Amazon VPC

Neal

Please use the menu below to navigate the article sections: Hide article menu

Amazon VPC lets you provision a logically isolated section of the Amazon Web Services (AWS) cloud where you can launch AWS resources in a virtual network that you define.

Analogous to having your own DC inside AWS.

Provides complete control over the virtual networking environment including selection of IP ranges, creation of subnets, and configuration of route tables and gateways.

A VPC is logically isolated from other VPCs on AWS.

Possible to connect the corporate data center to a VPC using a hardware VPN (site-to-site).

VPCs are region wide.

A default VPC is created in each region with a subnet in each AZ.

By default you can create up to 5 VPCs per region.

You can define dedicated tenancy for a VPC to ensure instances are launched on dedicated hardware (overrides the configuration specified at launch).

A default VPC is automatically created for each AWS account the first time Amazon EC2 resources are provisioned.

The default VPC has all-public subnets.

Public subnets are subnets that have:

- “Auto-assign public IPv4 address” set to “Yes”.
- The subnet route table has an attached Internet Gateway.

Instances in the default VPC always have both a public and private IP address.

AZs names are mapped to different zones for different users (i.e. the AZ "ap-southeast-2a" may map to a different physical zone for a different user).

Components of a VPC:

- **A Virtual Private Cloud:** A logically isolated virtual network in the AWS cloud. You define a VPC's IP address space from ranges you select.
- **Subnet:** A segment of a VPC's IP address range where you can place groups of isolated resources (maps to an AZ, 1:1).
- **Internet Gateway:** The Amazon VPC side of a connection to the public Internet.
- **NAT Gateway:** A highly available, managed Network Address Translation (NAT) service for your resources in a private subnet to access the Internet.
- **Hardware VPN Connection:** A hardware-based VPN connection between your Amazon VPC and your datacenter, home network, or co-location facility.
- **Virtual Private Gateway:** The Amazon VPC side of a VPN connection.
- **Customer Gateway:** Your side of a VPN connection.
- **Router:** Routers interconnect subnets and direct traffic between Internet gateways, virtual private gateways, NAT gateways, and subnets.
- **Peering Connection:** A peering connection enables you to route traffic via private IP addresses between two peered VPCs.
- **VPC Endpoints:** Enables private connectivity to services hosted in AWS, from within your VPC without using an Internet Gateway, VPN, Network Address Translation (NAT) devices, or firewall proxies.
- **Egress-only Internet Gateway:** A stateful gateway to provide egress only access for IPv6 traffic from the VPC to the Internet.

Options for connecting to a VPC are:

- Hardware based VPN.
- Direct Connect.
- VPN CloudHub.
- Software VPN.

Want to learn how to create a VPC hands-on? In the AWS Hands-On Labs video tutorial below, we' show you how to create a Custom Amazon Virtual Private Cloud (VPC) on AWS. You will also learn how to create subnets, route tables, and Internet Gateways. We launch some EC2 instances into the new VPC and test connectivity.

Routing

The VPC router performs routing between AZs within a region.

The VPC router connects different AZs together and connects the VPC to the Internet Gateway.

Each subnet has a route table the router uses to forward traffic within the VPC.

Route tables also have entries to external destinations.

Up to 200 route tables per VPC.

Up to 50 route entries per route table.

Each subnet can only be associated with one route table.

Can assign one route table to multiple subnets.

If no route table is specified a subnet will be assigned to the main route table at creation time.

Cannot delete the main route table.

You can manually set another route table to become the main route table.

There is a default rule that allows all VPC subnets to communicate with one another – this cannot be deleted or modified.

Routing between subnets is always possible because of this rule – any problems communicating is more likely to be security groups or NACLs.

Subnets and Subnet Sizing

Types of subnet:

- If a subnet’s traffic is routed to an internet gateway, the subnet is known as a **public subnet.**
- 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 internet gateway, but has its traffic routed to a virtual private gateway for a VPN connection, the subnet is known as a **VPN-only subnet.**

The VPC is created with a master address range (CIDR block, can be anywhere from 16-28 bits), and subnet ranges are created within that range.

New subnets are always associated with the default route table.

Once the VPC is created you cannot change the CIDR block.

You cannot create additional CIDR blocks that overlap with existing CIDR blocks.

You cannot create additional CIDR blocks in a different RFC 1918 range.

Subnets with overlapping IP address ranges cannot be created.

The first 4 and last 1 IP addresses in a subnet are reserved.

Subnets are created within availability zones (AZs).

Each subnet must reside entirely within one Availability Zone and cannot span zones.

Availability Zones are distinct locations that are engineered to be isolated from failures in other Availability Zones.

Availability Zones are connected with low latency, high throughput, and highly redundant networking.

Can create private, public or VPN subnets.

Subnets map 1:1 to AZs and cannot span AZs.

You can only attach one Internet gateway to a custom VPC.

IPv6 addresses are all public and the range is allocated by AWS.

It is recommended these come from the private IP ranges specified in RFC 1918:

- 10.0.0.0 – 10.255.255.255 (10/8 prefix)
- 172.16.0.0 – 172.31.255.255 (172.16/12 prefix)
- 192.168.0.0 – 192.168.255.255 (192.168/16 prefix)

However, it is possible to create a VPC with publicly routable CIDR block.

