Vertical scalability:

use case : database expansion

Con:

1. main server goes down ,whole system is down

2. for vertical scaling, we have to 1st shut down the server then only changes can be made, which is a challenge bcoz of downtime

Horizontal:

use case: scaling of application servers

cons: not recommended to be used for db expansion

Overview of Key Based Authentication

in new console for key pair aws asks which type of key pair to download ppk or pem, in older version only pem was available

Basics of Firewalls

* security groups are virtual firewalls
* command to check port associated with diff software : netstat -ntlp
* ssh port should not be allowed from internet but for specific ids only
* Also to connect to instance via browser you will have to allow for all ports that is 0.0.0.0

Budget Types: (RUSC)

Cost Budget : when a cost threshold is crossed like 1$  
Usage budget:  based on expected usage %, like 300 hrs  
Savings Plan Budget: associated with the savings plans  
Reservation budget: associated with reserved services

Virtual Private Cloud(VPC)

A VPC consists of:

-Internet Gateways or Virtual Private Gateways

A VPC can have only one internet gateway

-Route Tables

A route table gives direction based on the destination IP.

By default, Every VPC have a route table. Whenever a VPC is created a route table is also created.

-NACLs (Network Access Control lists)

-Security Groups

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

SUBNETS

* Subnets are partitions of a VPC
* you cannot launch unlimited servers into a subnet , limited decided by IPV4 availability
* you cannot launch an instance into a VPC without subnets,  instances are created inside a subnet only and not on VPC level
* 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.

IP Addressing in AWS

* Ec2 actual Private IP is used only for internal communications within the VPC, for comm. via internet we need a different IP, this is done via Public IP or Elastic IP.
* Public IP: dynamic Ip assigned to Ec2, value changes after shut down/stop of instance, also the public IP disappears if ec2 is terminated. No public IP is associated with a stopped instance, The IP section remains empty.
* Elastic IP: associated with aws account, then we associate with the EC2. even if ec2 is terminated, It is still available
* Elastic Ip is hourly charged if it's unassociated or associated with a stopped instance. No charge for IP associated with running instances.
* NOTE: Always assign elastic IP for critical servers in prod.

Understanding and Creating Subnets

VPC spans across all AZs i.e single VPC can be used for all AZs in a single region. But Diff VPCs for diff regions.

subnets are created by dividing the VPC CIDR range .

A subnet must be created in an availability zone, if no zone created while creating subnet, was automatically select any AZ and assign subnet in that AZ

/16- 65,356 available IPS

/24 - 251 IPS

/28 -11 Ips

After launching resources in a subnet, if you go to the subnet console and check for available IPs, its count will decrease to show the remaining IPS available

INTERNET GATEWAYS

* By default Ec2 cannot connect to the internet, nor can anyone from the internet can connect to the EC2.
* To enable internet connectivity, we use Internet Gateway: it allows users to connect to ec2 via the internet and also allows ec2 to connect to the internet to download patches, software, etc.
* To use IG, Ensure that the instances have Public / EIP attached, because users from internet can only connect via public ip/EIP.
* A default IG is also present with the default VPC.
* You can only attach one Internet gateway to a custom VPC.

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

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

Egress-only Internet Gateway:

