Amazon VPC

What is Amazon VPC?

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch Amazon Web Services (AWS) resources into a **virtual network** that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

Amazon VPC Concepts

What is CIDR block?

A system called **Classless Inter-Domain Routing**, or CIDR, was developed as an alternative to traditional subnetting. The idea is that **you can add a specification in the IP address itself as to the number of significant bits that make up the routing or networking portion**.

For example, we could express the idea that the IP address `192.168.0.15` is associated with the netmask `255.255.255.0` by using the CIDR notation of `192.168.0.15/24`. This means that the first 24 bits of the IP address given are considered significant for the network routing.

Resolving /28

Ip Address Octets

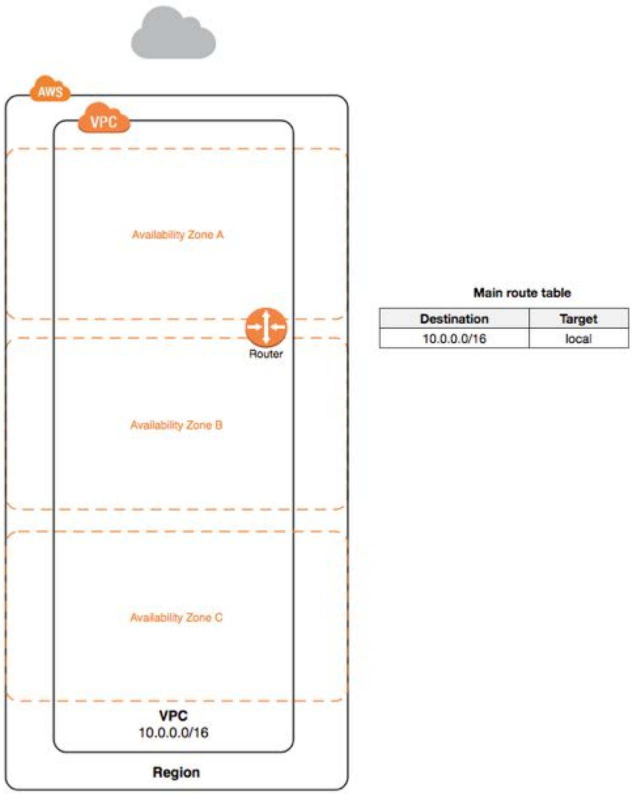
11111111	11111111	11111111	11110000
255	255	255	240 (255-240+1 = 16) Ip addresses available

VPCs and Subnets

A virtual private cloud (VPC) is a **virtual network dedicated to your AWS account**. It is logically isolated from other virtual networks in the AWS Cloud. You can launch your AWS resources, such as Amazon EC2 instances, into your VPC.

When you create a VPC, you must specify a range of IPv4 addresses for the VPC in the form of a Classless Inter-Domain Routing (CIDR) block; for example, `10.0.0.0/16`.

- **VPC spans all the Availability Zones in the region**
- **You can add one or more subnets in each Availability Zone**
- **Subnet CIDR block is a subset of the VPC CIDR block**
- **Subnet must reside entirely within one Availability Zone**
- **Each subnet is assigned an unique ID**



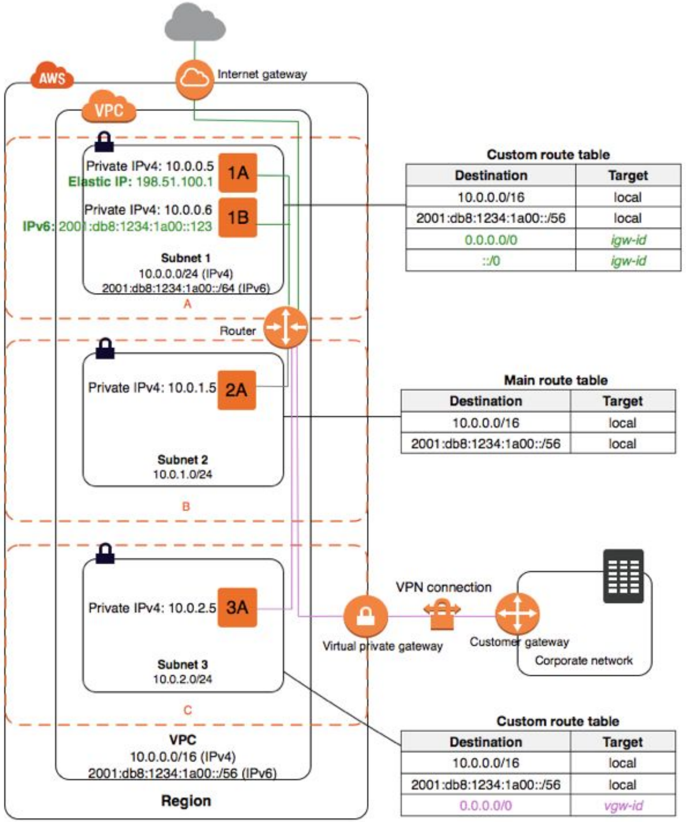
Destination	Target
10.0.0.0/16	local

VPC Components

- An Amazon VPC consists of the following components:
  - Subnets
  - Route tables
  - Dynamic Host Configuration Protocol (DHCP) option sets
  - Security groups
  - Network Access Control Lists (ACLs)
- An Amazon VPC has the following optional components:
  - Internet Gateways (IGWs)
  - Elastic IP (EIP) addresses
  - Elastic Network Interfaces (ENIs)
  - Endpoints
  - Peering
  - Network Address Translation (NATs) instances and NAT gateways
  - Virtual Private Gateway (VPG), Customer Gateways (CGWs), and Virtual Private Networks (VPNs)

Sample VPC Setup

- The following diagram shows a VPC that has been configured with subnets in multiple Availability Zones.
- An IPv6 CIDR block is associated with the VPC, and an IPv6 CIDR block is associated with subnet 1
- 1A, 1B, 2A, and 3A are instances in your VPC.
- An Internet gateway enables communication over the Internet, and a virtual private network (VPN) connection enables communication with your corporate network.
- Subnet 1 is a public subnet (has IPv4 address and can communicate with Internet)
- Subnet 2&3 are private subnets.



Custom route table

Destination	Target
10.0.0.0/16	local
2001:db8:1234:1a00::/56	local
0.0.0.0/0	igw-id
::/0	igw-id

Main route table

Destination	Target
10.0.0.0/16	local
2001:db8:1234:1a00::/56	local

Custom route table

Destination	Target
10.0.0.0/16	local
2001:db8:1234:1a00::/56	local
0.0.0.0/0	vgw-id

What is a subnet?

The process of dividing a network into **smaller network sections** is called subnetting. This can be useful for many different purposes and helps isolate groups of hosts together and deal with them easily. A **subnet is a segment of an Amazon VPC's IP address range** where you can launch Amazon EC2 instances, RDS and other AWS resources. CIDR blocks define subnets (for example, 10.0.1.0/24 and 192.168.0.0/24). The smallest subnet that you can create is a /28 (16 IP addresses).

AWS reserves the first four IP addresses and the last IP address of every subnet for internal networking purposes. For example, a subnet defined as a /28 has 16 available IP addresses; subtract the 5 IPs needed by AWS to yield 11 IP addresses for your use within the subnet.

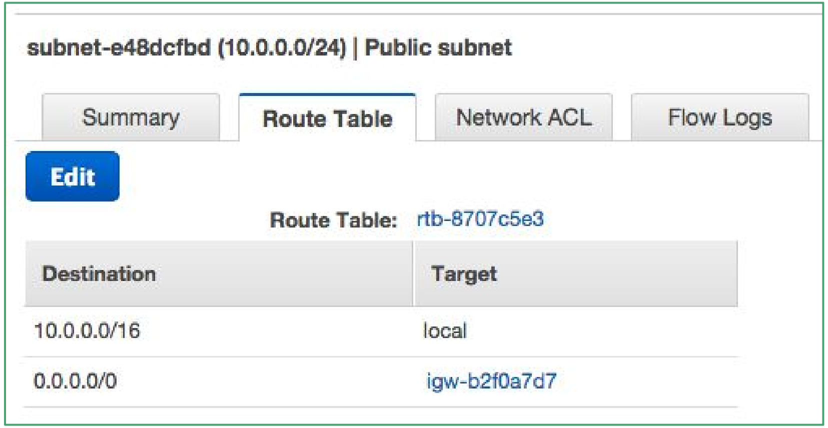
After creating an Amazon VPC, you can add one or more subnets in each Availability Zone. **Subnets reside within one Availability Zone and cannot span zones**. You can have **multiple subnets in one Availability Zone**.

Subnets can be classified as **public, private, or VPN-only**.

- A public subnet is one in which the associated route table directs the subnet's traffic to the Amazon VPC's IGW.
- A private subnet is one in which the associated route table does not direct the subnet's traffic to the Amazon VPC's IGW.
- A VPN-only subnet is one in which the associated route table directs the subnet's traffic to the Amazon VPC's VPG and does not have a route to the IGW.
- Regardless of the type of subnet, the internal IP address range of the subnet is always private (that is, non-routable on the Internet).
- Default Amazon VPCs contain one public subnet in every Availability Zone within the region, with a netmask of /20.

What are Route tables?

A route table is a l**ogical construct within an Amazon VPC that contains a set of rules** (called routes) that are a**pplied to the subnet** and used to **determine where network traffic is directed**. A route table's routes are what permit Amazon EC2 instances within different subnets within an Amazon VPC to communicate with each other. You can modify route tables and add your own custom routes. You can also use route tables to specify which **subnets are public (by directing Internet traffic to the IGW)** and which **subnets are private (by not having a route that directs traffic to the IGW)**. **Each route table contains a default route called the local route, which enables communication within the Amazon VPC**, and this route cannot be modified or removed.