The allowed block size is between a /28 netmask and /16 netmask.

The CIDR blocks of the subnets within a VPC cannot overlap.

The first four IP addresses and the last IP address in each subnet CIDR block are not available for you to use

For example, in a subnet with CIDR block 10.0.0.0/24, the following five IP addresses are reserved:

- 10.0.0.0: Network address.
- 10.0.0.1: Reserved by AWS for the VPC route.
- 10.0.0.2: Reserved by AWS.
- 10.0.0.3: Reserved by AWS for future use.
- 10.0.0.255: Network broadcast address (broadcast not supported).

For further information, check out this AWS article.

Internet Gateways

An Internet Gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet.

An Internet Gateway serves two purposes: .

- To provide a target in your VPC route tables for internet-routable traffic.
- To perform network address translation (NAT) for instances that have been assigned public IPv4 addresses.

Internet Gateways (IGW) must be created and then attached to a VPC, be added to a route table, and then associated with the relevant subnet(s).

No availability risk or bandwidth constraints.

If your subnet is associated with a route to the Internet, then it is a public subnet.

You cannot have multiple Internet Gateways in a VPC.

IGW is horizontally scaled, redundant and HA.

IGW performs NAT between private and public IPv4 addresses.

IGW supports IPv4 and IPv6.

IGWs must be detached before they can be deleted.

Can only attach 1 IGW to a VPC at a time.

Gateway terminology:

- Internet gateway (IGW) – AWS VPC side of the connection to the public Internet.

- Virtual private gateway (VPG) – VPC endpoint on the AWS side.
- Customer gateway (CGW) – representation of the customer end of the connection.

To enable access to or from the Internet for instances in a VPC subnet, you must do the following:

- Attach an Internet Gateway to your VPC.
- Ensure that your subnet's route table points to the Internet Gateway (see below).
- Ensure that instances in your subnet have a globally unique IP address (public IPv4 address, Elastic IP address, or IPv6 address).
- Ensure that your network access control and security group rules allow the relevant traffic to flow to and from your instance.

Must update subnet route table to point to IGW, either:

- To all destinations, e.g. 0.0.0.0/0 for IPv4 or ::/0for IPv6.
- To specific public IPv4 addresses, e.g. your company's public endpoints outside of AWS.

Egress-only Internet Gateway:

- Provides outbound Internet access for IPv6 addressed instances.
- Prevents inbound access to those IPv6 instances.
- IPv6 addresses are globally unique and are therefore public by default.
- Stateful – forwards traffic from instance to Internet and then sends back the response.
- Must create a custom route for ::/0 to the Egress-Only Internet Gateway.
- Use Egress-Only Internet Gateway instead of NAT for IPv6.

VPC Wizard

VPC with a Single Public Subnet:

- Your instances run in a private, isolated section of the AWS cloud with direct access to the Internet.
- Network access control lists and security groups can be used to provide strict control over inbound and outbound network traffic to your instances.
- Creates a /16 network with a /24 subnet. Public subnet instances use Elastic IPs or Public IPs to access the Internet.

VPC with Public and Private Subnets:

- In addition to containing a public subnet, this configuration adds a private subnet whose instances are not addressable from the Internet.
- Instances in the private subnet can establish outbound connections to the Internet via the public subnet using Network Address

Translation (NAT).
- Creates a /16 network with two /24 subnets.
- Public subnet instances use Elastic IPs to access the Internet.
- Private subnet instances access the Internet via Network Address Translation (NAT).

VPC with Public and Private Subnets and Hardware VPN Access:

- This configuration adds an IPsec Virtual Private Network (VPN) connection between your Amazon VPC and your data center – effectively extending your data center to the cloud while also providing direct access to the Internet for public subnet instances in your Amazon VPC.
- Creates a /16 network with two /24 subnets.
- One subnet is directly connected to the Internet while the other subnet is connected to your corporate network via an IPsec VPN tunnel.

VPC with a Private Subnet Only and Hardware VPN Access:

- Your instances run in a private, isolated section of the AWS cloud with a private subnet whose instances are not addressable from the Internet.
- You can connect this private subnet to your corporate data center via an IPsec Virtual Private Network (VPN) tunnel.
- Creates a /16 network with a /24 subnet and provisions an IPsec VPN tunnel between your Amazon VPC and your corporate network.

NAT Instances

NAT instances are managed **by** you.

Used to enable private subnet instances to access the Internet.

NAT instance must live on a public subnet with a route to an Internet Gateway.

Private instances in private subnets must have a route to the NAT instance, usually the default route destination of 0.0.0.0/0.

When creating NAT instances always disable the source/destination check on the instance.

NAT instances must be in a single public subnet.

NAT instances need to be assigned to security groups.

Security groups for NAT instances must allow HTTP/HTTPS inbound from the private subnet and outbound to 0.0.0.0/0.

There needs to be a route from a private subnet to the NAT instance for it to work.

The amount of traffic a NAT instance can support is based on the instance type.

Using a NAT instance can lead to bottlenecks (not HA).

HA can be achieved by using Auto Scaling groups, multiple subnets in different AZ’s and a script to automate failover.

Performance is dependent on instance size.

Can scale up instance size or use enhanced networking.

Can scale out by using multiple NATs in multiple subnets.

Can use as a bastion (jump) host.

