VPC

* + Overview
    - Virtual Private Cloud
    - Lets you provision a logiacally isolated section of the AWS cloud where you can launch resources in a virtual network that you define
    - Alalagous to haveing your own DC inside AWS
    - Provides complete control over virtual networking environment including selection of IP ranges, creation of subents and configuration of route tables and gateways
    - A VPC is logically isolated from other VPCs on AWS
    - Possible to connect the corprorate data centre 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 launche on dedicated hardware(overrides the configuration specified at launch)
    - A default VPC is automatically created for each AWS account the first time EC2 resources are provisioned
    - A 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(ie the AZ ap-southeast-2a' may map to a different physical zone for a different user
    - Components of a VPC
      * **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 resouces(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 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
    - **Routing**
      * The VPC router performs routing between AZs within a region
      * The VPC router connects different AZs together and connects the VPC to the IG
      * 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 anwyere 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 AZs
      * **Each subnet must reside entirely within one AZ and cannot span zones**
      * Availability Zones are distinct locations that are engineered to be isolated from failures in other AZ's
      * AZ's are connected with low latency, high throughput and highly redundatn networking
      * Can create private, public or VPN subnets
      * Subnets map 1:1 to AZs and cannot span AZs
      * You can only attach one IG to a custom VPC
      * IPv6 addresses are all public and the range is allocated by AWS
    - **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(IGWs) 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 constraings
      * **If your subnet is associated with a route to the Internet, then it is a public subnet**
      * **You cannot have multiple IGs 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
      * GW terminology:
        + **Internet Gateway (IGW):** AWS VPC side of the connection to the publc 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
        + Ensure that instances in your subnet have a globally unique IP address(public IPv4, Elastic IP, or IPv6)
        + Ensure that your network access control and SG 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 ::/0 for IPv6
        + To specific public IPv4 addresses, e.g. your company's public endpoints outside of AWS
      * Egress-only IG
        + 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 IG
        + Use Egress-Only IG instead of NAT for IPv6
    - **VPC Wizard**
      * VPC within a Single Public Subnet
        + Your instances run in a private isolated section of the AWS cloud with direct access to the Interent
        + Network access controls lists and SGs 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 within 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 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 Interent 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 instaces to access the Interent
      * NAT instance must live on a public subnet with a route to an IG
      * Private instances in private subnets must have a route to the NAT instance, usually the defaultt route destionation 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 SGs
      * SGs 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 ASGs, multiple subnets in different AZs and a script to automate failover
      * Performance is dependent on instance size
      * Can scale up instnce size or use enhanced networking
      * Can scale out by using multiple NATs in multiple subnets
      * Can use bastion(jump) hosts
      * Can monitor traffic metrics
      * Not supported for IPv6(use Egress-Only IG)
    - **NAT Gateways**
      * NAT GW's are mamanged for you by AWS
      * Fully-managed NAT service that replaces the need for NAT instances on EC2
      * Mulst be created in a public subnet
      * Uses and 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 GWs 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 GW to access VPC peering, VPN or Direct COnnect, so be sure to include specific routes to those in your route table
      * NAT GW's are highly available in each AZ into which they are deployed
      * They are preferred by enterprises
      * No need to patch
      * Nota associated with any SGs
      * Automatically assigned public IP addresses
      * Remember to update route tables and point towards your gateway
      * More seucre(you cannot access with SSH and there are no SG's 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 GW:

Graphical user interface, table

Description automatically generated

* + **Security Groups**
    - SGs act like a firewall at the instance level
    - Specifically SGs operate at the network interface level
    - Can only assign permit rules in a SG, cannot assign deny rules
    - There is an implicit deny rule at the end of the SG
    - All rules are evaluated until a permit is encountered or continues until the implicit deny
    - Can control ingress and egress traffic
    - SGs are stateful
    - By default, custom SGs do not have inbound allow rules(all inbound traffic is denied by default)
    - By default, default SGs do not have inbound allow rules(allowing traffic from within the group)
    - All outbound traffic is allowed by default in custom and default SGs
    - You cannot delete the SG that's created by default within a VPC
    - You can use SG names as the source or destination in other SGs
    - You can use the SG name as a source or destination in other SGs
    - You can use the SG group name as a source in its own inbound rules
    - SG members can be within any AZ or subnet within the VPC
    - SG membership can be changed whilst instances are running
    - Any changes made will take effect immediately
    - Up to 5 SGs can be added per EC2 instance interface
    - There is no limit on the number of EC2 instances within a SG
    - You cannot block specific IP addresses using SGs, 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 tat 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
    - NACLS are stateless so responses are subject to the rules for the direction of traffic
    - NACLs only apply to traffic that ingress or egress to the subnet to traffic within the subnet
    - A VPC automatically comes with a default NACL which allows all inbound/outbound traffic
    - A custom NACL denies all traffic from both inbound and oubound by default
    - All subnets must be associated with a NACL
    - You can create custom NACLs. 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 do this manually it will be associated with the default NACL
    - You can associate a NACL with multiple subnets; however a subnet can only be associated with one NACL at a time
    - NACLs do not filter traffic between instances in the same subnet
    - NACLs are preferred option for blocking specific IPs or ranges
    - SGs cannot be used to block specific ranges of IPs
    - NACLS is the first line of defense, the SG is the second line
    - Also recommended to have sotware firewalls installed on your instances'
    - Changes to NACs take effect immediately
    - Table