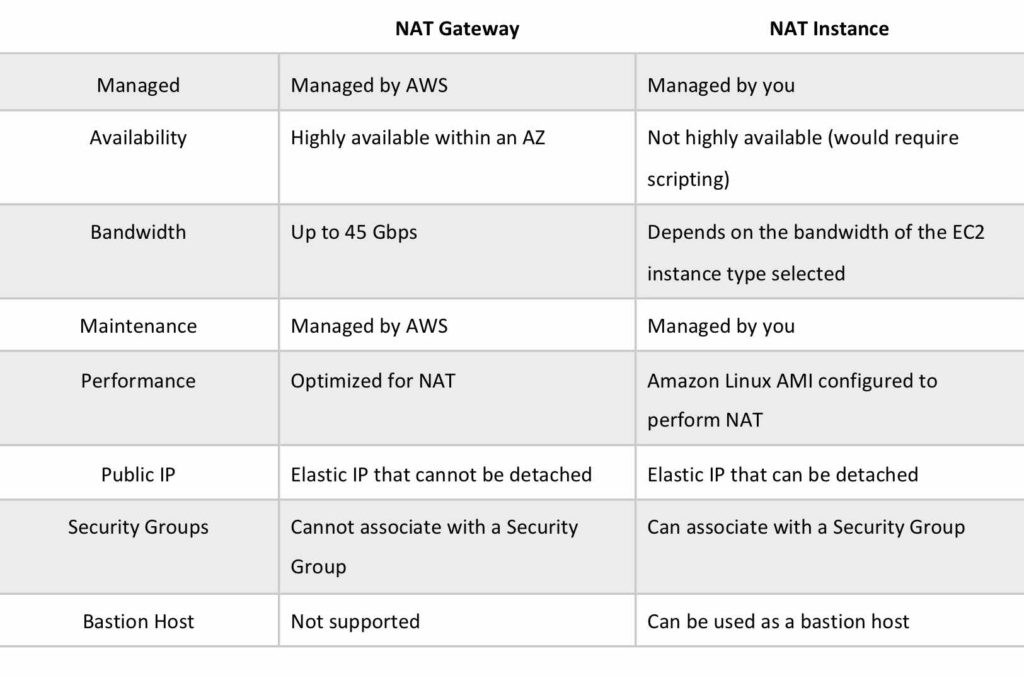
* Provides outbound Internet access for IPv6 addressed instances.
* Prevents inbound access to those IPv6 instances.
* IPv6 addresses are globally unique and are therefore public by default.
* Stateful – forwards traffic from instance to Internet and then sends back the response.
* Must create a custom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

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

* You can use a network address translation (NAT) gateway to enable instances in a private subnet to connect to the internet
* With NAT gateway you have one way connectivity to internet, EC2 can send request to internet but request from internet to EC2 are blocked.
* 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:



TRANSIT GATEWAY:

* A transit gateway is a network transit hub that you can use to interconnect your virtual private clouds (VPC) and on-premises networks.
* Transit Gateway controls how traffic is routed among all the connected spoke networks using route tables.
* Only A vpc or Vpn or direct connect gateway can be attached to a transit Gateway.
* To attach a VPN connection to your transit gateway, you must specify the customer gateway
* Direct connect can be added using direct connect gateway, which is then connected to transit gateway.
* For static VPNs, add the static routes to the transit gateway route table.
* You can peer two transit gateways and route traffic between them, which includes IPv4 and IPv6 traffic. To do this, create a peering attachment on your transit gateway, and specify a transit gateway in another Region. The peer transit gateway can be in your account or a different AWS account.

Limits:

* When you attach a VPC to a transit gateway, you must specify one subnet from each Availability Zone to be used by the transit gateway to route traffic, otherwise zones whose subnet is not attached to transit gateway cannot connect to transit gateway.
* The resources in a VPC attached to a transit gateway cannot access the security groups of a different VPC that is also attached to the same transit gateway.
* A transit gateway doesn't support routing between VPCs with identical CIDRs.

Network ACL (NACL)

NACL - Network Access control list

* With NACLs you can have **permit and deny rules**.
* In NACL the lower the Rule number , the more its priority. Say if the rule no 100 matches the conditions, it will be applied and rules of higher no will not be checked. Recommended to number in multiples of 10.
* **Network ACLs are stateless:** This means **any changes applied to an incoming rule will not be applied to the outgoing rule**. e.g. If you allow an incoming port 80, you would also need to apply the rule for outgoing traffic.
* **NACL default : allow all traffic IN & OUT**
* **Each subnet in your VPC must be associated with a network ACL.** If you don't explicitly associate a subnet with a network ACL, the subnet is automatically associated with the default network ACL.
* You can associate a network ACL with multiple subnets. However, a subnet can be associated with only one network ACL at a time. When you associate a network ACL with a subnet, the previous association is removed.
* **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 in a VPC. If not, they automatically attach to the default ACL.

