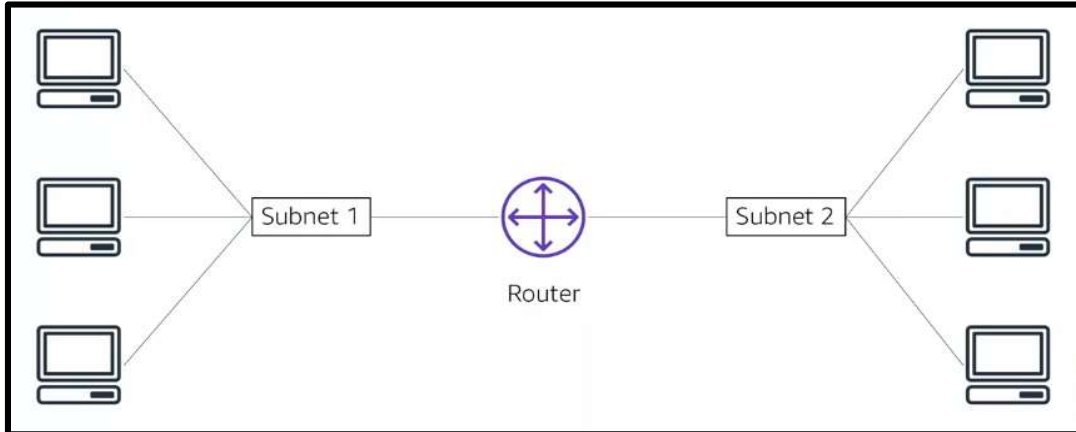


Networks

- A computer network is two or more client machines that are connected together to share resources.
- A network can be logically partitioned into subnets.
- Networking requires a networking device (such as a router or switch) to connect all the clients together and enable communication between them.



IP Addresses

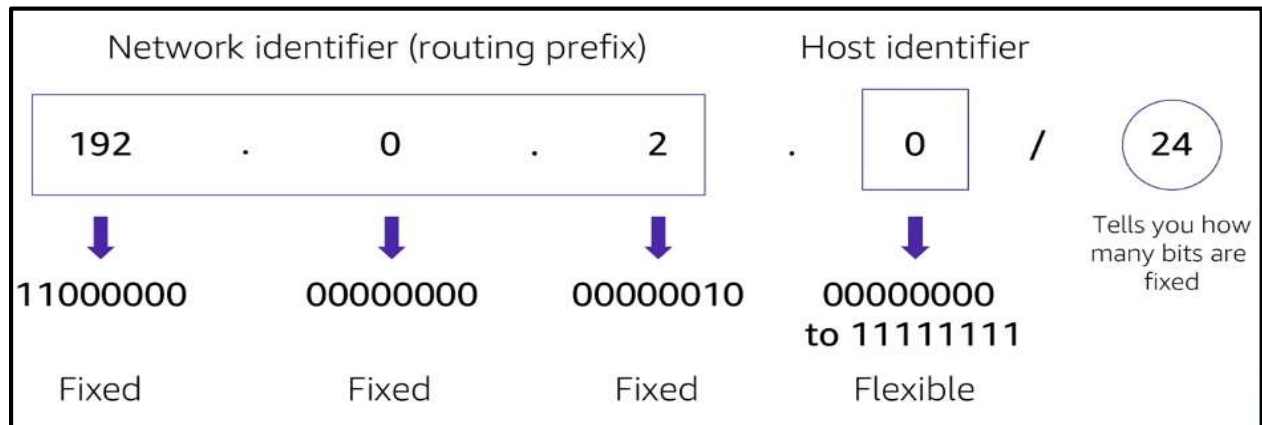
- Each client machine in a network has a unique Internet Protocol (IP) address that identifies it.
- An IP address is a numerical label in decimal format. Machines convert that decimal number to a binary format.
- In this example, the IP address is 192.0.2.0. Each of the four dot (.)-separated numbers of the IP address represents 8 bits in octal number format. That means each of the four numbers can be anything from 0 to 255. The combined total of the four numbers for an IP address is 32 bits in binary format.

192	.	0	.	2	.	0
↓		↓		↓		↓
11000000		00000000		00000010		00000000

IPv4 (32-bit) address: 192.0.2.0

IPv6 (128-bit) address: 2600:1f18:22ba:8c00:ba86:a05e:a5ba:00FF

CIDR (Classless Inter-Domain Routing)



Public IPv4 address

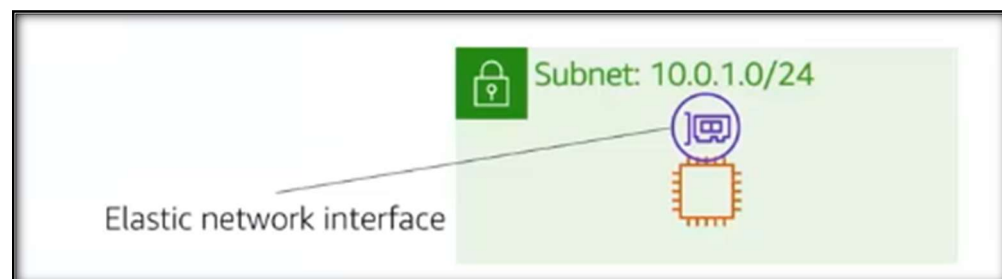
- Manually assigned through an Elastic IP address
- Automatically assigned through the auto-assign public IP address settings at the subnet level

Elastic IP address

- Associated with an AWS account
- Can be allocated and remapped anytime
- Additional costs might apply

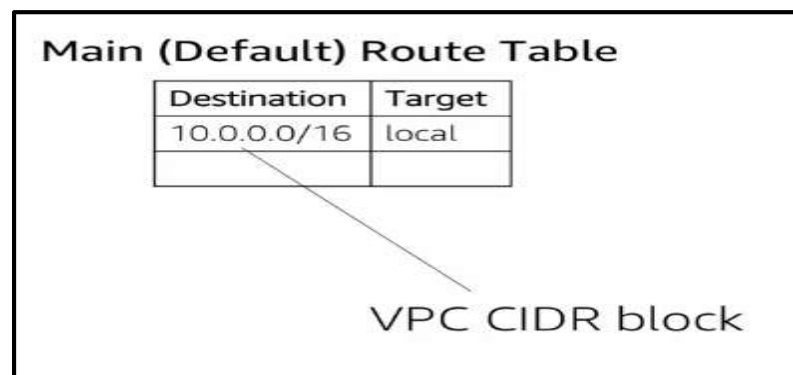
ENI interface

- An elastic network interface is a virtual network interface that you can attach or detach from an instance in a VPC.
- A network interface's attributes follow it when it is reattached to another instance.
- When you move a network interface from one instance to another, network traffic is redirected to the new instance.
- Each instance in your VPC has a default network interface (the primary network interface) that is assigned a private IPv4 address from the IPv4 address range of your VPC.
- You cannot detach a primary network interface from an instance. You can create and attach an additional network interface to any instance in your VPC.
- The number of network interfaces you can attach varies by instance type.



Route Tables

- A route table contains a set of rules (called routes) that directs network traffic from your subnet.
- Each route specifies a destination and a target.
- The destination is the destination CIDR block where you want traffic from your subnet to go.
- The target is the target that the destination traffic is sent through.
- By default, every route table that you create contains a local route for communication in the VPC.
- You can customize route tables by adding routes. You cannot delete the local route entry that is used for internal communications.
- Each subnet in your VPC must be associated with a route table.
- The main route table is the route table is automatically assigned to your VPC.
- It controls the routing for all subnets that are not explicitly associated with any other route table.
- A subnet can be associated with only one route table at a time, but you can associate multiple subnets with the same route table

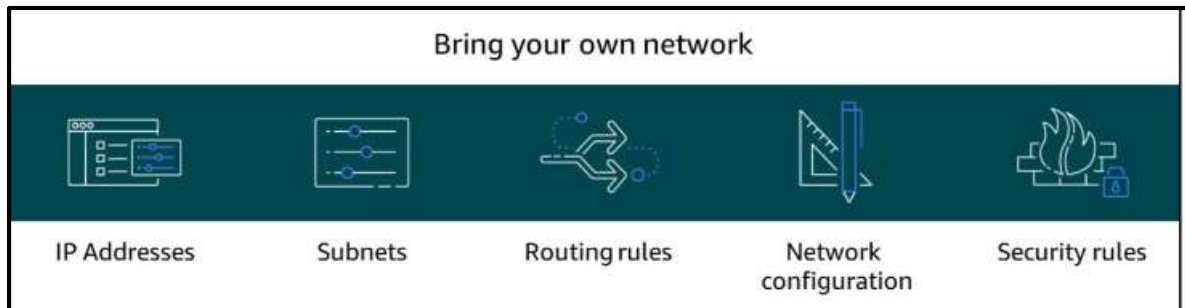


AWS VPC (Virtual Private Cloud)

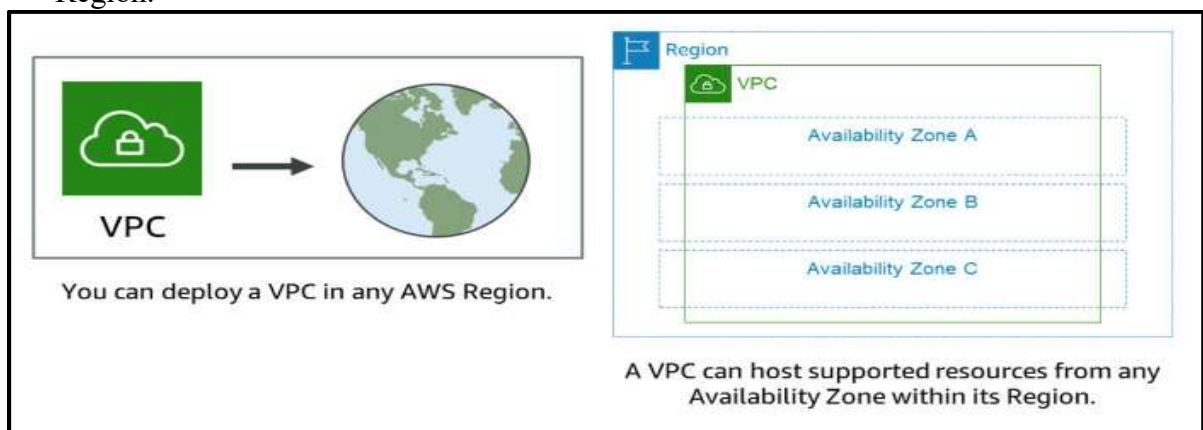
- Amazon Virtual Private Cloud (Amazon VPC) is a service that enables you to provision a logically isolated section of the AWS Cloud (called a virtual private cloud, or VPC) where you can launch your AWS resources.
- Amazon VPC gives you control over your virtual networking resources. For example, you can select your own IP address range, create subnets, and configure route tables and network gateways.
- You can use both IPv4 and IPv6 in your VPC for secure access to resources and applications.
- The default quota is 5 VPCs per Region. However, you can
- request an increase for this quota
- You can also customize the network configuration for your VPC. For example, you can create a public subnet for your web servers that can access the public internet. You can

place your backend systems (such as databases or application servers) in a private subnet with no public internet access.

- Finally, you can use multiple layers of security to help control access to Amazon Elastic Compute Cloud (Amazon EC2) instances in each subnet.
- These security layers include security groups and network access control lists (network ACLs)



- A VPC belongs to a single AWS Region. A VPC spans all the Availability Zones in a Region, so it can host supported resources from any Availability Zone within its Region.



VPC CIDR

- When you create a VPC, you provide the set of private IP addresses that you want instances in your VPC to use.
- You specify this set of addresses as a Classless Inter-Domain Routing (CIDR) block—for example, 10.0.0.0/16. This is the primary CIDR block for your VPC.
- You can assign block sizes of between /28 (16 IP addresses) and /16 (65,536 IP addresses).
- Amazon VPC supports IPv4 and IPv6 addressing and has different CIDR block size limits for each.
- By default, all VPCs and subnets must have IPv4 CIDR blocks—you can't change this behavior. You can optionally associate an IPv6 CIDR block with your VPC

0.0.0.0/0	= All IP addresses	
10.22.33.44/32	= 10.22.33.44	
10.22.33.0/24	= 10.22.33.*	
10.22.0.0/16	= 10.22.*.*	

CIDR	Total IP addresses
/28	16
...	...
/20	4,096
/19	8,192
/18	16,384
/17	32,768
/16	65,536

Subnets

- Amazon VPC enables you to provision virtual private clouds (VPCs).
- A VPC is a virtual network that is logically isolated from other virtual networks in the AWS Cloud.
- A VPC is dedicated to your account. VPCs belong to a single AWS Region and can span multiple Availability Zones.
- After you create a VPC, you can divide it into one or more subnets.
- A subnet is a segment or partition of a VPC's IP address range where you can allocate a group of resources.
- A subnet is a range of IP addresses in a VPC. Subnets belong to a single Availability Zone.
- You can create subnets in different Availability Zones for high availability.
- Subnets are generally classified as public or private.
- Public subnets have direct access to the internet, but private subnets do not.
- When you create a subnet, you specify the CIDR block for the subnet, which is a subset of the VPC CIDR block.
- Subnet CIDR blocks cannot overlap.
- Though each subnet must reside entirely within one Availability Zone and cannot span zones, each Availability Zone can have one or more subnets.