Can monitor traffic metrics.

Not supported for IPv6 (use Egress-Only Internet Gateway).

NAT Gateways

NAT gateways are managed **for** you by AWS.

Fully managed NAT service that replaces the need for NAT instances on EC2.

Must be created in a public subnet.

Uses an Elastic IP address for the public IP.

Private instances in private subnets must have a route to the NAT instance, usually the default route destination of 0.0.0.0/0.

Created in a specified AZ with redundancy in that zone.

For multi-AZ redundancy, create NAT Gateways in each AZ with routes for private subnets to use the local gateway.

Up to 5 Gbps bandwidth that can scale up to 45 Gbps.

Can’t use a NAT Gateway to access VPC peering, VPN or Direct Connect, so be sure to include specific routes to those in your route table.

NAT gateways are highly available in each AZ into which they are deployed.

They are preferred by enterprises.

No need to patch.

Not associated with any security groups.

Automatically assigned a public IP address.

Remember to update route tables and point towards your gateway.

More secure (e.g. you cannot access with SSH and there are no security groups to maintain).

No need to disable source/destination checks.

Egress only Internet gateways operate on IPv6 whereas NAT gateways operate on IPv4.

Port forwarding is not supported.

Using the NAT Gateway as a Bastion host server is not supported.

Traffic metrics are not supported.

The table below highlights the key differences between both types of gateway:

	NAT Gateway	**NAT Instance**
Managed	Managed by AWS	Managed
Availability	Highly available within an AZ	Not highly available (would require scripting)
Bandwidth	Up to 45 GPS	Depends on the bandwidth of the EC2 instance you selected
Maintenance	Managed by AWS	Managed by you
Performance	Optimized for NAT	Amazon Linux 2 AMI configured to perform NAT
Public IP	Elastic IP cannot be detached	Elastic IP that can be detached
Security Groups	Cannot associate with a security group	Can associate with a security group
Bastion Host	Not supported	Can be used as a bastion host

Security Groups

Security groups act like a firewall at the instance level.

Specifically security groups operate at the network interface level.

Can only assign permit rules in a security group, cannot assign deny rules.

There is an implicit deny rule at the end of the security group.

All rules are evaluated until a permit is encountered or continues until the implicit deny.

Can control ingress and egress traffic.

Security groups are stateful.

By default, custom security groups do not have inbound allow rules (all inbound traffic is denied by default).

By default, default security groups do have inbound allow rules (allowing traffic from within the group).

All outbound traffic is allowed by default in custom and default security groups.

You cannot delete the security group that's created by default within a VPC.

You can use security group names as the source or destination in other security groups.

You can use the security group name as a source in its own inbound rules.

Security group members can be within any AZ or subnet within the VPC.

Security group membership can be changed whilst instances are running.

Any changes made will take effect immediately.

Up to 5 security groups can be added per EC2 instance interface.

There is no limit on the number of EC2 instances within a security group.

You cannot block specific IP addresses using security groups, use NACLs instead.

Network ACL's

Network ACL's function at the subnet level.

The VPC router hosts the network ACL function.

With NACLs you can have permit and deny rules.

Network ACLs contain a numbered list of rules that are evaluated in order from the lowest number until the explicit deny.

Recommended to leave spacing between network ACL numbers.

Network ACLs have separate inbound and outbound rules and each rule can allow or deny traffic.

Network ACLs are stateless, so responses are subject to the rules for the direction of traffic.

NACLs only apply to traffic that is ingress or egress to the subnet not to traffic within the subnet.

A VPC automatically comes with a default network ACL which allows all inbound/outbound traffic.

A custom NACL denies all traffic both inbound and outbound by default.

All subnets must be associated with a network ACL.

You can create custom network ACL's. By default, each custom network ACL denies all inbound and outbound traffic until you add rules.

Each subnet in your VPC must be associated with a network ACL. If you don't do this manually it will be associated with the default network ACL.

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

Network ACLs do not filter traffic between instances in the same subnet.

NACLs are the preferred option for blocking specific IPs or ranges.

Security groups cannot be used to block specific ranges of IPs.

NACL is the first line of defense, the security group is the second line.

Also recommended to have software firewalls installed on your instances.

Changes to NACLs take effect immediately.

Security Group	Network ACL

Operates at the instance (interface level)	Operates at the subnet level
Supports allow rules only	Supports allow and deny rules
Stateful	Stateless
Evaluates all rules	Processes rules in order
Applies to an instance only if associated with a group	Automatically applies to all instances in the subnet it is associated with

VPC Connectivity

There are several methods of connecting to a VPC. These include:

- AWS Managed VPN.
- AWS Direct Connect.
- AWS Direct Connect plus a VPN.
- AWS VPN CloudHub.
- Software VPN.
- Transit VPC.
- VPC Peering.
- AWS PrivateLink.
- VPC Endpoints.

Each of these will be further detailed below.

