whether the DNS resolution is supported.

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

- Instances with a public IP address receive corresponding public DNS hostnames.
- 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:

- Instances with a public IP address do not receive corresponding public DNS hostnames.
- The Amazon-provided DNS server cannot resolve Amazon-provided private DNS hostnames.
- Instances receive custom private DNS hostnames if there is a custom domain name in the DHCP options set. If you are not using the Amazon-provided DNS, your custom domain name servers must resolve the hostname as appropriate.

If you use custom DNS domain names defined in a private hosted zone in Route 53, or use private DNS with interface VPC endpoints (AWS PrivateLink), you must set the enableDnsHostnames and enableDnsSupport attributes to true.

Using Private Hosted Zones:

If you want to access the resources in your VPC using custom DNS domain names, such as example.com, instead of using private IPv4 addresses or AWS-provided private DNS hostnames, you can create a private hosted zone in Route 53. A private hosted zone is a container that holds information about how you want to route traffic for a domain and its subdomains within one or more VPCs without exposing your resources to the Internet.

VPC Networking components - VPC Endpoints

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.

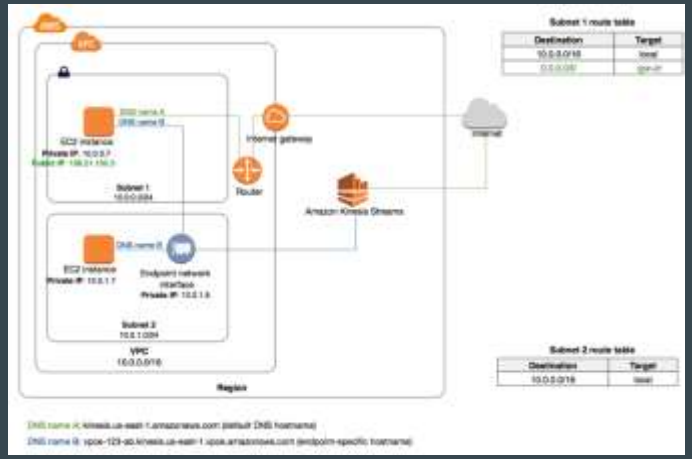
Endpoints are virtual devices. They are horizontally scaled, redundant, and highly available VPC components that allow communication between instances in your VPC and services without imposing availability risks or bandwidth constraints on your network traffic.

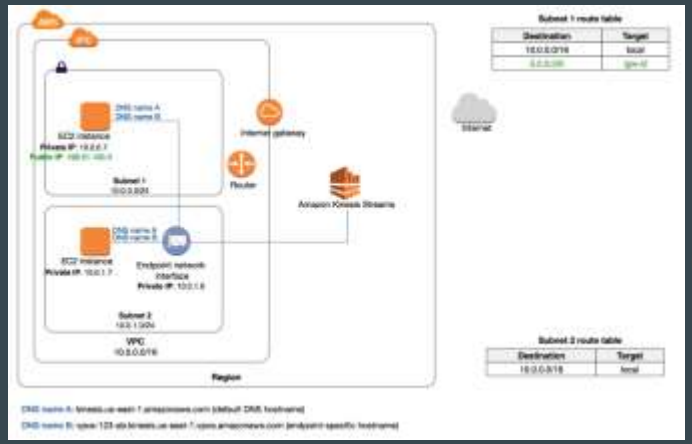
There are two types of VPC endpoints: **interface endpoints** and **gateway endpoints**.

An interface endpoint is an elastic network interface with a private IP address from the IP address range of your subnet that serves as an entry point for traffic destined to a supported service. These services include some AWS services, services hosted by other AWS customers and partners in their own VPCs (referred to as endpoint services), and supported AWS Marketplace partner services.

A gateway endpoint is a gateway that you specify as a target for a route in your route table for traffic destined to a supported AWS service. The following AWS services are supported: S3 & DynamoDB

VPC Networking components - Interface Endpoints - Private DNS





When you create an interface endpoint, we generate endpoint-specific DNS hostnames that you can use to communicate with the service.

For AWS services and AWS Marketplace partner services, private DNS (enabled by default) associates a private hosted zone with your VPC. The hosted zone contains a record set for the default DNS name for the service that resolves to the private IP addresses of the endpoint network interfaces in your VPC.

This enables you to make requests to the service using its default DNS hostname instead of the endpoint-specific DNS hostnames.

eXAMPLE 1: You have **not enabled private DNS** for the interface endpoint. Instances in either subnet can communicate with Kinesis using an endpoint-specific DNS hostname (DNS name B). Instance in subnet 1 can communicate with Kinesis over public IP address in the AWS Region using the default DNS name for the service (DNS name A).

eXAMPLE 2: you **have enabled private DNS** for the endpoint. Instances in either subnet can communicate with Kinesis using an endpoint-specific DNS hostname (DNS name B) or the default DNS name for the service (DNS name A).

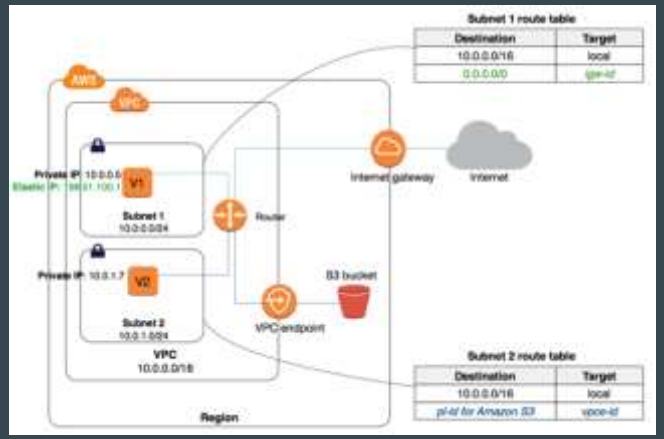
To use private DNS, you must set the following VPC attributes to true: enableDnsHostnames and enableDnsSupport.

VPC Networking components - Gateway Endpoints

To create and set up a gateway endpoint:

1. Specify the VPC in which to create the endpoint, and the service to which you're connecting.
2. Attach an *endpoint policy* to your endpoint that allows access to some or all of the service to which you're connecting.
3. Specify one or more route tables in which to create routes to the service.

In this example, instances in subnet 2 can access Amazon S3 through the gateway endpoint.



Routing for Gateway Endpoints

We use the most specific route that matches the traffic to determine how to route the traffic (longest prefix match). If you have an existing route in your route table for all internet traffic (0.0.0.0/0) that points to an IGW, the endpoint route takes precedence for all traffic destined for the service, because the IP address range for the service is more specific than 0.0.0.0/0. All other internet traffic goes to your IGW, including traffic that's destined for the service in other Regions. However, if you have existing, more specific routes to IP address ranges that point to an internet gateway or a NAT device, those routes take precedence.