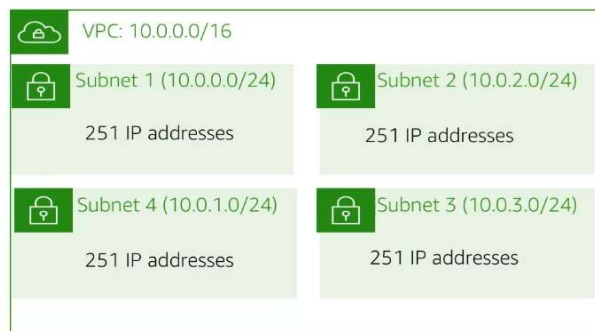


- AWS reserves the first four IP addresses and the last IP address in each subnet CIDR block. For example, in a subnet with CIDR block 10.0.0.0/24, AWS reserves the following five IP addresses for:
 - 10.0.0.0: Network address
 - 10.0.0.1: VPC local router or for Internal Communication
 - 10.0.0.2: Domain Name System (DNS) resolution
 - 10.0.0.3: Future use
 - 10.0.0.255: Network broadcast address

CIDR


- A common method to describe networks is Classless Inter-Domain Routing (CIDR).
- The CIDR address is expressed as follows:
 - An IP address (which is the first address of the network)
 - Next, a slash character (/)
 - Finally, a number that tells you how many bits of the routing prefix must be fixed or allocated for the network identifier.
- In this example, the CIDR address is 192.0.2.0/24.
 - The last number (24) tells you that the first 24 bits must be fixed.
 - The last 8 bits are flexible, which means that 2^8 (or 256) IP addresses are available for the network, which range from 192.0.2.0 to 192.0.2.255.
 - The fourth decimal digit is allowed to change from 0 to 255.
- If the CIDR was 192.0.2.0/16,
 - the last number (16) tells you that the first 16 bits must be fixed.
 - The last 16 bits are flexible, which means that 2^{16} (or 65,536) IP addresses are available for the network, ranging from 192.0.0.0 to 192.0.255.255.
 - The third and fourth decimal digits can each change from 0 to 255.

Example: A VPC with an IPv4 CIDR block of 10.0.0.0/16 has 65,536 total IP addresses. The VPC has four equal-sized subnets. Only 251 IP addresses are available for use by each subnet.



IP Addresses for CIDR block 10.0.0.0/24	Reserved for
10.0.0.0	Network address
10.0.0.1	Internal communication
10.0.0.2	Domain Name System (DNS) resolution
10.0.0.3	Future use
10.0.0.255	Network broadcast address

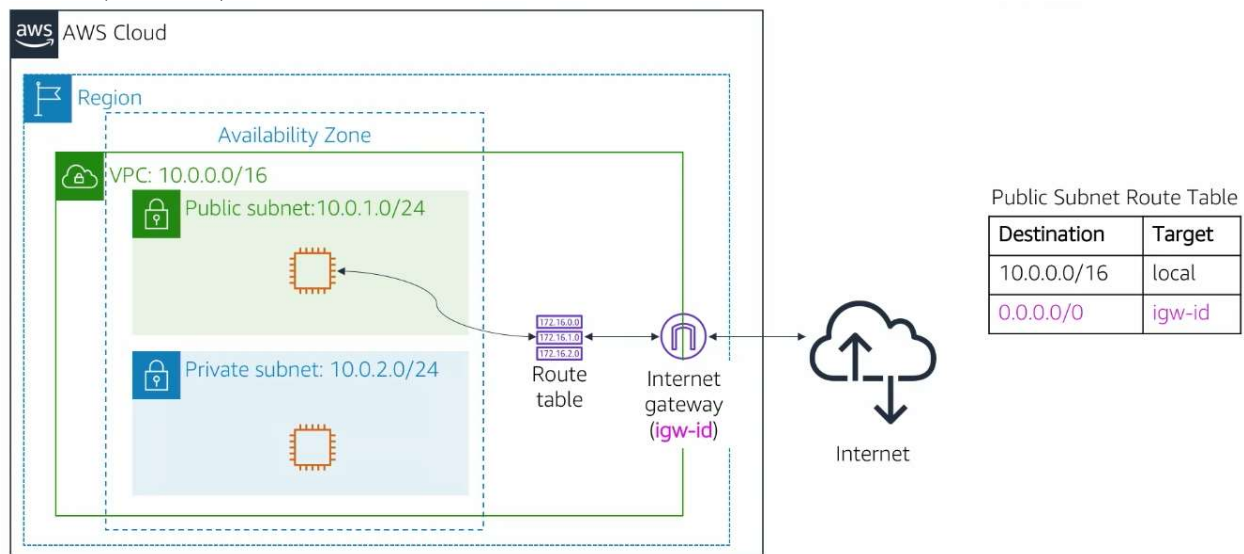
VPC Connectivity Components

VPC connectivity scenarios and solutions 		
If You Need to:	Consider Using:	Solution Category:
Connect a private subnet to the internet	<ul style="list-style-type: none">NAT gatewayNAT instance	<ul style="list-style-type: none">EC2 instance connectivity
Connect a VPC to another VPC	<ul style="list-style-type: none">VPC peering	<ul style="list-style-type: none">Amazon VPC to Amazon VPC
Connect a VPC to an external network	<ul style="list-style-type: none">AWS Site-to-Site VPNAWS Direct Connect plus VPNAWS VPN CloudHub	<ul style="list-style-type: none">Network to Amazon VPCVPN connectivity
Connect a VPC to AWS services without leaving the AWS network	<ul style="list-style-type: none">AWS PrivateLinkVPC Gateway endpoint	<ul style="list-style-type: none">Amazon VPC to Amazon VPCVPC Gateway endpoint
Connect a VPC to multiple VPCs and external networks	<ul style="list-style-type: none">AWS Transit Gateway	<ul style="list-style-type: none">Network to Amazon VPCAmazon VPC to Amazon VPC

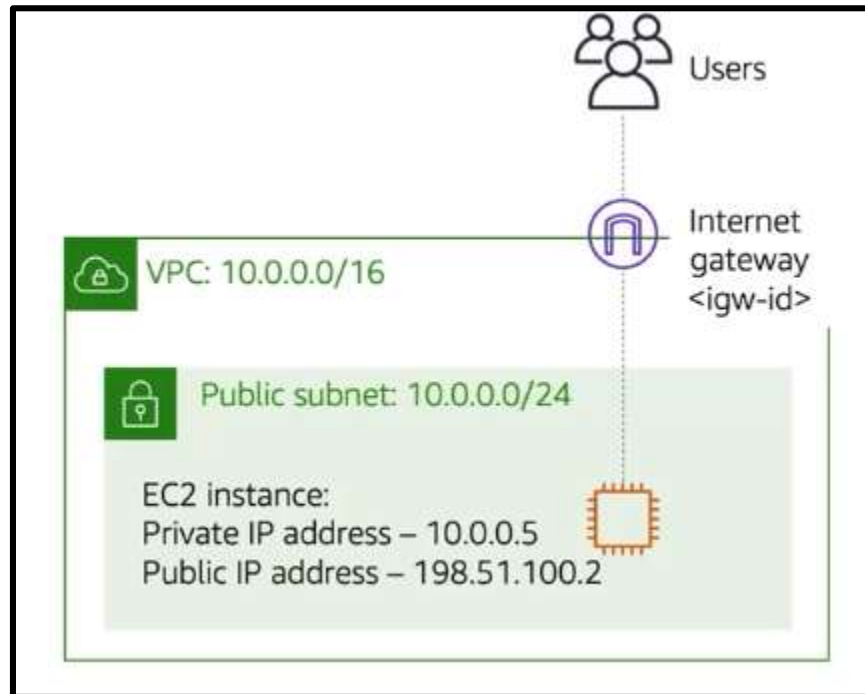
© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Internet Gateway

- An internet gateway connects public subnet instances to the internet.
- An internet gateway is a scalable, 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, and
 - to perform network address translation for instances that were assigned public IPv4 addresses
- To make a subnet public, you attach an internet gateway to your VPC and add a route to the route table to send non-local traffic through the internet gateway to the internet (0.0.0.0/0).

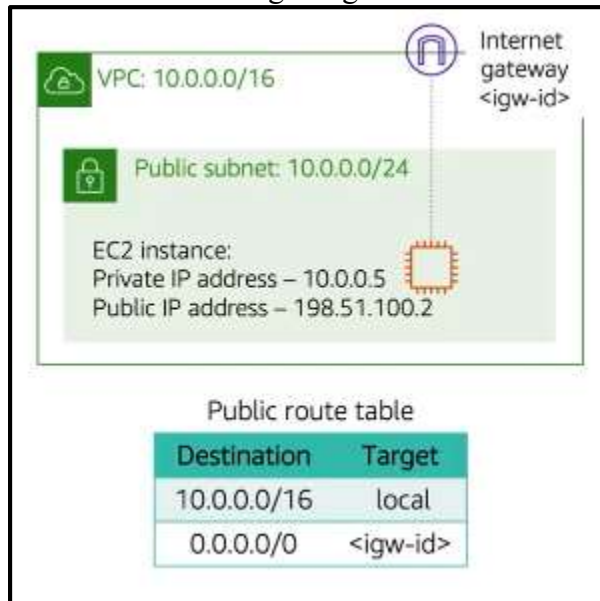


- An internet gateway supports IPv4 and IPv6 traffic.
- An internet gateway serves two purposes:
 - First, it provides a target in your VPC route tables for internet-routable traffic.
 - Second, the internet gateway performs network address translation (NAT) for instances that were assigned public IPv4 addresses.
- To make a subnet public, you must first create an internet gateway and attach it to your VPC.



- Next, you must update the route table associated with the subnet you want to connect to the internet.
- A route table contains a set of rules, called routes.
- Routes are used to determine where network traffic is directed.
- When you create a VPC, it automatically has a main route table.
- Initially, the main route table (and every route table in a VPC) contains only a single local route that enables communication for all the resources in the VPC.
- You can't modify the local route in a route table.
- When you launch an instance in the VPC, the local route automatically covers that instance.
- You don't need to add the new instance to a route table.
- You can create additional custom route tables for your VPC.
- Each subnet in your VPC must be associated with a route table, which controls the routing for the subnet.
- If you don't explicitly associate a subnet with a particular route table, the subnet is implicitly associated with and uses the main route table.
- A subnet can be associated with only one route table at a time, but you can associate multiple subnets with the same route table.
- You can create custom route tables for each subnet to enable granular routing for destinations.

- To send non-local traffic through the internet gateway to the internet, create a route with
- destination 0.0.0.0/0 and target <igw-id> in the route table associated with the subnet.



- To create a public subnet to allow communication between instances in your VPC and the internet, you must:
 - Attach an internet gateway to your VPC
 - Add a route to your subnet's route table that directs internet-bound traffic to the internet gateway
 - Make sure that your instances have public IP or Elastic IP addresses
 - Make sure that your security groups and network ACLs allow relevant traffic to flow

To create a **public subnet** to allow communication between instances in your VPC and the internet, you must:



Attach an **internet gateway** to your VPC.

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	<igw-id>

Point your instance subnet's **route table** to the internet gateway.