vs

SG- Security group

* A security group can be understood as a firewall at the Resource level, eg: Ec2 servers, Db servers.
* SG default: **all inbound traffic is denied by default**. When you first 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 evaluate all the rules in them before allowing traffic whereas NACLs do it in the number order, from top to bottom.
* You can specify allow rules, but not deny rules i.e Security group rules are always permissive; you can't create rules that deny access.
* You can specify separate rules for inbound and outbound traffic.
* Security group rules enable you to filter traffic based on protocols and port numbers.
* **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.
* 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 group
* 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.
* Stale security group rules

If your VPC has a VPC peering connection with another VPC, a security group rule can reference another security group in the peer VPC. This allows instances that are associated with the referenced security group and those that are associated with the referencing security group to communicate with each other.

If the owner of the peer VPC deletes the referenced security group, or if you or the owner of the peer VPC deletes the VPC peering connection, the security group rule is marked as stale. You can delete stale security group rules as you would any other security group rule.

Combining Security Group and NACL

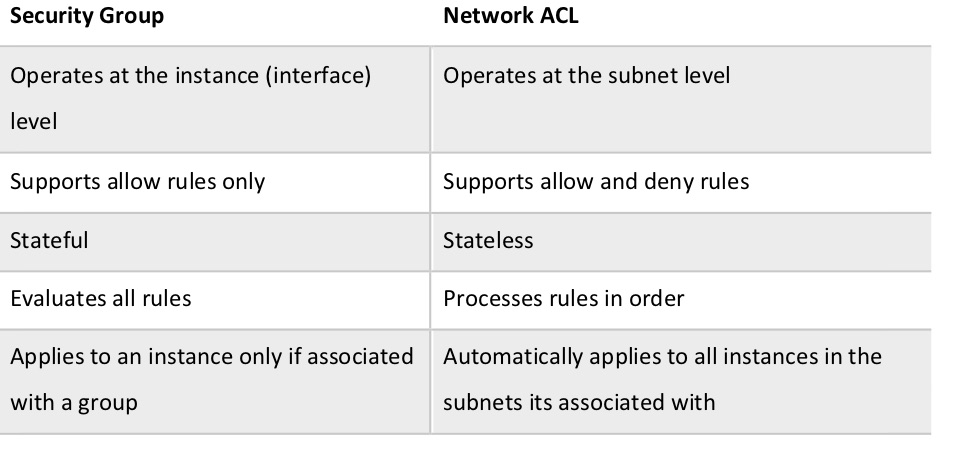
Counts below are default values.

Maximum number of rules that exist per NACL: 20

Maximum number of rules that can exist per Security Group: 60

Maximum number of Security Groups that can exist per instance: 5

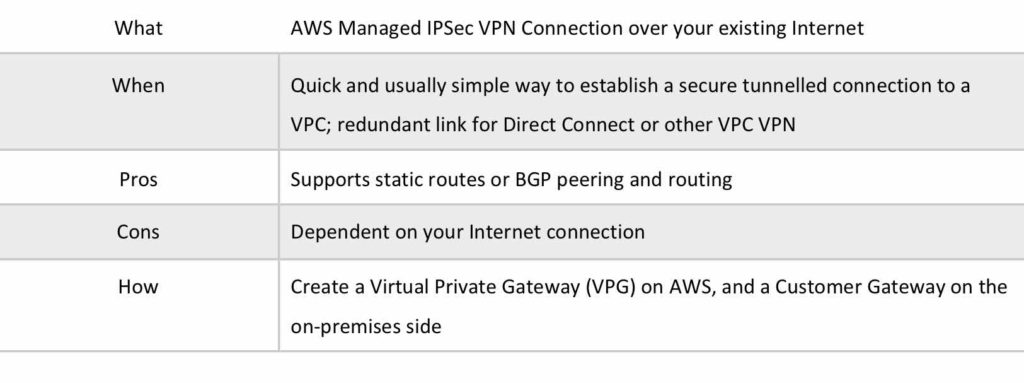
Maximum number of rules that can exist per instance: 5\*60 + 20 = 320



Virtual Private Networks(VPN):

VPN enables you to route traffic from yourself towards destination through itself. Something similar to Proxy.