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	igw-1a2b3c4d
pl-1a2b3c4d	vpce-11bb22cc

Any traffic from the subnet that's destined for that AWS service in the same Region goes to the endpoint, and does not go to the internet gateway. All other internet traffic goes to your internet gateway, including traffic that's destined for other services, and destined for the AWS service in other Regions.

Destination	Target
10.0.0.0/16	Local
54.123.165.0/24	igw-1a2b3c4d
pl-1a2b3c4d	vpce-11bb22cc

You've added a route with 54.123.165.0/24 as a destination (assume this is an IP address range currently within Amazon S3), and the internet gateway as the target. You then create an endpoint, and associate this route table with the endpoint. An endpoint route is automatically added to the route table. You then use the describe-prefix-lists command to view the IP address range for Amazon S3. The range is 54.123.160.0/19, which is less specific than the range that's pointing to your internet gateway. This means that any traffic destined for the 54.123.165.0/24 IP address range continues to use the internet gateway, and does not use the endpoint (for as long as this remains the public IP address range for Amazon S3).

VPC Networking components - Gateway Endpoints

Gateway Endpoint Limitations

To use gateway endpoints, you need to be aware of the current limitations:

- You cannot use a prefix list ID in an outbound rule in a network ACL to allow or deny outbound traffic to the service specified in an endpoint. If your network ACL rules restrict traffic, you must specify the CIDR block (IP address range) for the service instead. You can, however, use a prefix list ID in an outbound security group rule.
- Endpoints are supported within the same region only. You cannot create an endpoint between a VPC and a service in a different region.
- Endpoints support IPv4 traffic only.
- You cannot transfer an endpoint from one VPC to another, or from one service to another.
- You have a limit on the number of endpoints you can create per VPC.
- Endpoint connections cannot be extended out of a VPC. Resources on the other side of a VPN connection, VPC peering connection, AWS Direct Connect connection, or ClassicLink connection in your VPC cannot use the endpoint to communicate with resources in the endpoint service.
- You must enable DNS resolution in your VPC, or if you're using your own DNS server, ensure that DNS requests to the required service (such as Amazon S3) are resolved correctly to the IP addresses maintained by AWS.

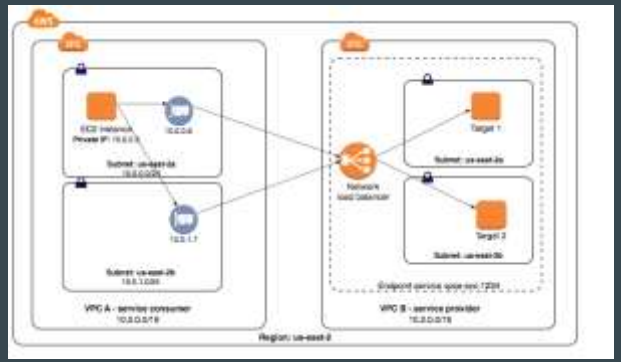
VPC Networking components - VPC Endpoint Services (AWS PrivateLink) ...creating a Service Provider

You can create your own application in your VPC and configure it as an AWS PrivateLink-powered service (referred to as an endpoint service). Other AWS principals can create a connection from their VPC to your endpoint service using an interface VPC endpoint. You are the service provider, and the AWS principals that create connections to your service are service consumers. Following are general steps to create:

1. Create a Network Load Balancer for your application in your VPC and configure it for each subnet (Availability Zone) in which the service should be available.
2. Create a VPC endpoint service configuration and specify your Network Load Balancer.

How to enable service consumers to connect to your service ?

1. Grant permissions to specific service consumers (AWS accounts, IAM users, and IAM roles) to create a connection to your endpoint service.
2. A service consumer that has been granted permissions creates an interface endpoint to your service, optionally in each Availability Zone in which you configured your service.
3. To activate the connection, accept the interface endpoint connection request. By default, connection requests must be manually accepted. However, you can configure the acceptance settings for your endpoint service so that any connection requests are automatically accepted.



In the following diagram, the owner of VPC B is the service provider, and it has configured a Network Load Balancer with targets in two different Availability Zones. The service consumer (VPC A) has created interface endpoints in the same two Availability Zones in their VPC. Requests to the service from instances in VPC A can use either interface endpoint.

VPC Networking components - VPC Connection options

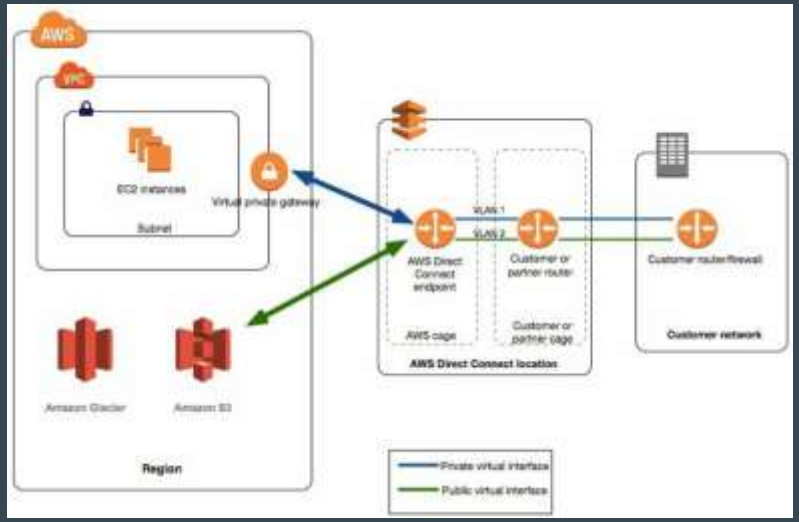
VPN connectivity option	Description
AWS Site-to-Site VPN	You can create an IPsec VPN connection between your VPC and your remote network. On the AWS side of the Site-to-Site VPN connection, a *virtual private gateway* provides two VPN endpoints (tunnels) for automatic failover. You configure your *CG* on the remote side of the Site-to-Site VPN connection.
AWS Client VPN	AWS Client VPN is a managed client-based VPN service that enables you to securely access your AWS resources in your on-premises network. With AWS Client VPN, you configure an endpoint to which your users can connect to establish a secure TLS VPN session. This enables clients to access resources in AWS or an on-premises from any location using an OpenVPN-based VPN client.
AWS VPN CloudHub	If you have more than one remote network (for example, multiple branch offices), you can create multiple AWS Site-to-Site VPN connections via your virtual private gateway to enable communication between these networks.
Third party software VPN appliance	You can create a VPN connection to your remote network by using an Amazon EC2 instance in your VPC that's running a third party software VPN appliance. AWS does not provide or maintain third party software VPN appliances; however, you can choose from a range of products provided by partners and open source communities. Find third party software VPN appliances on the AWS Marketplace.