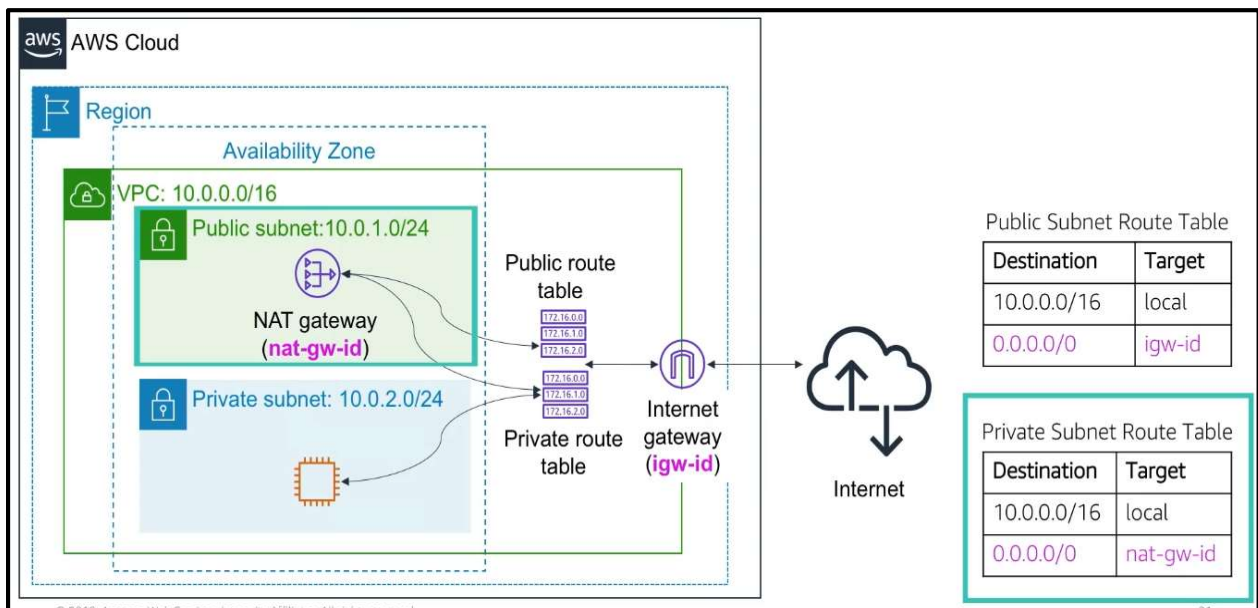
Make sure that your instances have **public IP or Elastic IP** addresses.

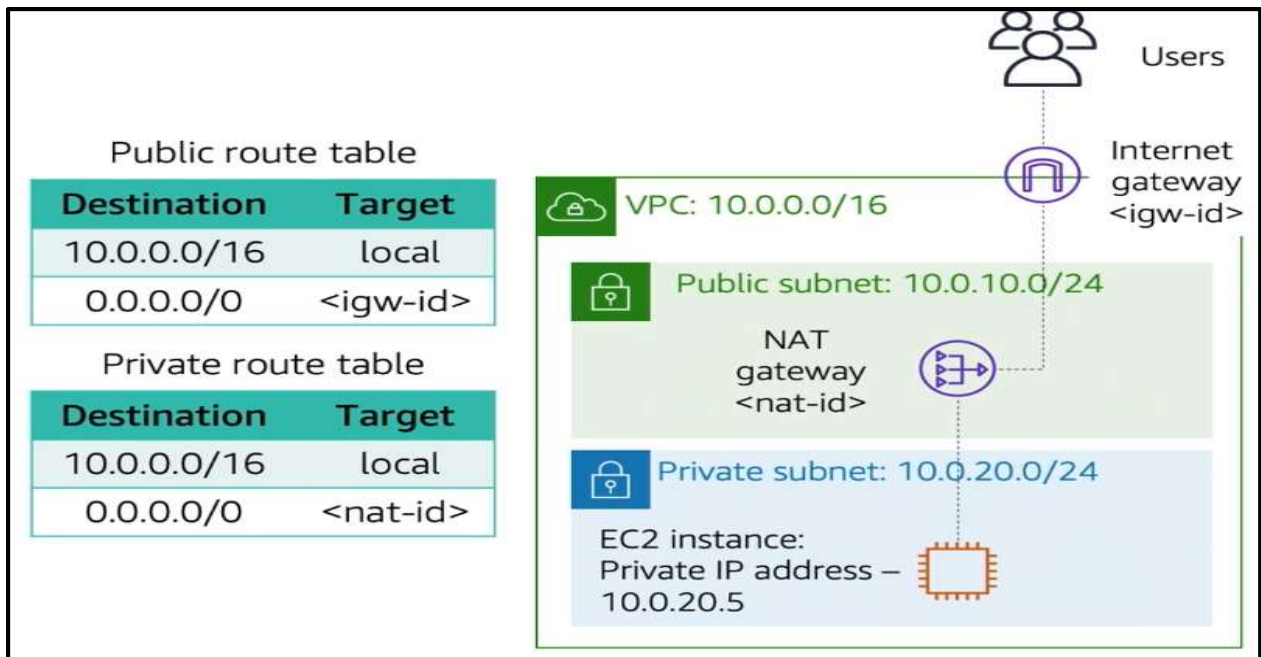


Make sure that your **security groups** and **network ACLs** allow relevant traffic to flow.

NAT Gateway

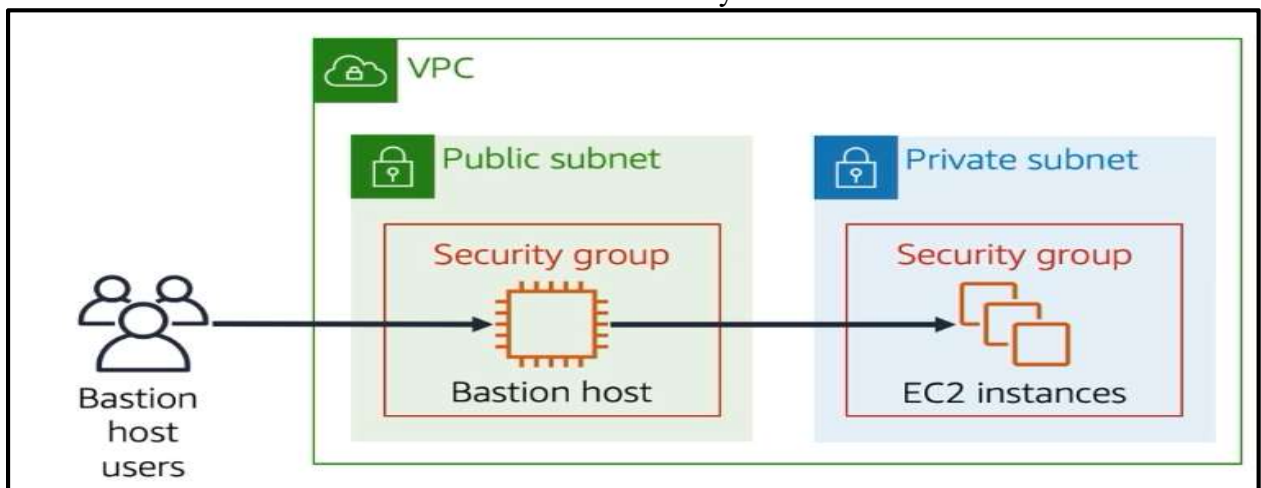
- To connect instances in your private subnet to the internet or other AWS services, you need a network address translation (NAT) gateway.
- A network address translation (NAT) gateway enables instances in a private subnet to connect to the internet or other AWS services, but prevents the internet from initiating a connection with those instances.
- To create a NAT gateway, you must specify the public subnet in which the NAT gateway should reside.
- You must also specify an Elastic IP address to associate with the NAT gateway when you create it.
- After you create a NAT gateway, you must update the route table that is associated with one or more of your private subnets to point internet-bound traffic to the NAT gateway. Thus, instances in your private subnets can communicate with the internet.
- You can also use a NAT instance in a public subnet in your VPC instead of a NAT gateway.
- However, a NAT gateway is a managed NAT service that provides better availability, higher bandwidth, and less administrative effort.
- For common use cases, AWS recommends that you use a NAT gateway instead of a NAT instance.





Bastion Hosts

- A bastion host is a server that provides access to a private network from an external network, such as the internet.
- You can use a bastion host to minimize the chances of penetration and potential attack on resources in your private network.
- For example, suppose you want to allow connections from an external network to Linux instances in a private subnet of your VPC via Secure Shell, or SSH.
- You can use a bastion host to mitigate the risk of allowing these external SSH connections to the instances in the private subnet.
- A bastion host typically runs on an EC2 instance in a public subnet of your VPC, as shown in this example.
- The Linux instances in the private subnet are in a security group that allows SSH access from the security group attached to the bastion host.
- Bastion host users connect to the bastion host so they can connect to the Linux instances.



SUMMARY

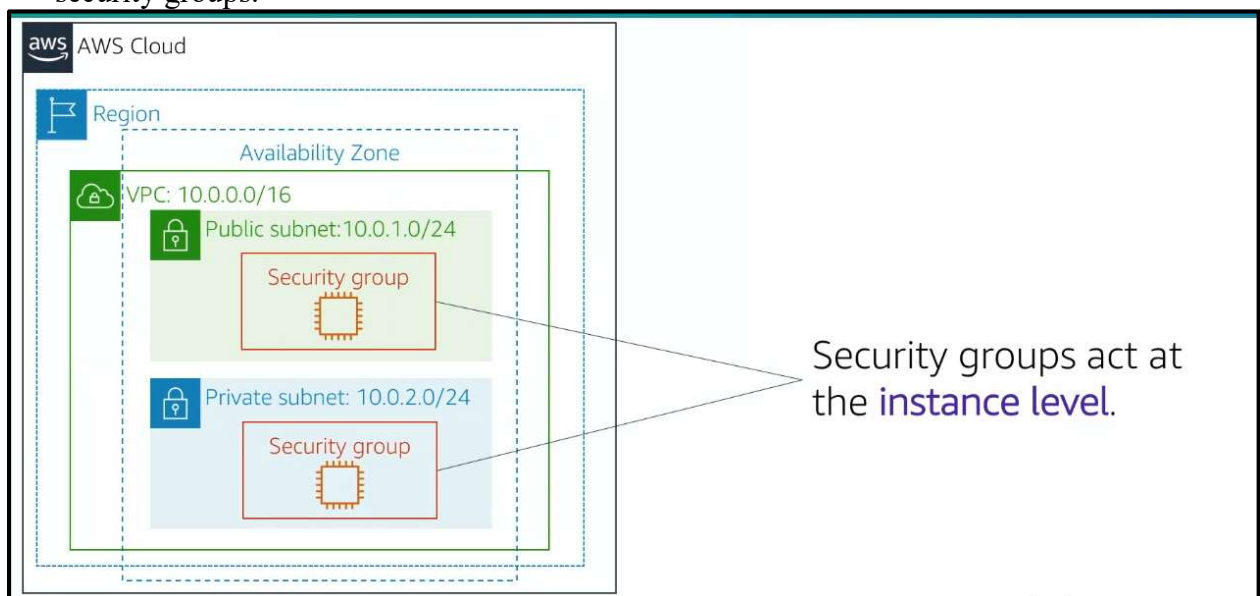
- An internet gateway allows communication between instances in your VPC and the internet.
- Route tables control traffic from your subnet or gateway.
- Elastic IP addresses are static, public IPv4 addresses that can be associated with an instance or elastic network interface.
- They can be remapped to another instance in your account.
- NAT gateways enable instances in the private subnet to initiate outbound traffic to the internet or other AWS services.
- A bastion host is a server whose purpose is to provide access to a private network from an external network, such as the internet

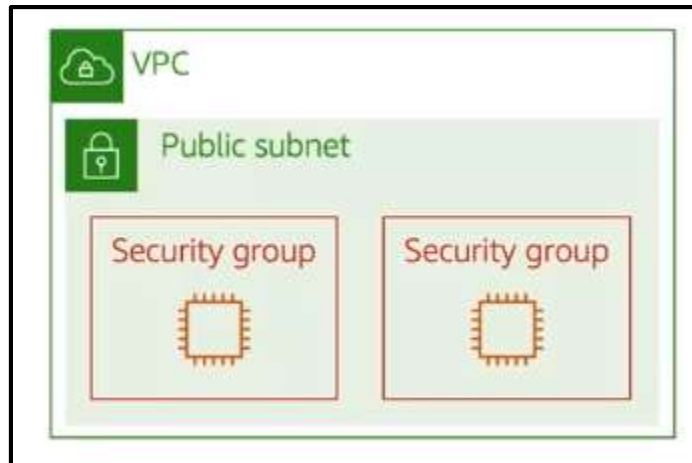
VPC Security

- Security Groups
- Network Access Control List (Network ACL or NACL)

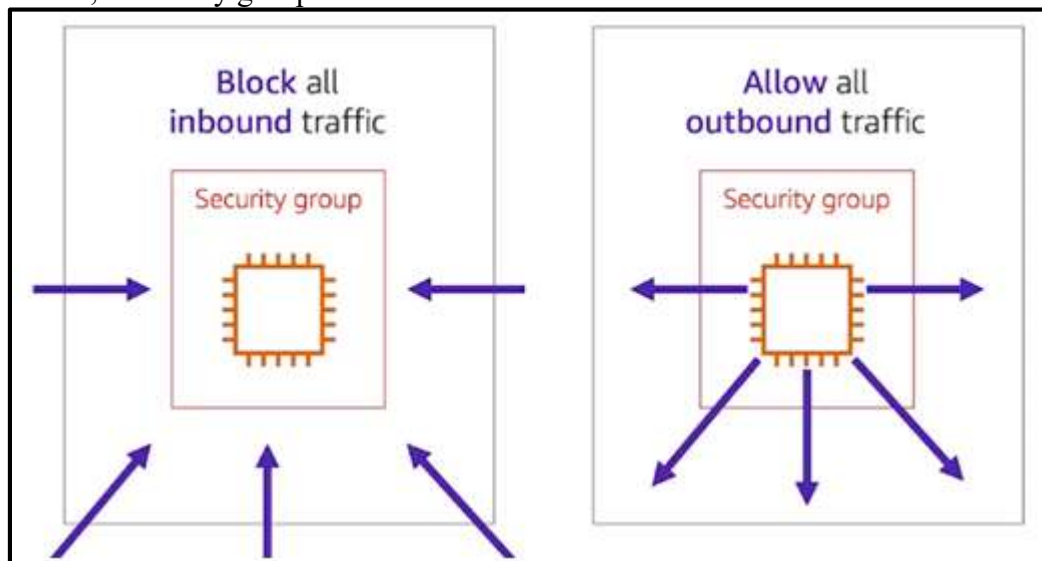
Security Groups

- A security group acts as a virtual firewall for your instance, and it controls inbound and outbound traffic.
- Security groups act at the instance level, not the subnet level.
- Therefore, each instance in a subnet in your VPC can be assigned to a different set of security groups.