AWS Managed VPN

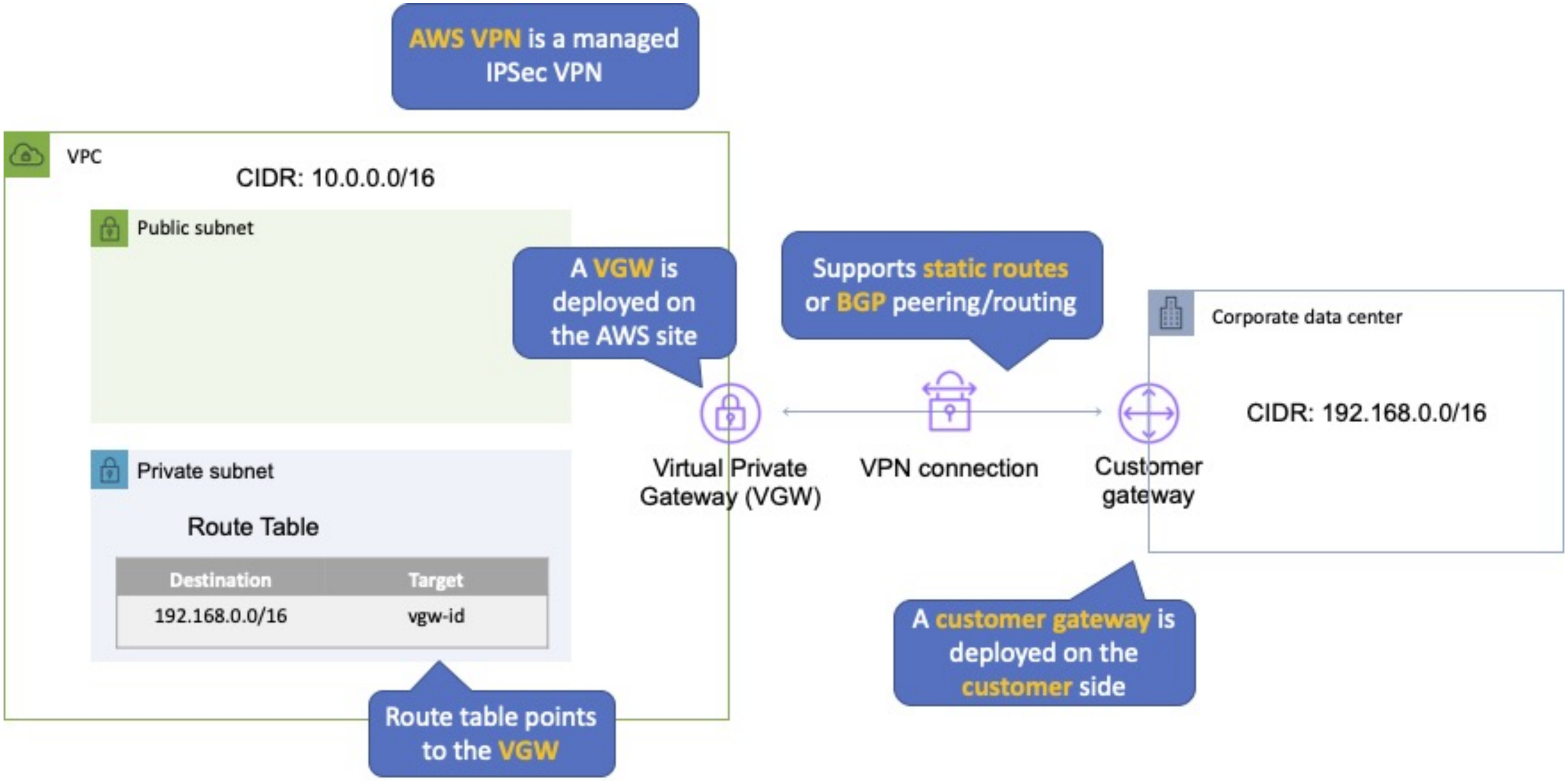
What	AWS-provided network connectivity between two VPCs
When	Multiple VPCs need to communicate or access each other’s resources
Pros	Uses AWS backbone without traversing the public internet
Cons	Transitive peering is not supported
How	VPC Peering request made; acceptor accepts request (either within or across accounts)

VPNs are quick, easy to deploy, and cost effective.

A Virtual Private Gateway (VGW) is required on the AWS side.

A Customer Gateway is required on the customer side.

The diagram below depicts an AWS S2S VPN configuration:


AWS VPN is a managed IPSec VPN
VPC
CIDR: 10.0.0.0/16
Public subnet
A VGW is deployed on the AWS site
Supports static routes or BGP peering/routing
Corporate data center
CIDR: 192.168.0.0/16
Virtual Private Gateway (VGW)
VPN connection
Customer gateway
Private subnet
Route Table
A customer gateway is deployed on the customer side
Route table points to the VGW

Destination	Target
192.168.0.0/16	vgw-id

An Internet routable IP address is required on the customer gateway.

Two tunnels per connection must be configured for redundancy.

You cannot use a NAT gateway in AWS for clients coming in via a VPN.

For route propagation you need to point your VPN-only subnet’s route tables at the VGW.

Must define the IP prefixes that can send/receive traffic through the VGW.

VGW does not route traffic destined outside of the received BGP advertisements, static route entries, or its attached VPC CIDR.

Cannot access Elastic IPs on your VPC via the VPN – Elastic IPs can only be connected to via the Internet.