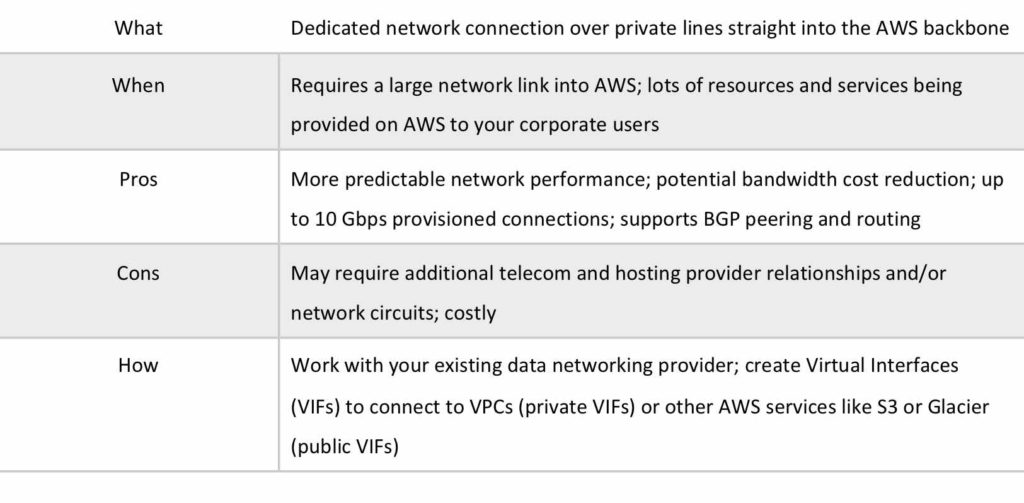
In Corporate environments, VPN is used to connect to instances in a Private Subnet.



* Virtual Private Gateway: The Amazon VPC side of a VPN connection.
* Customer Gateway: Your side of a VPN connection.
* 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**
* 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

* 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.



AWS Direct Connect Plus VPN

* With AWS Direct Connect plus VPN, you can combine one or more AWS Direct Connect dedicated network connections with the Amazon VPC VPN.
* Provides a Backup to your Direct Connect Network.
* 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.

AWS VPN CloudHub

* 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.
* Branches can talk to each other (and provides redundancy).
* Can have Direct Connect connections.
* Hourly rates plus data egress charges.

Software VPN

* 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.

VPC Peering

* 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 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.
* 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.
* ClassicLink allows you to link an EC2-Classic instance to a VPC in your account, within the same Region. This allows you to associate the VPC security groups with the EC2-Classic instance, enabling communication between your EC2-Classic instance and instances in your VPC using private IPv4 addresses. ClassicLink removes the need to make use of public IPv4 addresses or Elastic IP addresses to enable communication between instances in these platforms. For more information about private and public IPv4 addresses, see IP Addressing in your VPC.
* ClassicLink is available to all users with accounts that support the EC2-Classic platform, and can be used with any EC2-Classic instance.
* There is no additional charge for using ClassicLink. Standard charges for data transfer and instance hour usage apply.

Elastic Network Interface (ENI)

* Also referred to as a “Network Interface”, is a virtual network interface that you can attach to an EC2 instance in a VPC. Network interfaces are available only for instances running in a VPC.
* You can attach a network interface to an instance:
* When it’s running (hot attach)
* When it’s stopped (warm attach)
* When the instance is being launched (cold attach)
* Note that an Elastic Network Interface can only be reassigned within the same AZ, and cannot be assigned to an EC2 instance in a different AZ

VPC Endpoints

A VPC endpoint enables connections between a virtual private cloud (VPC) and supported services, without requiring that you use an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Therefore, your VPC is not exposed to the public internet.

VPC endpoints are virtual devices. They are horizontally scaled, redundant, and highly available VPC components. The following are the **different types of 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.
* 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 DyanmoDB
  + 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 have to 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.
* **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.

**Building Scalable Applications**

162. SQS - Practical

when message is recieved by consumer in SQS queue, it is not automatically deleted from the queue. It resides there unless customer deletes it, or its retention periods end.