You can also use **AWS Direct Connect** to create a dedicated private connection from a remote network to your VPC. You can combine this connection with an AWS Site-to-Site VPN to create an IPsec-encrypted connection.

AWS Direct Connect

Border Gateway Protocol >> BGP

What is AWS Direct Connect?



AWS Direct Connect links your internal network to an AWS Direct Connect location over a standard Ethernet fiber-optic cable. One end of the cable is connected to your router, the other to an AWS Direct Connect router. With this connection, you can create virtual interfaces directly to public AWS services (e.g.to S3) or to VPC, bypassing ISPs in your network path.

AWS Direct Connect Components:

- Connections: Create a connection in an AWS Direct Connect location to establish a network connection from your premises to an AWS Region.
- Virtual interfaces: Create a virtual interface to enable access to AWS services. A public virtual interface enables access to public services, such as S3. A private virtual interface enables access to your VPC.

Network Requirements:

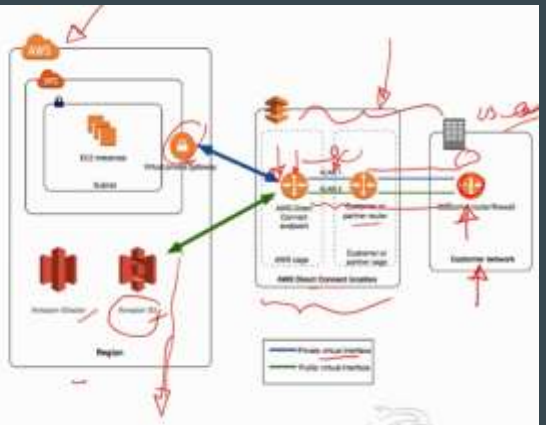
- Your network is colocated with an existing AWS Direct Connect location.
- You are working with an AWS Direct Connect partner who is a member of the APN.
- You are working with an independent service provider to connect to AWS Direct Connect.

DC locations in public Regions can access public services OR VPC in your account in any other public Region. You can therefore use a single DC connection to build multi-Region services.

To access public resources in a remote Region, set up a **public virtual interface >> establish a BGP session.** To access VPCs in a remote Region, create a DC gateway >> private virtual interface to VPCs in your account that are located in different Regions or to a transit gateway.

What is AWS Direct Connect?

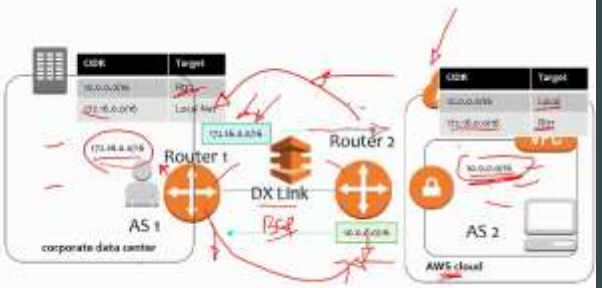
- Border Gateway Routing Protocol must be used to route between then two ends of the DX connection.
- Customer Router must be capable of 802.1Q and BGP
- You can use the connection to establish Private Virtual InterFaces (VIFs) to connect to a VPC in the region (via the VPC's VGW)
- You can establish multiple Private VIFs to multiple VPCs in a region (one per VPC)
- You can establish Public VIF to connect to any public AWS endpoints in ANY region
- Supports both IPv4 and IPv6 (requires public VIFs)
- All networking traffic remains on the AWS global network backbone, regardless of whether you access public AWS services or a VPC in another Region.
- Any data transfer out of a remote Region is billed at the remote Region data transfer rate.



So what is a VIF, 802.1Q, and what is BGP?

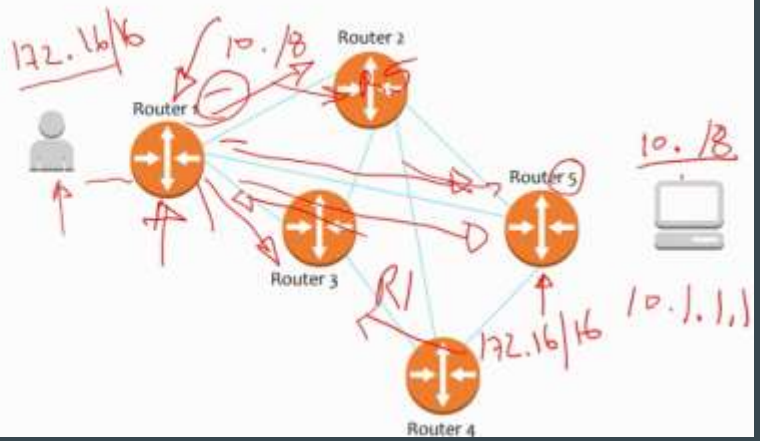
Border Gateway Protocol (BGP)

- BGP between Corporate Data Center and AWS VPC configured using a direct connect link
- BGP setup, Routing updates are propagated dynamically
- Each side builds its own route table based on these updates and BGP communities configured, if any.
- BGP is the only routing option on Direct connect links



Static vs Dynamic routing (BGP)

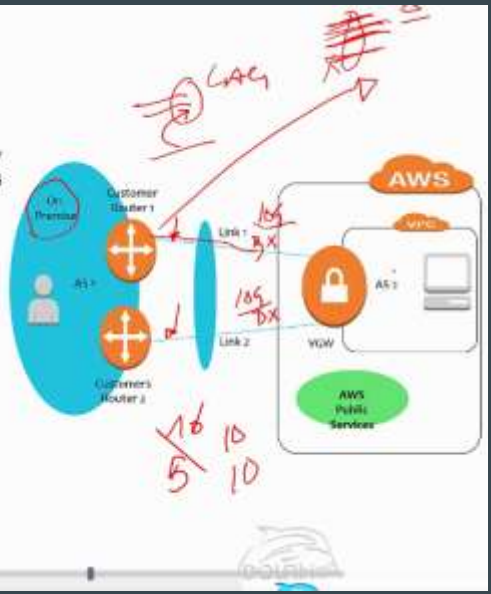
- You need routing in order for Routers to be able to forward your traffic correctly
- In simple networks, you can define the routing manually, and this is called static routing (like we do on AWS Route tables)
- When the network grows, we need a dynamic mechanism to take care of updating and exchanging routing information such that we do not have to face the administration challenges of static routing at scale (like when we enable Route Propagation on AWS route tables for VPN or Direct Connect).