AWS Direct Connect

What	Dedicated network connection over private line straight into the AWS backbone
When	Requires a large network link into AWS; lots of resources and services being provided on AWS to your corporate users
Pros	More predictable network performance; potential bandwidth cost reduction; up to 10 Gbps provisioned connections; supports BGP peering and routing
Cons	May require additional telecom and hosting provider relationships and/or network circuits; costly
How	Work with your existing data networking provider; create Virtual Interfaces (VIFs) to connect to VPCs (private VIFs) or other AWS services like S3 or Glacier (public VIFs)

AWS Direct Connect makes it easy to establish a dedicated connection from an on-premises network to Amazon VPC.

Using AWS Direct Connect, you can establish private connectivity between AWS and your data center, office, or collocated environment.

This private connection can reduce network costs, increase bandwidth throughput, and provide a more consistent network experience than internet-based connections.

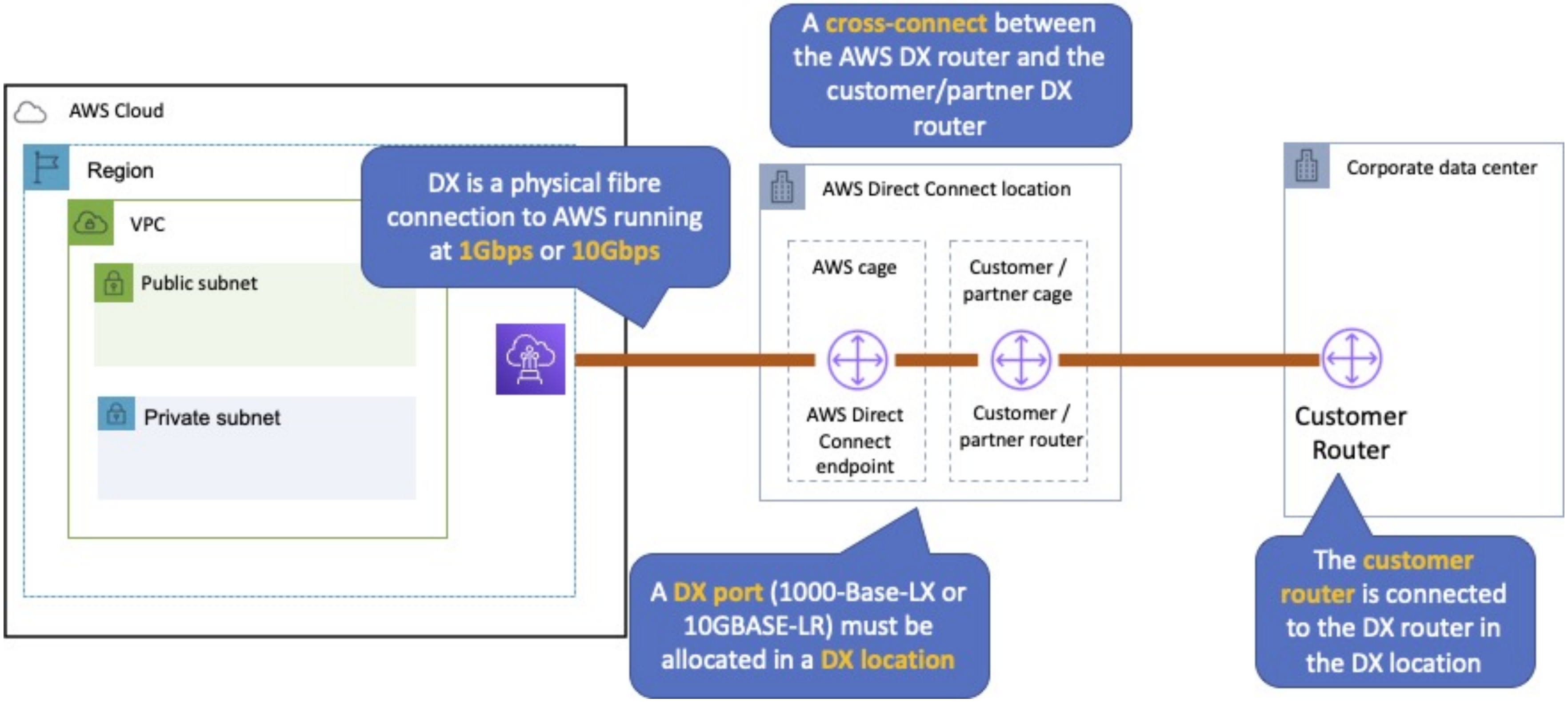
AWS Direct Connect lets you establish 1 Gbps or 10 Gbps dedicated network connections (or multiple connections) between AWS networks and one of the AWS Direct Connect locations.

It uses industry standard VLANs to access Amazon Elastic Compute Cloud (Amazon EC2) instances running within an Amazon VPC using private IP addresses.

AWS Direct Connect does not encrypt your traffic that is in transit.

You can use the encryption options for the services that traverse AWS Direct Connect.