- At the most basic level, a security group is a way for you to filter traffic to your instances.
- Security groups have rules that control the inbound and outbound traffic.
- When you create a security group, it has no inbound rules. Therefore, no inbound traffic that originates from another host to your instance is allowed until you add inbound rules to the security group.
- By default, a security group includes an outbound rule that allows all outbound traffic.

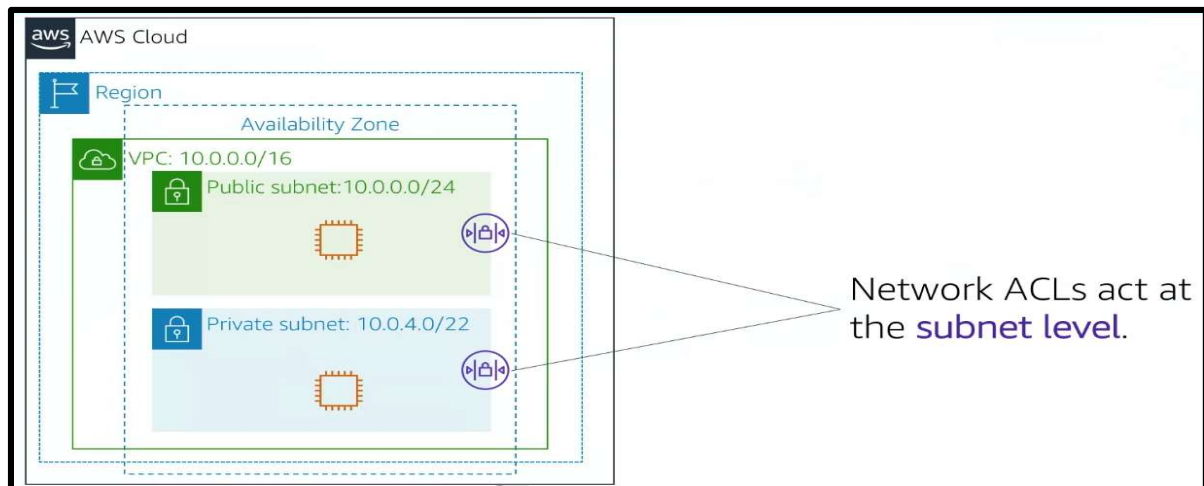


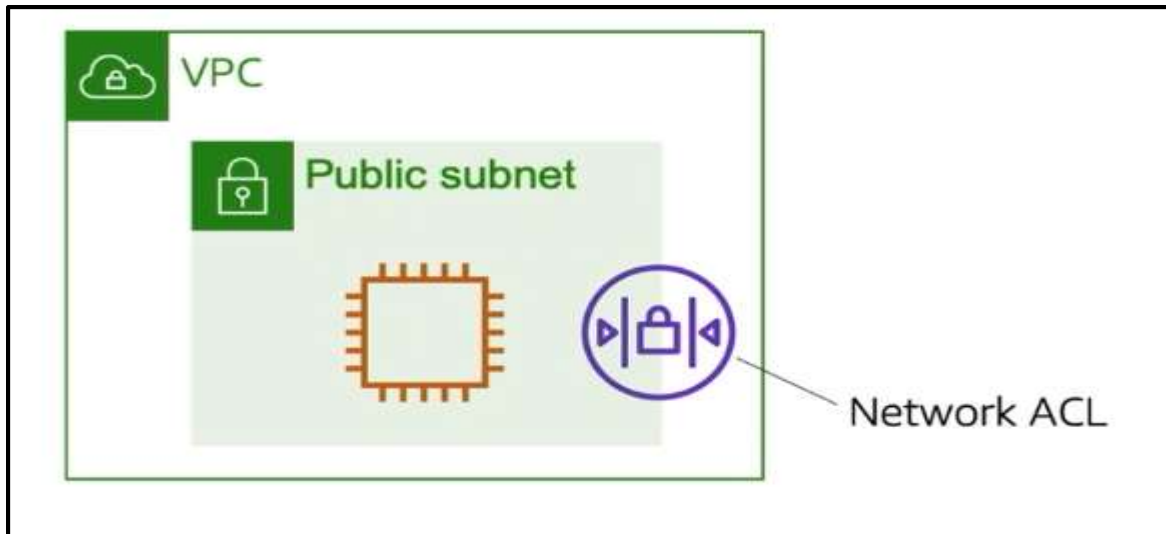
- You can remove the rule and add outbound rules that allow specific outbound traffic only.
- If your security group has no outbound rules, no outbound traffic that originates from your instance is allowed.
- When you create a custom security group, you can specify allow rules, but not deny rules.
- For example, when you create a public subnet for the instances that host your web application, the last step is to create a security group that allows HTTP, HTTPS and SSH traffic to those instances and allows all outgoing traffic.

Inbound				
Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	All web traffic
HTTPS	TCP	443	0.0.0.0/0	All web traffic
SSH	TCP	22	54.24.12.19/32	Office address
Outbound				
Type	Protocol	Port Range	Source	Description
All traffic	All	All	0.0.0.0/0	
All traffic	All	All	::/0	

Network ACL (NACL)

- A network access control list (Network ACL) is an optional layer of security for your Amazon VPC.
- NACL act at the subnet level.
- It acts as a firewall for controlling traffic in and out of one or more subnets.
- To add another layer of security to your VPC, you can set up network ACLs with rules that are similar to your security groups.
- 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.
- A network ACL has separate inbound and outbound rules, and each rule can either allow or deny traffic.
- Your VPC automatically comes with a modifiable default network ACL. By default, it allows all inbound and outbound IPv4 traffic and, if applicable, IPv6 traffic.
- The table shows a default network ACL:

Inbound					
Rule #	Type	Protocol	Port Range	Source	Allow/Deny
100	All IPv4 traffic	All	All	0.0.0.0/0	ALLOW
*	All IPv4 traffic	All	All	0.0.0.0/0	DENY
Outbound					
Rule #	Type	Protocol	Port Range	Source	Allow/Deny
100	All IPv4 traffic	All	All	0.0.0.0/0	ALLOW
*	All IPv4 traffic	All	All	0.0.0.0/0	DENY

- Network ACLs are stateless, which means that no information about a request is maintained after a request is processed. Return traffic must be explicitly allowed by rules.
- You can create a custom network ACL and associate it with a subnet. By default, each custom network ACL denies all inbound and outbound traffic until you add rules.

Nacl-11223344

Inbound:

Rules # 100: SSH 172.31.1.2/32 **ALLOW**

Rules # *: ALL traffic 0.0.0.0/0 **DENY**

Outbound:

Rules # 100: Custom TCP 172.31.1.2/31 **ALLOW**

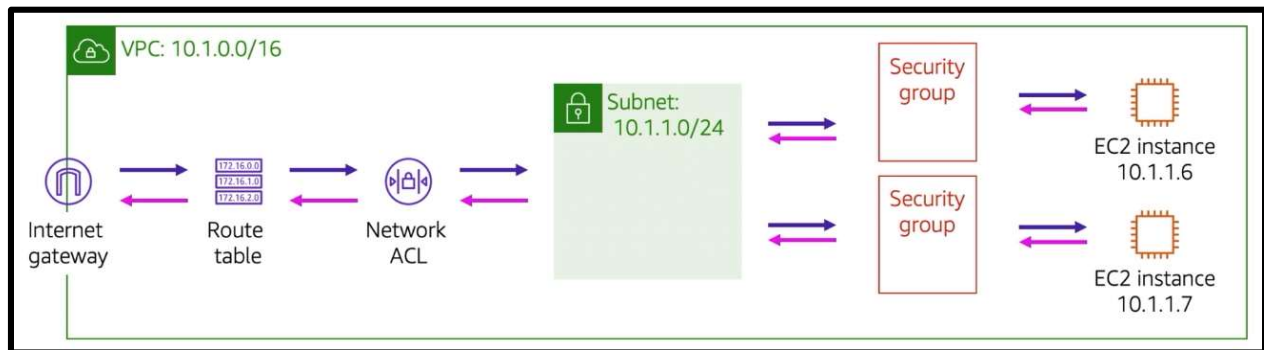
Rules # *: All traffic 0.0.0.0/0 **DENY**

- A network ACL contains a numbered list of rules that are evaluated in order, starting with the lowest numbered rule.
- The purpose is to determine whether traffic is allowed in or out of any subnet that is associated with the network ACL.
- The highest number that you can use for a rule is 32,766.
- AWS recommends that you create rules in increments (for example, increments of 10 or 100) so that you can insert new rules where you need them later.

Security Groups Vs Network ACL

Attribute	Security Groups	Network ACLs
Scope	Instance level	Subnet level
Supported Rules	Allow rules only	Allow and deny rules
State	Stateful (return traffic is automatically allowed, regardless of rules)	Stateless (return traffic must be explicitly allowed by rules)
Order of Rules	All rules are evaluated before decision to allow traffic	Rules are evaluated in number order before decision to allow traffic

VPC Multi-layer Defense



- As a best practice, we should secure your VPC Environment with multiple layers of defense.
- By running your infrastructure in a VPC, you can control which instances are exposed to the internet.
- You can define both security groups and network ACLs to further protect your infrastructure at the infrastructure and subnet levels, respectively.
- Additionally, you should secure your instances with a firewall at the operating system level, and follow other security best practices.
- When you implement both network ACLs and security groups as a defense-in-depth way of controlling traffic, a mistake in the configuration of one of these controls will not expose the host to unwanted traffic.

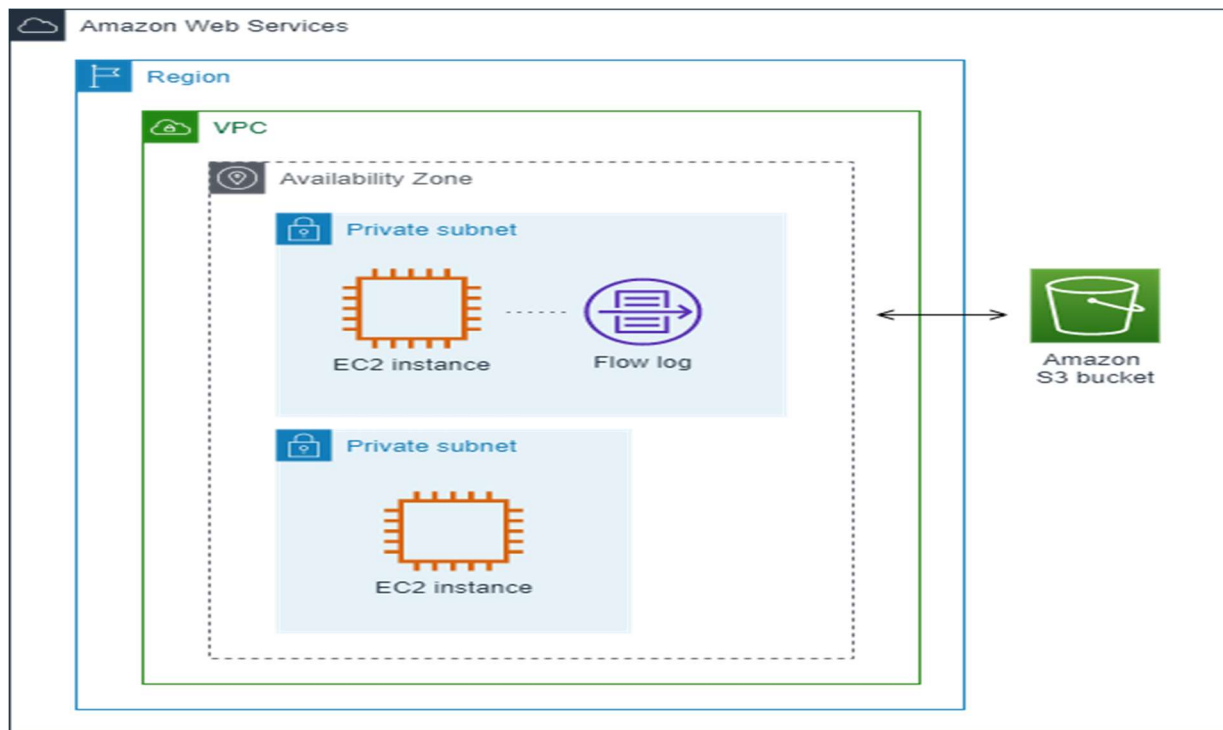
VPC Reachability Analyzer

- Reachability Analyzer is a configuration analysis tool that enables you to perform connectivity testing between a source resource and a destination resource in your virtual private clouds (VPCs).
- When the destination is reachable, Reachability Analyzer produces hop-by-hop details of the virtual network path between the source and the destination.
- When the destination is not reachable, Reachability Analyzer identifies the blocking component. For example, paths can be blocked by configuration issues in a security group, network ACL, route table, or load balancer.
- You can use Reachability Analyzer to do the following:
 - Troubleshoot connectivity issues caused by network misconfiguration.
 - Verify that your network configuration matches your intended connectivity.
 - Automate the verification of your connectivity intent as your network configuration changes.
- To use Reachability Analyzer, you specify the path for the traffic from a source to a destination.
- For example, you could specify an internet gateway as the source, an EC2 instance as the destination, 22 as the destination port, and TCP as the protocol. This would allow you to verify that you can connect to the EC2 instance through the internet gateway using SSH.
- If there are multiple reachable paths between a source and a destination, Reachability Analyzer identifies and displays the shortest path.
- VPC Reachability Analyzer supports the following resource types as sources and destinations:
 - Instances
 - Internet gateways
 - Network interfaces
 - Transit gateways
 - Transit gateway attachments
 - VPC endpoints
 - VPC peering connections
 - VPN gateways