- Connects different autonomous systems together
- Each Autonomous System (AS) requires an Autonomous system number (ASN)
- TCP based sessions between neighbors, on port 179
- The Main internet routing protocol today
- Can carry large number of routes, with granular route control
- Has many route/path attributes to control routing
- Has IBGP (within an AS) and EBGP (between ASs) peering types
- Transit AS capability

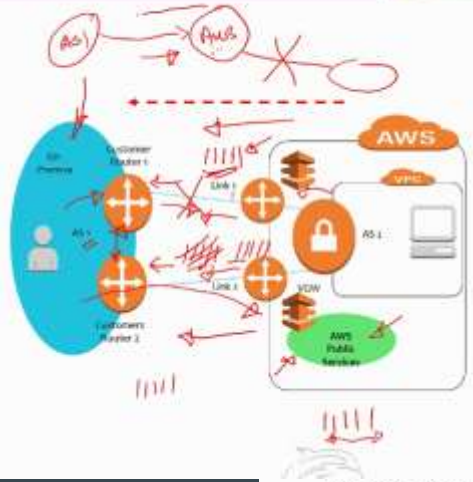
Link Aggregation group

- A link aggregation group (LAG) is a logical interface that uses the Link Aggregation Control Protocol (LACP) to aggregate multiple connections at a single AWS Direct Connect endpoint, allowing you to treat them as a single, managed connection.
  - All connections in a LAG operate in active/active mode.
- You can create a LAG from existing connections, or you can provision new connections. After you've created the LAG, you can associate existing connections (whether standalone or part of another LAG) with the LAG.
- The following rules apply:
  - All connections in the LAG must use the same bandwidth.
  - You can have a maximum of four connections in a LAG.
  - All connections in the LAG must terminate at the same AWS Direct Connect endpoint.
- All LAGs have an attribute that determines the minimum number of connections in the LAG that must be operational for the LAG itself to be operational.
  - By default, new LAGs have this attribute set to 0.
  - You can update your LAG to specify a different value—doing so means that your entire LAG becomes nonoperational if the number of operational connections falls below this threshold.
  - This attribute can be used to prevent over-utilization of the remaining connections.



AWS Direct Connect – BGP Communities in Outbound Policies – AWS Sid

- AWS Direct Connect advertises all local and remote AWS Region prefixes where available and includes on-net prefixes from other AWS non-Region points of presence (PoP) where available; for example, CloudFront and Route 53.
  - No access to non-Amazon prefixes (i.e no access to the global internet over the direct links).
- AS_PATH is used to determine the routing path, and **AWS Direct Connect is the preferred path** for traffic sourced from Amazon (Preference over HA VPN links)
- Only public ASNs are used internally for route selection.
- AWS Direct Connect advertises prefixes with a minimum path length of 3.
- When **multiple AWS Direct Connect connections** are there, the load-sharing of inbound traffic can be adjusted by advertising prefixes with **similar path** attributes.
- AWS Direct Connect advertises all public prefixes with the well-known NO_EXPORT BGP community.
  - The prefixes advertised by AWS Direct Connect must not be advertised beyond the network boundaries of your connection.



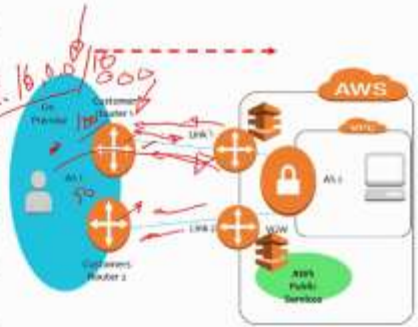
AWS Direct Connect – BGP Communities – On Premise side

Scope BGP Communities

- BGP community tags can be applied on the public prefixes that you advertise to Amazon to indicate how far to propagate your prefixes in the Amazon network
  - For the local AWS Region only, all Regions within a continent, or all public Regions.
  - If community tags are applied, prefixes are advertised to all public AWS Regions (global) by default.

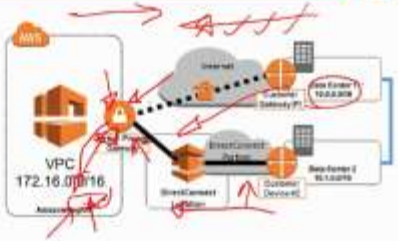
Local Preference BGP Communities

- **Local preference BGP community tags are supported for private virtual interfaces, and transit virtual interfaces.**
- Local preference BGP community tags can be used to achieve load balancing and route preference for incoming traffic(sourced from AWS).
- For each advertised prefix over a BGP session, a community tag can be applied to indicate the priority of the associated path for returning traffic.
- To load balance traffic across multiple AWS Direct Connect connections, apply the same community tag across the prefixes for the connections.
- To support failover across multiple AWS Direct Connect connections, apply a community tag with **a higher preference to the prefixes for the primary or active virtual interface.**



Route Tables and Route Priority - From AWS to On Premise

- Route tables determine where network traffic is directed.
- A route(s) must be added in the VPC route table (for the CIDRs on premise that will be reached from the VPC), the VGW will be the Target (next hop)
  - This is true for both VPN and Direct Connect VPC connectivity
- Alternatively, route propagation can be enabled for the route table to automatically propagate the on premise CIDRs (received via BGP or configured by Static routes on the VGW (VPN case) to the table.

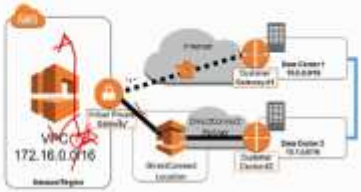


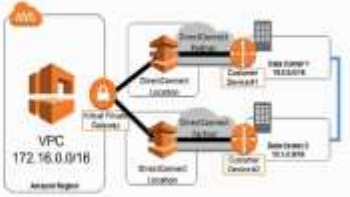
- If overlapping routes within a Site-to-Site VPN connection and longest prefix match cannot be applied, then AWS prioritize the routes as follows, from most preferred to least preferred:
  - BGP propagated routes from an AWS Direct Connect connection
  - Manually added static routes for a Site-to-Site VPN connection
  - BGP propagated routes from a Site-to-Site VPN connection

IMP: You cannot add a route in the VPC route table to point directly to a DX connection, you need to add an entry pointing to the VGW

Traffic Forwarding - From AWS to On Premise

- When a virtual private gateway (VGW) receives routing information, it uses **path selection** to determine how to route traffic to your remote network.
- First it applies **Longest prefix match** applies;
- If all routes are of equal length (subnet mask) then, the following rules apply:
  - If any propagated routes from a Site-to-Site VPN connection or AWS Direct Connect connection **overlap** with the local route for your VPC,
    - Forwarding will use the local route is most preferred even if the propagated routes are more specific.
  - If any propagated routes from a Site-to-Site VPN connection or AWS Direct Connect connection have the same destination CIDR block as other existing static routes, AWS prioritizes the static routes whose targets are an Internet gateway, a virtual private gateway, a network interface, an instance ID, a VPC peering connection, a NAT gateway, or a VPC endpoint.