The diagram below depicts an AWS Direct Connect configuration:


AWS Cloud
Region
VPC
Public subnet
Private subnet
DX is a physical fibre connection to AWS running at 1Gbps or 10Gbps
A cross-connect between the AWS DX router and the customer/partner DX router
AWS Direct Connect location
AWS cage
Customer / partner cage
AWS Direct Connect endpoint
Customer / partner router
Corporate data center
Customer Router
A DX port (1000-Base-LX or 10GBASE-LR) must be allocated in a DX location
The customer router is connected to the DX router in the DX location

AWS Direct Connect Plus VPN

What	IPSec VPN connection over private lines (Direct Connect)
When	Need the added security of encrypted tunnels over Direct Connect
Pros	More secure (in theory) than Direct Connect alone
Cons	More complexity introduced by VPN layer
How	Work with your existing data networking provider

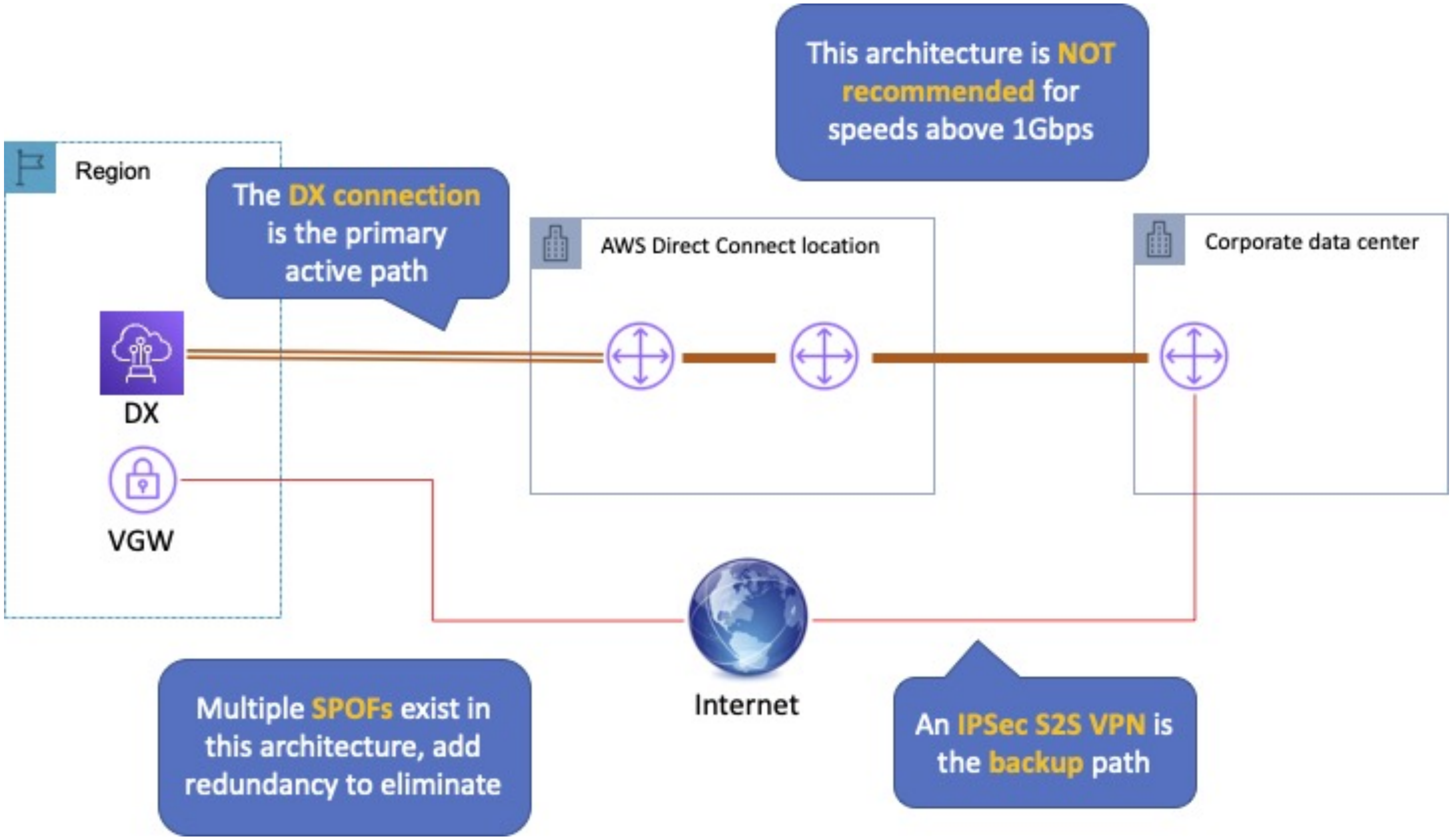
With AWS Direct Connect plus VPN, you can combine one or more AWS Direct Connect dedicated network connections with the Amazon VPC VPN.

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 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.

The diagram below depicts an AWS Direct Connect plus VPN configuration:


This architecture is NOT recommended for speeds above 1Gbps
Region
The DX connection is the primary active path
AWS Direct Connect location
Corporate data center
DX
VGW
Internet
Multiple SPOFs exist in this architecture, add redundancy to eliminate
An IPSec S2S VPN is the backup path

AWS VPN CloudHub

What	Connect locations in a hub and spoke manner using AWSs Virtual Private Gateway
When	Link remote offices for backup or primary WAN access to AWS resources and each other
Pros	Reuses existing Internet connections; supports BGP routes to direct traffic
Cons	Dependent on Internet connections; no inherent redundancy

How Assign multiple Customer Gateways to a Virtual Private Gateway, each with their own BGP ASN and unique IP ranges

The AWS VPN CloudHub operates on a simple hub-and-spoke model that you can use with or without a VPC.