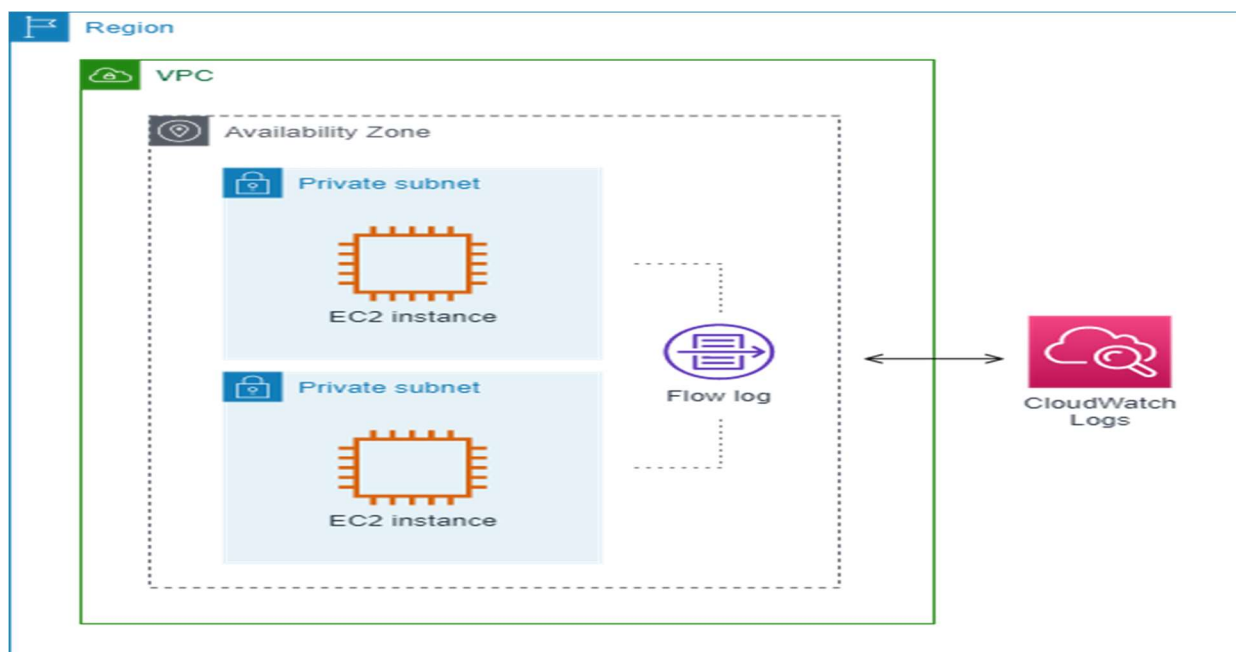
VPC Flow Logs

- VPC Flow Logs is a feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC.
- Flow log data can be published to the following locations: Amazon CloudWatch Logs, Amazon S3, or Amazon Kinesis Data Firehose.
- After you create a flow log, you can retrieve and view the flow log records in the log group, bucket, or delivery stream that you configured.
- Flow logs can help you with a number of tasks, such as:
 - Diagnosing overly restrictive security group rules
 - Monitoring the traffic that is reaching your instance
 - Determining the direction of the traffic to and from the network interfaces
- Flow log data is collected outside of the path of your network traffic, and therefore does not affect network throughput or latency.
- You can create or delete flow logs without any risk of impact to network performance.
- You can create a flow log for a VPC, a subnet, or a network interface. If you create a flow log for a subnet or VPC, each network interface in that subnet or VPC is monitored.
- Flow log data for a monitored network interface is recorded as *flow log records*, which are log events consisting of fields that describe the traffic flow.
- To create a flow log, you specify:
 - The resource for which to create the flow log
 - The type of traffic to capture (accepted traffic, rejected traffic, or all traffic)
 - The destinations to which you want to publish the flow log data

In the following example, you create a flow log that captures accepted traffic for the network interface for one of the EC2 instances in a private subnet and publishes the flow log records to an Amazon S3 bucket.

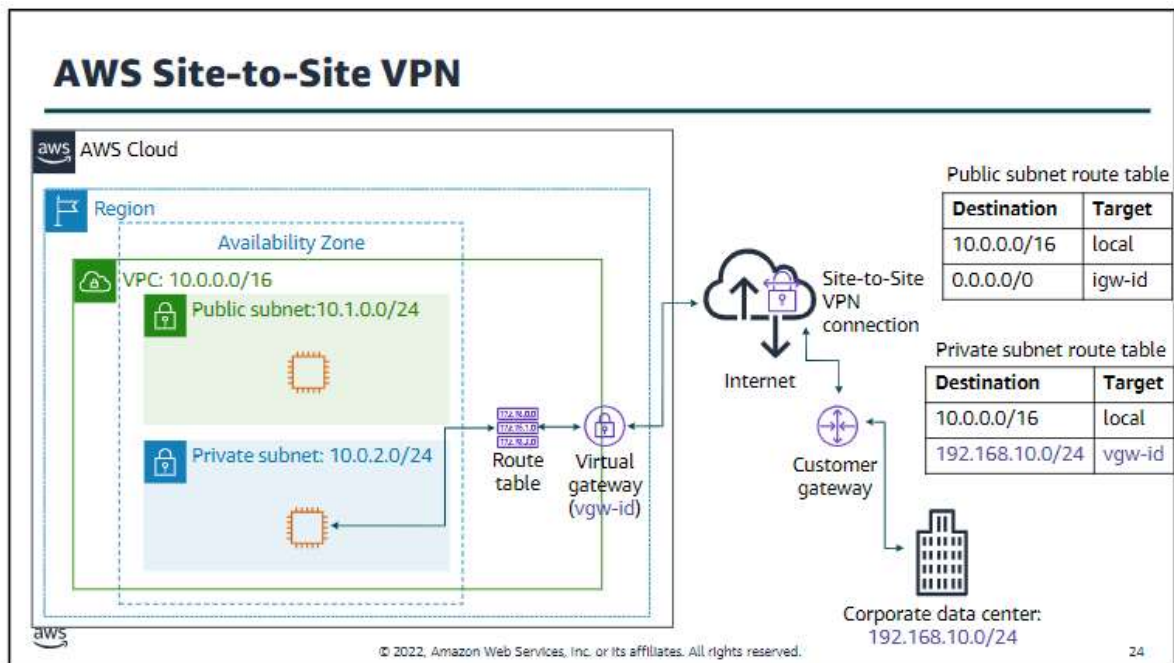


In the following example, a flow log captures all traffic for a subnet and publishes the flow log records to Amazon CloudWatch Logs. The flow log captures traffic for all network interfaces in the subnet.



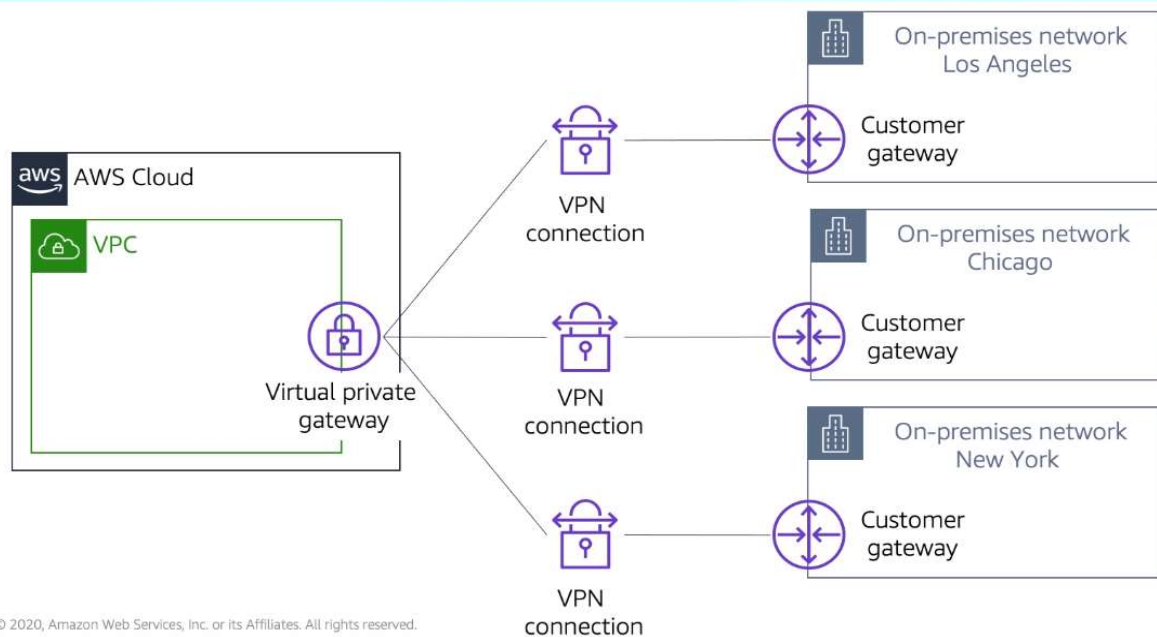
AWS Site to Site VPN, Virtual Private Gateway and Customer gateway

- By default, instances that you launch into a VPC cannot communicate with a remote network.



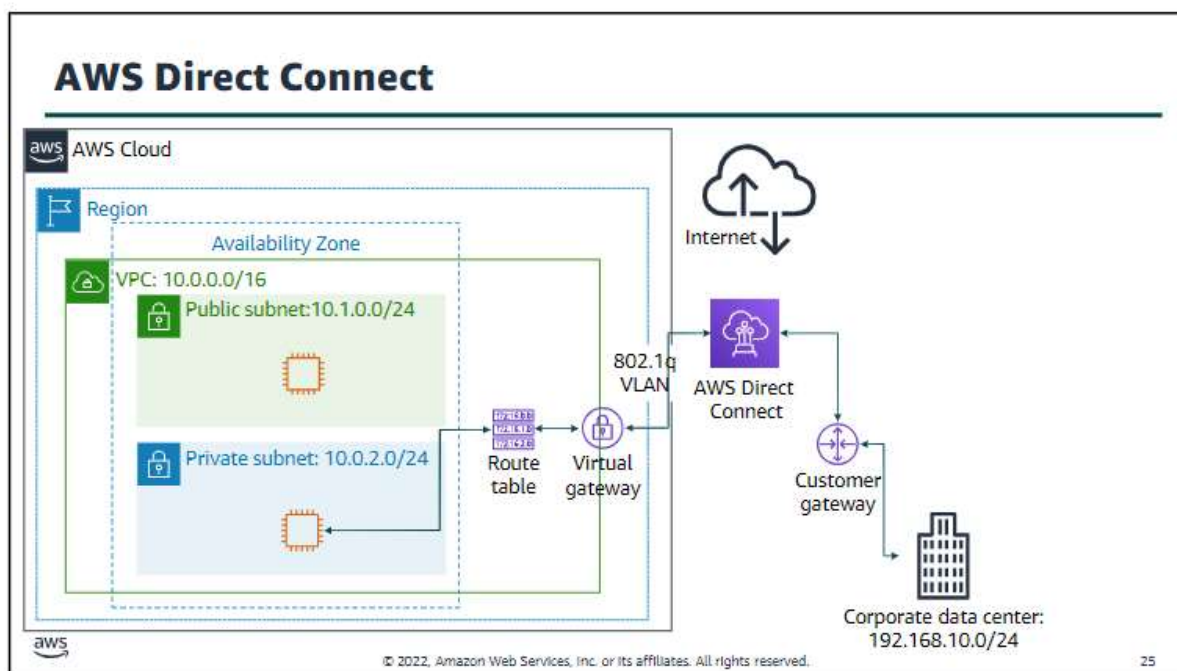
- By default, instances that you launch into a virtual private cloud (VPC) on AWS cannot communicate with your on-premises network.
- You can use AWS Site-to-Site Virtual Private Network (AWS Site-to-Site VPN) to securely connect your on-premises network or branch office site to your VPC.
- Each AWS Site-to-Site VPN connection uses internet protocol security (IPSec) communications to create encrypted VPN tunnels between two locations.
- A VPN tunnel is an encrypted link where data can pass from the customer network to or from AWS.
- The AWS side of the connection is the virtual private gateway. (Note that instead of a virtual private gateway, you can also create a Site-to-Site VPN connection as an attachment on a transit gateway.)
- The on-premises side of the connection is the customer gateway.
- AWS Site-to-Site VPN supports two types of routing.
 - If your VPN device supports Border Gateway Protocol (BGP), specify dynamic routing when you configure your Site-to-Site VPN connection. Dynamic routing uses the BGP to advertise routes to the virtual private gateway.
 - If your VPN device does not support BGP, specify static routing. static routing requires that you specify the routes (that is, IP prefixes) for your network that should be communicated to the virtual private gateway.