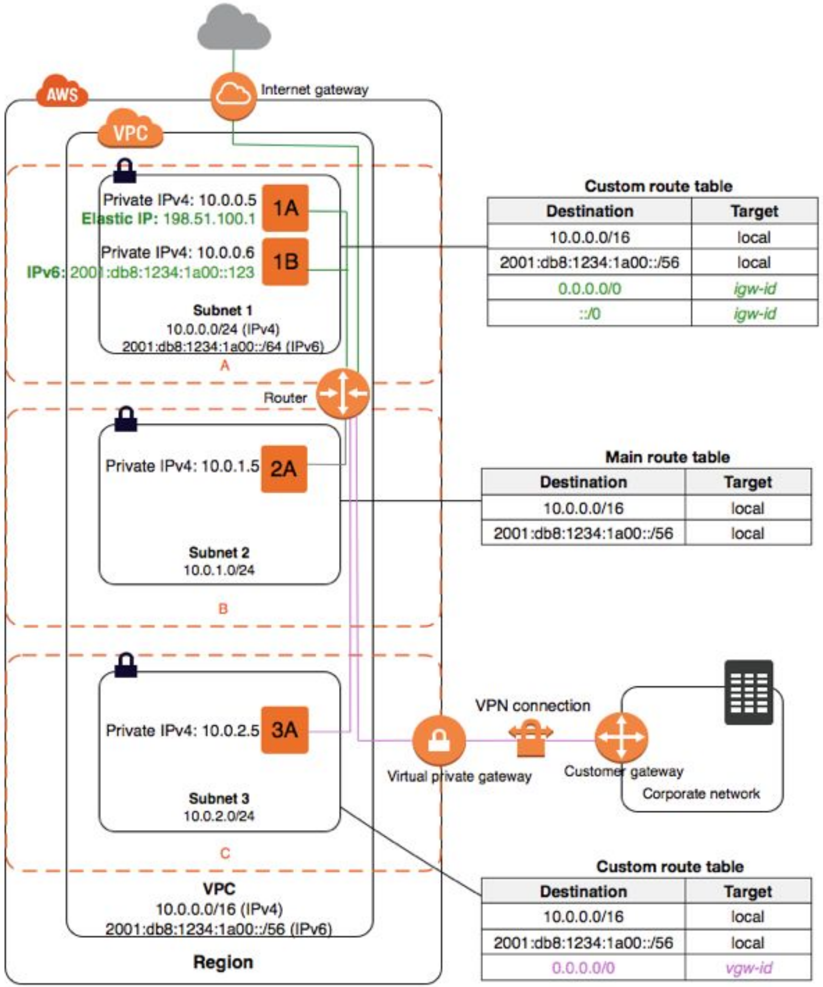


subnet-e48dcfbd (10.0.0.0/24) | Public subnet

Route Table: rtb-8707c5e3

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-b2f0a7d7

Decode the Route tables



Custom route table

Destination	Target
10.0.0.0/16	local
2001:db8:1234:1a00::/56	local
0.0.0.0/0	*igw-id*
::/0	*igw-id*

Main route table

Destination	Target
10.0.0.0/16	local
2001:db8:1234:1a00::/56	local

Custom route table

Destination	Target
10.0.0.0/16	local
2001:db8:1234:1a00::/56	local
0.0.0.0/0	*vgw-id*

Route Table Tips

You should remember the following points about route tables:

- Your VPC has an implicit router.
- Your VPC automatically comes with a main route table that you can modify.
- You can create additional custom route tables for your VPC.
- Each subnet 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 uses the main route table.
- You can replace the main route table with a custom table that you’ve created so that each
- new subnet is automatically associated with it.
- Each route in a table specifies a destination CIDR and a target; for example, traffic destined for 172.16.0.0/12 is targeted for the VPG. AWS uses the most specific route that matches the traffic to determine how to route the traffic.

What is Dynamic Host Configuration Protocol (DHCP) option sets

Dynamic Host Configuration Protocol (DHCP) is a client/server protocol that automatically **provides an Internet Protocol (IP) host with its IP address** and other related configuration information such as the subnet mask and default gateway.

AWS automatically creates and associates a DHCP option set for your Amazon VPC upon creation and sets two options: domain-name-servers (defaulted to AmazonProvidedDNS) and domain-name (defaulted to the domain name for your region). AmazonProvidedDNS is an Amazon Domain Name System (DNS) server, and this option enables DNS for instances that need to communicate over the Amazon VPC's IGW.

The DHCP option sets element of an Amazon VPC allows you to direct Amazon EC2 hostname assignments to your own resources. To assign your own domain name to your instances, create a custom DHCP option set and assign it to your Amazon VPC.

Every Amazon VPC must have only one DHCP option set assigned to it.

What are Security groups?

A security group is a **virtual stateful firewall that controls inbound and outbound network traffic** to AWS resources and Amazon EC2 instances. All Amazon EC2 instances must be launched into a security group. If a security group is not specified at launch, then the instance will be launched into the default security group for the Amazon VPC. The default security group allows communication between all resources within the security group, allows all outbound traffic, and denies all other traffic. You may change the rules for the default security group, but you may not delete the default security group

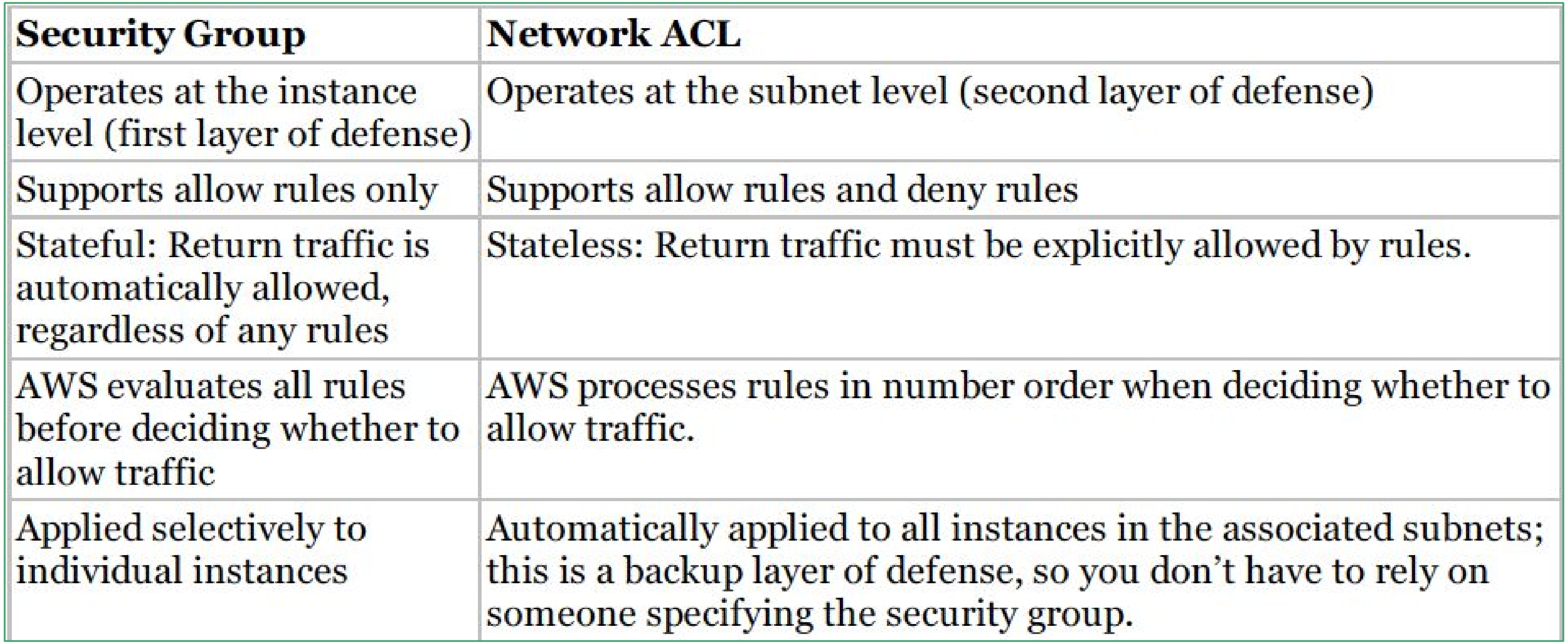
Security Group Exam Tips

- 500 security groups for each Amazon VPC.
- You can add up to 50 inbound and 50 outbound rules to each security group. If you need to apply more than 100 rules to an instance, you can associate up to five security groups with each network interface.
- You can specify allow rules, but not deny rules. This is an important difference between security groups and ACLs.
- You can specify separate rules for inbound and outbound traffic.
- By default, no inbound traffic is allowed until you add inbound rules to the security group.
- By default, new security groups have an outbound rule that allows all outbound traffic. You can remove the rule and add outbound rules that allow specific outbound traffic only.
- Security groups are stateful. This means that responses to allowed inbound traffic are allowed to flow outbound regardless of outbound rules and vice versa. This is an important difference between security groups and network ACLs.
- Instances associated with the same security group can’t talk to each other unless you add rules allowing it (with the exception being the default security group).
- You can change the security groups with which an instance is associated after launch, and the changes will take effect immediately.

What are Network Access Control Lists (ACLs)?

A network access control list (ACL) is another layer of security that acts as a **stateless firewall on a subnet level**. A network ACL is a numbered list of rules that AWS evaluates in order, starting with the lowest numbered rule, to determine whether traffic is allowed in or out of any subnet associated with the network ACL. Amazon VPCs are created with a modifiable default network ACL associated with every subnet that allows all inbound and outbound traffic. When you create a custom network ACL, its initial configuration will deny all inbound and outbound traffic until you create rules that allow otherwise. You may set up network ACLs with rules similar to your security groups in order to add a layer of security to your Amazon VPC, or you may choose to use the default network ACL that does not filter traffic traversing the subnet boundary. Overall, every subnet must be associated with a network ACL.