Use this design if you have multiple branch offices and existing internet connections and would like to implement a convenient, potentially low-cost hub-and-spoke model for primary or backup connectivity between these remote offices.

VPN CloudHub is used for hardware-based VPNs and allows you to configure your branch offices to go into a VPC and then connect that to the corporate DC (hub and spoke topology with AWS as the hub).

Can have up to 10 IPSec tunnels on a VGW by default.

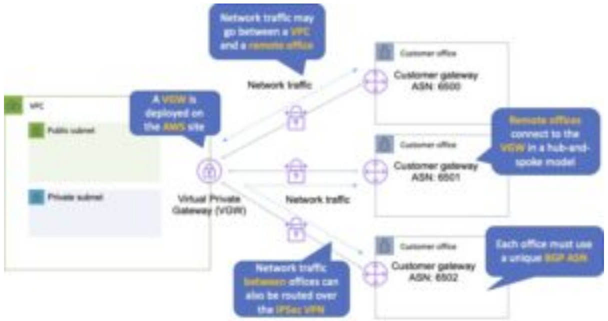
Uses eBGP.

Branches can talk to each other (and provides redundancy).

Can have Direct Connect connections.

Hourly rates plus data egress charges.

The diagram below depicts an AWS VPN CloudHub configuration:


Network traffic may go between a VPC and a remote office
VPC
Public subnet
Private subnet
A VGW is deployed on the AWS site
Virtual Private Gateway (VGW)
Network traffic
Network traffic
Customer office
Customer gateway ASN: 6500
Customer office
Customer gateway ASN: 6501
Remote offices connect to the VGW in a hub-and-spoke model
Customer office
Customer gateway ASN: 6502
Each office must use a unique BGP ASN
Network traffic between offices can also be routed over the IPSec VPN

Software VPN

What You must provide your own endpoint and software

When You must manage both ends of the VPN connection for compliance reasons or you want to use a VPN option not supported by AWS

Pros Ultimate flexibility and manageability

Cons	You must design for any needed redundancy across the whole chain
How	Install VPN software via Marketplace on an EC2 instance

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.

Transit VPC

What	Common strategy for connecting geographically dispersed VPCs and locations to create a global network transit center
When	Locations and VPC-deployed assets across multiple regions that need to communicate with one another
Pros	Ultimate flexibility and manageability but also AWS-managed VPN hub-and-spoke between VPCs
Cons	You must design for any redundancy across the whole chain
How	Providers like Cisco, Juniper Networks, and Riverbed have offerings which work with their equipment and AWS VPC

Building on the Software VPN design mentioned above, you can create a global transit network on AWS.

A transit VPC is a common strategy for connecting multiple, geographically disperse VPCs and remote networks to create a global network transit center.

A transit VPC simplifies network management and minimizes the number of connections required to connect multiple VPCs and remote networks.

VPC Peering

What	AWS-provided network connectivity between VPCs
When	Multiple VPCs need to connect with one another and access their resources
Pros	Uses AWS backbone without traversing the internet
Cons	Transitive peering is not supported
How	VPC Peering request made; accepter request (either within or across accounts)

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses.

Instances in either VPC can communicate with each other as if they are within the same network.

You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account.

The VPCs can be in different regions (also known as an inter-region VPC peering connection).

Data sent between VPCs in different regions is encrypted (traffic charges apply).

For inter-region VPC peering there are some limitations:

- You cannot create a security group rule that references a peer security group.
- Cannot enable DNS resolution.
- Maximum MTU is 1500 bytes (no jumbo frames support).
- Limited region support.

AWS uses the existing infrastructure of a VPC to create a VPC peering connection.

It is neither a gateway nor a VPN connection and does not rely on a separate piece of physical hardware.

There is no single point of failure for communication or a bandwidth bottleneck.

A VPC peering connection helps you to facilitate the transfer of data.

Can only have one peering connection between any two VPCs at a time.

Can peer with other accounts (within or between regions).

Cannot have overlapping CIDR ranges.

A VPC peering connection is a one-to-one relationship between two VPCs.

You can create multiple VPC peering connections for each VPC that you own, but transitive peering relationships are not supported.

You do not have any peering relationship with VPCs that your VPC is not directly peered with.

Limits are 50 VPC peers per VPC, up to 125 by request.

DNS is supported.

Must update route tables to configure routing.

Must update the inbound and outbound rules for VPC security group to reference security groups in the peered VPC.

When creating a VPC peering connection with another account you need to enter the account ID and VPC ID from the other account.

Need to accept the pending access request in the peered VPC.

The VPC peering connection can be added to route tables – shows as a target starting with “pcx-“.

AWS PrivateLink

AWS PrivateLink simplifies the security of data shared with cloud-based applications by eliminating the exposure of data to the public Internet.

AWS PrivateLink provides private connectivity between VPCs, AWS services, and on-premises applications, securely on the Amazon network.

AWS PrivateLink makes it easy to connect services across different accounts and VPCs to significantly simplify the network architecture.