Connecting multiple VPNs



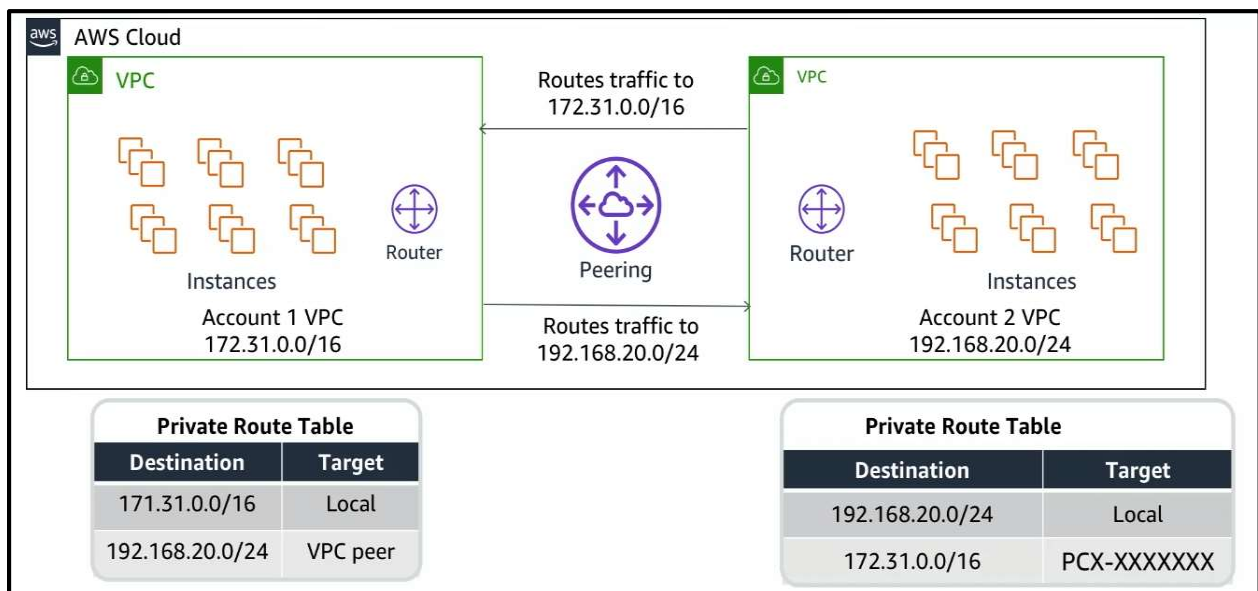
AWS Direct Connect and Direct Connect Gateway

- One of the challenges of network communication is network performance.
- Performance can be negatively affected if your data center is located far away from your AWS Region.
- For such situations, AWS offers AWS Direct Connect, or DX.



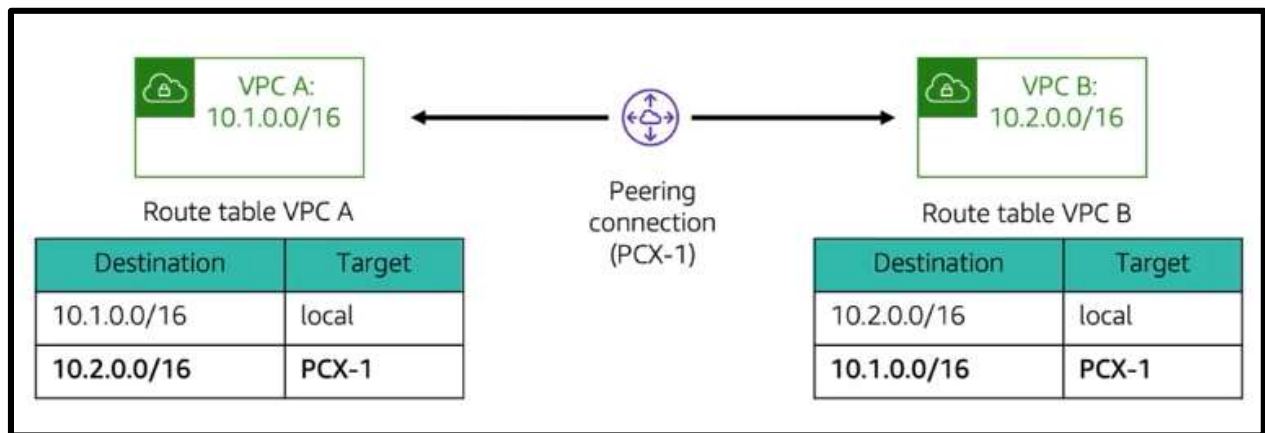
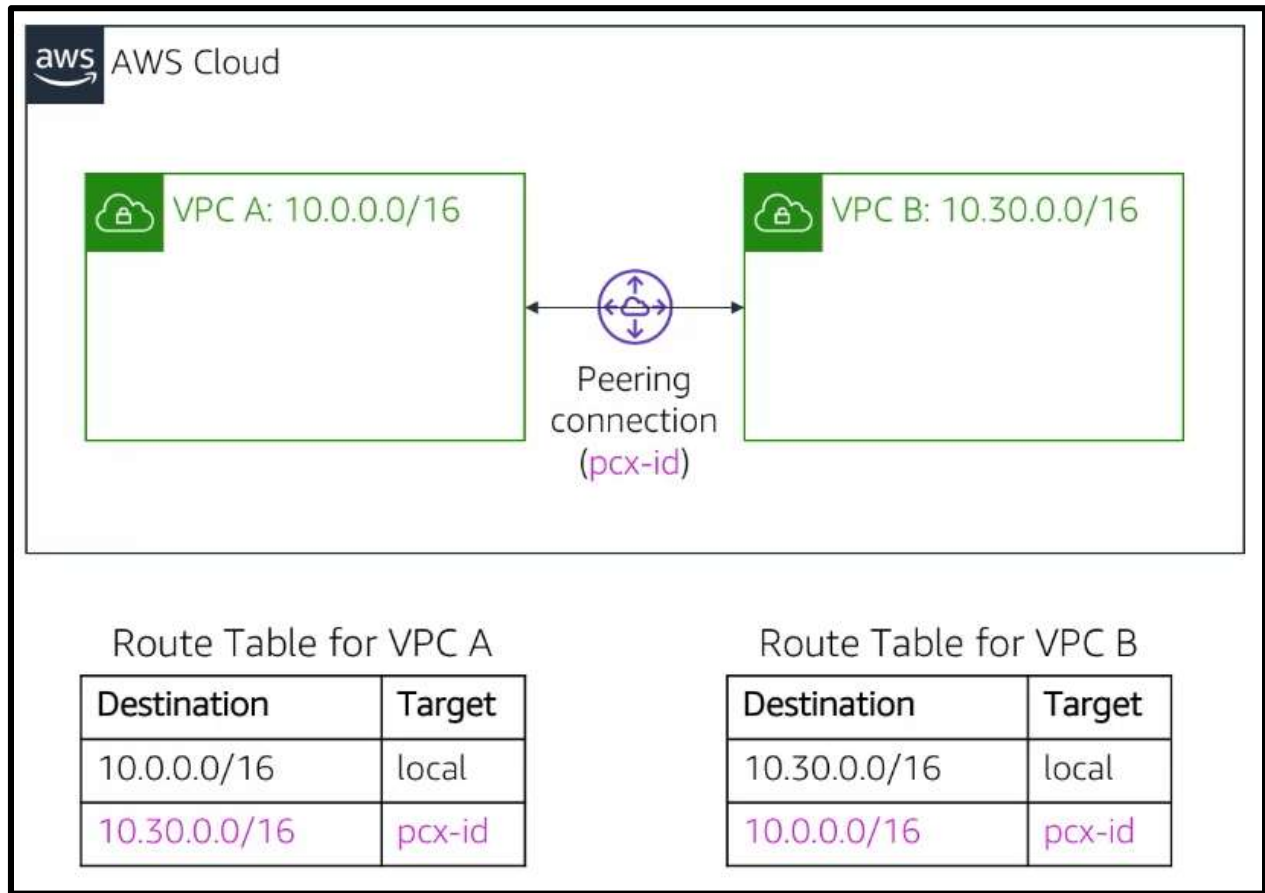
- AWS Site-to-Site VPN is one option for connecting your on-premises network to the AWS global network. With this option, your data is transferred through encrypted tunnels over the public internet.
- AWS Direct Connect (or DX) is another solution that goes beyond simple connectivity over the internet.
- DX uses open standard 802.1q virtual local area networks (VLANs) so you can establish a dedicated, private network connection from your premises to AWS.
- This private connection can reduce network costs, increase bandwidth throughput, and provide a more consistent network experience than internet-based connections.
- Dedicated connections are available with 1-Gbps and 10-Gbps capacity.

VPC Peering

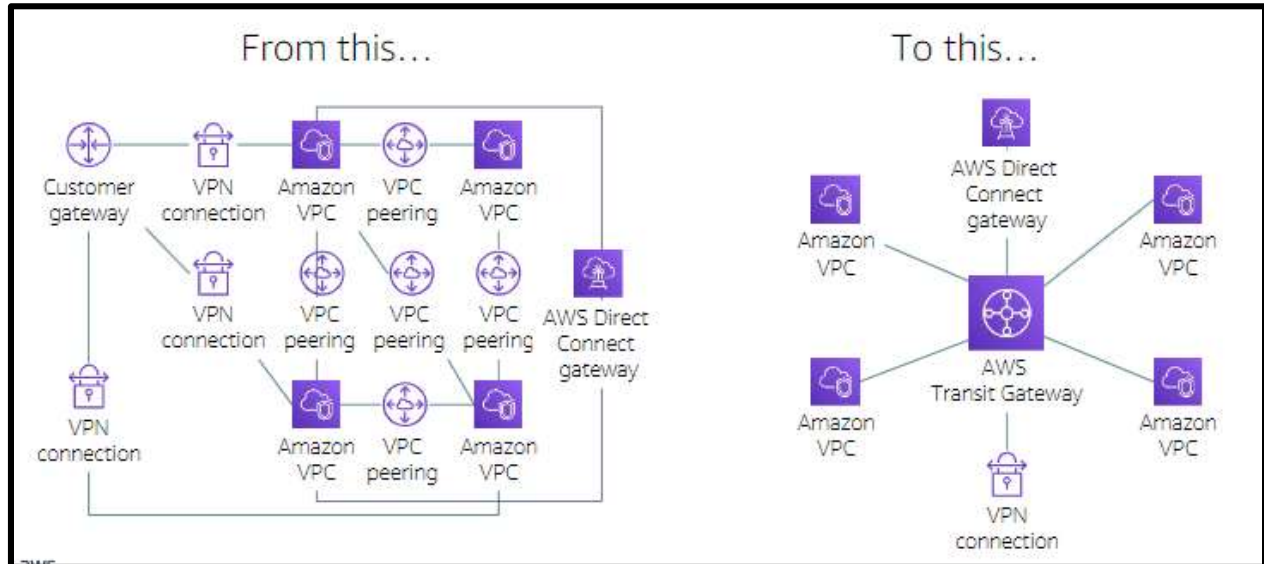


- A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them privately.
- Instances in either VPC can communicate with each other as if they are within the same network.
- You can create a VPC peering connection between your own VPCs, with a VPC in another AWS account, or with a VPC in a different AWS Region.
- When you set up the peering connection, you create rules in your route table to allow the VPCs to communicate with each other through the peering resource.
- For example, suppose that you have two VPCs. In the route table for VPC A, you set the destination to be the IP address of VPC B and the target to be the peering resource ID. In the route table for VPC B, you set the destination to be the IP address of VPC A and the target to be the peering resource ID.
- VPC peering has some restrictions:
 - IP address ranges cannot overlap.
 - Transitive peering is not supported. For example, suppose that you have three VPCs: A, B, and C.

- VPC A is connected to VPC B, and VPC A is connected to VPC C. However, VPC B is not connected to VPC C implicitly.
- To connect VPC B to VPC C, you must explicitly establish that connectivity.
- You can only have one peering resource between the same two VPCs

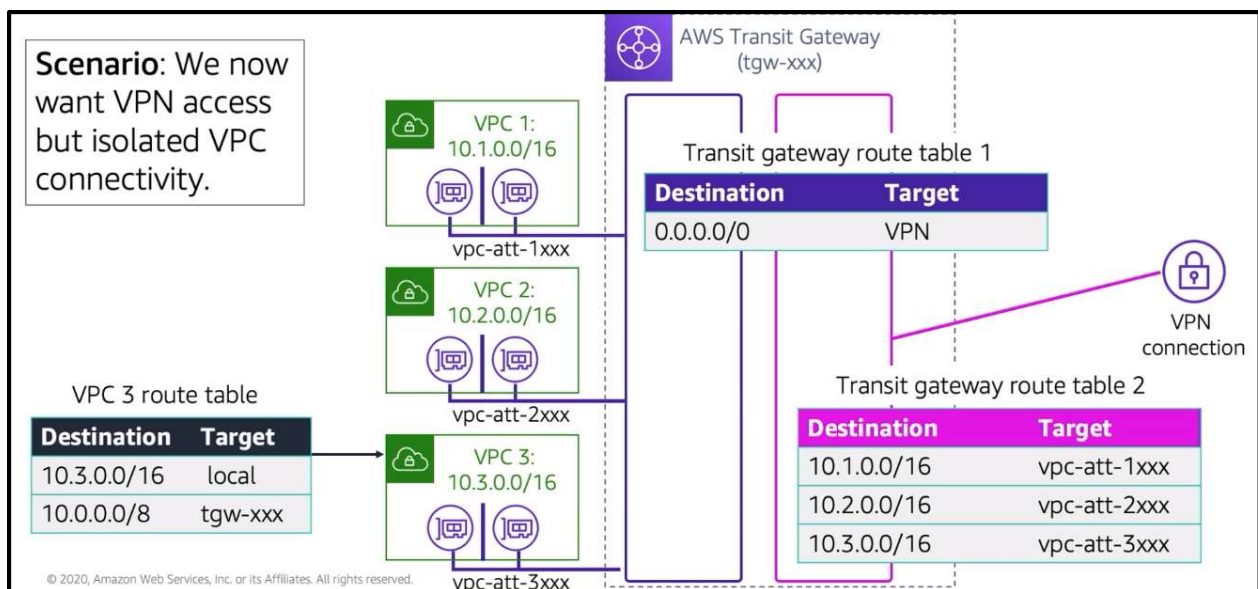
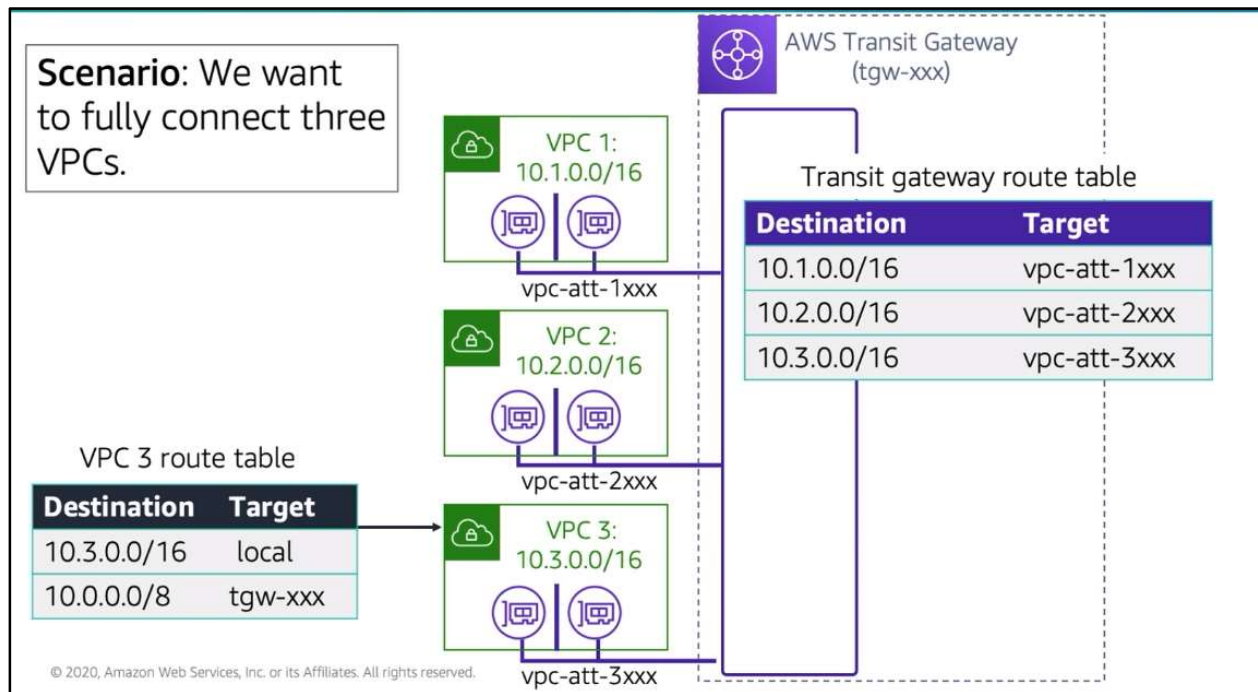


AWS Transit Gateway



- You can configure your VPCs in several ways, and take advantage of numerous connectivity options and gateways.
- These options and gateways include AWS Direct Connect (via DX gateways), NAT gateways, internet gateways, VPC peering, etc.
- It is not uncommon to find AWS customers with hundreds of VPCs distributed across AWS accounts and Regions to serve multiple lines of business, teams, projects, and so forth. Things get more complex when customers start to set up connectivity between their VPCs.
- All the connectivity options are strictly point-to-point, so the number of VPC-to-VPC connections can grow quickly.
- To solve this problem, you can use AWS Transit Gateway to simplify your networking model.
- With AWS Transit Gateway, you only need to create and manage a single connection from the central gateway into each VPC, on-premises data center, or remote office across your network.
- A transit gateway acts as a hub that controls how traffic is routed among all the connected networks, which act like spokes.
- This hub-and-spoke model significantly simplifies management and reduces operational costs because each network only needs to connect to the transit gateway and not to every other network.
- Any new VPC is connected to the transit gateway, and is then automatically available to every other network that is connected to the transit gateway.
- This ease of connectivity makes it easier to scale your network as you grow.
- AWS Transit Gateway is a service that enables you to connect your VPCs and on-premises networks to a single gateway (called a transit gateway). With AWS Transit Gateway, you only need to create and manage a single connection from the central gateway into each VPC, on-premises data center, or remote office across your network.

- AWS Transit Gateway uses a hub-and-spoke model. This model significantly simplifies management and reduces operational costs because each network only needs to connect to the transit gateway and not to every other network.
- Any new VPC is connected to the transit gateway, and is then automatically available to every other network that is connected to the transit gateway.
- This ease of connectivity makes it easier to scale your network as you grow.
- You can use AWS Transit Gateway to connect up to 5,000 VPCs and on-premises networks

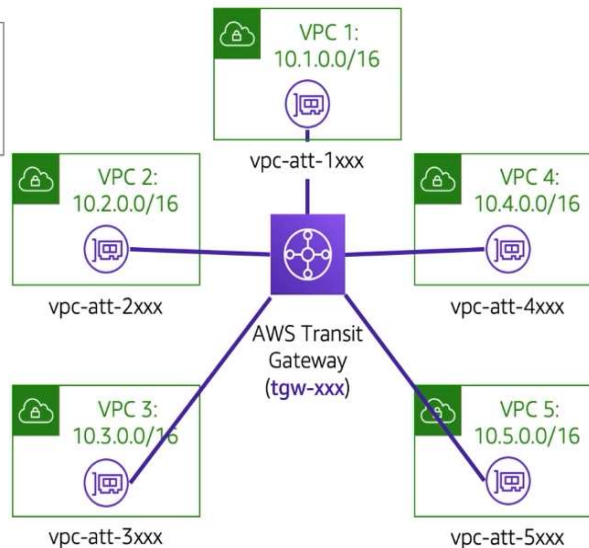


AWS Transit Gateway: Challenge

Scenario: How do you connect these five VPCs?

VPC # route table

Destination	Target
10.#.0.0/16	local
?	?



Transit gateway route table

Destination	Target
?	?

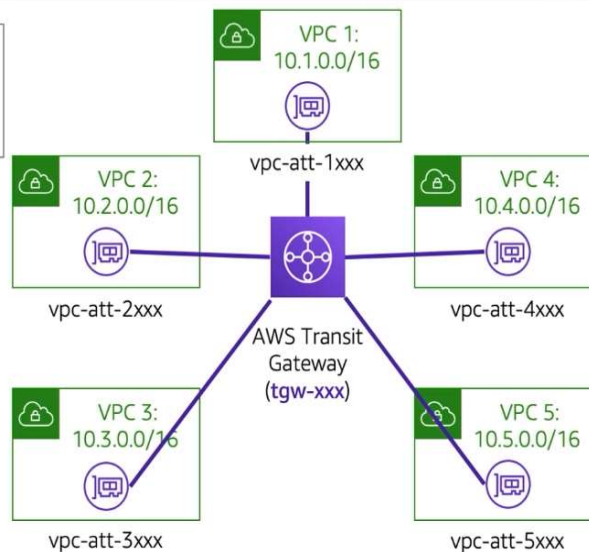
© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

AWS Transit Gateway activity: Solution

Scenario: How do you connect these five VPCs?

VPC 3 route table

Destination	Target
10.3.0.0/16	local
10.0.0.0/8	tgw-xxx



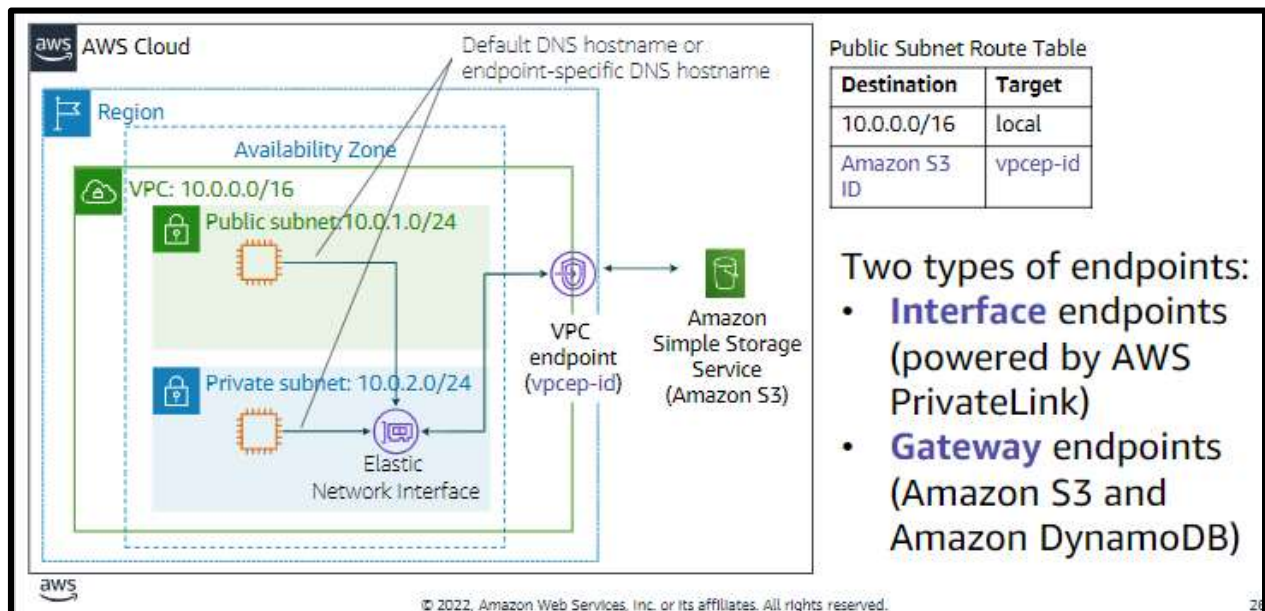
Transit gateway route table

Destination	Target
10.1.0.0/16	vpc-att-1xxx
10.2.0.0/16	vpc-att-2xxx
10.3.0.0/16	vpc-att-3xxx
10.4.0.0/16	vpc-att-4xxx
10.5.0.0/16	vpc-att-5xxx

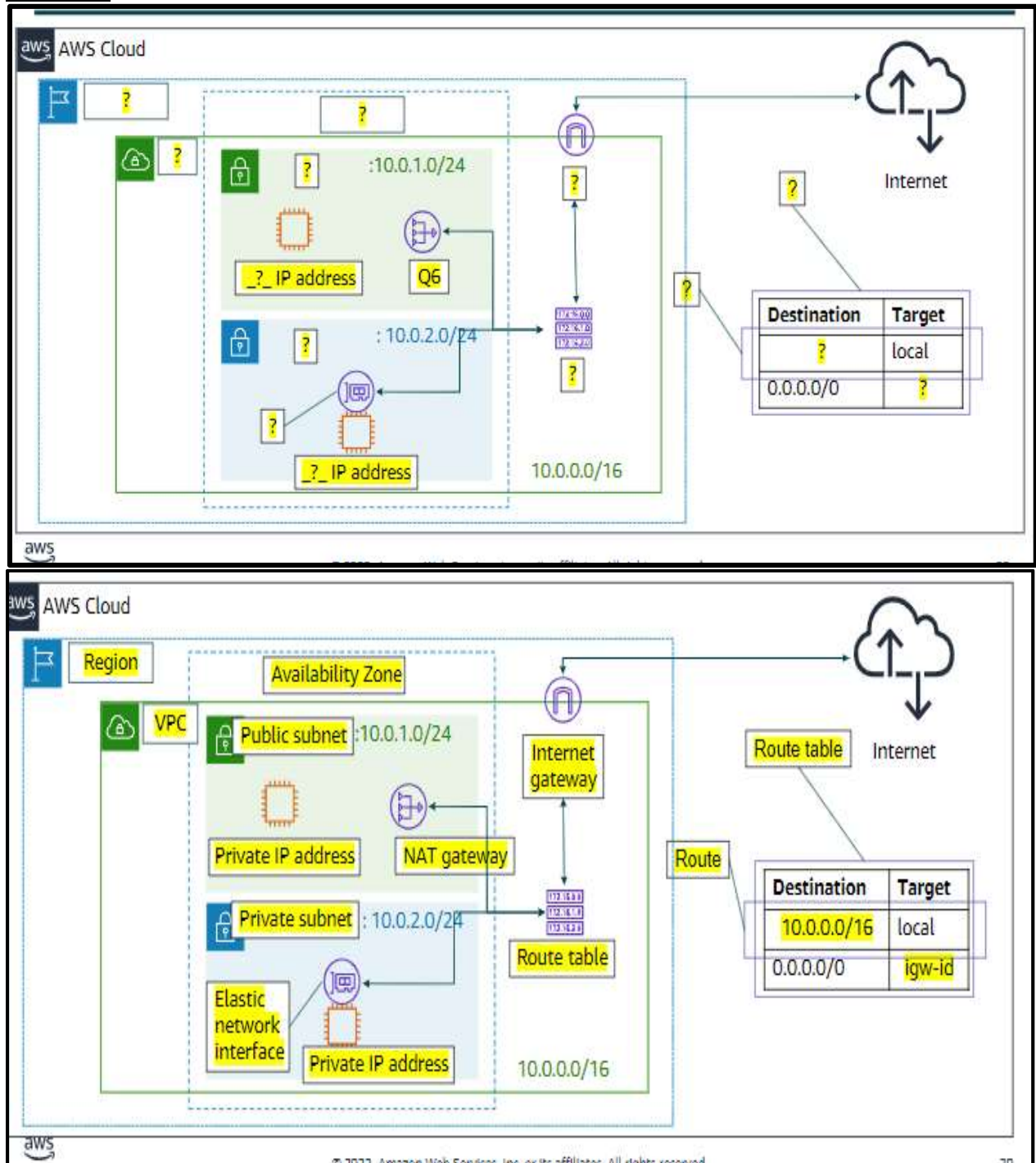
© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

VPC Endpoints

- A VPC endpoint is a virtual device that enables you to privately connect your VPC to supported AWS services and VPC endpoint services.
- Connection to these services does not require an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection.
- Instances in your VPC do not require public IP addresses to communicate with resources in the service.
- Traffic between your VPC and the other service does not leave the Amazon network.
- There are two types of VPC endpoints:
 - An interface VPC endpoint (interface endpoint) enables you to connect to services.
 - These services include some AWS services, services that are hosted by other AWS.
 - The owner of the service is the service provider, and you as the principal who creates the interface endpoint—are the service consumer.
- Gateway endpoints: The use of gateway endpoints incurs no additional charge. Standard charges for data transfer and resource usage apply.



Activity



Activity: Design a VPC

Scenario: You have a small business with a website that is hosted on an Amazon Elastic Compute Cloud (Amazon EC2) instance. You have customer data that is stored on a backend database that you want to keep private. You want to use Amazon VPC to set up a VPC that meets the following requirements:

- Your web server and database server must be in separate subnets.
- The first address of your network must be 10.0.0.0. Each subnet must have 256 total IPv4 addresses.
- Your customers must always be able to access your web server.
- Your database server must be able to access the internet to make patch updates.
- Your architecture must be highly available and use at least one custom firewall layer.

Lab 2: Scenario

In this lab, you use Amazon VPC to **create your own VPC** and add some components to produce a customized network. You **create a security group** for your VPC. You also **create an EC2 instance and configure it** to run a web server and to use the security group. You then launch the EC2 instance into the VPC.



Amazon
VPC



Amazon
EC2

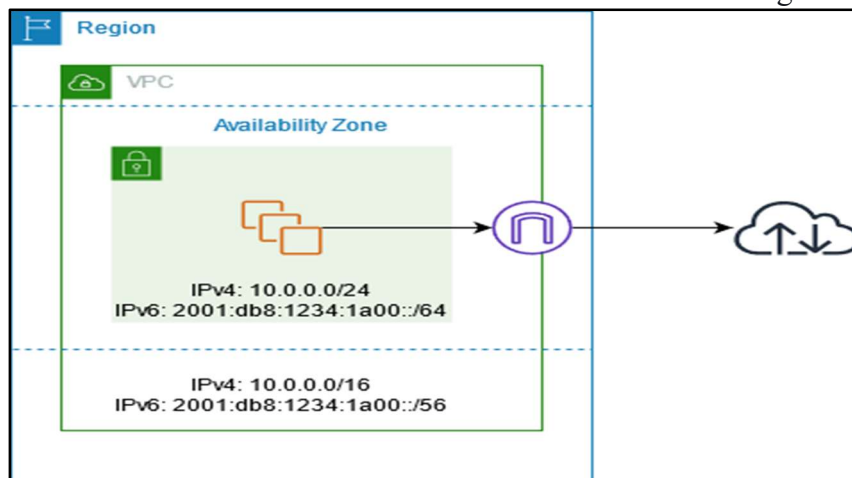
IPv6 for VPC

- If you have an existing VPC that supports IPv4 only, and resources in your subnet that are configured to use IPv4 only, you can enable IPv6 support for your VPC and resources.
- Your VPC can operate in dual-stack mode — your resources can communicate over IPv4, or IPv6, or both. IPv4 and IPv6 communication are independent of each other.
- You cannot disable IPv4 support for your VPC and subnets; this is the default IP addressing system for Amazon VPC and Amazon EC2.
- Steps to enable your VPC and subnets to use IPv6.
 - Step 1: Associate an IPv6 CIDR block with your VPC and subnets
 - Step 2: Update your route tables

- Step 3: Update your security group rules
- Step 4: Change your instance type
- Step 5: Assign IPv6 addresses to your instances
- Step 6: (Optional) Configure IPv6 on your instances

Egress only Internet Gateway

- IPv6 addresses are globally unique, and are therefore public by default.
- If you want your instance to be able to access the internet, but you want to prevent resources on the internet from initiating communication with your instance, you can use an egress-only internet gateway.
- To do this, create an egress-only internet gateway in your VPC, and then add a route to your route table that points all IPv6 traffic (::/0) or a specific range of IPv6 address to the egress-only internet gateway.
- IPv6 traffic in the subnet that's associated with the route table is routed to the egress-only internet gateway.
- An egress-only internet gateway is stateful: it forwards traffic from the instances in the subnet to the internet or other AWS services, and then sends the response back to the instances.
- An egress-only internet gateway has the following characteristics:
 - You cannot associate a security group with an egress-only internet gateway. You can use security groups for your instances in the private subnet to control the traffic to and from those instances.
 - You can use a network ACL to control the traffic to and from the subnet for which the egress-only internet gateway routes traffic.
- In the following diagram, the VPC has both IPv4 and IPv6 CIDR blocks, and the subnet both IPv4 and IPv6 CIDR blocks. The VPC has an egress-only internet gateway.

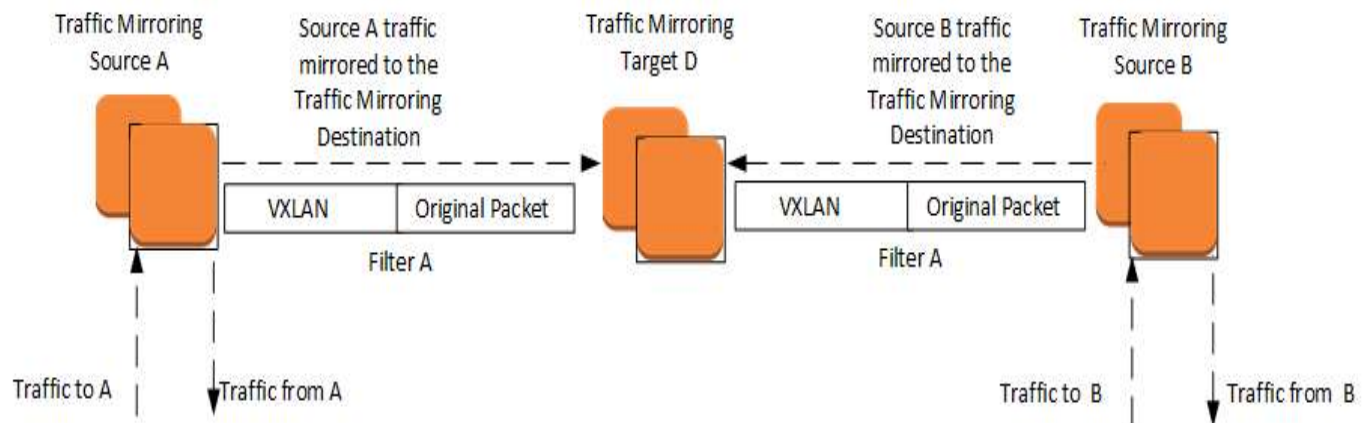


- The following is an example of the route table associated with the subnet. There is a route that sends all internet-bound IPv6 traffic (::/0) to the egress-only internet gateway.

Destination	Target
10.0.0.0/16	Local
2001:db8:1234:1a00:/64	Local
::/0	<i>eigw-id</i>

VPC Traffic Mirroring

- Traffic Mirroring is an Amazon VPC feature that you can use to copy network traffic from an elastic network interface of type interface.
- You can then send the traffic for:
 - Content inspection
 - Threat monitoring
 - Troubleshooting
- Traffic Mirroring supports filters and packet truncation, so that you only extract the traffic of interest to monitor by using monitoring tools of your choice.
- The following are the key concepts for Traffic Mirroring:
 - Source** — The network interface to monitor.
 - Target** — The destination for mirrored traffic.
 - Filter** — A set of rules that defines the traffic that is copied in a traffic mirror session.
 - Session** — An entity that describes Traffic Mirroring from a source to a target using filter



- **Benefits:**
 - **Enhanced security** — Capture packets at the elastic network interface, which cannot be disabled or tampered with from a user space.
 - **Increased monitoring options** — Send your mirrored traffic to any security device.

AWS Private Link

- AWS PrivateLink establishes private connectivity between virtual private clouds (VPC), and supported AWS services, services hosted by other AWS accounts, and supported AWS Marketplace services.
- You do not need to use an internet gateway, NAT device, AWS Direct Connect connection, or AWS Site-to-Site VPN connection to communicate with the service.
- To use AWS PrivateLink, create a VPC endpoint in your VPC, specifying the name of the service and a subnet. This creates an elastic network interface in the subnet that serves as an entry point for traffic destined to the service.
- You can create your own VPC endpoint service, powered by AWS PrivateLink and enable other AWS customers to access your service.