- Only IP prefixes that are known to the virtual private gateway, whether through BGP advertisements or static route entry, can receive traffic from your VPC.
  - The virtual private gateway does not route any other traffic destined outside of received BGP advertisements, static route entries, or its attached VPC CIDR.

Multiple Data Center HA Network Connectivity

https://aws.amazon.com/answers/networking/aws-multiple-data-center-ha-network-connectivity/

General best practices:

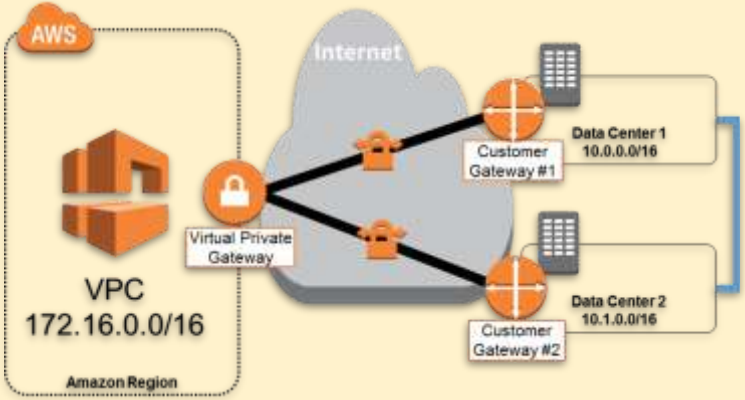
- Leverage multiple dynamically routed, rather than statically routed, connections to AWS. This will allow remote connections to failover automatically between redundant connections. Dynamic routing also enables remote connections to automatically leverage available preferred routes, if applicable, to the on-premises network.
- When selecting AWS Direct Connect network service providers, consider a dual-vendor ISP approach, if financially feasible, to ensure private network diversity.
- Leverage more specific BGP route advertisements or BGP AS-path prepending to ensure AWS uses the most efficient routes to send traffic to your remote data centers.
- Provision sufficient network capacity to ensure that the failure of one network connection does not overwhelm and degrade redundant connections.

Multiple Data Center HA Network Connectivity - 1:Redundant Active/Active VPN Connections

https://aws.amazon.com/answers/networking/aws-multiple-data-center-ha-network-connectivity/

Setup:

The best practice for making VPN connections highly available is to use redundant CGs and dynamic routing for automatic failover between AWS and customer VPN endpoints.



Configuration details:

- Setup 4 fully meshed, dynamically routed IPsec tunnels between both VGW endpoints and two CGs.
- The VGW will prefer to send 10.0.0.0/16 traffic to DC1 through CG1, and only reroute this traffic through DC2 if the connection to DC1 is down. Likewise, 10.1.0.0/16 traffic will prefer the VPN connection originating from DC2

AWS recommends using one of the following approaches for communicating these route preferences:

1. **More specific routes:** With this approach, both CG 1 and CG 2 advertise a **summary route of 10.0.0.0/15**. In addition, CG 1 advertises 10.0.0.0/16 and CG 2 advertises 10.1.0.0/16. AWS will use the more specific routes to send traffic to the appropriate data center, and will fail back to the other data center following the summarized route if the more specific route becomes temporarily unavailable.

2. **AS-path prepending:** With this approach, both CG 1 and CG 2 advertise 10.0.0.0/16 and 10.1.0.0/16. However, CG 1 uses AS-path prepending when advertising the 10.1.0.0/16 network to make this route less preferred. Likewise, CG 2 uses AS-path prepending when advertising the 10.0.0.0/16 network to make this route less preferred. AWS will use the preferred routes to send traffic to the appropriate data center, and will fail back to the other data center following the less preferred routes when necessary. If your organization already leverages AS-path prepending for influencing route preferences, then the latter approach will likely align more closely with your existing routing policies.

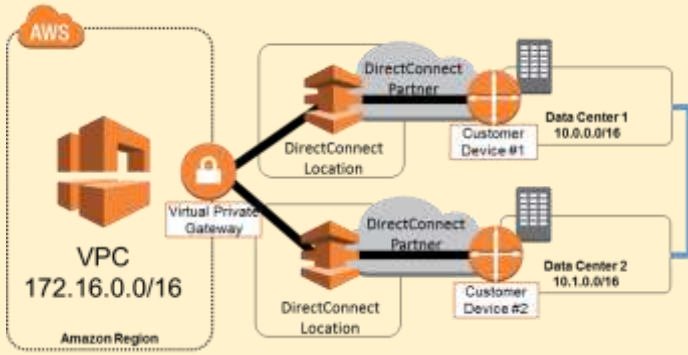
This configuration relies on the Internet to carry traffic between on-premises networks and VPC. Although AWS leverages multiple Internet Service Providers (ISPs), and even if the customer leverages multiple ISPs, an Internet service disruption can still affect the availability of VPN network connectivity due to the interdependence of ISPs and Internet routing. **The only way to control the exact network path of your traffic is to provision private network connectivity with AWS Direct Connect**

Multiple Data Center HA Network Connectivity - 2:Redundant Active/Active AWS Direct Connect Connections

https://aws.amazon.com/answers/networking/aws-multiple-data-center-ha-network-connectivity/

Setup:

Use AWS Direct Connect to reduce network costs, increase bandwidth throughput, or provide a more consistent network experience than Internet-based connections. Because each dedicated, physical connection is in one AWS Direct Connect location, multiple dynamically routed AWS Direct Connect connections are necessary to achieve high availability. Architectures with the highest levels of availability will leverage different AWS Direct Connect partner networks to ensure network-provider redundancy.



Configuration details:

- Setup AWS Direct Connect connections to separate AWS Direct Connect routers in two locations from two independently configured customer devices.
- The VGW will prefer to send 10.0.0.0/16 traffic to DC1, and only reroute this traffic to DC2 if connectivity to Customer Device 1 is down. Likewise, 10.1.0.0/16 traffic will prefer the AWS Direct Connect connection from DC2

AWS recommends using one of the following approaches for communicating these route preferences :

- **More specific routes:** With this approach, both Customer Device 1 and Customer Device 2 advertise a summary route of 10.0.0.0/15.
- **AS-path prepending:** With this approach, both Customer Device 1 and Customer Device 2 advertise 10.0.0.0/16 and 10.1.0.0/16. If your organization already leverages AS-path prepending for influencing route preferences, then the latter approach will likely align more closely with your existing routing policies.

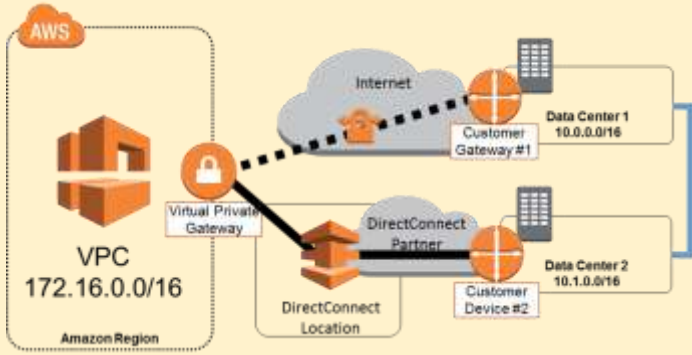
AWS Direct Connect allows you to create resilient connections to AWS because you have full control over the network path and network providers between your remote networks and AWS. Choose network providers and AWS Direct Connect locations that align with your organization's risk tolerance, financial expectations, and data-center-connectivity policies. For example, if your current data centers leverage multiple network providers to reduce the risk associated with an individual network-provider outage, then you should consider using different network providers for each AWS Direct Connect connection. Likewise, leveraging different AWS Direct Connect locations (e.g. CoreSite and Equinix in US East) will reduce the risk that a facility failure will interrupt your network connectivity with AWS.

Multiple Data Center HA Network Connectivity - 3:AWS Direct Connect with Backup VPN Connection

https://aws.amazon.com/answers/networking/aws-multiple-data-center-ha-network-connectivity/

Setup:

Some AWS customers would like the benefits of one or more AWS Direct Connect connections for their primary connectivity to AWS, coupled with a lower-cost backup connection.



Configuration details:

- Setup 2 dynamically routed connections, one using AWS Direct Connect and the other using a VPN connection from two different customer devices.
- By default, AWS will always prefer to send traffic over an AWS Direct Connect connection, so no additional configuration is required to define primary and backup connections. In this example, both Customer Gateway 1 and Customer Device 2 advertise a summary route of 10.0.0.0/15 and AWS will send all traffic to Customer Device 2 as long as this network path is available.

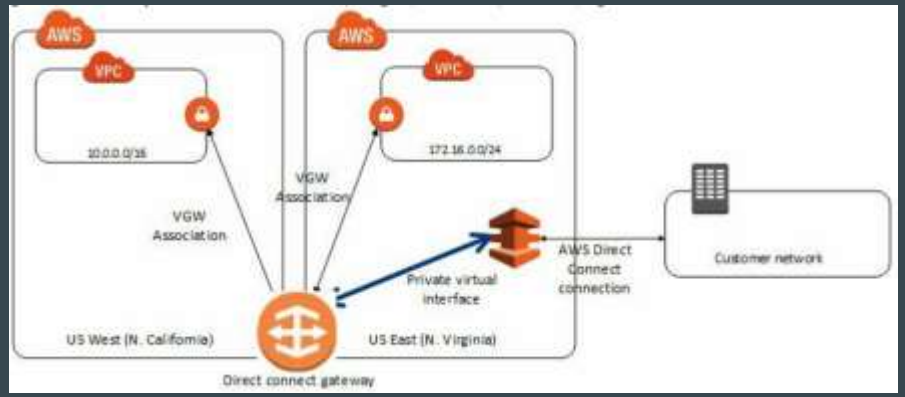
Although AWS prefers AWS Direct Connect to VPN connections by default, note that you still have the ability to influence AWS routing decisions by advertising more specific routes. If, for example, you want to leverage your backup VPN connection for a subset of traffic (e.g., developer traffic versus production traffic), you can advertise specific routes from Customer Gateway 1.

This approach allows you to choose the primary network path and network provider for your AWS traffic, with the option of using a different provider for a backup VPN connection. Choose network providers and AWS Direct Connect locations that align with your organization's risk tolerance, financial expectations, and data-center connectivity policies. For example, if you are concerned about the risk associated with an individual network-provider outage, consider different network providers for AWS Direct Connect and Internet connectivity. Additionally, make sure to monitor AWS Direct Connect utilization to ensure that a VPN connection will be a sufficient backup to support your application's latency and bandwidth requirements.

Direct Connect Gateways

Use AWS Direct Connect gateway to connect your VPCs. You associate an AWS Direct Connect gateway with 1 of the following
- A transit gateway when you have multiple VPCs in the same Region
- A virtual private gateway

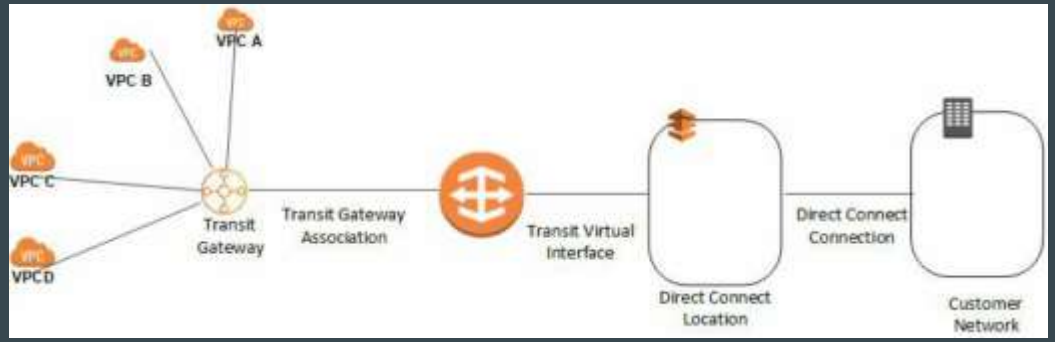
A Direct Connect gateway is a globally available resource. You can create the Direct Connect gateway in any public Region and access it from all other public Regions.



- You associate a Direct Connect gateway with the virtual private gateway for the VPC.
- Then, you 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 DC gateway. In the eg., the DC gateway enables you to use your AWS DC connection in the US East Region to access VPCs in your account in both the US East and US West Regions.

You can associate a DC gateway with a virtual private gateway that's in a different AWS account. The owner of the virtual private gateway creates an association proposal and the owner of the Direct Connect gateway must accept the association proposal.



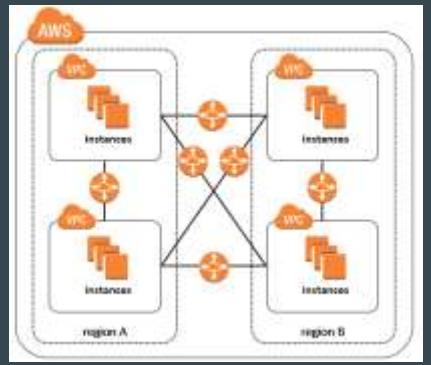
You can use a DC gateway to connect your DC connection over a transit virtual interface to the VPCs or VPNs that are attached to your transit gateway. You associate a DC gateway with the transit gateway. Then, create a transit virtual interface for your DC connection to the DC gateway. Benefits?:
- Manage a single connection for multiple VPCs or VPNs that are in the same Region.
- Advertise prefixes from on-premises to AWS and from AWS to on-premises.

How do I connect multiple VPCs in different AWS Regions?

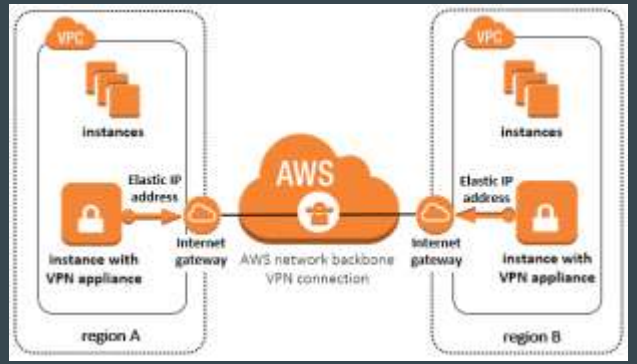
https://aws.amazon.com/answers/networking/aws-multiple-region-multi-vpc-connectivity/

1. Routing Over AWS Networks

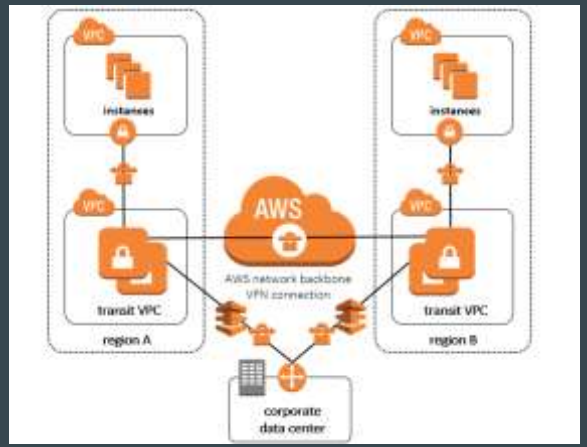
a. Inter-region VPC peering



b. Software VPN Appliances

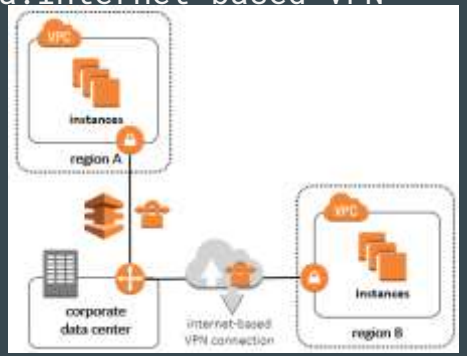


c. Transit VPC

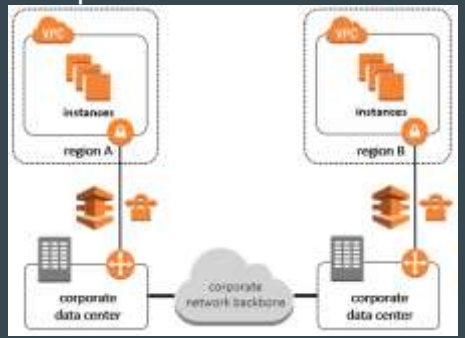


2. Routing Over Non-AWS Networks

a. Internet based VPN



b. Corporate NW backbone

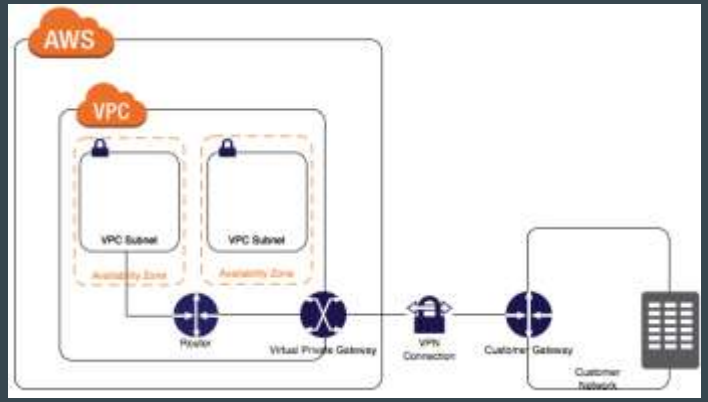


AWS VPN

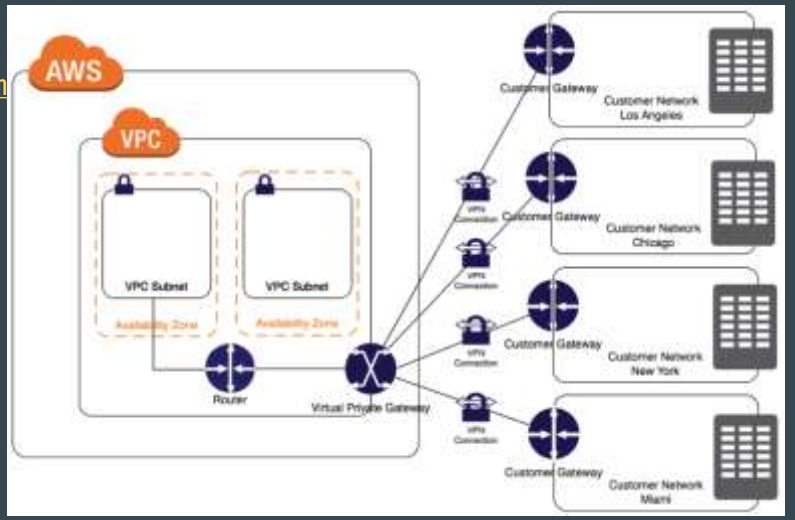
Site-to-Site VPN Single and Multiple Connection Examples

https://docs.aws.amazon.com/vpn/latest/s2svpn/Examples.htm

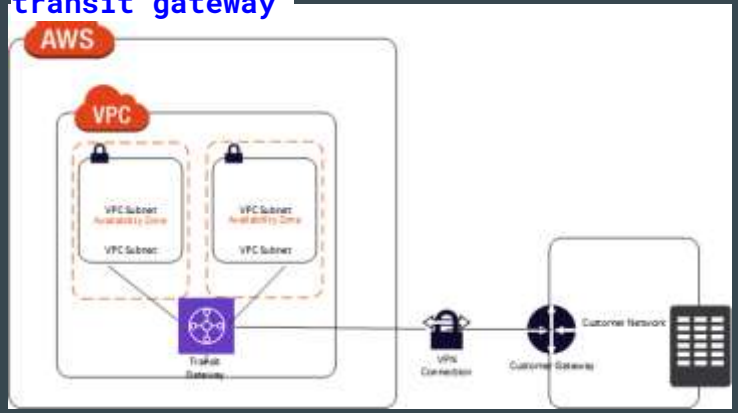
Single S2S VPN Connection



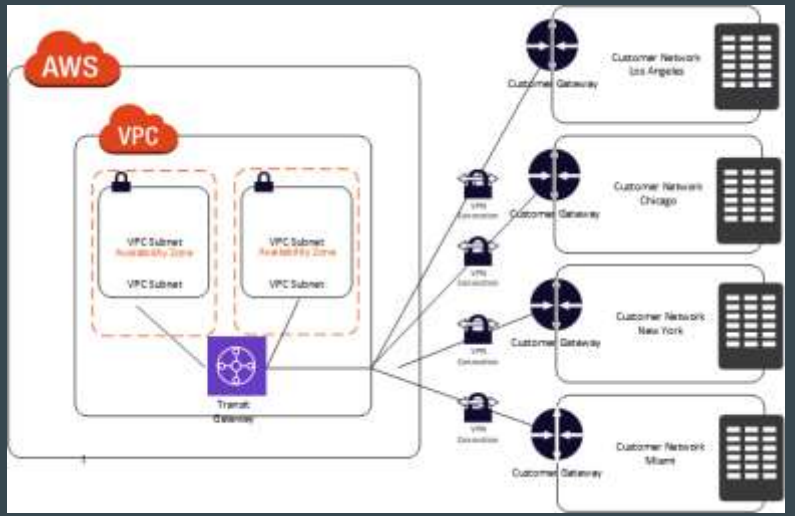
Multiple S2S VPN Connections



Single S2S VPN Connection with a transit gateway

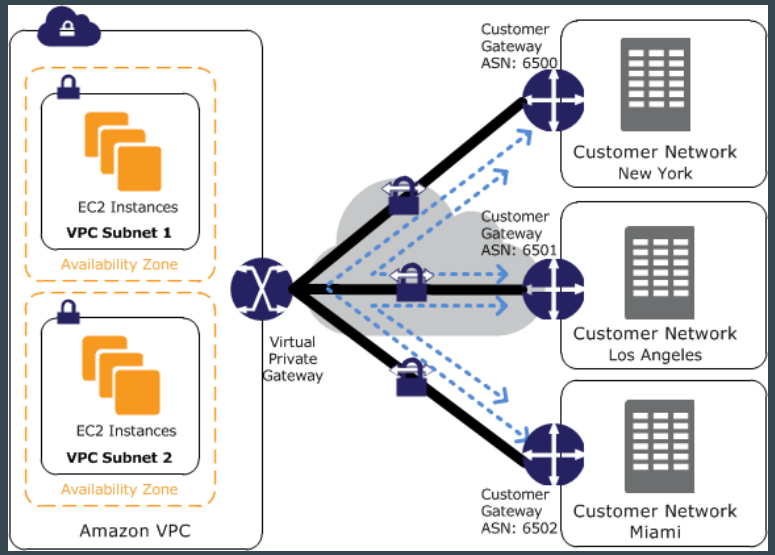


Multiple S2S VPN Connections w/a transit gateway



Providing Secure Communication Between Sites Using VPN CloudHub

https://docs.aws.amazon.com/vpn/latest/s2svpn/VPN_CloudHub.html



If you have multiple AWS S2S 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.

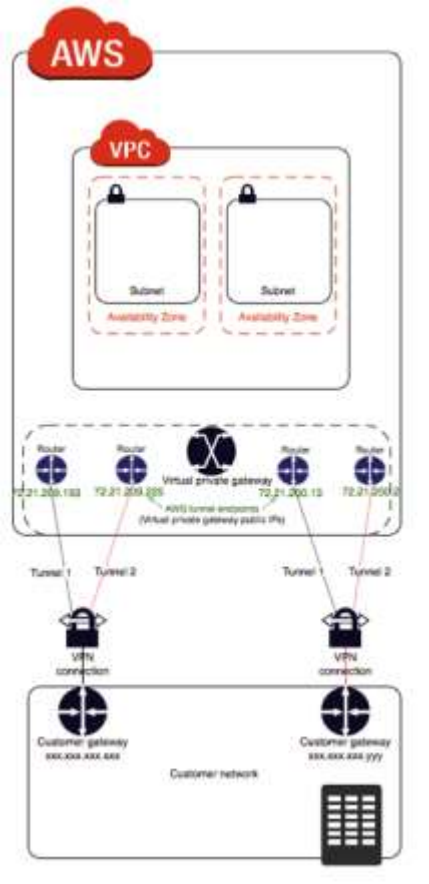
This design is suitable for customers with multiple branch offices and existing internet connections who'd like to implement a convenient, potentially low-cost hub-and-spoke model for primary or backup connectivity between these remote offices.

To use the AWS VPN CloudHub, you must create a VGW with multiple customer gateways. You must use a unique Border Gateway Protocol (BGP) Autonomous System Number (ASN) for each customer gateway. Customer gateways advertise the appropriate routes (BGP prefixes) over their Site-to-Site VPN connections. These routing advertisements are received and re-advertised to each BGP peer, enabling each site to send data to and receive data from the other sites. The sites must not have overlapping IP ranges. Each site can also send and receive data from the VPC as if they were using a standard Site-to-Site VPN connection.

Sites that use AWS Direct Connect connections to the virtual private gateway can also be part of the AWS VPN CloudHub.

Using Redundant Site-to-Site VPN Connections to Provide Failover

https://docs.aws.amazon.com/vpn/latest/s2svpn/VPNConnections.html



a Site-to-Site VPN connection has two tunnels to help ensure connectivity in case one of the Site-to-Site VPN connections becomes unavailable. To protect against a loss of connectivity in case your customer gateway becomes unavailable, you can set up a second Site-to-Site VPN connection to your VPC and virtual private gateway by using a second customer gateway. By using redundant Site-to-Site VPN connections and customer gateways, you can perform maintenance on one of your customer gateways while traffic continues to flow over the second customer gateway's Site-to-Site VPN connection. To establish redundant Site-to-Site VPN connections and customer gateways on your remote network, you need to set up a second Site-to-Site VPN connection. The customer gateway IP address for the second Site-to-Site VPN connection must be publicly accessible.

Dynamically routed Site-to-Site VPN connections use the Border Gateway Protocol (BGP) to exchange routing information between your customer gateways and the virtual private gateways. Statically routed Site-to-Site VPN connections require you to enter static routes for the remote network on your side of the customer gateway. BGP-advertised and statically entered route information allow gateways on both sides to determine which tunnels are available and reroute traffic if a failure occurs.

We recommend that you configure your network to use the routing information provided by BGP (if available) to select an available path. The exact configuration depends on the architecture of your network.