Security Group vs Network ACLs



Security Group	Network ACL
Operates at the instance level (first layer of defense)	Operates at the subnet level (second layer of defense)
Supports allow rules only	Supports allow rules and deny rules
Stateful: Return traffic is automatically allowed, regardless of any rules	Stateless: Return traffic must be explicitly allowed by rules.
AWS evaluates all rules before deciding whether to allow traffic	AWS processes rules in number order when deciding whether to allow traffic.
Applied selectively to individual instances	Automatically applied to all instances in the associated subnets; this is a backup layer of defense, so you don't have to rely on someone specifying the security group.

What is NAT Instance and NAT Gateways?

By default, any instance that you launch into a private subnet in an Amazon VPC is not able to communicate with the Internet through the IGW. This is problematic if the instances within private subnets need direct access to the Internet from the Amazon VPC in order to apply security updates, download patches, or update application software. AWS provides NAT instances and NAT gateways to allow instances deployed in private subnets to gain Internet access. For common use cases, we recommend that you use a NAT gateway instead of a NAT instance. The NAT gateway provides better availability and higher bandwidth, and requires less administrative effort than NAT instances.

NAT Instance

A network address translation (NAT) instance is an Amazon Linux Amazon Machine Image (AMI) that is designed to accept traffic from instances within a private subnet, translate the source IP address to the public IP address of the NAT instance, and forward the traffic to the IGW. In addition, the NAT instance maintains the state of the forwarded traffic in order to return response traffic from the Internet to the proper instance in the private subnet. These instances have the string amzn-ami-vpc-nat in their names, which is searchable in the Amazon EC2 console.

NAT Gateway

A NAT gateway is an Amazon managed resource that is designed to operate just like a NAT instance, but it is simpler to manage and highly available within an Availability Zone.

Attribute	NAT instance	NAT gateway
Availability	Use a script to manage failover between instances.	Highly available. NAT gateways in each Availability Zone are implemented with redundancy. Create a NAT gateway in each Availability Zone to ensure zone-independent architecture.
Bandwidth	Depends on the bandwidth of the instance type.	Supports bursts of up to 10Gbps.
Maintenance	Managed by you, for example, by installing software updates or operating system patches on the instance.	Managed by AWS.You do not need to perform any maintenance.
Performance	A generic Amazon Linux AMI that's configured to perform NAT.	Software is optimized for handling NAT traffic.
Cost	Charged depending on the number of NAT instances that you use, duration of usage, and instance type and size.	Charged depending on the number of NAT gateways you use, duration of usage, and amount of data that you send through the NAT gateways.
Type and size	Choose a suitable instance type and size, according to your predicted workload.	Uniform offering; you don’t need to decide on the type or size.
Public IP addresses	Use an Elastic IP address or a public IP address with a NAT instance. You can change the public IP address at any time by associating a new Elastic IP address with the instance.	Choose the Elastic IP address to associate with a NAT gateway at creation.
Private IP addresses	Assign a specific private IP address from the subnet's IP address range when you launch the instance.	Automatically selected from the subnet's IP address range when you create the gateway.
Security groups	Associate with your NAT instance and the resources behind your NAT instance to control inbound and outbound traffic.	Cannot be associated with a NAT gateway. You can associate security groups with your resources behind the NAT gateway to control inbound and outbound traffic.
Network ACLs	Use a network ACL to control the traffic to and from the subnet in which your NAT instance resides.	Use a network ACL to control the traffic to and from the subnet in which your NAT gateway resides.
Traffic metrics	View CloudWatch metrics.	Not supported.

What is Internet Gateway (IGW)?

An Internet Gateway (IGW) is a horizontally scaled, redundant, and highly available Amazon VPC component that **allows communication between instances in your Amazon VPC and the Internet.** An IGW provides a target in your Amazon VPC route tables for Internet-routable traffic, and it performs network address translation for instances that have been assigned public IP addresses.

Amazon EC2 instances within an Amazon VPC are only aware of their private IP addresses. When traffic is sent from the instance to the Internet, the IGW translates the reply address to the instance’s public IP address and maintains the one-to-one map of the instance private IP address and public IP address. When an instance receives traffic from the Internet, the IGW translates the destination address (public IP address) to the instance’s private IP address and forwards the traffic to the Amazon VPC.

What is Elastic IP (EIP) addresses?

AWS maintains a pool of public IP addresses in each region and makes them available for you to associate to resources within your Amazon VPCs. An Elastic IP Addresses (EIP) is a static, public IP address in the pool for the region that you can allocate to your account (pull from the pool) and release (return to the pool). EIPs allow you to maintain a set of IP addresses that remain fixed while the underlying infrastructure may change over time.

Important points to understand about EIPs for the exam:

- You must first allocate an EIP for use within a VPC and then assign it to an instance.
- EIPs are specific to a region (that is, an EIP in one region cannot be assigned to an instance within an Amazon VPC in a different region).
- There is a one-to-one relationship between network interfaces and EIPs.
- You can move EIPs from one instance to another, either in the same Amazon VPC or a different Amazon VPC within the same region.
- EIPs remain associated with your AWS account until you explicitly release them.
- There are charges for EIPs allocated to your account, even when they are not associated with a resource.

What is Elastic Network Interfaces (ENIs)?

An Elastic Network Interface (ENI) is a **virtual network interface that you can attach to an instance in an Amazon VPC**. ENIs are **only available within an Amazon VPC**, and they are **associated with a subnet upon creation**. They can have **one public IP address and multiple private IP addresses**. If there are multiple private IP addresses, **one of the private ip address is primary**.

You can create a network interface, attach it to an instance, detach it from an instance, and attach it to another instance. The attributes of a network interface follow it as it's attached or detached from an instance and reattached to another instance. When you move a network interface from one instance to another, network traffic is redirected to the new instance.

Every instance in a VPC has a default network interface, called the primary network interface (eth0). You cannot detach a primary network interface from an instance. You can create and attach additional network interfaces.

What are Endpoints?

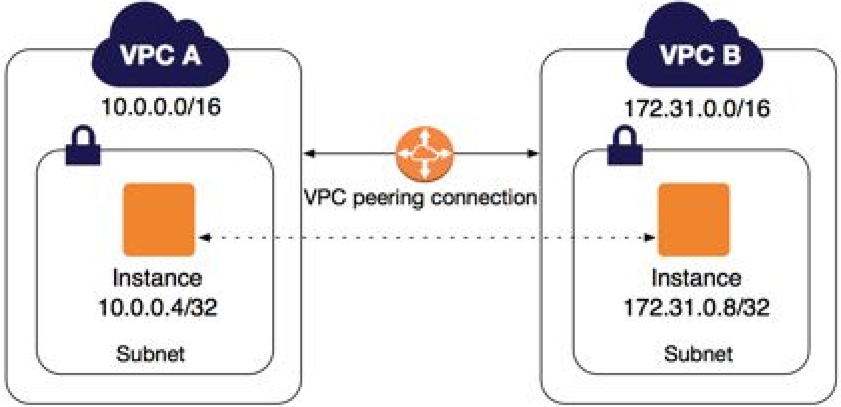
An Amazon VPC endpoint **enables you to create a private connection between your Amazon VPC and another AWS service without requiring access over the Internet or through a NAT instance, VPN connection, or AWS Direct Connect**.

You can create multiple endpoints for a single service, and you can use different route tables to enforce different access policies from different subnets to the same service.

Amazon VPC endpoints currently support communication with Amazon Simple Storage Service (Amazon S3).

What is Peering?

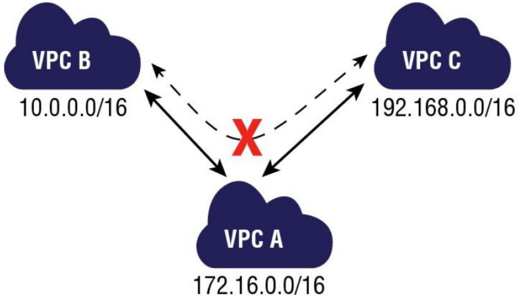
A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them. **Instances in either VPC can communicate with each other as if they are within the same network**. You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account. In both cases, the VPCs must be in the same region.



If you have more than one AWS account, you can peer the VPCs across those accounts to create a file sharing network. You can also use a VPC peering connection to allow other VPCs to access resources you have in one of your VPCs.

Peering Exam Tips

- You cannot create a peering connection between Amazon VPCs that have matching or overlapping CIDR blocks.
- You cannot create a peering connection between Amazon VPCs in different regions.
- Amazon VPC peering connections do not support transitive routing. A-B && B-C pairing does not mean A-C pairing. They need to be explicitly paired up.



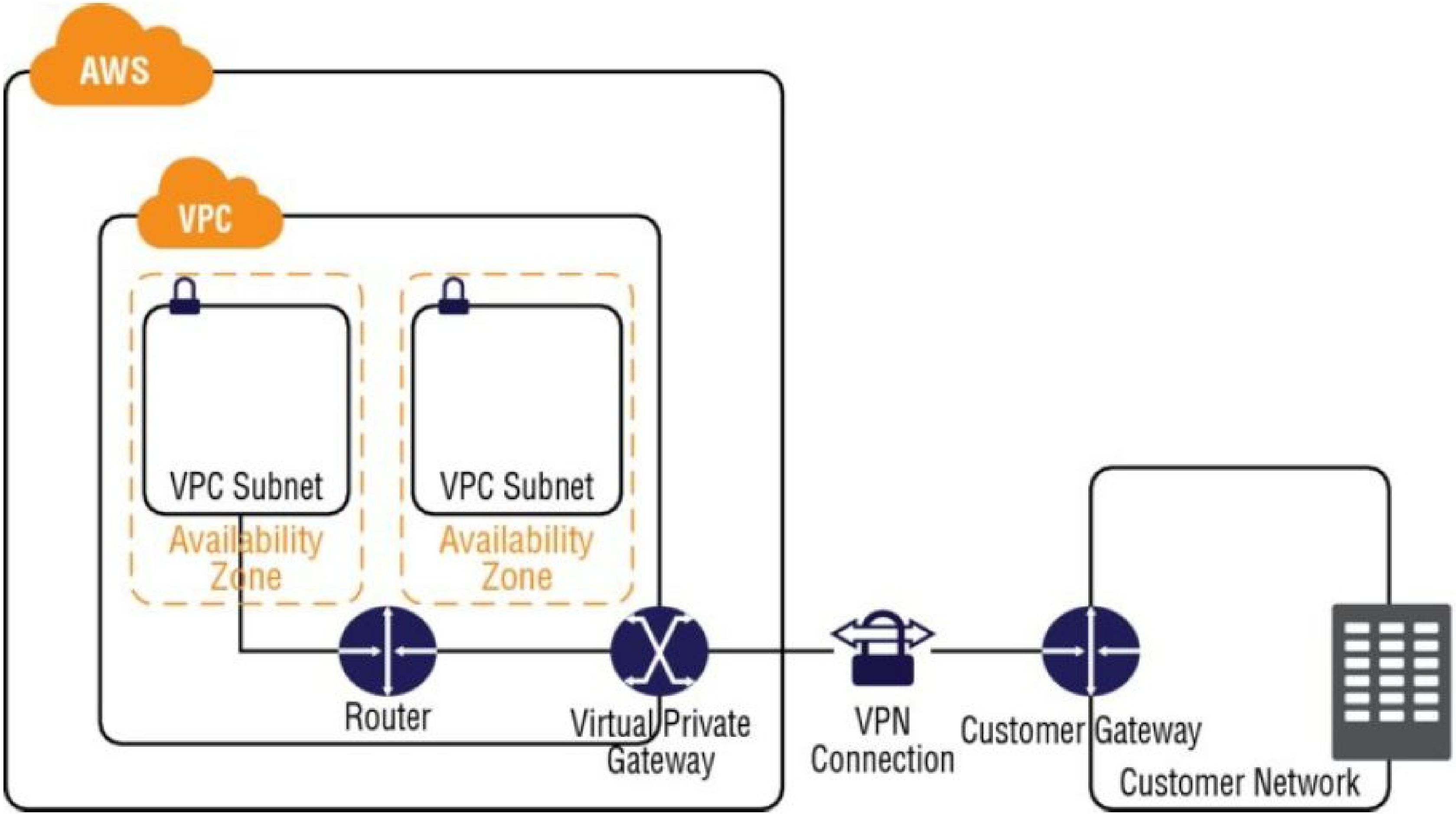
- You cannot have more than one peering connection between the same two Amazon VPCs at the same time.

Virtual Private Gateway (VPG), Customer Gateways (CGWs), and Virtual Private Networks (VPNs)

A virtual private gateway (VPG) is the virtual private network (VPN) concentrator on the AWS side of the VPN connection between the two networks. A customer gateway (CGW) represents a physical device or a software application on the customer's side of the VPN connection. After these two elements of an Amazon VPC have been created, the last step is to create a VPN tunnel. The VPN tunnel is established after traffic is generated from the customer's side of the VPN connection

Tips

- The VPG is the AWS end of the VPN tunnel.
- The CGW is a hardware or software application on the customer's side of the VPN tunnel.
- You must initiate the VPN tunnel from the CGW to the VPG.
- VPGs support both dynamic routing with Border Gateway Protocol(BGP) and static routing. Border Gateway Protocol (BGP) is a standardized exterior gateway protocol designed to exchange routing and reachability information among autonomous systems (AS) on the Internet
- The VPN connection consists of two tunnels for higher availability to the VPC.


AWS
VPC
VPC Subnet
Availability Zone
VPC Subnet
Availability Zone
Router
Virtual Private Gateway
VPN Connection
Customer Gateway
Customer Network

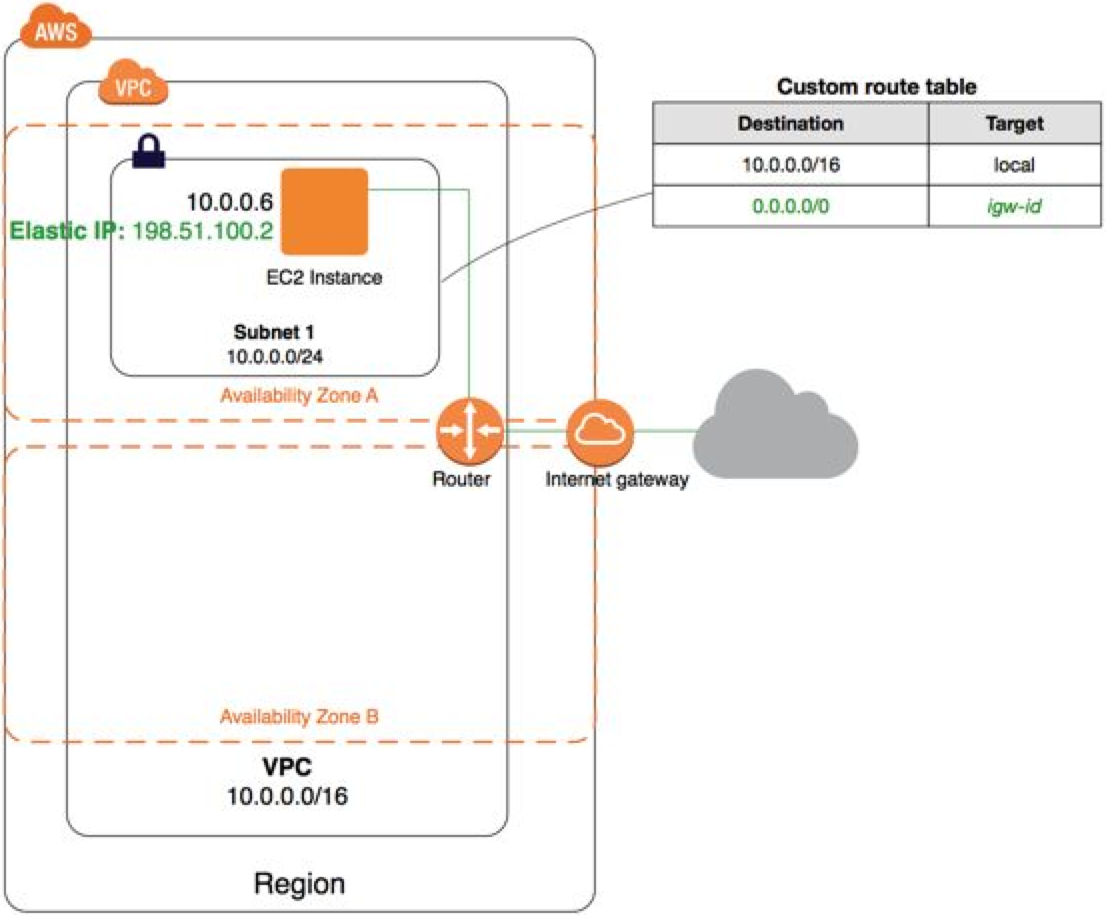
Practical: VPC with a Single Public Subnet

Problem Statement

Scenario: I want to run a single-tier, public-facing web application, such as a blog or a simple website.

What you would need?

- Create a non default VPC with a single public subnet. Subnets enable you to group instances based on your security and operational needs. A public subnet is a subnet that has access to the Internet through an Internet gateway.
- Create a security group for your instance that allows traffic only through specific ports.
- Launch an Amazon EC2 instance into your subnet.
- Associate an Elastic IP address with your instance. This allows your instance to access the Internet.



Custom route table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	*igw-id*

Configuration

- A virtual private cloud (VPC) with a size /16 IPv4 CIDR block (example: 10.0.0.0/16). This provides 65,536 private IPv4 addresses.
- A subnet with a size /24 IPv4 CIDR block (example: 10.0.0.0/24). This provides 256 private IPv4 addresses.
- An Internet gateway. This connects the VPC to the Internet and to other AWS services.
- An instance with a private IPv4 address in the subnet range (example: 10.0.0.6), which enables the instance to communicate with other instances in the VPC, and an Elastic IPv4 address (example: 198.51.100.2), which is a public IPv4 address that enables the instance to be reached from the Internet.
- A custom route table associated with the subnet. The route table entries enable instances in the subnet to use IPv4 to communicate with other instances in the VPC, and to communicate directly over the Internet. A subnet that's associated with a route table that has a route to an Internet gateway is known as a public subnet.

Steps

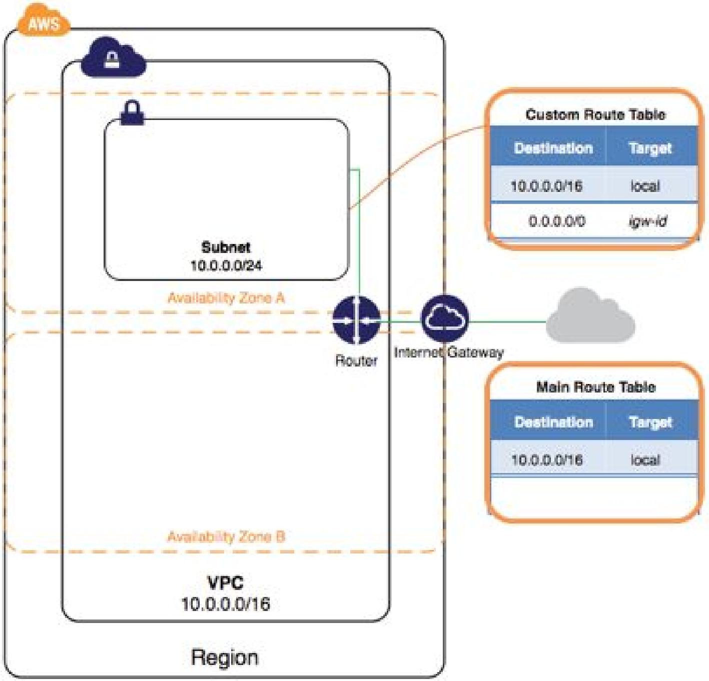
- Create the VPC
- Create a Security Group
- Launch an Instance into Your VPC
- Assign an Elastic IP Address to Your Instance
- Clean Up

Step 1: Create the VPC

Note: We will use the Amazon VPC wizard in the Amazon VPC console to create a VPC.

- Creates a VPC with a /16 IPv4 CIDR block (a network with 65,536 private IP addresses).
- Attaches an Internet gateway to the VPC.
- Creates a size /24 IPv4 subnet (a range of 256 private IP addresses) in the VPC.
- Creates a custom route table, and associates it with your subnet, so that traffic can flow between the subnet and the Internet gateway.

The following diagram represents the architecture of your VPC after you've completed this step.



Custom Route Table

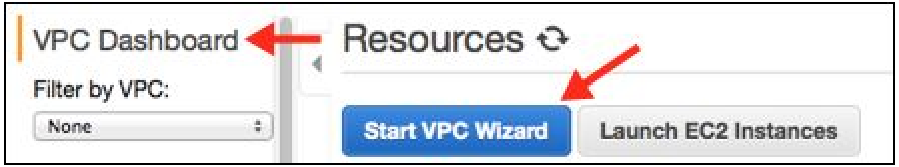
Destination	Target
10.0.0.0/16	local
0.0.0.0/0	*igw-id*

Main Route Table

Destination	Target
10.0.0.0/16	local

Steps

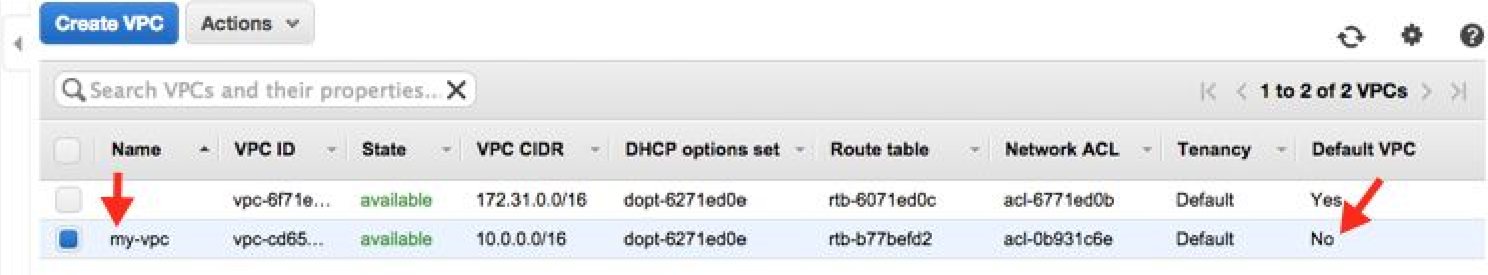
- Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
- In the navigation bar, on the top-right, take note of the region in which you'll be creating the VPC. Ensure that you continue working in the same region for the rest of this exercise, as you cannot launch an instance into your VPC from a different region.
- In the navigation pane, choose **VPC dashboard**, and then choose **Start VPC Wizard**.



- Choose the first option, **VPC with a Single Public Subnet**, and then choose **Select**.
- On the configuration page, enter a name for your VPC in the **VPC name** field; for example, `my-vpc`, and enter a name for your subnet in the **Subnet name** field. This helps you to identify the VPC and subnet in the Amazon VPC console after you've created them. For this exercise, you can leave the rest of the configuration settings on the page, and choose **Create VPC**.

Steps

- A status window shows the work in progress. When the work completes, choose **OK** to close the status window.
- The **Your VPCs** page displays your default VPC and the VPC that you just created. The VPC that you created is a nondefault VPC, therefore the **Default VPC** column displays **No**.



□	Name	VPC ID	State	VPC CIDR	DHCP options set	Route table	Network ACL	Tenancy	Default VPC
□		vpc-6f71e...	available	172.31.0.0/16	dopt-6271ed0e	rtb-6071ed0c	acl-6771ed0b	Default	Yes
☑	my-vpc	vpc-cd65...	available	10.0.0.0/16	dopt-6271ed0e	rtb-b77befd2	acl-0b931c6e	Default	No

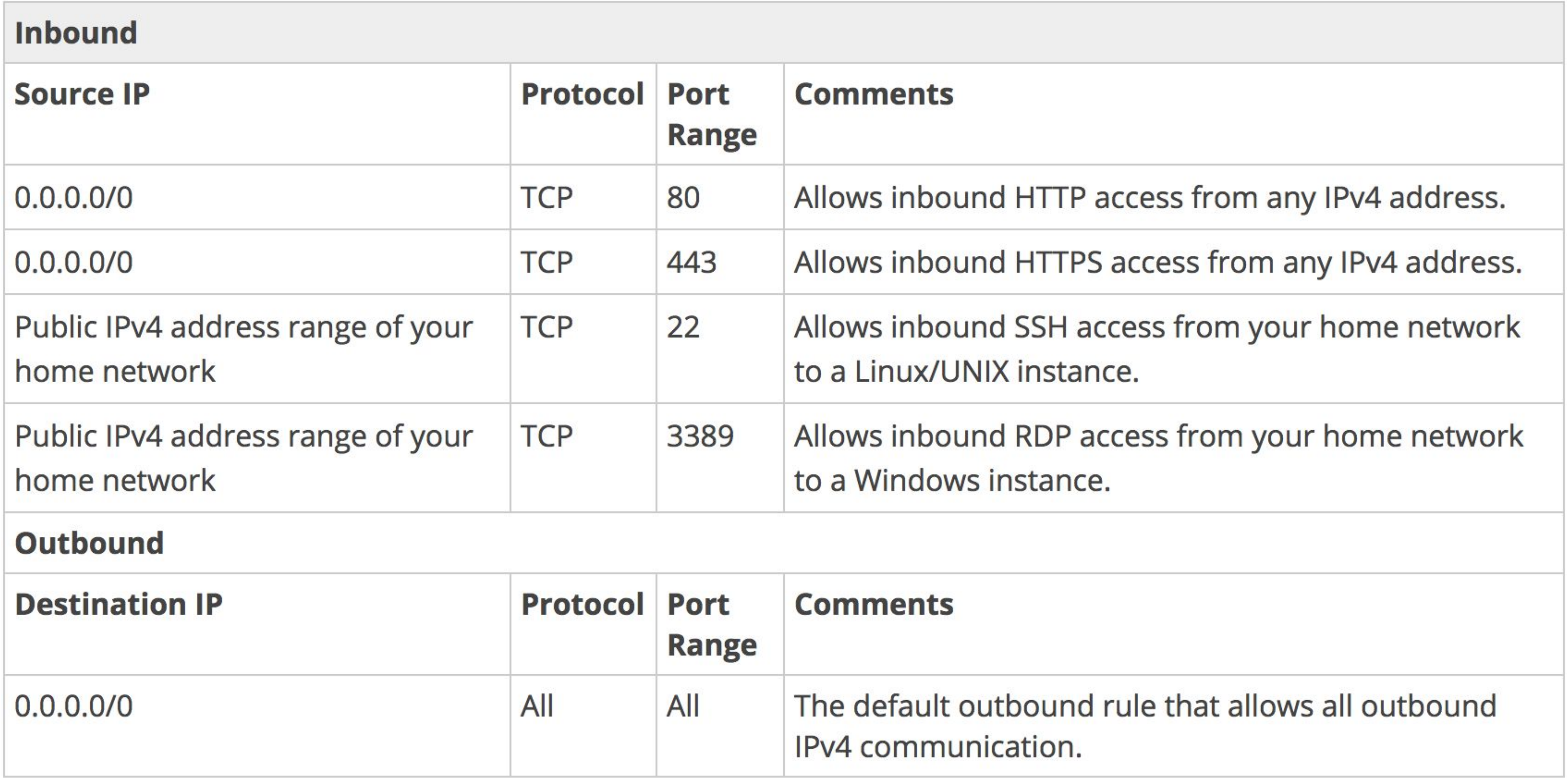
View information about your VPC

- In the navigation pane, choose **Your VPCs**. Take note of the name and the ID of the VPC that you created (look in the **Name** and **VPC ID** columns). You will use this information to identify the components that are associated with your VPC.
- In the navigation pane, choose **Subnets**. The console displays the subnet that was created when you created your VPC. You can identify the subnet by its name in **Name** column, or you can use the VPC information that you obtained in the previous step and look in the **VPC** column.
- In the navigation pane, choose **Internet Gateways**. You can find the Internet gateway that's attached to your VPC by looking at the **VPC** column, which displays the ID and the name (if applicable) of the VPC.
- In the navigation pane, choose **Route Tables**. There are two route tables associated with the VPC. Select the custom route table (the **Main** column displays **No**), and then choose the **Routes** tab to display the route information in the details pane:
  - The first row in the table is the local route, which enables instances within the VPC to communicate. This route is present in every route table by default, and you can't remove it.
  - The second row shows the route that the Amazon VPC wizard added to enable traffic destined for an IPv4 address outside the VPC (`0.0.0.0/0`) to flow from the subnet to the Internet gateway.
- Select the main route table. The main route table has a local route, but no other routes.

Step 2: Create a Security Group

A security group acts as a virtual firewall to control the traffic for its associated instances. To use a security group, you add the inbound rules to control incoming traffic to the instance, and outbound rules to control the outgoing traffic from your instance. To associate a security group with an instance, you specify the security group when you launch the instance. If you add and remove rules from the security group, we apply those changes to the instances associated with the security group automatically.

Your VPC comes with a default security group. Any instance not associated with another security group during launch is associated with the default security group.

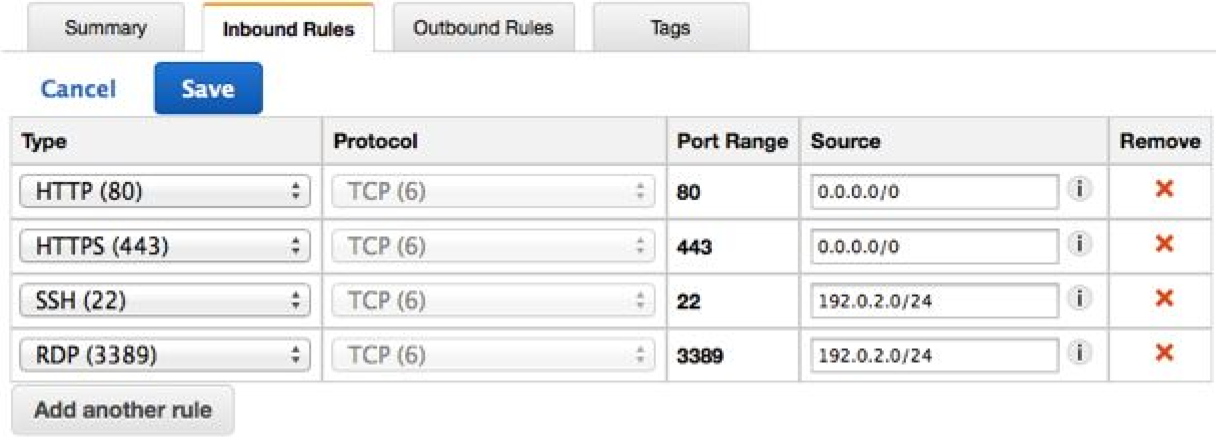


Inbound			
Source IP	**Protocol**	**Port Range**	**Comments**
0.0.0.0/0	TCP	80	Allows inbound HTTP access from any IPv4 address.
0.0.0.0/0	TCP	443	Allows inbound HTTPS access from any IPv4 address.
Public IPv4 address range of your home network	TCP	22	Allows inbound SSH access from your home network to a Linux/UNIX instance.
Public IPv4 address range of your home network	TCP	3389	Allows inbound RDP access from your home network to a Windows instance.
Outbound			
Destination IP	**Protocol**	**Port Range**	**Comments**
0.0.0.0/0	All	All	The default outbound rule that allows all outbound IPv4 communication.

Steps

- Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
- In the navigation pane, choose **Security Groups**.
- Choose **Create Security Group**.
- In the **Group name** field, enter `WebServerSG` as the name of the security group, and provide a description. You can optionally use the **Name tag** field to create a tag for the security group with a key of `Name` and a value that you specify.
- Select the ID of your VPC from the **VPC** menu, and then choose **Yes, Create**.
- Select the `WebServerSG` security group that you just created (you can view its name in the **Group Name**column).
- On the **Inbound Rules** tab, choose **Edit** and add rules for inbound traffic as follows, and then choose **Save**when you're done:
  - Select **HTTP** from the **Type** list, and enter `0.0.0.0/0` in the **Source** field.
  - Choose **Add another rule**, then select **HTTPS** from the **Type** list, and enter `0.0.0.0/0` in the **Source**field.
  - Choose **Add another rule**. If you're launching a Linux instance, select **SSH** from the **Type** list, or if you're launching a Windows instance, select **RDP** from the **Type** list. Enter your network's public IP address range in the **Source** field. If you don't know this address range, you can use `0.0.0.0/0` for this exercise.

Steps

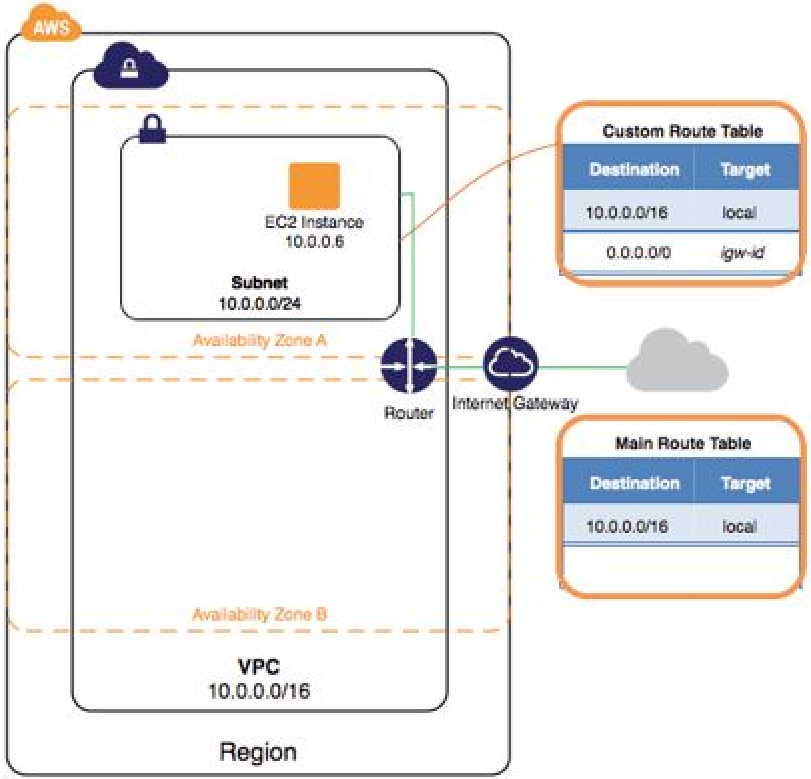


Type	Protocol	Port Range	Source	Remove
HTTP (80)	TCP (6)	80	0.0.0.0/0	✕
HTTPS (443)	TCP (6)	443	0.0.0.0/0	✕
SSH (22)	TCP (6)	22	192.0.2.0/24	✕
RDP (3389)	TCP (6)	3389	192.0.2.0/24	✕

Note: If you use 0.0.0.0/0, you enable all IP addresses to access your instance using SSH or RDP. This is acceptable for the short exercise, but it's unsafe for production environments. In production, you'll authorize only a specific IP address or range of addresses to access your instance.

Step 3: Launch an Instance into Your VPC

When you launch an EC2 instance into a VPC, you must specify the subnet in which to launch the instance. In this case, you'll launch an instance into the public subnet of the VPC you created. You'll use the Amazon EC2 launch wizard in the Amazon EC2 console to launch your instance.



Custom Route Table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	*igw-id*

Main Route Table

Destination	Target
10.0.0.0/16	local

Steps

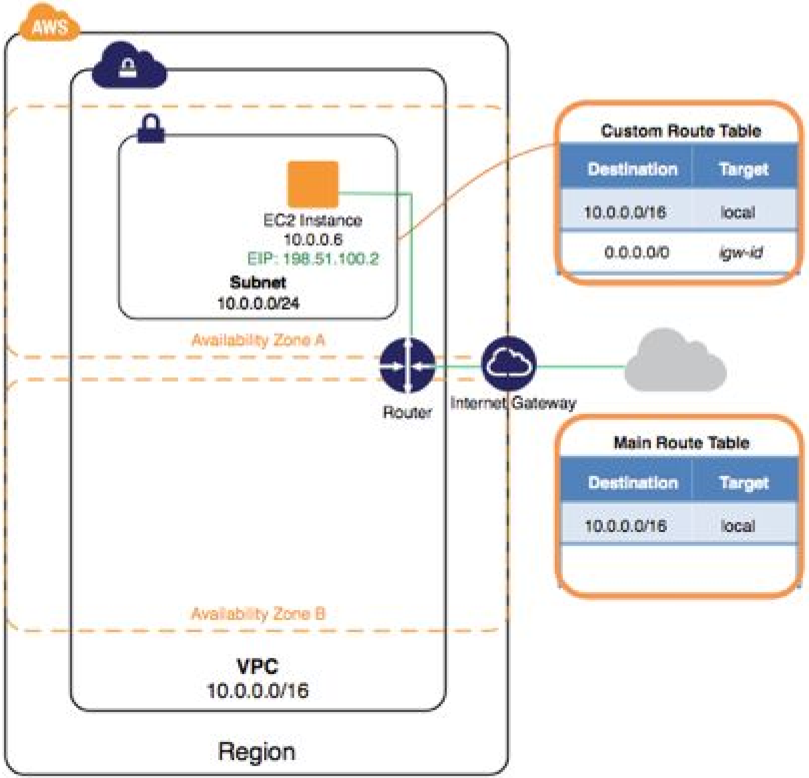
- Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
- In the navigation bar, on the top-right, ensure that you select the same region in which you created your VPC and security group.
- From the dashboard, choose **Launch Instance**.
- On the first page of the wizard, choose the AMI that you want to use. For this exercise, we recommend that you choose an Amazon Linux AMI or a Windows AMI.
- On the **Choose an Instance Type** page, you can select the hardware configuration and size of the instance to launch. By default, the wizard selects the first available instance type based on the AMI you selected. You can leave the default selection, and then choose **Next: Configure Instance Details**.
- On the **Configure Instance Details** page, select the VPC that you created from the **Network** list, and the subnet from the **Subnet** list. Leave the rest of the default settings, and go through the next pages of the wizard until you get to the **Add Tags** page.
- On the **Add Tags** page, you can tag your instance with a `Name` tag; for example `Name=MyWebServer`. This helps you to identify your instance in the Amazon EC2 console after you've launched it. Choose **Next: Configure Security Group** when you are done.

Steps

- On the **Configure Security Group** page, the wizard automatically defines the launch-wizard-*x* security group to allow you to connect to your instance. Instead, choose the **Select an existing security group**option, select the **WebServerSG** group that you created previously, and then choose **Review and Launch**.
- On the **Review Instance Launch** page, check the details of your instance, and then choose **Launch**.
- In the **Select an existing key pair or create a new key pair** dialog box, you can choose an existing key pair, or create a new one. If you create a new key pair, ensure that you download the file and store it in a secure location. You'll need the contents of the private key to connect to your instance after it's launched.
- To launch your instance, select the acknowledgment check box, and then choose **Launch Instances**.
- On the confirmation page, choose **View Instances** to view your instance on the **Instances** page. Select your instance, and view its details in the **Description** tab. The **Private IPs** field displays the private IP address that's assigned to your instance from the range of IP addresses in your subnet.

Step 4: Assign an Elastic IP Address to Your Instance

In the previous step, you launched your instance into a public subnet — a subnet that has a route to an Internet gateway. However, the instance in your subnet also needs a public IPv4 address to be able to communicate with the Internet. By default, an instance in a nondefault VPC is not assigned a public IPv4 address. In this step, you'll allocate an Elastic IP address to your account, and then associate it with your instance.



Custom Route Table

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

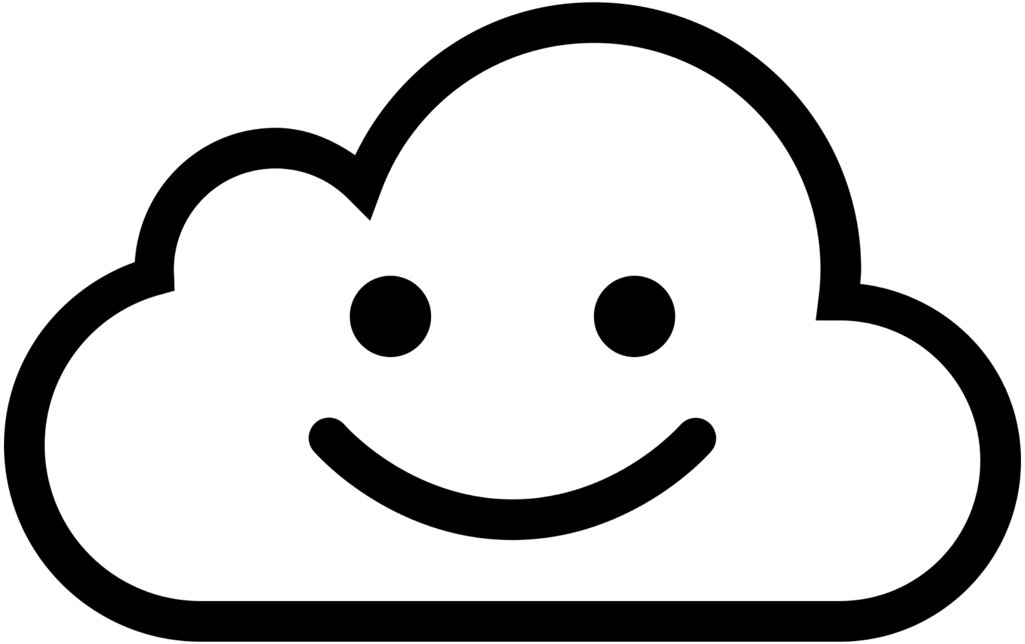
Main Route Table

Destination	Target
10.0.0.0/16	local

Steps

- Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
- In the navigation pane, choose **Elastic IPs**.
- Choose **Allocate New Address**, and then **Yes, Allocate**.
  - **Note**
  - If your account supports EC2-Classic, first select **EC2-VPC** from the **Network platform** list.
- Select the Elastic IP address from the list, choose **Actions**, and then choose **Associate Address**.
- In the dialog box, choose **Instance** from the **Associate with** list, and then select your instance from the**Instance** list. Choose **Yes, Associate** when you're done.

Your instance is now accessible from the Internet. You can connect to your instance through its Elastic IP address using SSH or Remote Desktop from your home network !!!!



Step 5: Clean Up

Before you can delete a VPC, you must terminate any instances that are running in the VPC. If you delete a VPC using the VPC console, it also deletes resources that are associated with the VPC, such as subnets, security groups, network ACLs, DHCP options sets, route tables, and Internet gateways.

Steps

- Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
- In the navigation pane, choose **Instances**.
- Select your instance, choose **Actions**, then **Instance State**, and then select **Terminate**.
- In the dialog box, expand the **Release attached Elastic IPs** section, and select the check box next to the Elastic IP address. Choose **Yes, Terminate**.
- Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
- In the navigation pane, choose **Your VPCs**.
- Select the VPC, choose **Actions**, and then choose **Delete VPC**.
- When prompted for confirmation, choose **Yes, Delete**.

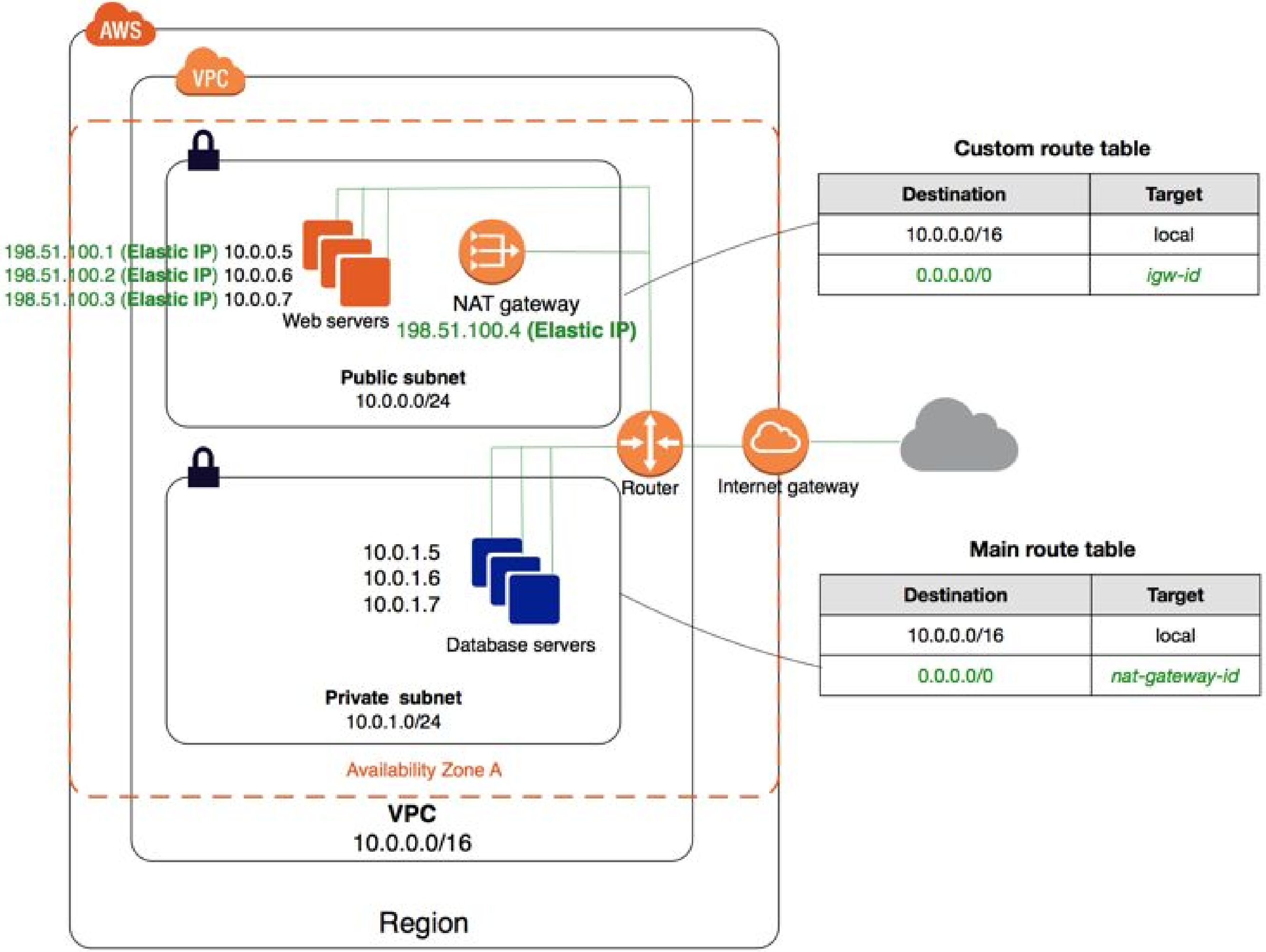
Practical: VPC with Public and Private Subnets

Problem Statement

Scenario: I want to run a public-facing web application, while maintaining back-end servers that aren't publicly accessible

What you would need?

- Create a non default VPC with a single public subnet and a private subnet. Subnets enable you to group instances based on your security and operational needs. A public subnet is a subnet that has access to the Internet through an Internet gateway.
- Create the web server EC2 instance in a public subnet and the database server EC2 instance in a private subnet.
- Setup security and routing so that the web servers can communicate with the database servers.
- Instances in the private subnet can access the Internet by using a network address translation (NAT) gateway that resides in the public subnet. The database servers can connect to the Internet for software updates using the NAT gateway, but the Internet cannot establish connections to the database servers.



Custom route table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	*igw-id*

Main route table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	*nat-gateway-id*

Configuration

- A VPC with a size /16 IPv4 CIDR block (example: 10.0.0.0/16). This provides 65,536 private IPv4 addresses.
- A public subnet with a size /24 IPv4 CIDR block (example: 10.0.0.0/24). This provides 256 private IPv4 addresses. A public subnet is a subnet that's associated with a route table that has a route to an Internet gateway.
- A private subnet with a size /24 IPv4 CIDR block (example: 10.0.1.0/24). This provides 256 private IPv4 addresses.
- An Internet gateway. This connects the VPC to the Internet and to other AWS services.
- Instances with private IPv4 addresses in the subnet range (examples: 10.0.0.5, 10.0.1.5). This enables them to communicate with each other and other instances in the VPC.
- Instances in the public subnet with Elastic IPv4 addresses (example: 198.51.100.1), which are public IPv4 addresses that enable them to be reached from the Internet. The instances can have public IP addresses assigned at launch instead of Elastic IP addresses. Instances in the private subnet are back-end servers that don't need to accept incoming traffic from the Internet and therefore do not have public IP addresses; however, they can send requests to the Internet using the NAT gateway (see the next bullet).
- A NAT gateway with its own Elastic IPv4 address. Instances in the private subnet can requests to the Internet through the NAT gateway over IPv4 (for example, for software updates).
- A custom route table associated with the public subnet. This route table contains an entry that enables instances in the subnet to communicate with other instances in the VPC over IPv4, and an entry that enables instances in the subnet to communicate directly with the Internet over IPv4.
- The main route table associated with the private subnet. The route table contains an entry that enables instances in the subnet to communicate with other instances in the VPC over IPv4, and an entry that enables instances in the subnet to communicate with the Internet through the NAT gateway over IPv4.

Step 1: Create an Elastic IP

To allocate an Elastic IP address for the NAT gateway (IPv4)

- Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
- In the navigation pane, choose **Elastic IPs**.
- Choose **Allocate New Address**.
- Choose **Yes, Allocate**.

Step 2: Create a VPC

- Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
- On the VPC dashboard, choose **Start VPC Wizard**.
- Choose the second option, **VPC with Public and Private Subnets**, and **Select**.
- For **VPC name**, **Public subnet name** and **Private subnet name**, you can name your VPC and subnets to help you identify them later in the console. You can specify your own IPv4 CIDR block range for the VPC and subnets, or you can leave the default values.
- In the **Specify the details of your NAT gateway** section, specify the allocation ID for an Elastic IP address in your account.
- You can leave the rest of the default values on the page, and choose **Create VPC**.

Step 3: Create Security Groups

To create the WebServerSG and DBServerSG security groups

- Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
- In the navigation pane, choose **Security Groups**, **Create Security Group**.
- Provide a name and description for the security group. In this topic, the name `WebServerSG` is used as an example. For **VPC**, select the ID of the VPC you created and choose **Yes, Create**.
- Choose **Create Security Group** again.
- Provide a name and description for the security group. In this topic, the name `DBServerSG` is used as an example. For **VPC**, select the ID of your VPC and choose **Yes, Create**.

Step 4: Add rules to the Security Groups

- Select the WebServerSG security group that you created. The details pane displays the details for the security group, plus tabs for working with its inbound and outbound rules.
- On the **Inbound Rules** tab, choose **Edit** and add rules for inbound traffic as follows:
  - Choose **Type**, **HTTP**. For **Source**, enter `0.0.0.0/0`.
  - Choose **Add another rule**, **Type**, **HTTPS**. For **Source**, enter `0.0.0.0/0`.
  - Choose **Add another rule**, **Type**, **SSH**. For **Source**, enter your network's public IPv4 address range.
  - Choose **Add another rule**, **Type**, **RDP**. For **Source**, enter your network's public IPv4 address range.
  - Choose **Save**.

Step 4: Add rules to the Security Groups

- On the **Outbound Rules** tab, choose **Edit** and add rules for outbound traffic as follows:
  - Locate the default rule that enables all outbound traffic and choose **Remove**.
  - Choose **Type**, **MS SQL**. For **Destination**, specify the ID of the DBServerSG security group.
  - Choose **Add another rule**, **Type**, **MySQL**. For **Destination**, specify the ID of the DBServerSG security group.
  - Choose **Add another rule**, **Type**, **HTTPS**. For **Destination**, enter `0.0.0.0/0`.
  - Choose **Add another rule**, **Type**, **HTTP**. For **Destination**, enter `0.0.0.0/0`.
  - Choose **Save**.

Step 4: Add rules to the Security Groups

To add the recommended rules to the DBServerSG security group

1. Select the DBServerSG security group that you created. The details pane displays the details for the security group, plus tabs for working with its inbound and outbound rules.
2. On the **Inbound Rules** tab, choose **Edit** and add rules for inbound traffic as follows:
   a. Choose **Type**, **MS SQL**. For **Source**, specify the ID of your WebServerSG security group.
   b. Choose **Add another rule**, **Type**, **MYSQL**. For **Source**, specify the ID of your WebServerSG security group.
   c. Choose **Save**.
3. On the **Outbound Rules** tab, choose **Edit** and add rules for outbound traffic as follows:
   a. Locate the default rule that enables all outbound traffic and choose **Remove**.
   b. Choose **Type**, **HTTP**. For **Destination**, enter `0.0.0.0/0`.
   c. Choose **Add another rule**, **Type**, **HTTPS**. For **Destination**, enter `0.0.0.0/0`.
   d. Choose **Save**.

You can now launch instances into your VPC.

Step 5: Launch Instances

- From the EC2 dashboard, choose **Launch Instance**.
- Select an AMI and an instance type and choose **Next: Configure Instance Details**.
- On the **Configure Instance Details** page, for **Network**, select the VPC that you created earlier and then select a subnet. For example, launch a web server into the public subnet and the database server into the private subnet.
- (Optional) By default, instances launched into a non default VPC are not assigned a public IPv4 address. To be able to connect to your instance in the public subnet, you can assign a public IPv4 address now, or allocate an Elastic IP address and assign it to your instance after it's launched. To assign a public IPv4 address now, ensure that you choose **Enable** from the **Auto-assign Public IP** list. You do not need to assign a public IP address to an instance in the private subnet.
  - **Note:** You can only use the auto-assign public IPv4 feature for a single, new network interface with the device index of eth0.
- On the next two pages of the wizard, you can configure storage for your instance, and add tags. On the **Configure Security Group** page, choose the **Select an existing security group** option, and select one of the security groups you created earlier (**WebServerSG** for web server / **DBServerSG** for DB server). Choose **Review and Launch**.
- Review the settings that you've chosen. Make any changes that you need and choose **Launch** to choose a key pair and launch your instance.
- If you did not assign a public IPv4 address to your instance in the public subnet in step 5, you will not be able to connect to it. Before you can access an instance in your public subnet, you must assign it an Elastic IP address.

Step 6: Allocate Elastic IP address to an instance

1. Select the Elastic IP address from the VPC -> **Elastic IPs** list and choose **Actions**, **Associate address**.
2. Select the network interface or instance. For **Private IP**, select the corresponding address to associate the Elastic IP address with and choose **Associate**.

Step 7: Test Internet Connectivity

Exam Essentials

- **Understand what a VPC is and its core and optional components.**
  - Logically isolated network in the AWS Cloud.
  - Spans all the Availability Zones in the region
  - Mandatory components subnets (public, private, and VPN-only), route tables, DHCP option sets, security groups, and network ACLs.
  - Optional elements include an IGW, EIP addresses, endpoints, peering connections, NAT instances, VPGs, CGWs, and VPN connections.
- **Understand the purpose of a subnet.**
  - Segment of an Amazon VPC’s IP address range where you can place groups of isolated resources.
  - Subnets are defined by CIDR blocks—for example, 10.0.1.0/24 and 10.0.2.0/24
  - Contained within an Availability Zone.
- **Identify the difference between a public subnet, a private subnet, and a VPN-Only subnet.**
  - Subnet’s traffic is routed to an IGW, then public subnet.
  - Doesn’t have a route to the IGW, then private subnet.
  - Doesn’t have a route to the IGW, but has its traffic routed to a VPG, then VPN-only subnet.
-

Exam Essentials

- **Understand the purpose of a route table.**
  - Set of rules (called routes) that are used to determine where network traffic is directed. A route table allows Amazon EC2 instances within different subnets to communicate with each other (within the same Amazon VPC). The Amazon VPC router also enables subnets, IGWs, and VPGs to communicate with each other.
- **Understand the purpose of an IGW.**
  - Horizontally scaled, redundant, and highly available Amazon VPC component
  - Allows communication between instances in your Amazon VPC and the Internet.
  - Fully redundant and have no bandwidth constraints.
  - Provides a target in your Amazon VPC route tables for Internet-routable traffic and performs network address translation for instances that have been assigned public IP addresses.
- **Understand what DHCP option sets provide to an Amazon VPC.**
  - Allows you to direct Amazon EC2 hostname assignment to your own resources.
  - You can specify the domain name for instances within an Amazon VPC and identify the IP addresses of custom DNS servers, NTP servers, and NetBIOS servers.

Exam Essentials

- **Know the difference between an Amazon VPC public IP address and an EIP address.**
  - Public IP address is an AWS-owned IP, automatically assigned to instances launched within a subnet.
  - EIP address is an AWS-owned public IP address, you allocate to your account and assign to instances or network interfaces on demand.
- **Understand what endpoints provide to an Amazon VPC.**
  - Amazon VPC endpoint enables you to create a private connection between your Amazon VPC and another AWS service without requiring access over the Internet or through a NAT instance, a VPN connection, or AWS Direct Connect. Endpoints support services within the region only.
- **Understand Amazon VPC peering.**
  - Is a networking connection between two Amazon VPCs that enables instances in either Amazon VPC to communicate with each other as if they are within the same network. Peering connections are created through a request/accept protocol. Transitive peering is not supported, and peering is only available between Amazon VPCs within the same region.

Exam Essentials

- **Know the difference between a security group and a network ACL.**
  - A security group applies at the instance level. You can have multiple instances in multiple subnets that are members of the same security groups. Security groups are stateful, which means that return traffic is automatically allowed, regardless of any outbound rules. A network ACL is applied on a subnet level, and traffic is stateless. You need to allow both inbound and outbound traffic on the network ACL in order for Amazon EC2 instances in a subnet to be able to communicate over a particular protocol.
- **Understand what a NAT provides to an Amazon VPC.**
  - A NAT instance or NAT gateway enables instances in a private subnet to initiate outbound traffic to the Internet. This allows outbound Internet communication to download patches and updates, for example, but prevents the instances from receiving inbound traffic initiated by someone on the Internet.
- **Understand the components needed to establish a VPN connection from a network to an Amazon VPC.**
  - A VPG is the VPN concentrator on the AWS side of the VPN connection between the two networks. A CGW represents a physical device or a software application on the customer’s side of the VPN connection. The VPN connection must be initiated from the CGW side, and the connection consists of two IPSec tunnels.

Test

Test