      Description automatically generated
  + **VPC Connectivity**
    - There are several methods of connecting to a VPC including:
      * AWS Managed VPN
      * Direct Connect
      * DC plus a VPN
      * VPN CloudHub
      * Software VPN
      * Transit VPC
      * VPC Peering
      * AWS PrivateLink
      * VPC Endpoints
    - **AWS Managed VPN**
    - Graphical user interface, text, application, table

      Description automatically generated with medium confidence
      * 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 Managed VPN configuration:
    - Timeline

      Description automatically generated
      * An internet routable IP address is required on the customer gateway
      * Two tunnels per connection must be configured for redundancy
      * You cannot use NAT gateway in AWS for lcients 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**
* Table

  Description automatically generated
  + - DC makes it easy to establish a dedicated connection from an on-premises network to Amazon VPC
    - Using DC, 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 DC lets you establish 1Gbps or 10Gbps dedicated network connections(or multiple connections) between AWS networks and one of the AWS DC locations
    - It uses industry-standard VLANs to access EC2 instances running within a VPC using private IP addresses
    - DC does not encrypt your traffic that is in transit
    - You can use encryption options for the services that travers DC
    - DC configuration:

Diagram

Description automatically generated

* + **AWS Direct Connect Plus VPN**
  + Table

    Description automatically generated
    - DC plus VPN, you can combine one or more DC dedicated network connections with the 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 DC 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 DC solution, and an end-to-end secure IPsec connection
    - DC plus VPN diagram:
  + Diagram, timeline

    Description automatically generated
  + **AWS VPN CloudHub**
  + Application, table

    Description automatically generated with medium confidence
    - 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 ineternet 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 yuour branch offices to go into a VPC and then connect that to the dcorporate DC
    - Can have 10 IPSec tunnels in 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
    - VPN CH config:
  + Diagram

    Description automatically generated
  + **Software VPN**
  + Text

    Description automatically generated with low confidence
    - VPC offers you the flexibility to manage both sides of your VPC connectivity by creating a VPN connection between your remote network and a software VPN appliance running in your VPN network
    - This option is recommended i you must manage both ends of the VPN connection either for compliance purposes or for leveraging gateway devices that are not currently supported by VPC's VPN solution
      * Software VPN config
    - Timeline

      Description automatically generated
  + **Transit VPC**
  + Table

    Description automatically generated
    - 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 disperce VPCs and remote networks in order 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**
  + Graphical user interface, application

    Description automatically generated
    - 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 SG rule that references a peer SG
      * 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 separate piece of physical hardware
    - There is no single point of failure for communication or a bandwidth bottlenck
    - 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 SG to reference SGs 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 startig with 'pcx-'
  + **PrivateLink**
    - PL simplifies the security of data shared with cloud-based applications by eliminating the exposure of data to the public internet
    - PL provides private connectivity between VPCs, AWS services, and on-premises applicaitons, securely on Amazon network
    - PL makes it easy to connect services across different accounts and VPCs to significantly simplify the network architecture

Graphical user interface, text, application, email

Description automatically generated

* + ***Exam Tip:*** *Know the difference between 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, CL may come up in the exam as a possible (likely incorrect) answer*
  + **VPC Endpoints**
    - An interface endpoint uses AWS PL and is an ENI with a private IP address that serves as an entry point for traffic destined to a supported service
    - Using PL 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 PL access over Inter-Region VPC Peering
      * Applications in an AWS VPC can securely access AWS PL endpoints across AWS Regions using inter-region VPC Peering
      * PL 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-Regioin VPC Peering stays on the global AWS backbone and never traverses the public Internet
    - A GW 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 PL
    - Graphical user interface, application

      Description automatically generated
  + 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
  + GW endpoints are only available for DynamoDB and S3
  + ***Exam Tip:*** *Know wich services use endpoints and GW endpoints. The easiest way to remember this is that GW Endpoints are for S3 and DynamoDB only*
  + **Shared Services VPCs**
    - You can allow other AWS accounts to create their application resources, such as EC2 instances, RDS databases, Redshift clusters, Lambda functions, into shared, centrally-managed 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 SGs
      * VPC sharing participants can reference SG IDs of each other
      * Efficiencies: higher density in subnets, efficient use of VPNs and DC
      * Hard limits can be avoided, for example, 50 VIFs per DC connection through simplified network architecture
      * Costs can be optimized through reuse of NAT GWs, VPC interface endpoints and intra-Availability Zone traffic
    - You can create separate VPCs for each account with the account ownder being responsible for connectivity and security of each VPC
    - With VPC sharing your IT team can own and manage your VPCs and your application developers no longer have to manage or configure VPCs, but they can access them as needed
    - Can also share 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 VPCs using connectivity features, such as PL, Transit Gateway and VPC Peering
    - Can also be used with PL to secure access to resouces shared such as applications behind a NLB
  + **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 CW 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, the following is excluded
      * Traffic that goes to Route53
      * Traffic genereated 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
    - DC is not HA by default, so you need to establish a secondary connection via another DC 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 GW's, create GW's in each AZ with routes for private subnets to use the local gateway