The table below provides more information on AWS PrivateLink and when to use it:

What	AWS-provided connectivity between VPCs and/or AWS services using interface endpoints
When	Keep private subnets truly private by using the AWS backbone rather than using the public internet
Pros	Redundant; uses AWS backbone
Cons	
How	Create endpoint for required AWS or Marketplace service in all required subnets; access via the provided DNS hostname

EXAM TIP: *Know the difference between AWS PrivateLink and ClassicLink. ClassicLink allows you to link EC2-Classic instances to a VPC in your account, within the same region. EC2-Classic is an old platform from before VPCs were introduced and is not available to accounts created after December 2013. However, ClassicLink may come up in exam questions as a possible (incorrect) answer, so you need to know what it is.*

VPC Endpoints

An Interface endpoint uses AWS PrivateLink and is an elastic network interface (ENI) with a private IP address that serves as an entry point for traffic destined to a supported service.

Using PrivateLink you can connect your VPC to supported AWS services, services hosted by other AWS accounts (VPC endpoint services) and supported AWS Marketplace partner services.

AWS PrivateLink access over Inter-Region VPC Peering:

- Applications in an AWS VPC can securely access AWS PrivateLink endpoints across AWS Regions using Inter-Region VPC Peering.
- AWS PrivateLink allows you to privately access services hosted on AWS in a highly available and scalable manner, without using public IPs, and without requiring the traffic to traverse the Internet.
- Customers can privately connect to a service even if the service endpoint resides in a different AWS Region.
- Traffic using Inter-Region VPC Peering stays on the global AWS backbone and never traverses the public Internet.

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.

An interface VPC endpoint (interface endpoint) enables you to connect to services powered by AWS PrivateLink.

The table below highlights some key information about both types of endpoints:

	Interface Endpoint	**Gateway Endpoint**
What	Elastic Network Interface with a private IP	A gateway that is a target for a specific route
How	Uses DNS entries to redirect traffic	Use prefix lists in the route table to redirect traffic
Which Services	A large amount of AWS services, for full list follow this link	Amazon S3, DynamoDB
Security	Security Groups	VPC Endpoint Policies

By default, IAM users do not have permission to work with endpoints.

You can create an IAM user policy that grants users the permissions to create, modify, describe, and delete endpoints.

There’s a long list of services that are supported by interface endpoints.

Gateway endpoints are only available for:

- Amazon DynamoDB
- Amazon S3

EXAM TIP: Know which services use interface endpoints and gateway endpoints. The easiest way to remember this is that Gateway Endpoints are for Amazon S3 and DynamoDB only.

Shared Services VPCs

You can allow other AWS accounts to create their application resources, such as EC2 instances, Relational Database Service (RDS) databases, Redshift clusters, and Lambda functions, into shared, centrally managed Amazon Virtual Private Clouds (VPCs).

VPC sharing enables subnets to be shared with other AWS accounts within the same AWS Organization. Benefits include:

- Separation of duties: centrally controlled VPC structure, routing, IP address allocation.
- Application owners continue to own resources, accounts, and security groups.
- VPC sharing participants can reference security group IDs of each other.
- Efficiencies: higher density in subnets, efficient use of VPNs and AWS Direct Connect.
- Hard limits can be avoided, for example, 50 VIFs per AWS Direct Connect connection through simplified network architecture.
- Costs can be optimized through reuse of NAT gateways, VPC interface endpoints, and intra-Availability Zone traffic.

You can create separate Amazon VPCs for each account with the account owner being responsible for connectivity and security of each Amazon VPC.

With VPC sharing, your IT team can own and manage your Amazon VPCs and your application developers no longer must manage or configure Amazon VPCs, but they can access them as needed.

Can also share Amazon VPCs to leverage the implicit routing within a VPC for applications that require a high degree of interconnectivity and are within the same trust boundaries.

This reduces the number of VPCs that need to be created and managed, while you still benefit from using separate accounts for billing and access control.

Customers can further simplify network topologies by interconnecting shared Amazon VPCs using connectivity features, such as AWS PrivateLink, AWS Transit Gateway, and Amazon VPC peering.

Can also be used with AWS PrivateLink to secure access to resources shared such as applications behind a Network Load Balancer.

VPC Flow Logs

Flow Logs capture information about the IP traffic going to and from network interfaces in a VPC.

Flow log data is stored using Amazon CloudWatch Logs.

Flow logs can be created at the following levels:

- VPC.
- Subnet.
- Network interface.

You can’t enable flow logs for VPC’s that are peered with your VPC unless the peer VPC is in your account.

You can’t tag a flow log.

You can’t change the configuration of a flow log after it’s been created.

After you’ve created a flow log, you cannot change its configuration (you need to delete and re-create).

Not all traffic is monitored, e.g. the following traffic is excluded:

- Traffic that goes to Route53.
- Traffic generated for Windows license activation.
- Traffic to and from 169.254.169.254 (instance metadata).
- Traffic to and from 169.254.169.123 for the Amazon Time Sync Service.
- DHCP traffic.
- Traffic to the reserved IP address for the default VPC router.

High Availability Approaches for Networking

By creating subnets in the available AZs, you create Multi-AZ presence for your VPC.

Best practice is to create at least two VPN tunnels into your Virtual Private Gateway.

Direct Connect is not HA by default, so you need to establish a secondary connection via another Direct Connect (ideally with another provider) or use a VPN.

Route 53’s health checks provide a basic level of redirecting DNS resolutions.

Elastic IPs allow you flexibility to change out backing assets without impacting name resolution.

For Multi-AZ redundancy of NAT Gateways, create gateways in each AZ with routes for private subnets to use the local gateway.

Related posts: