Grau en Enginyeria de Dades

Computer Architecture & Operating Systems Department

Cloud Computing (2023-24)

Session 5: Introduction to VPC

Professors: Daniel Franco / Carles Carrillo / José Gabriel Mesa

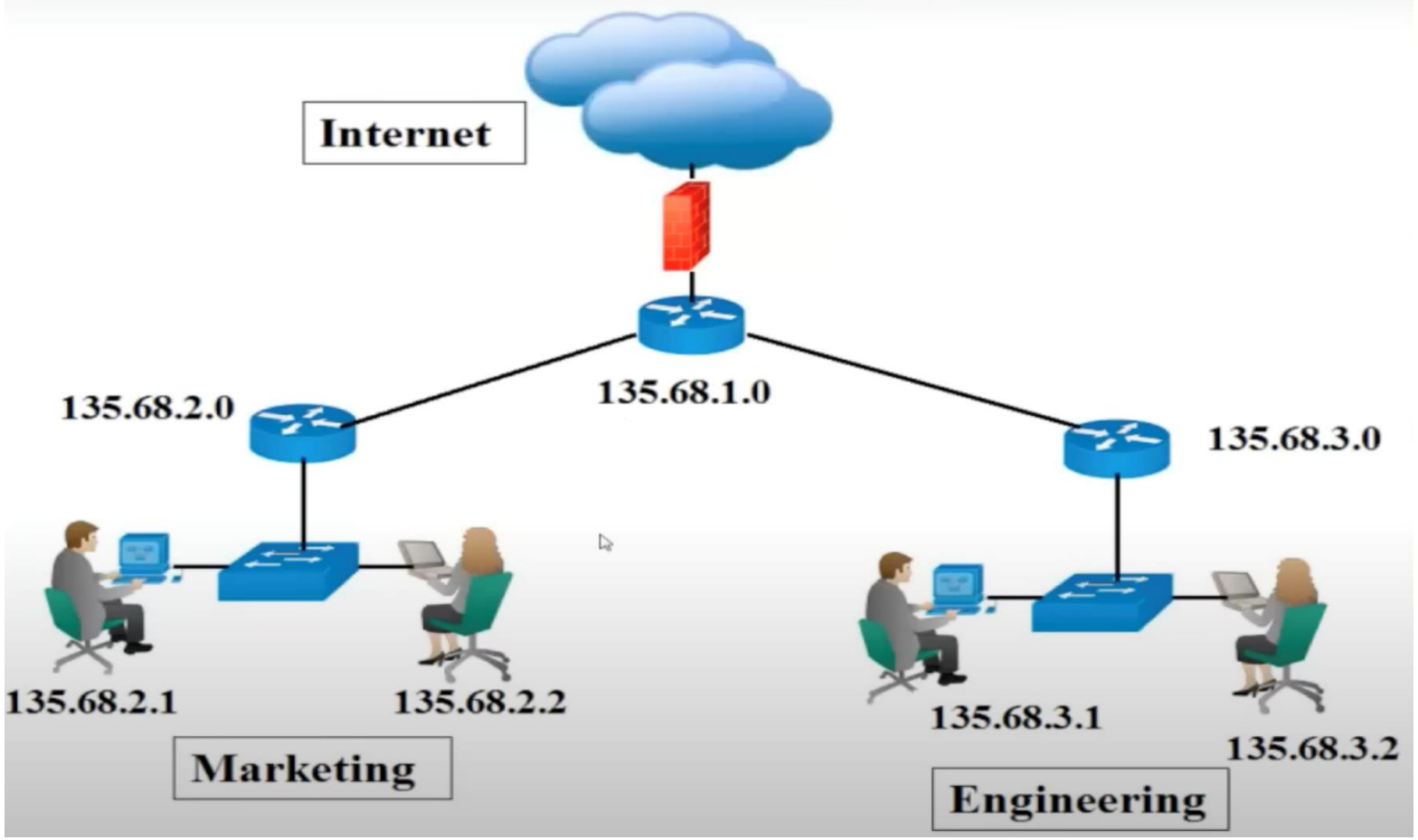
UAB
Universitat Autònoma
de Barcelona

Agenda

- Introduction to networking
- AWS Global Infraestructure
- VPC (Virtual Private Cloud)
- Internet Gateway
- Route Tables
- NAT (Network Adress Translation)
- Security Groups
- ACL’s (Acces Control List)
- Short demo About VPC

Introduction to networking: Computer Network

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

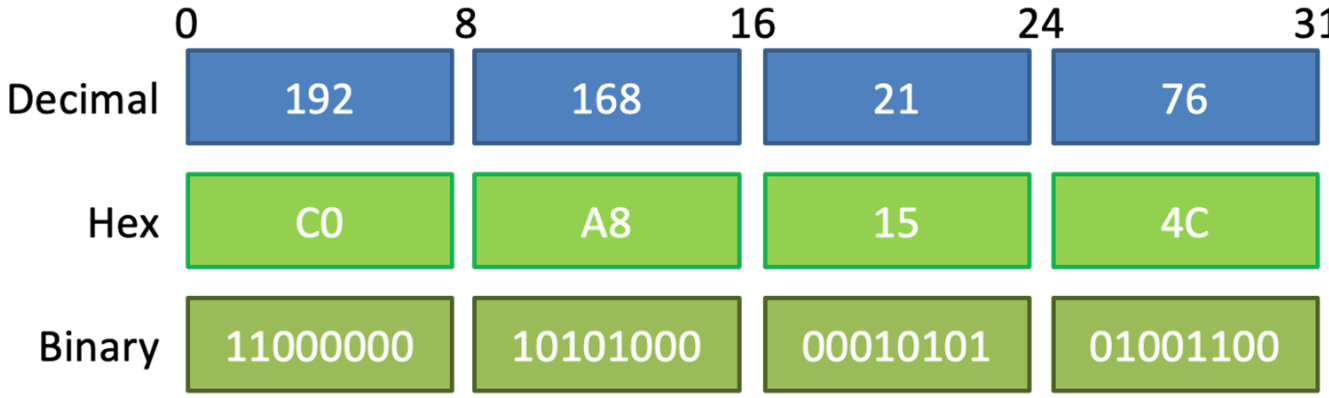


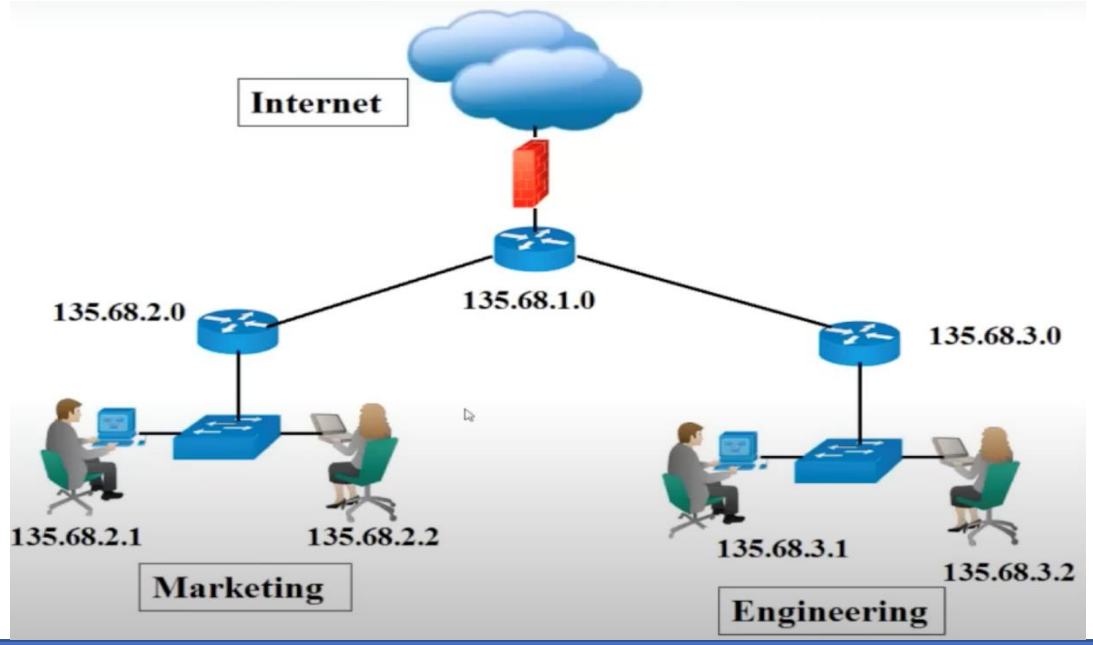
Introduction to networking: Computer Network

How can we identify a specific machine within a subnet ?

Introduction to networking: IP

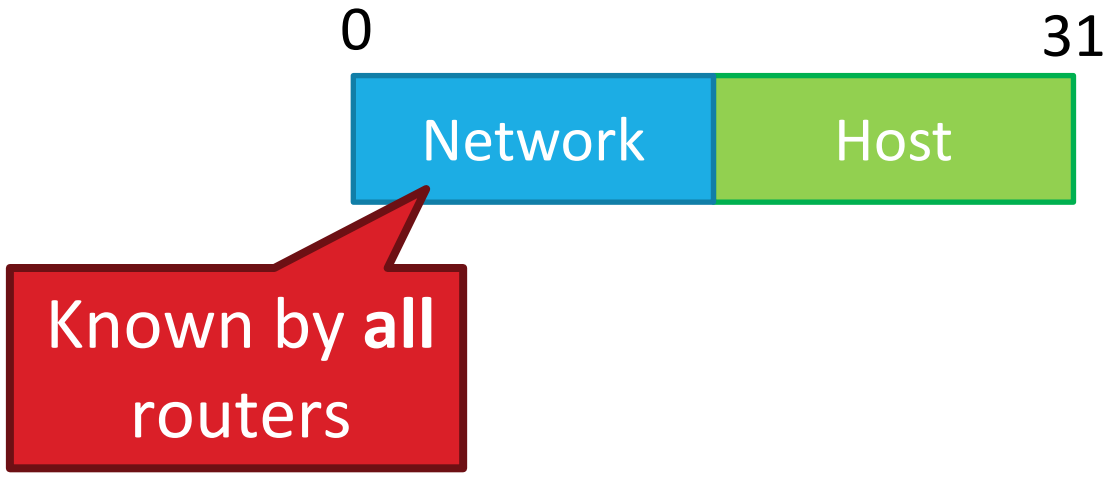
- Each machine in a network has a unique Internet Protocol (IP) address that identifies it.
- An IP address is a numerical label in decimal format. Machines convert that decimal number to a binary format.
- It consists of a series of four numbers separated by dots, for example, 192.168.21.76. Each number can range from 0 to 255.
- The combined total of the four numbers for an IP address is 32 bits in binary format. A 32-bit IP address is called an IPv4 Adress.
- All the IP addresses of computers on the same network start with the same number to identify the network and the host (computer).





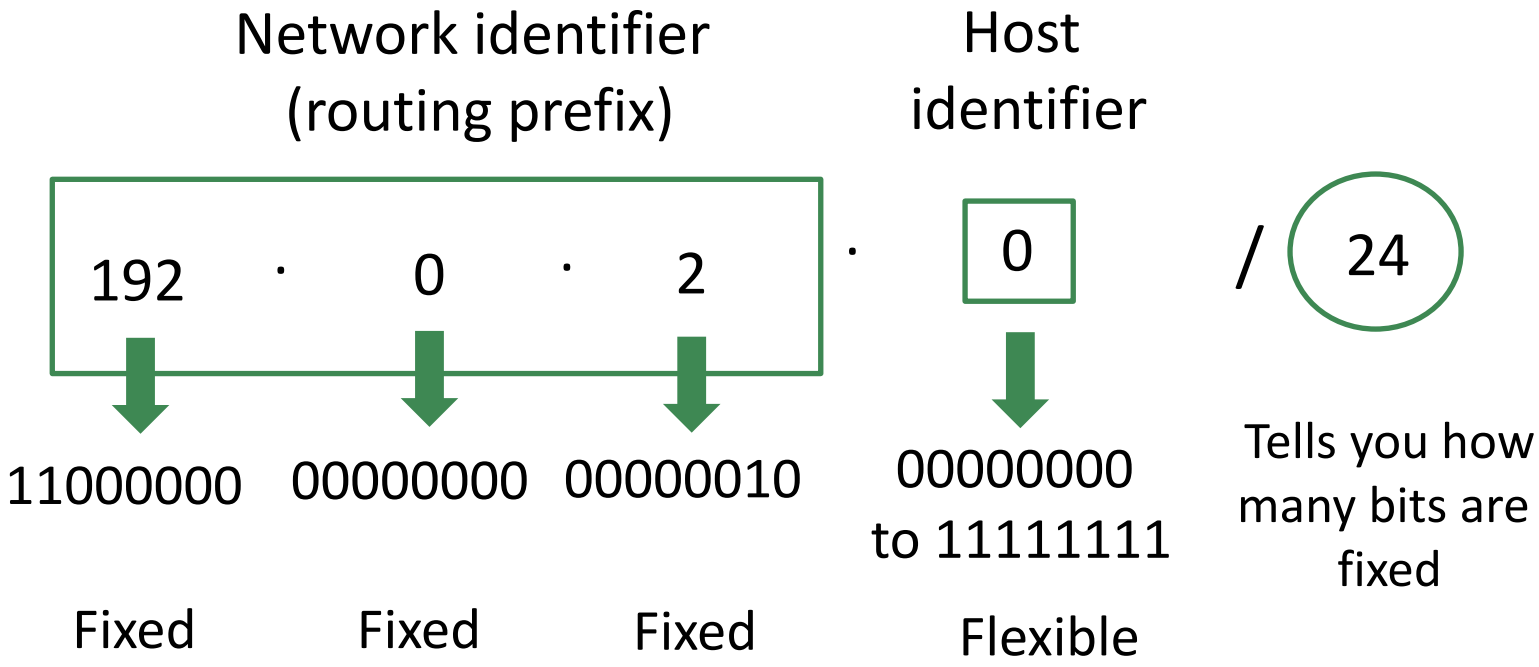
Introduction to networking: IP

- Each IP address consists of two parts, one part identifies the network where the system is connected, and the second part identifies the specific system within that network:
  - **Host Number**: One, two, or even three octets can be used to identify the network.
  - **Network Number**: Up to 3 octets can be used to identify devices on a network.
- The number of octets used to differentiate the network or devices (host) has given rise to a classification of IP addresses called classes.



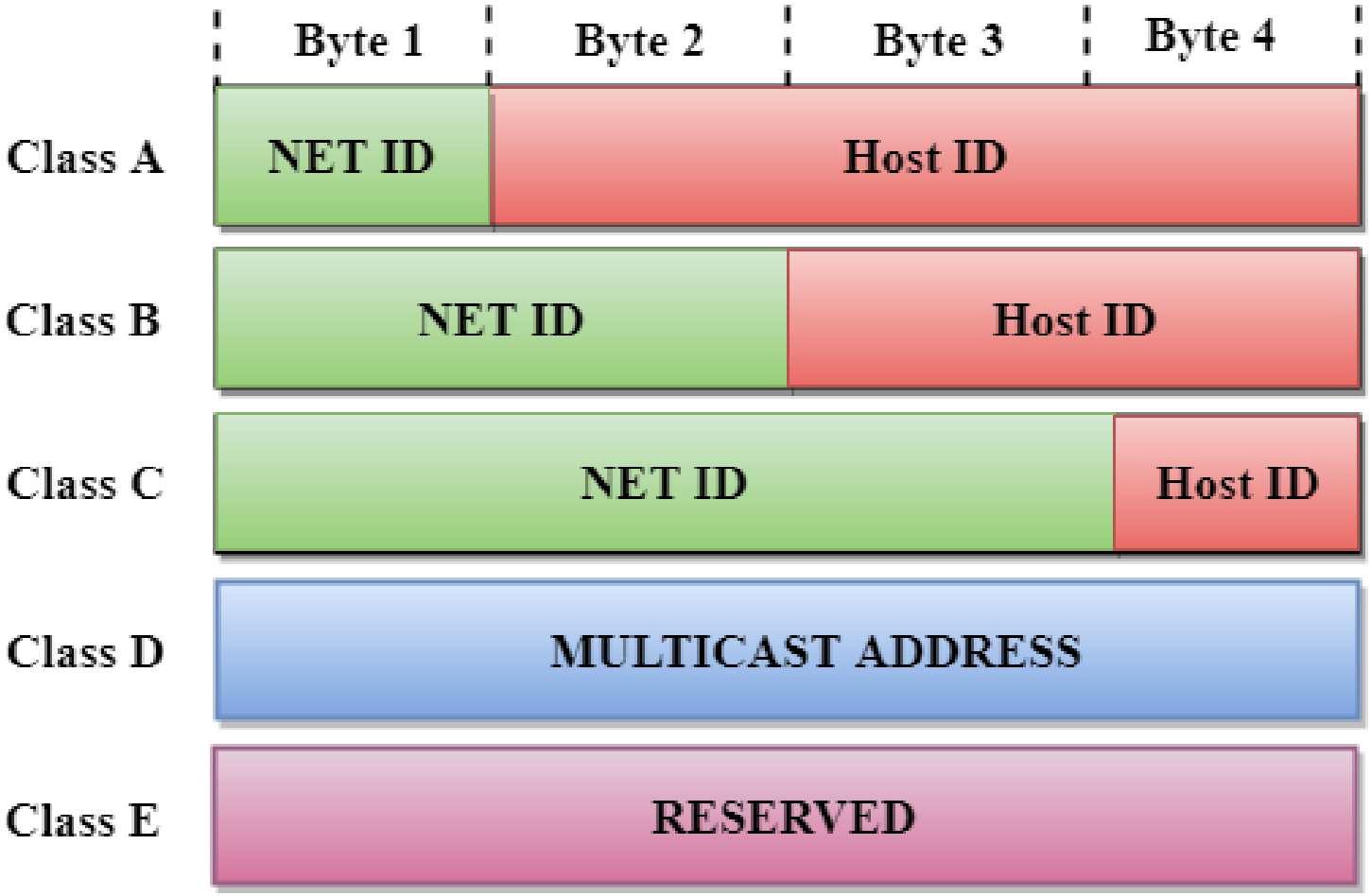
Introduction to networking: Classless Inter-Domain Routing (CIDR)

- A common method to describe networks and subnetworks is Classless Inter-Domain Routing (CIDR).
- The CIDR address is expressed as follows:
  - An IP address (which is the first address of the network)
  - A slash character (/)
  - A number that tells you how many bits of the routing prefix must be fixed or allocated for the network identifier

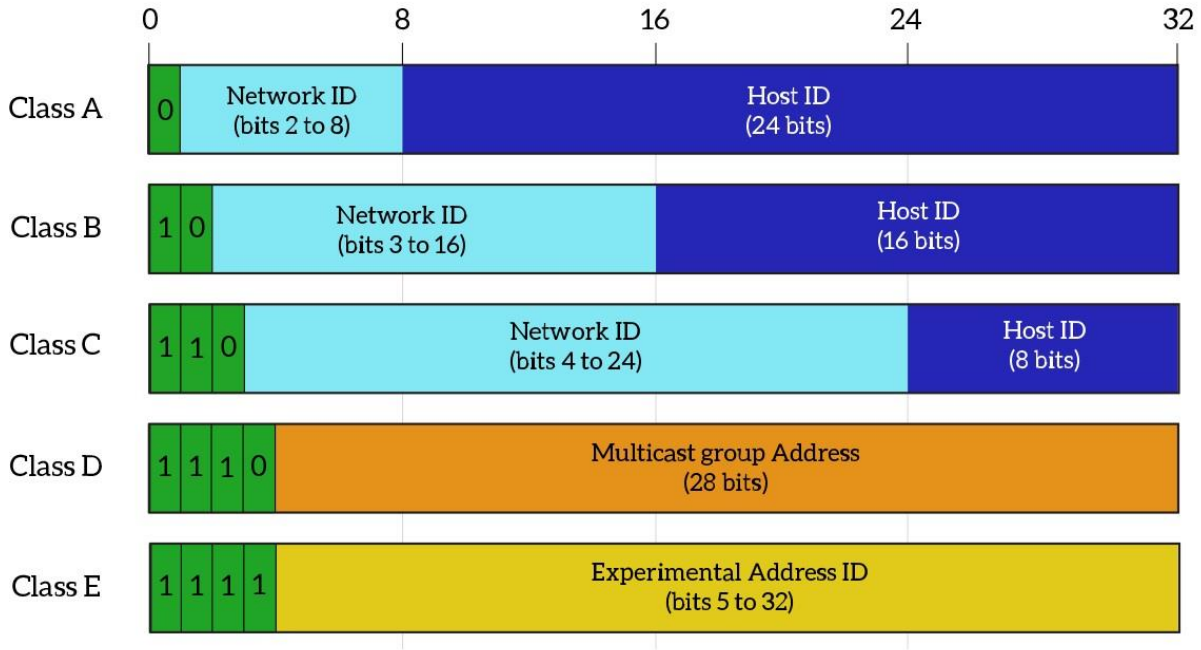


Introduction to networking: IP Classes

- The class of IP address is used to determine the bits used for network ID and host ID and the number of total networks and hosts possible in that particular class.
- There are five different classes, from A to E. A is used by huge companies and C by médium and small companies
- Classes D and E are reserved for multicast and experimental purposes respectively.
- To identify the network and the host we use the mask.



Introduction to networking: IP Classes

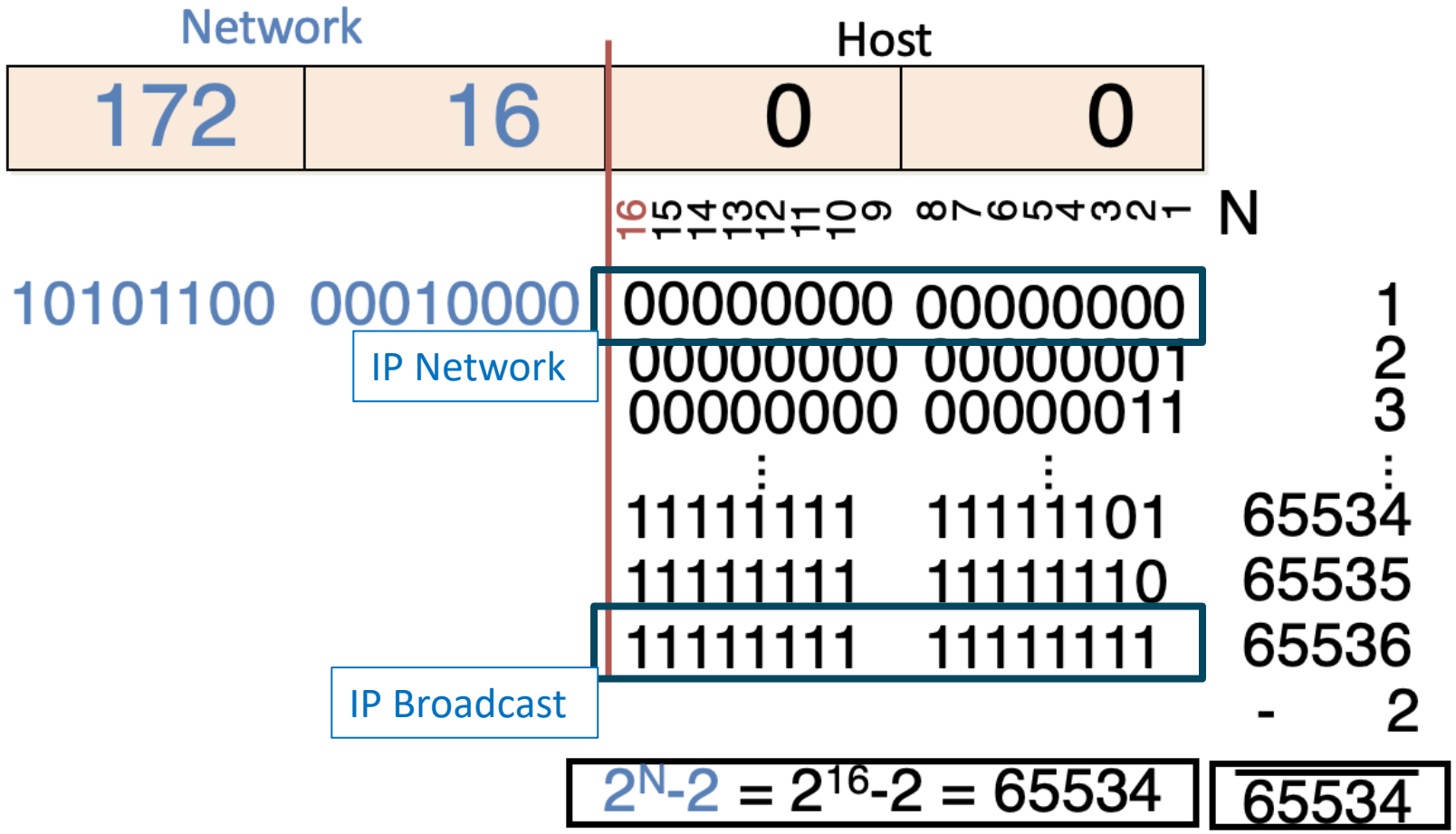


Address Class	RANGE	Default Subnet Mask
A	1.0.0.0 to 126.255.255.255	255.0.0.0
B	128.0.0.0 to 191.255.255.255	255.255.0.0
C	192.0.0.0 to 223.255.255.255	255.255.255.0
D	224.0.0.0 to 239.255.255.255	Reserved for Multicasting
E	240.0.0.0 to 254.255.255.255	Experimental

Note: Class A addresses 127.0.0.0 to 127.255.255.255 cannot be used and is reserved for loopback testing.

Introduction to networking: Host adresses

- As it (172.16.X.X) falls within the range between 128.0.0.0 and 223.255.255.255, it belongs to class B.
- Class B uses 16 bits to identify the network and 16 bits to identify the hosts (machines on the network).
- This network can have up to 65,534 different machines connected.

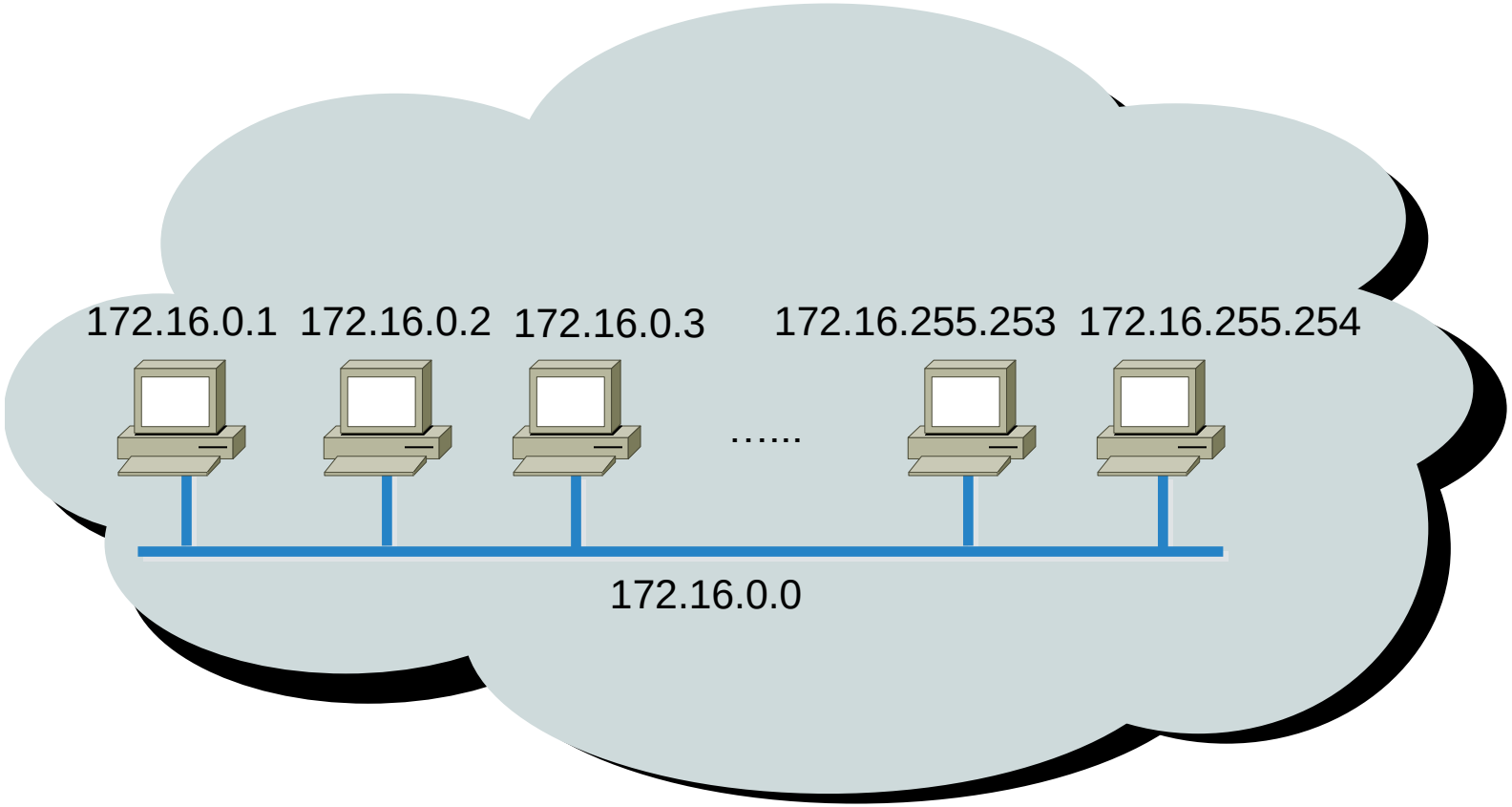


Introduction to networking: Host adresses

Class	Start	End	Subnet Mask	Number of Fixed Bits (R)	Number of Subnets 2^{m-R}	Number of Hosts $2^{m}-2$
A	0.0.0.0	127.255.255.255	255.0.0.0	1	128	16.777.214
B	128.0.0.0	191.255.255.255	255.255.0.0	2	16.384	65.534
C	192.0.0.0	223.255.255.255	255.255.255.0	3	2.097.152	254
D	224.0.0.0	239.255.255.255				
E	240.0.0.0	255.255.255.255				

Introduction to networking: Addressing without subnets

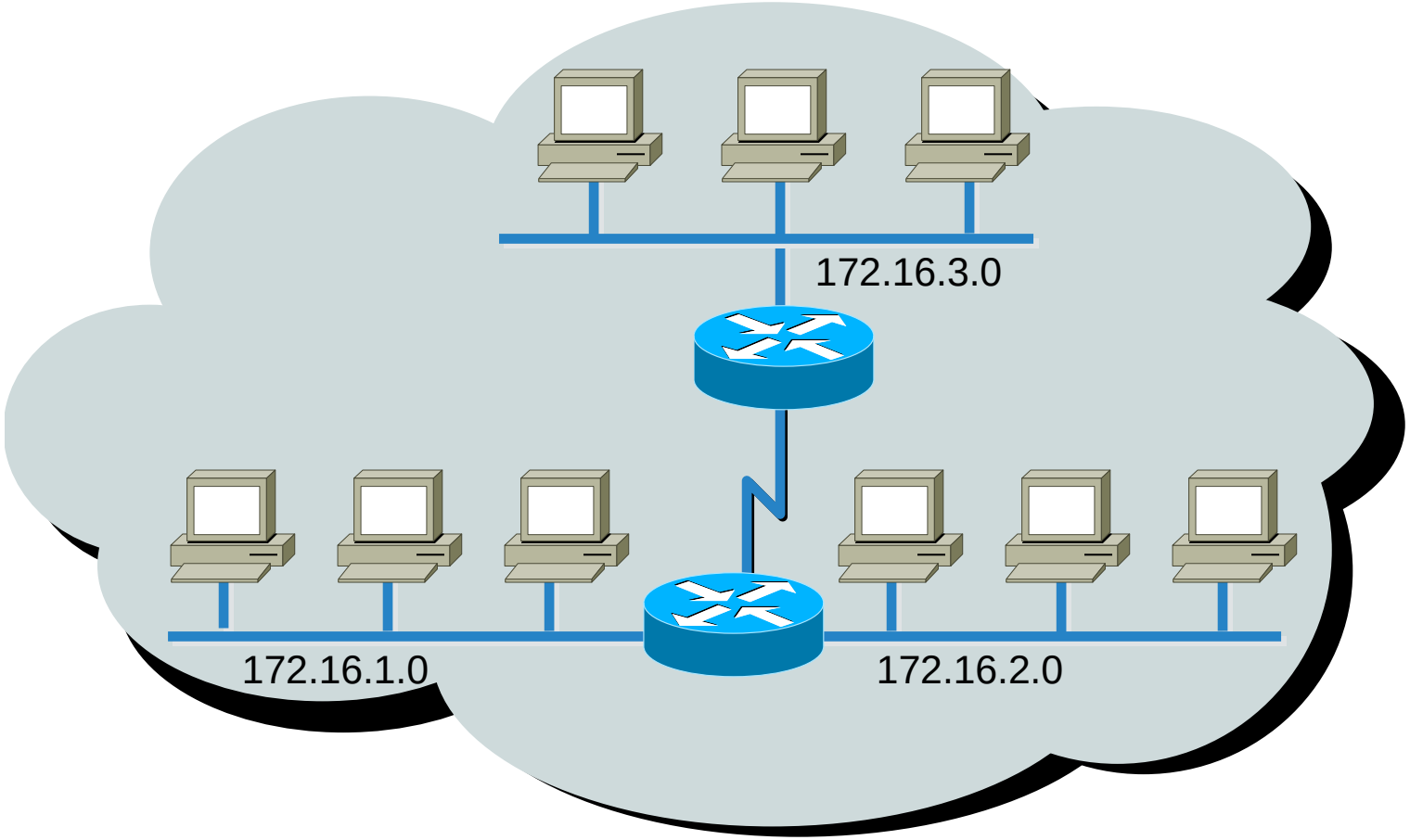
- Without subnets, the use of network addressing space is inefficient.
- The Class B network is like a highway with no exits, there is no place to exit, so all of the traffic is in one line (Bottleneck).



Network 172.16.0.0

Introduction to networking: Addressing with subnets

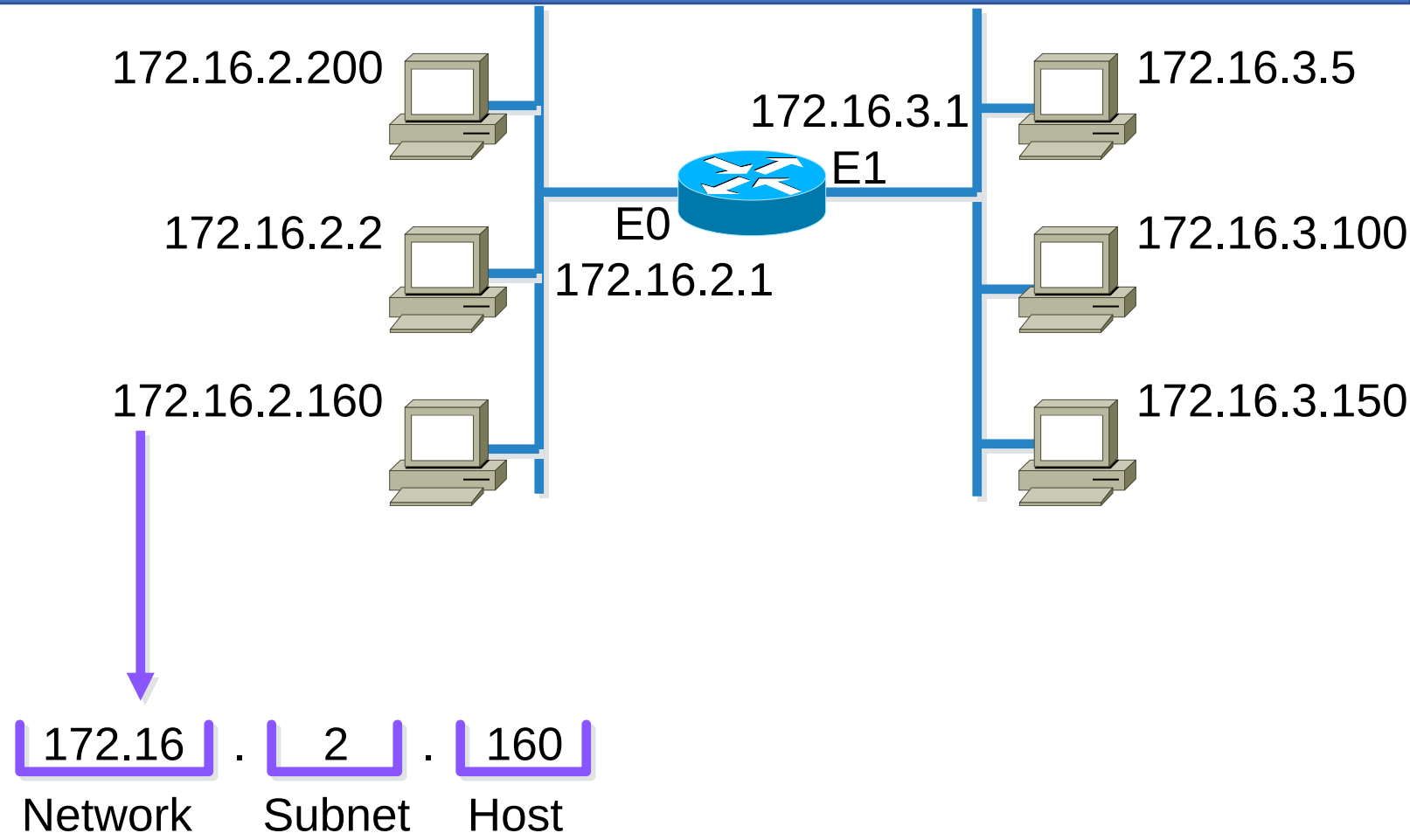
- The host bits of an IP address can be subdivided into a subnetwork section and a host section.
- The subnets in this example is identified with the third octet.
- The last octet represents the host



Network 172.16.0.0

Introduction to networking: Subnet Mask

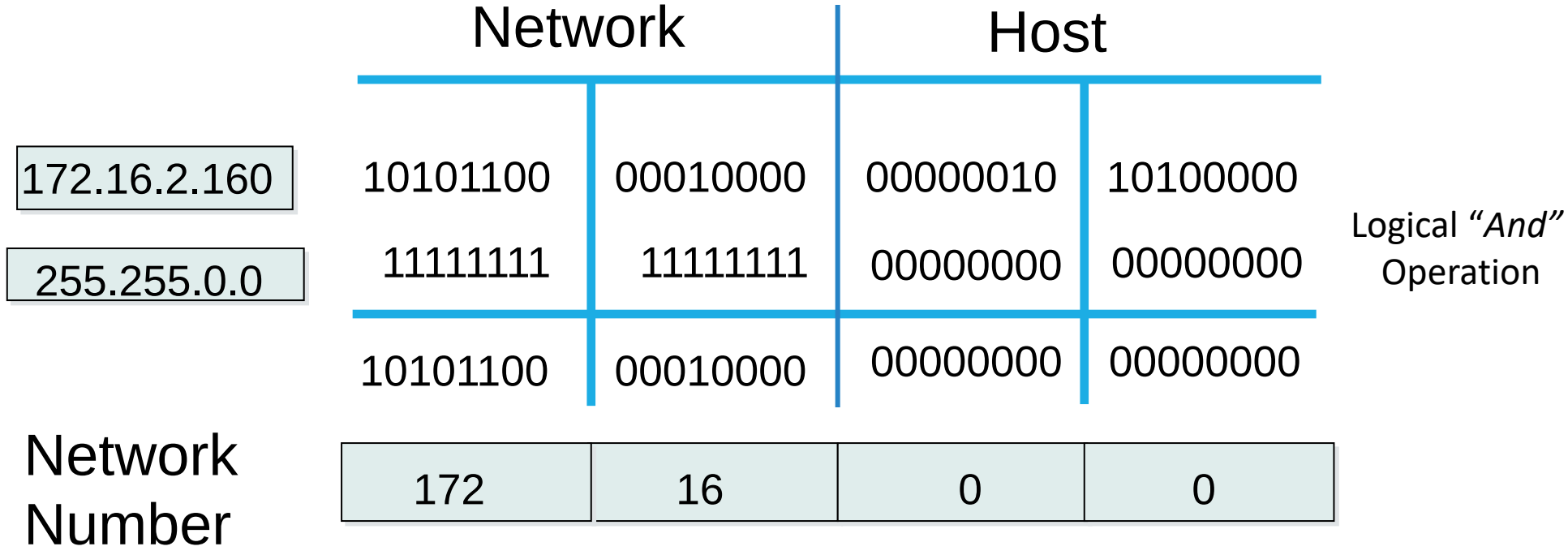
- We use part of the host bits (8 bits) to identify the subnet.
- Example: 172.16.2.16 (class B)
  - 172.16 -> Identify the network
  - 2.16 -> Identify the subnet and host
- The network mask is a combination of bits that serves to indicate to devices which part of the IP address is the network number, including the subnet, and which part corresponds to the host.



Introduction to networking: Subnet Mask

- The network mask consists of a series of contiguous binary 1s followed by a series of contiguous binary 0s. The 1s represent the network portion of the IP address, and the 0s represent the host portion

Mask without Subnets



Introduction to networking: Subnet Mask

- To specify a subnet, it is necessary to use part of the host bits.
- Example: 172.16.2.160
  - 172.16 → Network
  - 2.160 → host
- We extend the mask using some bits of the host to identify the subnet.

Mask with Subnet

	Network		Subnet	Host
172.16.2.160	10101100	00010000	00000010	10100000
255.255.255.0	11111111	11111111	11111111	00000000
	10101100	00010000	00000010	00000000

Network Number	172	16	2	0

Network number extended by eight bits

Introduction to networking: Subnet Mask

Mask with Subnet

128	64	32	16	8	4	2	1
1	1	0	0	0	0	0	0

	Network		Subnet	Host
172.16.2.160	10101100	00010000	00000010	10100000
255.255.255.192	11111111	11111111	11111111	11000000
	10101100	00010000	00000010	10000000

128 192 224 240 248 252 254 255 128 192 224 240 248 252 254 255

Network Number

172	16	2	128

IP Range of this subnet
172.16.2.129 - 172.16.2.191

Network number extended by ten bits

Introduction to networking: Subnet Mask

https://www.calculadoras.uno/binario-decimal/

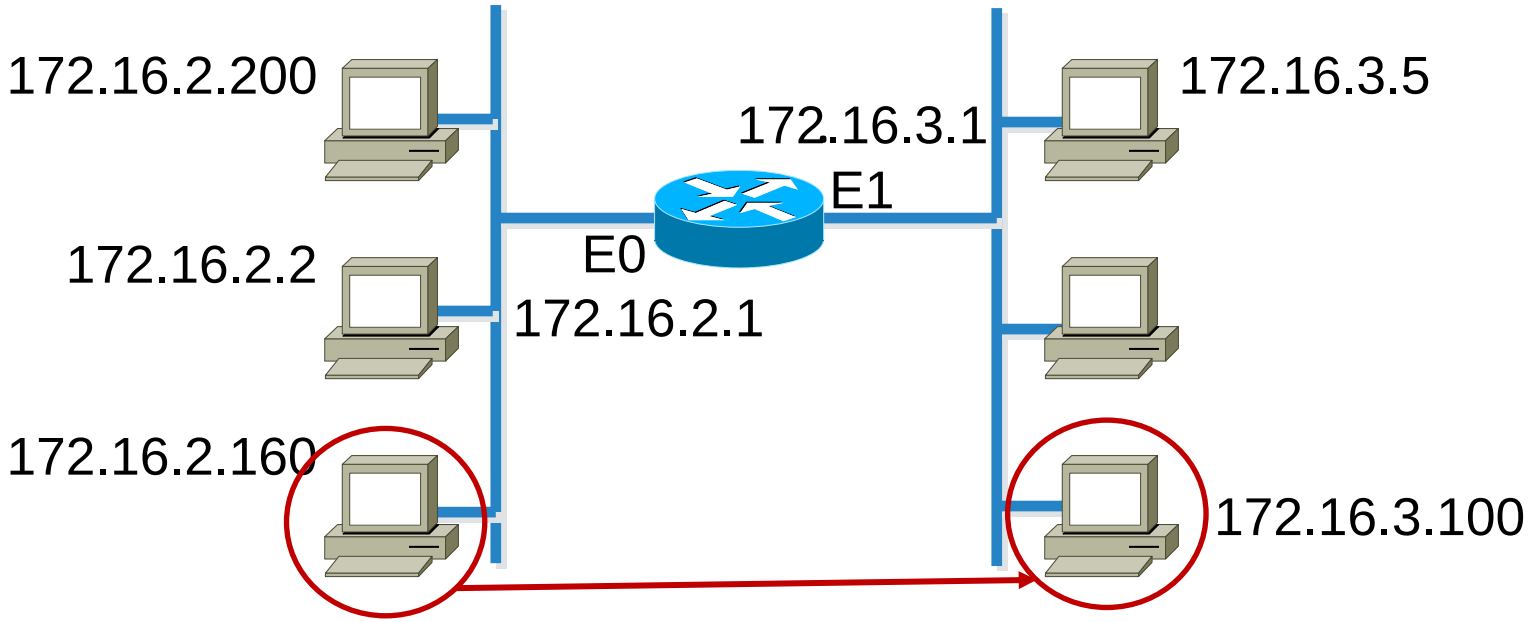
- **Extract network**: We apply an “and” logical operation between the IP and the Mask

	181	221	84	114
IP Address:	10110101	11011101	01010100	01110010
Subnet Mask: &	11111111	11111111	11000000	00000000
Result:	10110101	11011101	01000000	00000000
	181	221	64	0

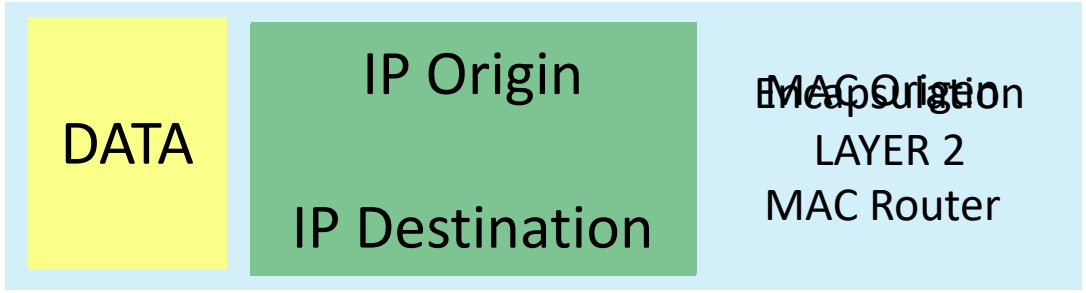
Introduction to networking: Open Systems Interconection (OSI) model

Layer	Number	Function	Protocol/Address
Application	7	Means for an application to access a computer network	HTTP(S), FTP, DHCP, LDAP
Presentation	6	• Ensures that the application layer can read the data • Encryption	ASCI, ICA
Session	5	Enables orderly exchange of data	NetBIOS, RPC
Transport	4	Provides protocols to support host-to-host communication	TCP, UDP
Network	3	Routing and packet forwarding (routers)	IP
Data link	2	Transfer data in the same LAN network (hubs and switches)	MAC
Physical	1	Transmission and reception of raw bitstreams over a physical medium	Signals (1s and 0s)

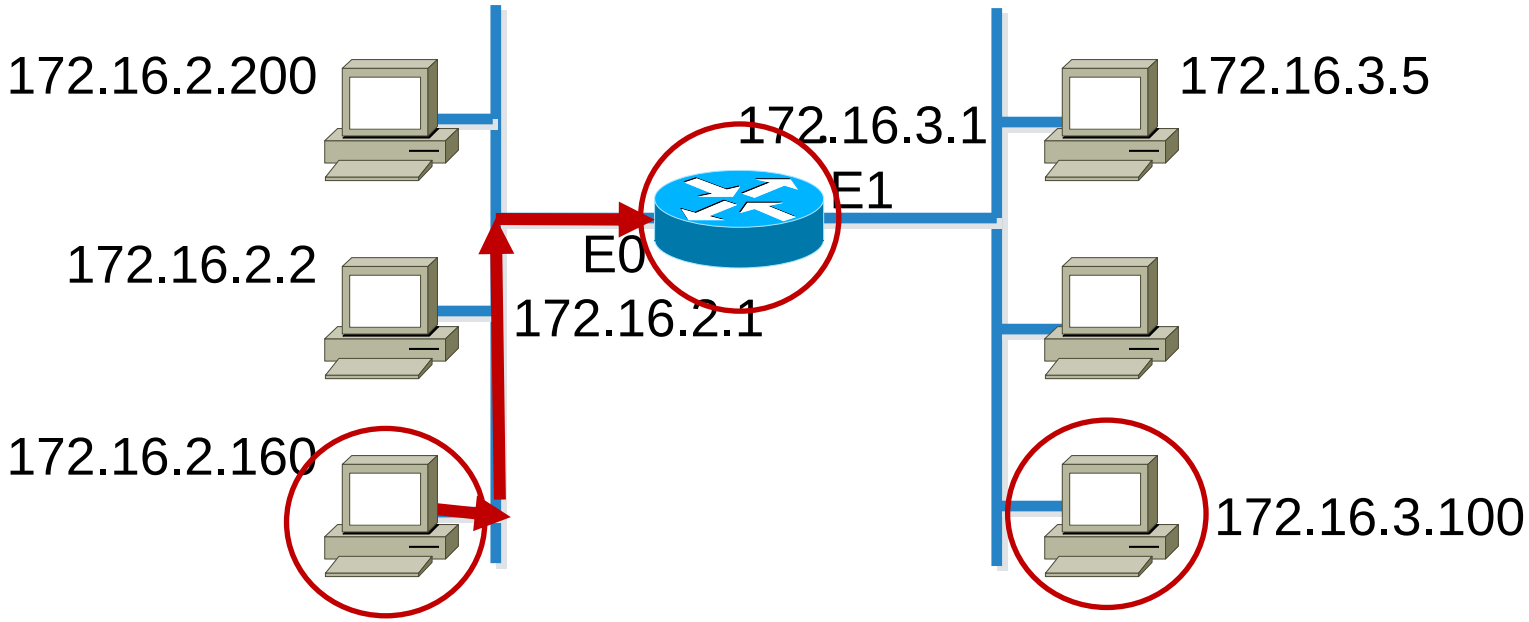
Introduction to networking: Open Systems Interconection (OSI) model



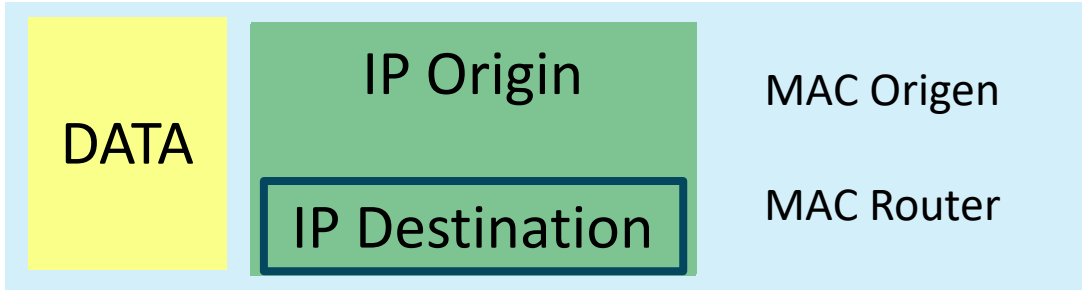
Encapsulation



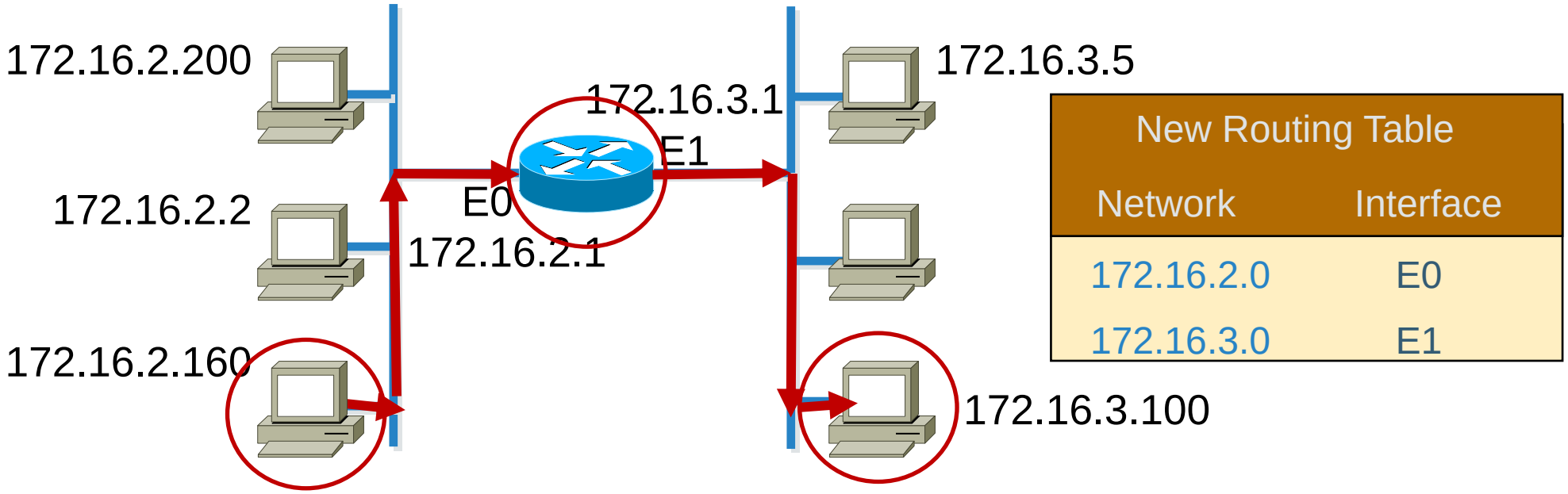
Introduction to networking: Open Systems Interconection (OSI) model



Encapsulation

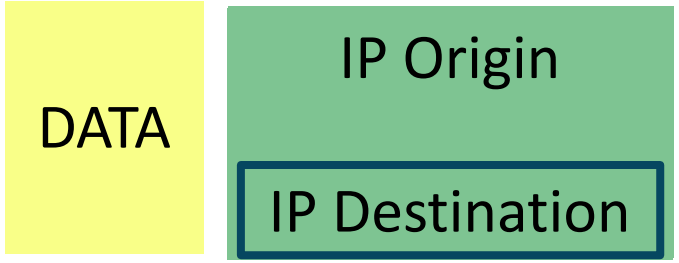


Introduction to networking: Open Systems Interconection (OSI) model

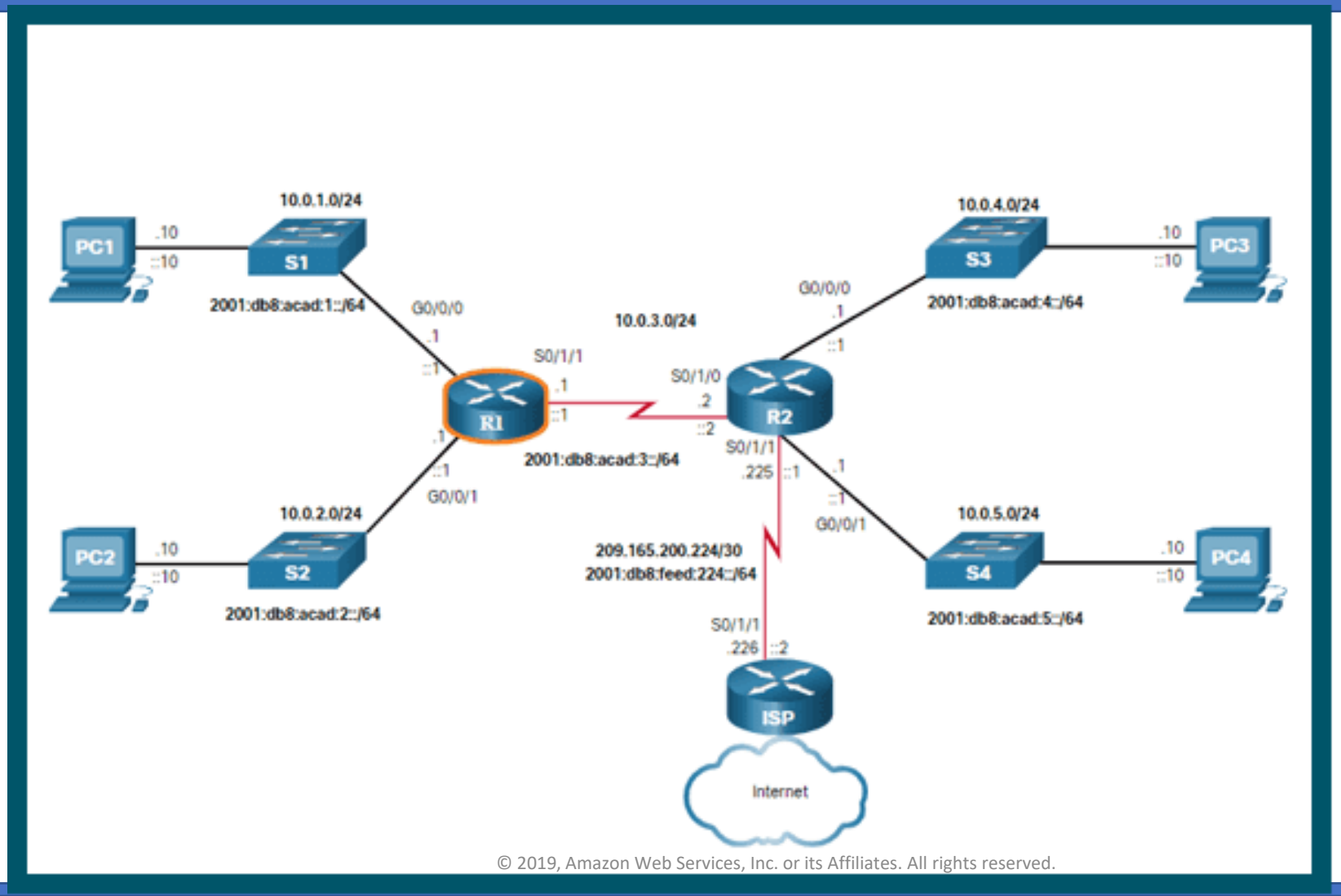


New Routing Table	
Network	Interface
172.16.2.0	E0
172.16.3.0	E1

Encapsulation



Introduction to networking: Open Systems Interconection (OSI) model



AWS Global infrastructure

- The **AWS Global Infrastructure** is designed and built to deliver a **flexible**, **reliable**, **scalable**, and **secure** cloud computing environment with high-quality **global network performance**.

- This map from https://infrastructure.aws shows the current **AWS Regions** and more that are coming soon.

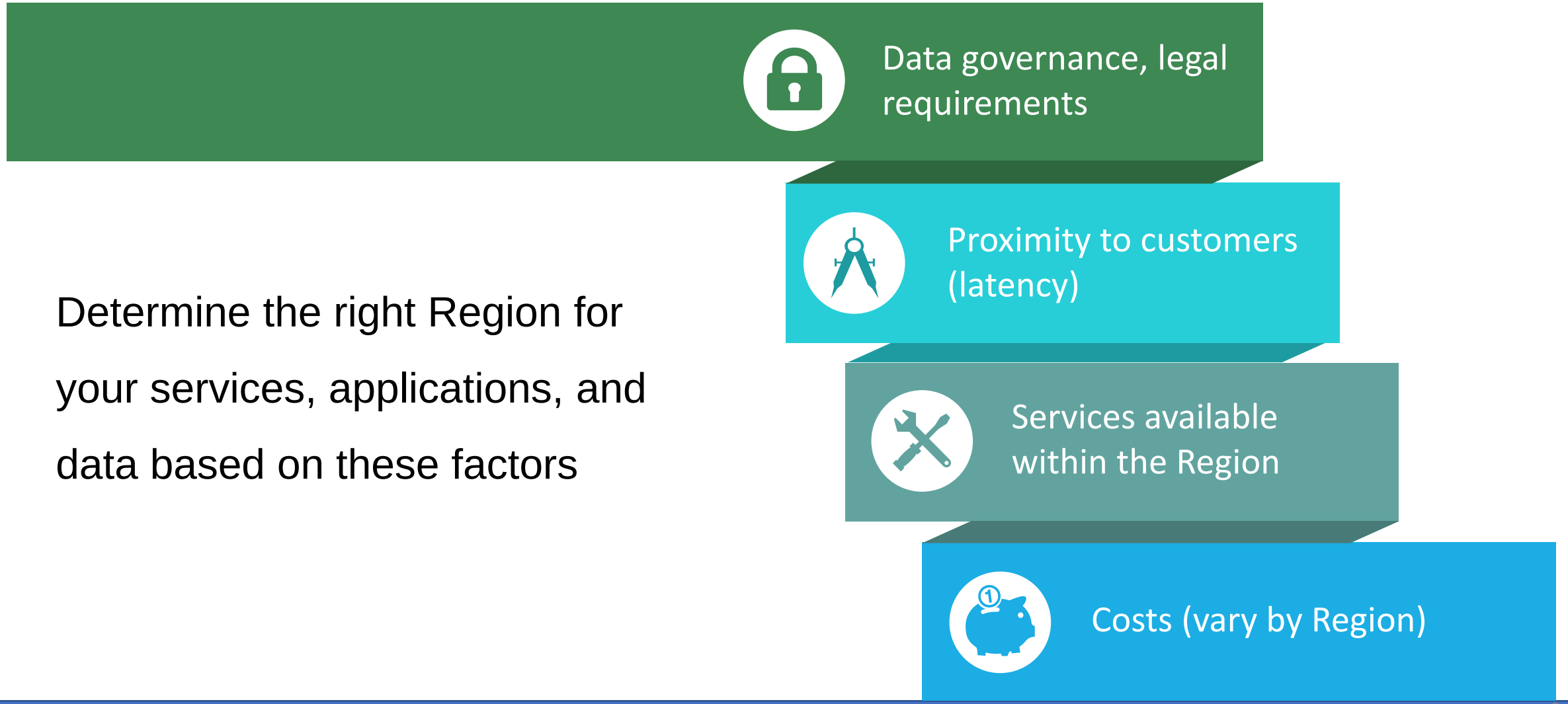
AWS Global infrastructure

Regions

- AWS regions are geographical áreas.
- Each Region provides full redundancy and connectivity to the network.
- A Region typically consists of two or more Availability Zones.
- Data replication across Regions is controled by the user.
- Communication between Regions uses AWS backbone network infrastructure.

AWS Global infrastructure

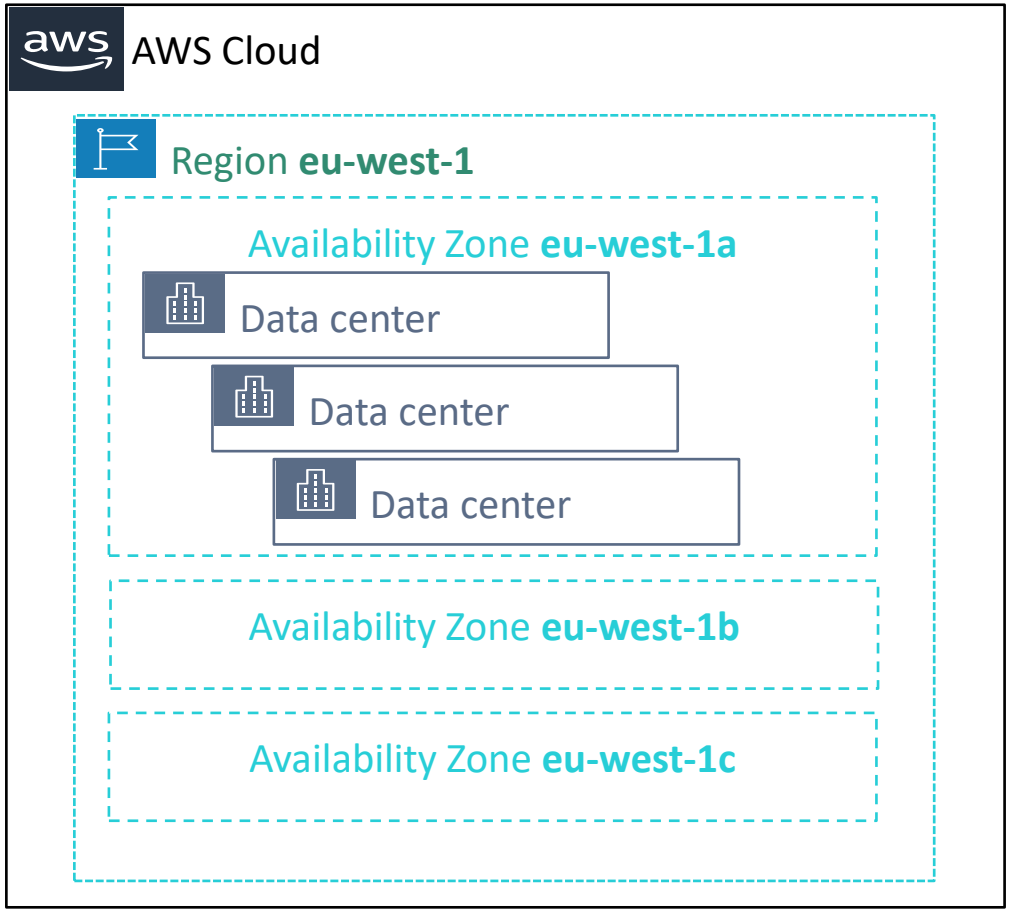
Determine the right Region for your services, applications, and data based on these factors



AWS Global infrastructure

Availability Zones

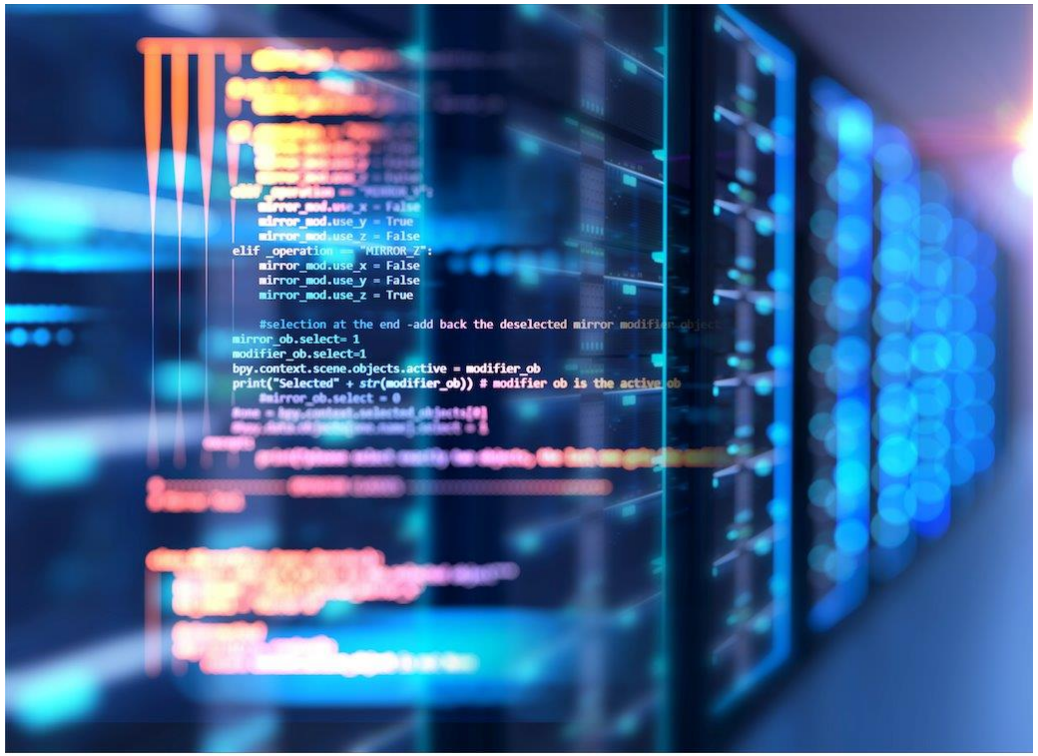
- Each Region has multiple Availability Zones.
- Each **Availability Zone** is a fully isolated partition of the AWS infrastructure.
  - Availability Zones consist of discrete **data centers**
  - They are designed for fault isolation
  - They are interconnected with other Availability Zones by using high-speed private networking
  - You choose your Availability Zones.
  - AWS recommends replicating data and resources across Availability Zones for resiliency.



AWS Global infrastructure

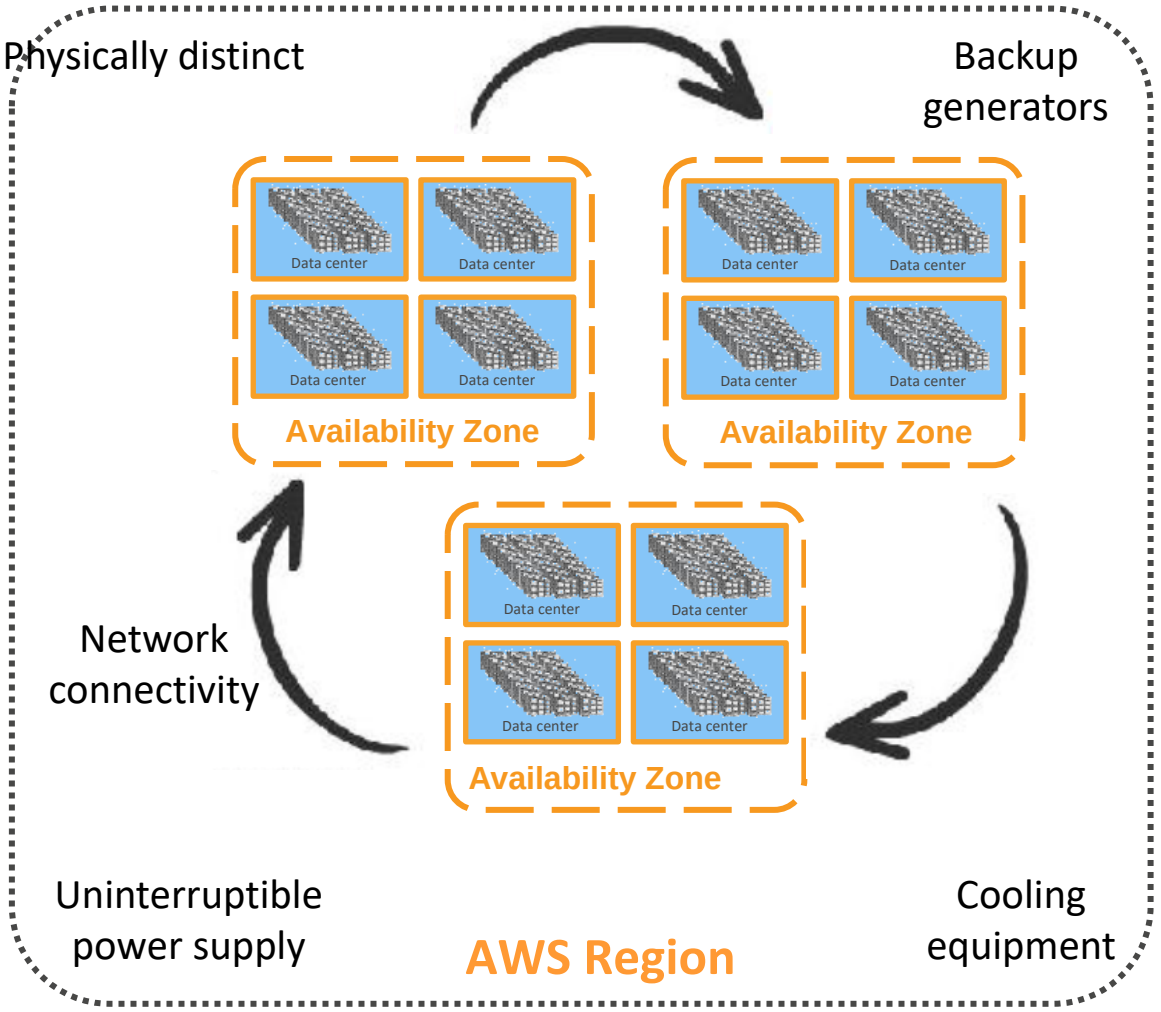
AWS data centers

- AWS data centers are designed for security.
- Data centers are where the data resides, and data processing occurs.
- Each data center has redundant power, networking, and connectivity, and is housed in a separate facility.
- A data center typically has 50,000 to 80,000 physical servers.



AWS Global infrastructure

- Elasticity and scalability
  - Elastic infrastructure; dynamic adaption of capacity
  - Scalable infrastructure; adapts to accommodate growth
- Fault-tolerance
  - Continues operating properly in the presence of a failure
  - Built-in redundancy of components
- High availability
  - High level of operational performance
  - Minimized downtime
  - No human intervention



Virtual Private Clouds (VPC)

- VPC: is a virtual network where the user will place your resources like VMs, Databases, etc.

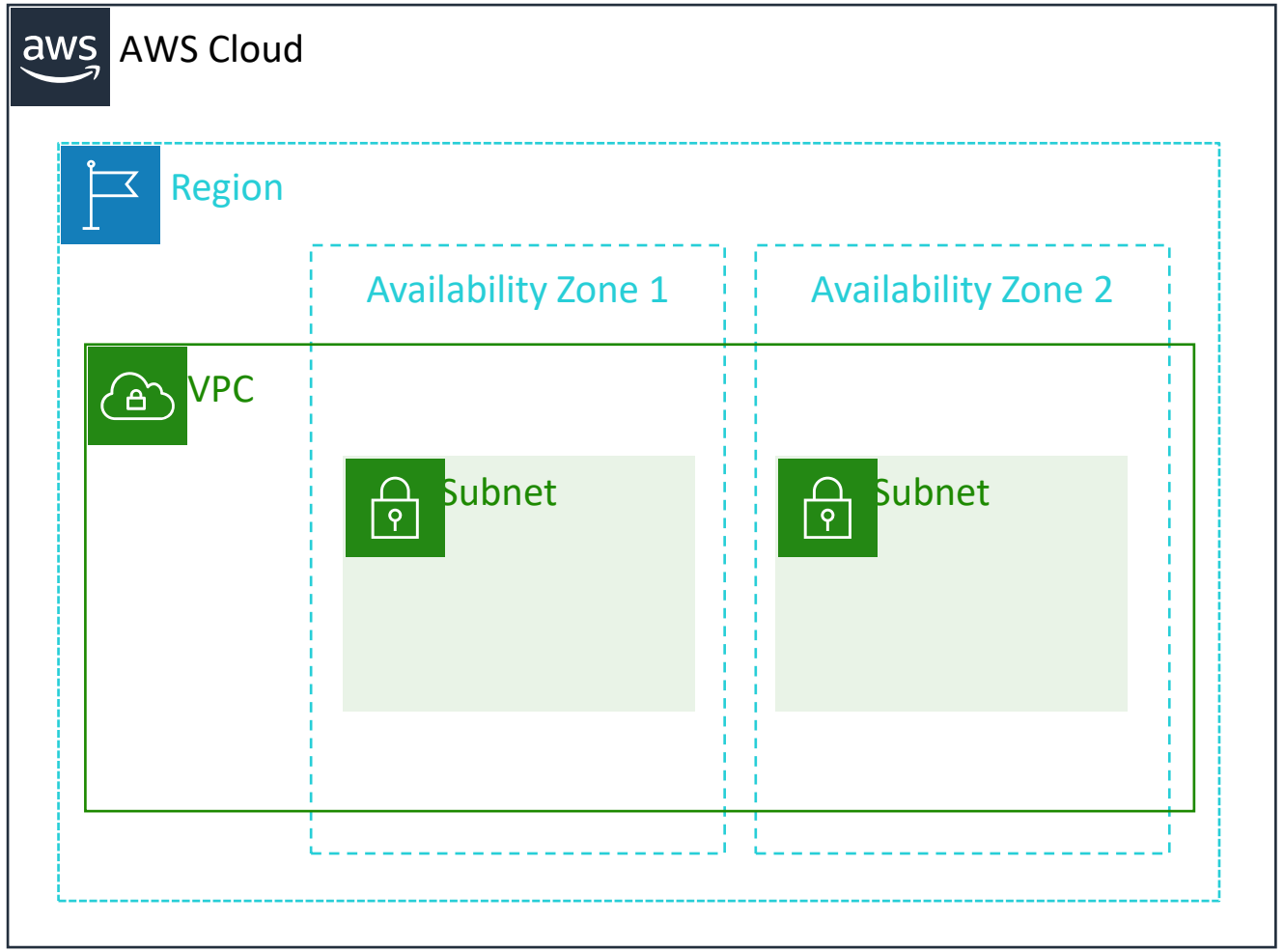
- Enables you to provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define.

- Gives you control over your virtual networking resources, including:
  - Selection of IP address range
  - Creation of subnets
  - Configuration of route tables and network gateways

- Enables you to use multiple layers of security

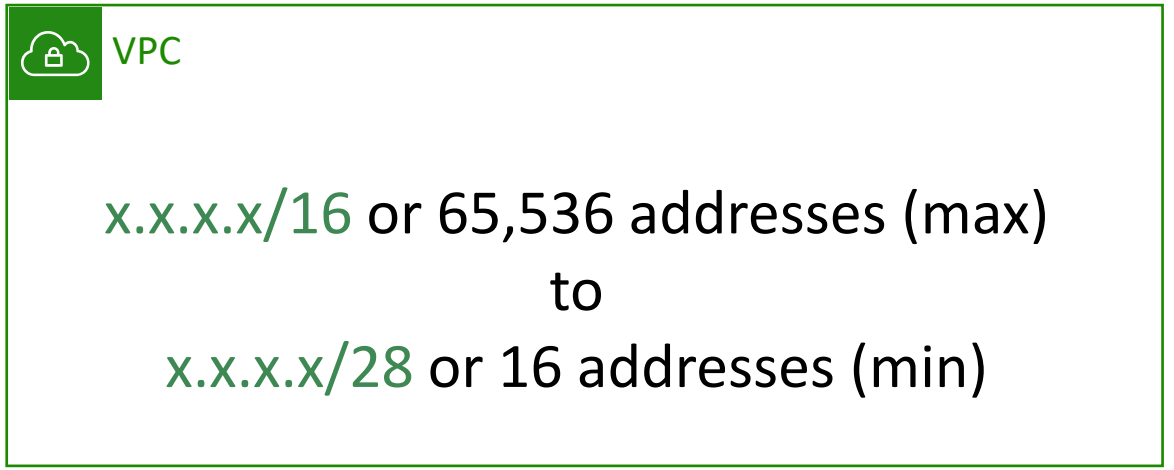
VPCs and subnets

- **VPCs**:
  - Logically isolated from other VPCs
  - Dedicated to your AWS account
  - Belong to a single AWS Region and can span multiple Availability Zones
- **Subnets**:
  - Range of IP addresses that divide a VPC
  - Belong to a single Availability Zone
  - Classified as public or private



VPCs and subnets

- When you create a VPC, you assign it to an IPv4 CIDR block (range of private IP addresses).
- You cannot change the address range after you create the VPC.
- The largest IP CIDR block size is /16.
- The smallest IP CIDR block size is /28.
- IPv6 is also supported (with a different block size limit). (IpV6 is out of scope of this course)
- CIDR blocks of subnets cannot overlap.



VPCs and subnets: Example

- **Example**: A VPC with an IPv4 CIDR block of 10.0.0.0/16 has 65.536 total IP addresses.
- For our application, we need four equal-sized subnets:

VPC: 10.0.0.0/16 -> 65.536 addresses

Subnet 1: 10.0.1.0/24 -> 256 addresses

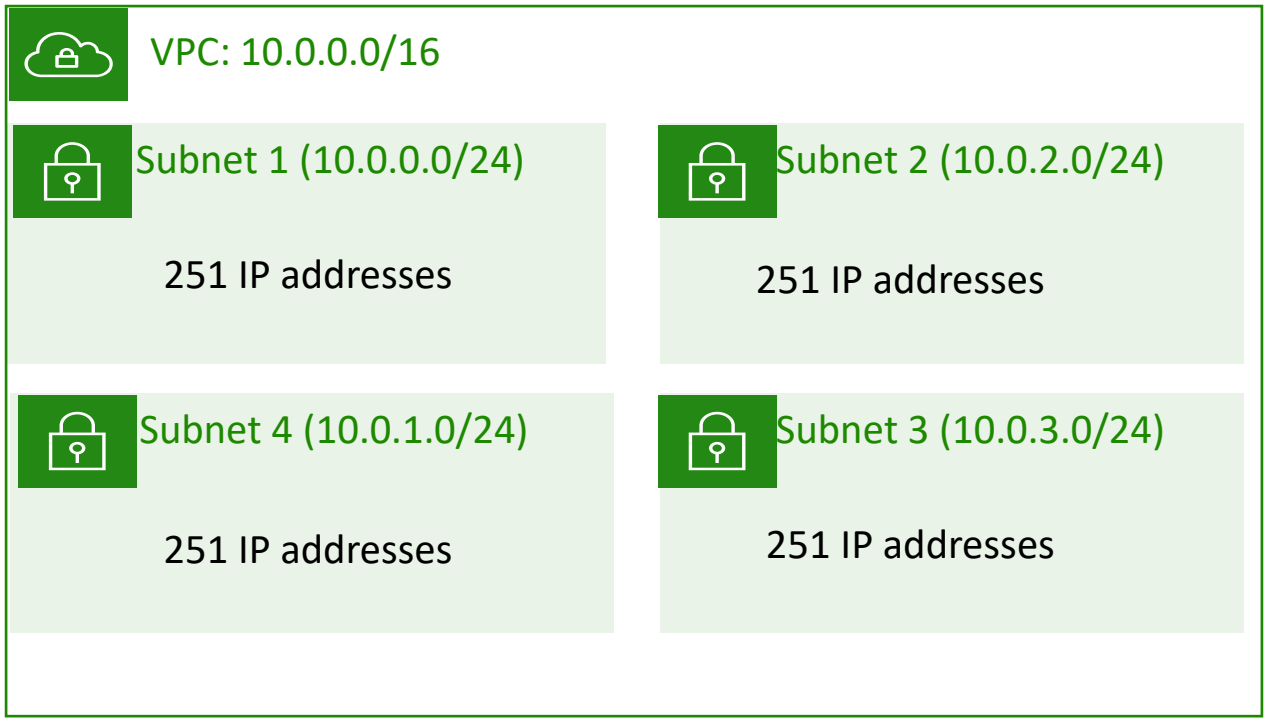
Subnet 2: 10.0.2.0/24 -> 256 addresses

Subnet 3: 10.0.3.0/24 -> 256 addresses

Subnet 4: 10.0.4.0/24 -> 256 addresses

VPCs and subnets: Example

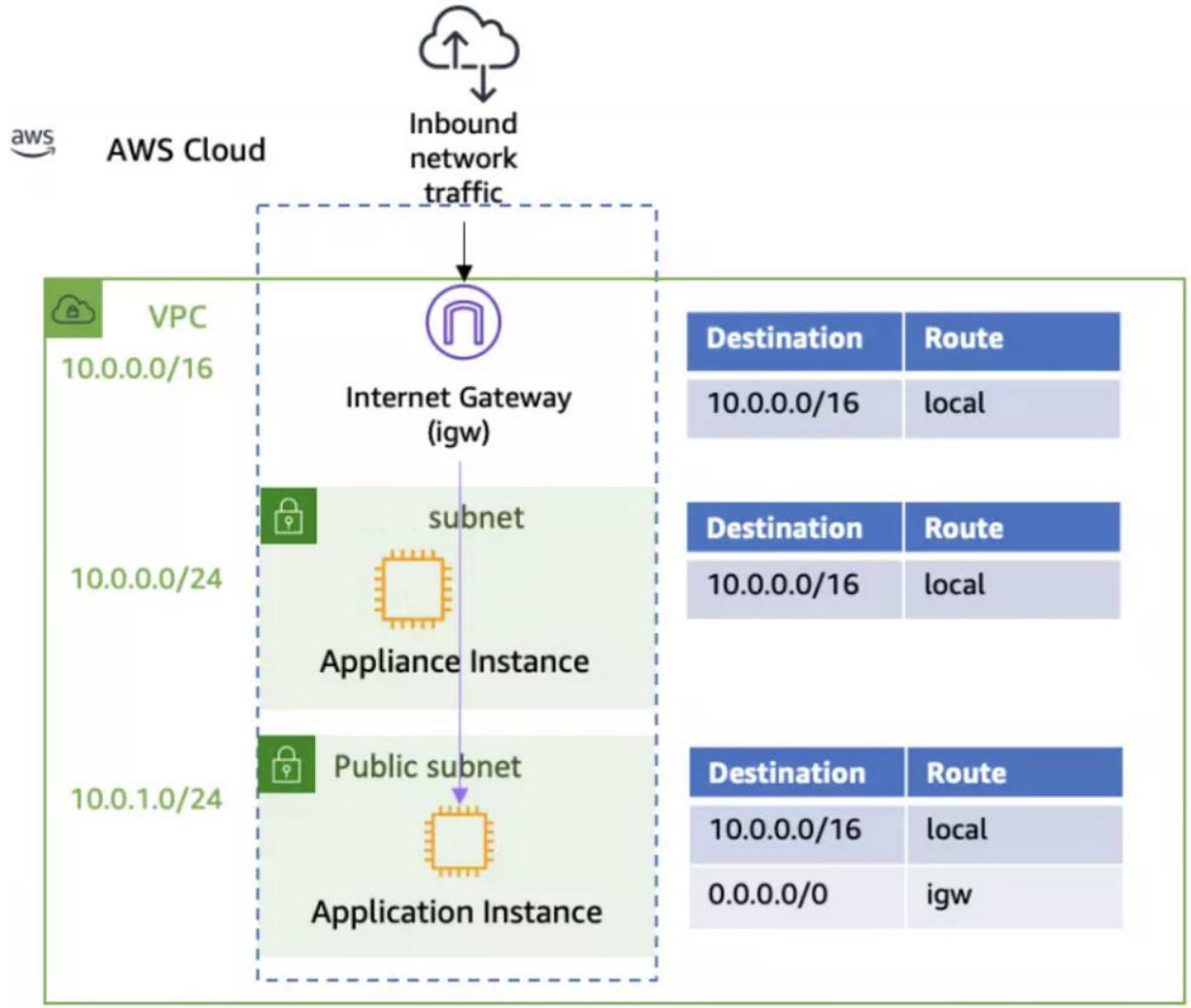
Example: A VPC with an IPv4 CIDR block of 10.0.0.0/16 has 65,536 total IP addresses. The VPC has four equal-sized subnets. Only 251 IP addresses are available for use by each subnet.



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

Route Tables and routes

- Route tables are required to direct traffic between VPC resources.
- A *route table* contains a set of rules, called *routes*, that determine where network traffic from your subnet or gateway is directed.
- Each subnet in your VPC must be associated with a route table.
- A subnet can be associated with only one route table at a time
- By default, every route table contains a local route for communication within the VPC.



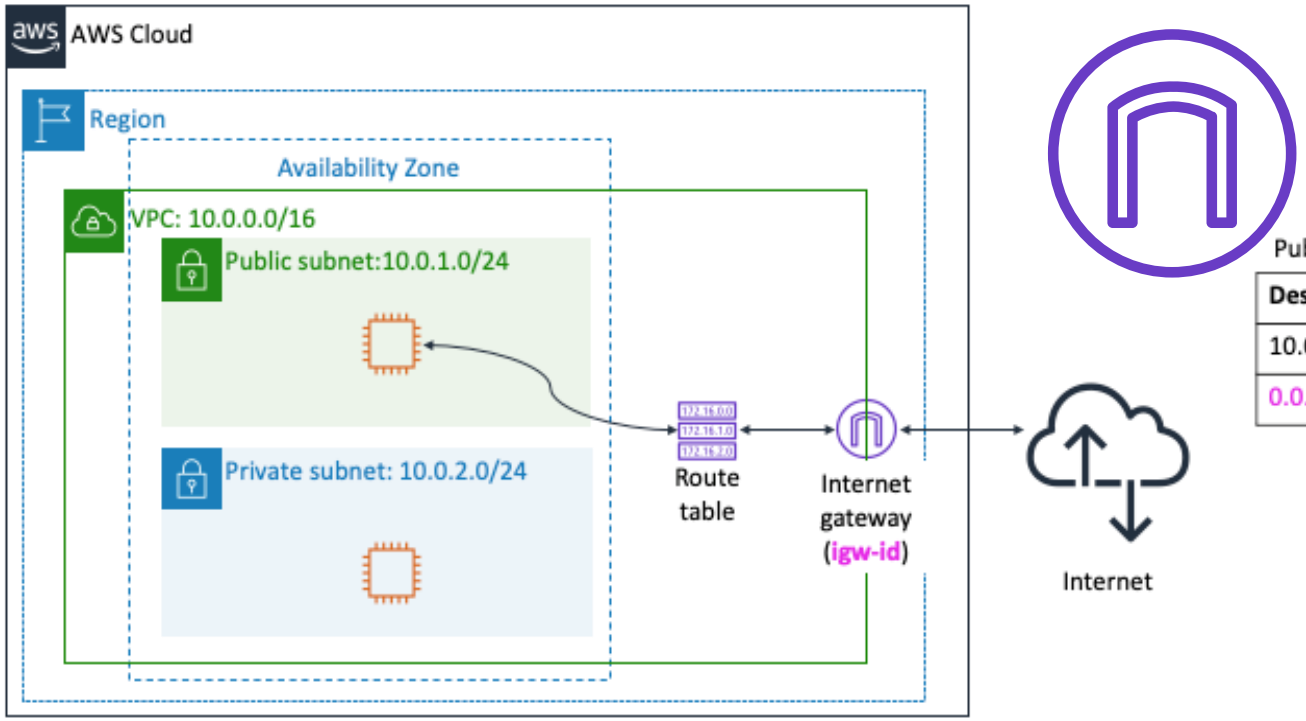
Route tables an routes

- 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.

Internet gateway

- An internet gateway is a scalable, redundant, and highly available VPC component that allows communication between instances in your VPC and internet.
- It is seen as a VPC level component.
- An internet gateway serves two purposes:
  - Provide a target in your VPC route tables for internet-routable traffic.
  - Perform network address translation for instances that were assigned public IPv4 addresses



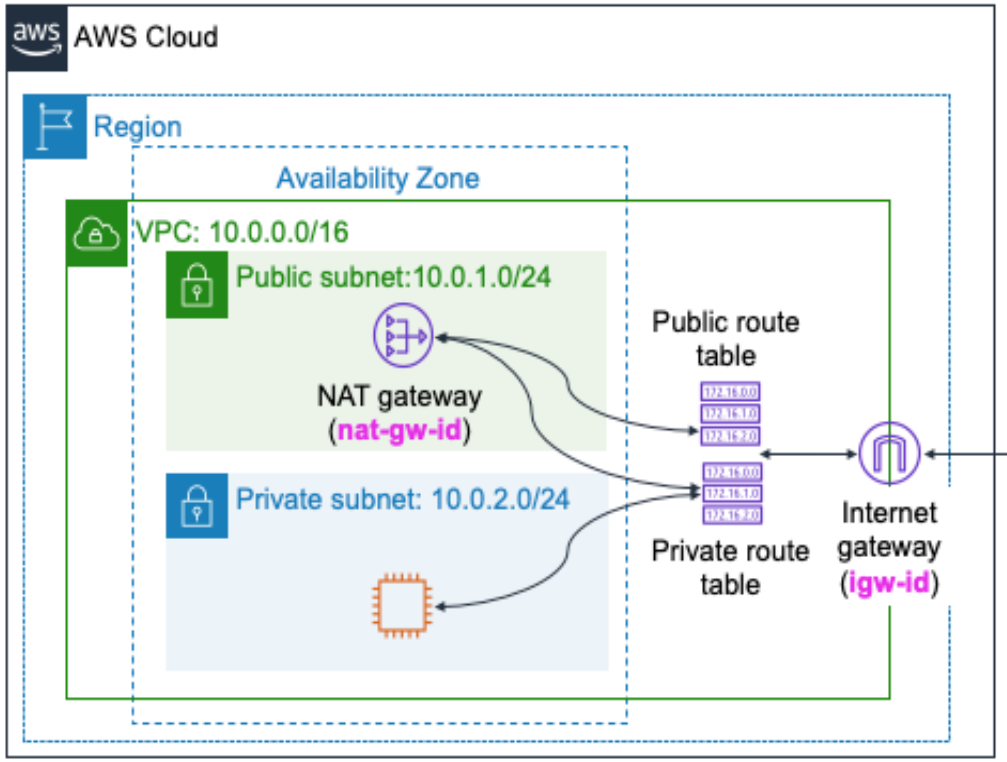
Public Subnet Route Table

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

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).

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.
- It prevents the internet from initiating a connection with those instances.
- It is a subnet level component



Public Subnet Route Table

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

Private Subnet Route Table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	nat-gw-id

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.

Network Address Translation (NAT) gateway

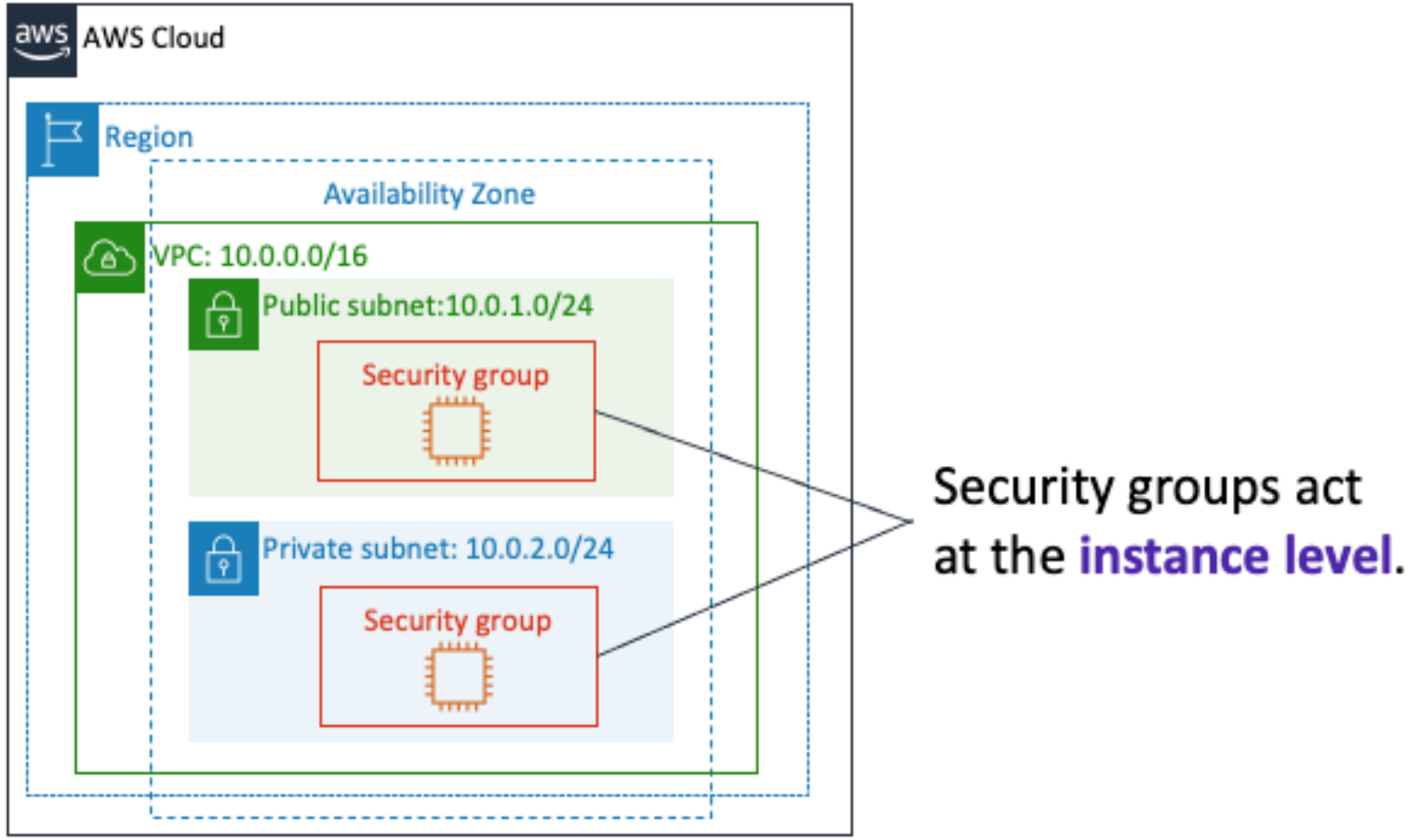
Elastic IP

- When you stop and then start an EC2 instance, it changes its public IP.
- If you need to have a fixed public IP, you need ab Elastic IP.
- An Elastic IP is a public IPv4 you own as long as you do not delete it.
- You can attach it to once instance at time.
- You can remap across instances.

Amazon Elastic Compute Cloud (EC2) provides on-demand, scalable computing capacity in the Amazon Web Services (AWS) Cloud.

Security Groups

- A security group acts as a virtual firewall for your instance.
- 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.



Security Groups

Inbound				
Type	Protocol	Port Range	Source	Description
All traffic	All	All	sg-xxxxxxxx	
Outbound				
Type	Protocol	Port Range	Source	Description
All traffic	All	All	sg-xxxxxxxx	

- Security groups have rules that control inbound and outbound instance traffic.
- Default security groups deny all inbound traffic and allow all outbound traffic.
- Security groups are stateful, which means that state information is kept even after a request is processed.

If you send a request from your instance, the response traffic for that request is allowed to flow in regardless of inbound security group rules. Responses to allowed inbound traffic are allowed to flow out, regardless of outbound rules.

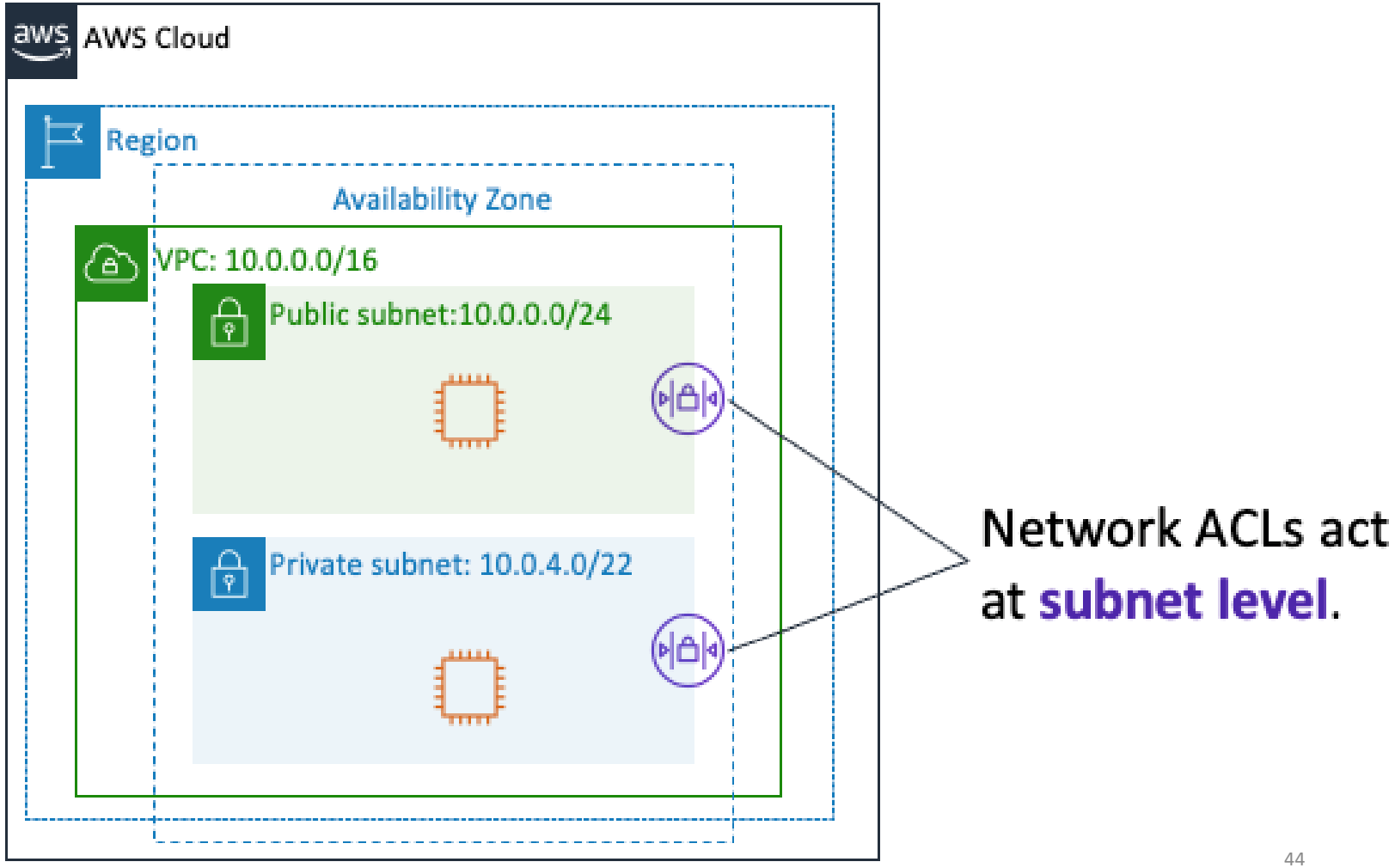
Security Groups

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	

- You can specify allow rules, but not deny rules.
- All rules are evaluated before the decision to allow traffic.

Network Access Control lists (network ACLs)

- A *Network Access Control List (ACL)* is an optional layer of security for your Amazon VPC. It acts as a firewall for controlling traffic in and out of one or more subnets.
- 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
- 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.



Network ACLs

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

- A network ACL has separate inbound and outbound rules, and each rule can either allow or deny traffic.
- Default network ACLs allow all inbound and outbound IPv4 traffic.
- Network ACLs are stateless , which means that no information about a request is maintained after a request is processed.

Security groups versus network ACLs

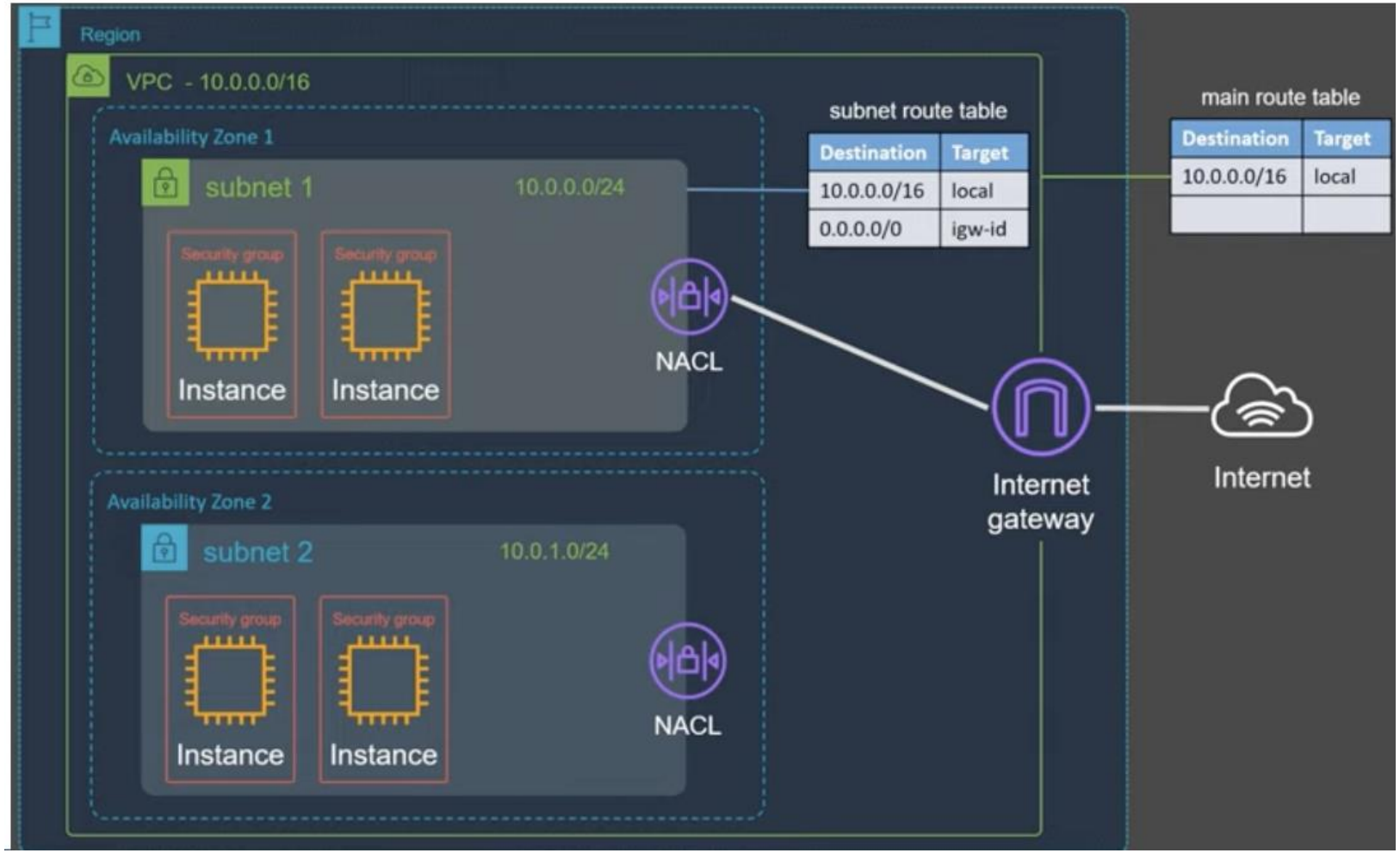
Attribute	Security Groups	Network ACLs
Scope	Instance level	Subnet level
Supported Rules	Allow rules only	Allow and deny rules
State	Stateful (return traffic is automatically allowed, regardless of rules)	Stateless (return traffic must be explicitly allowed by rules)
Order of Rules	All rules are evaluated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

Additional resources

- Amazon VPC overview page
- Amazon Virtual Private Cloud Connectivity Options whitepaper
- One to Many: Evolving VPC Design AWS Architecture blog post
- Amazon VPC User Guide
- Amazon CloudFront overview page

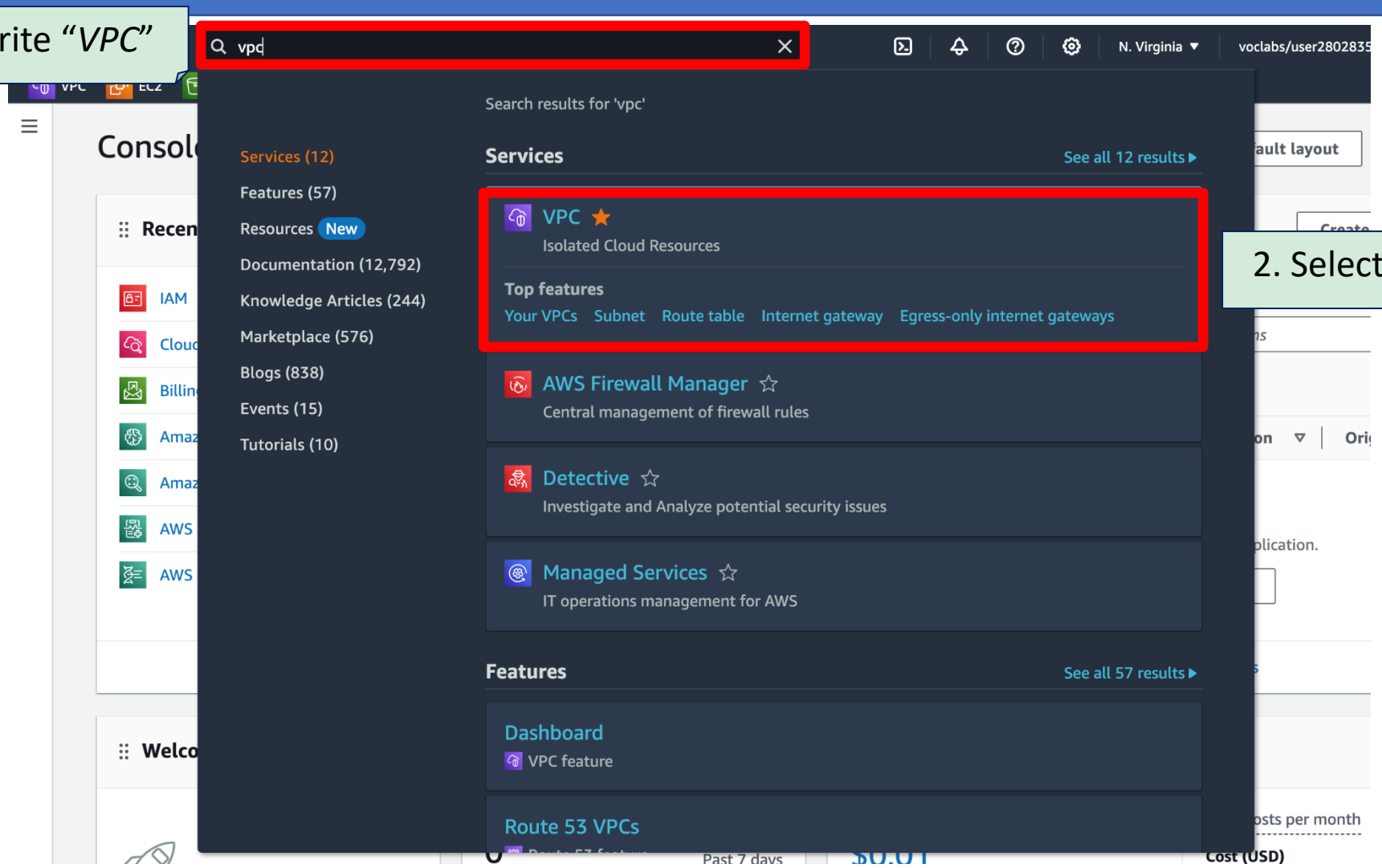
Short DEMO: Creating our first VPC

- **Objective**: Create a VPC with 2 subnets, one private and one public with internet access, where you can deploy computing services.
- The figure on this slide illustrates the final architecture of our VPC.
- Let’s go!!!!



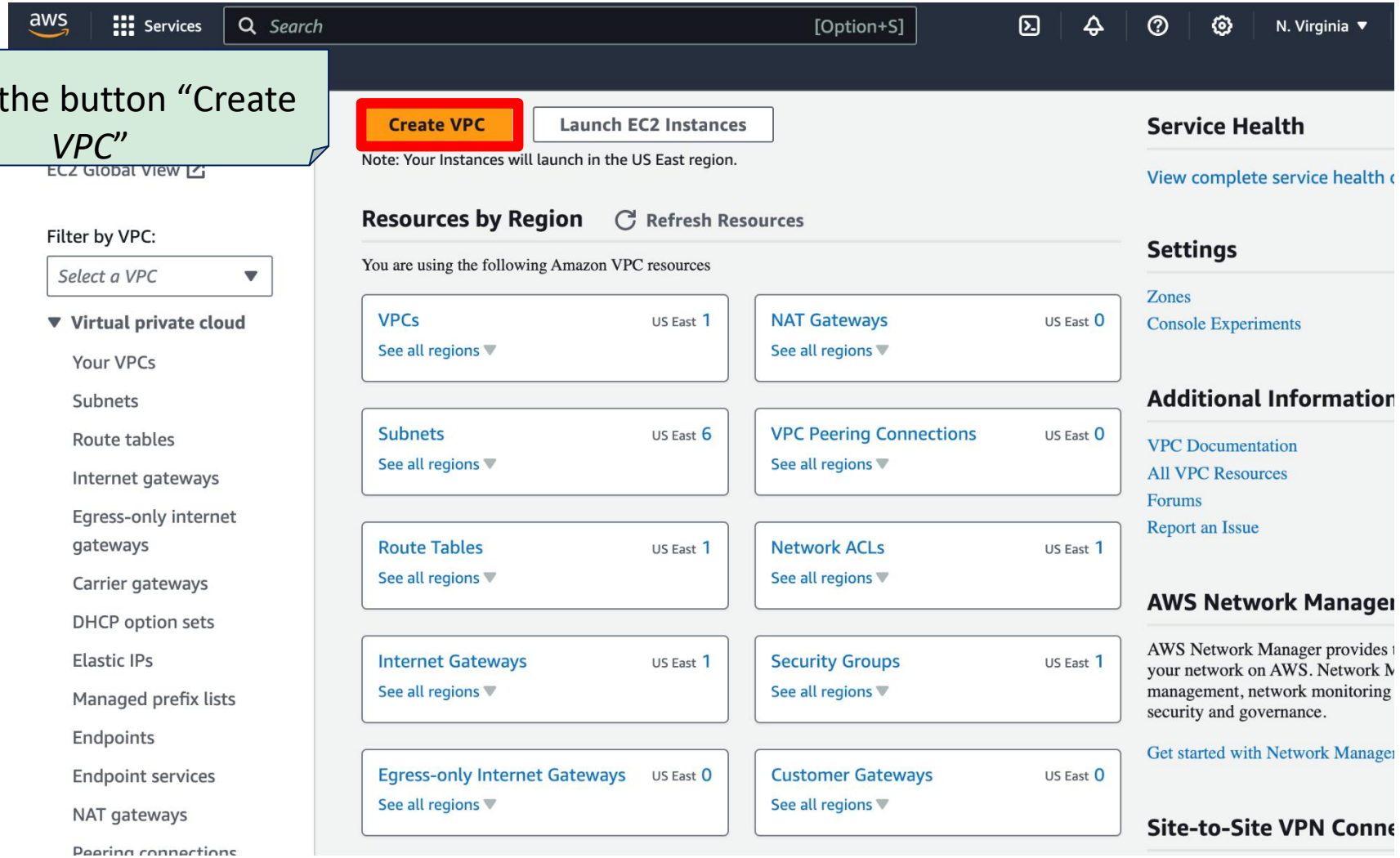
Short DEMO: Creating a VPC

- The first step is to open the main menu of AWS to create a VPC.
- In the search bar, type "VPC"



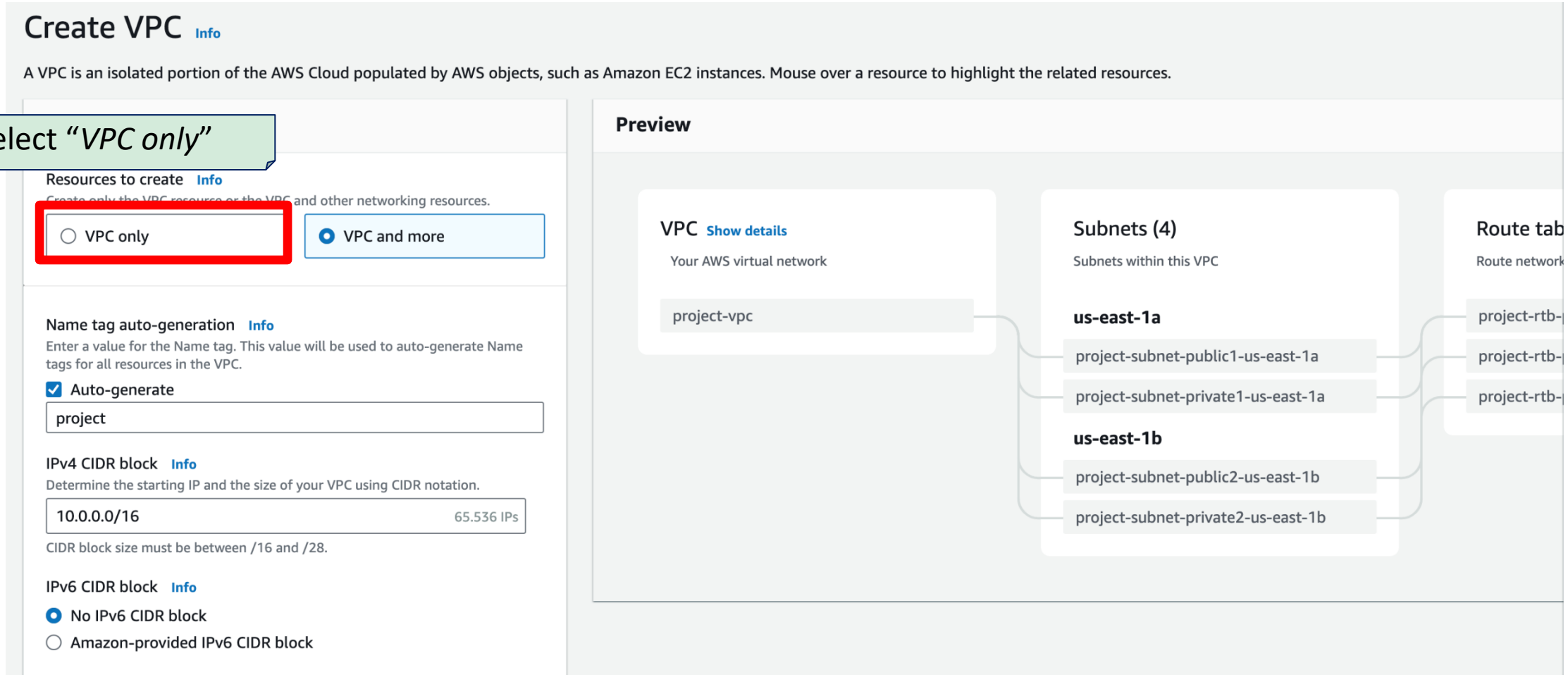
Short DEMO: Creating a VPC

Select the button “Create *VPC*”



Short DEMO: Creating a VPC

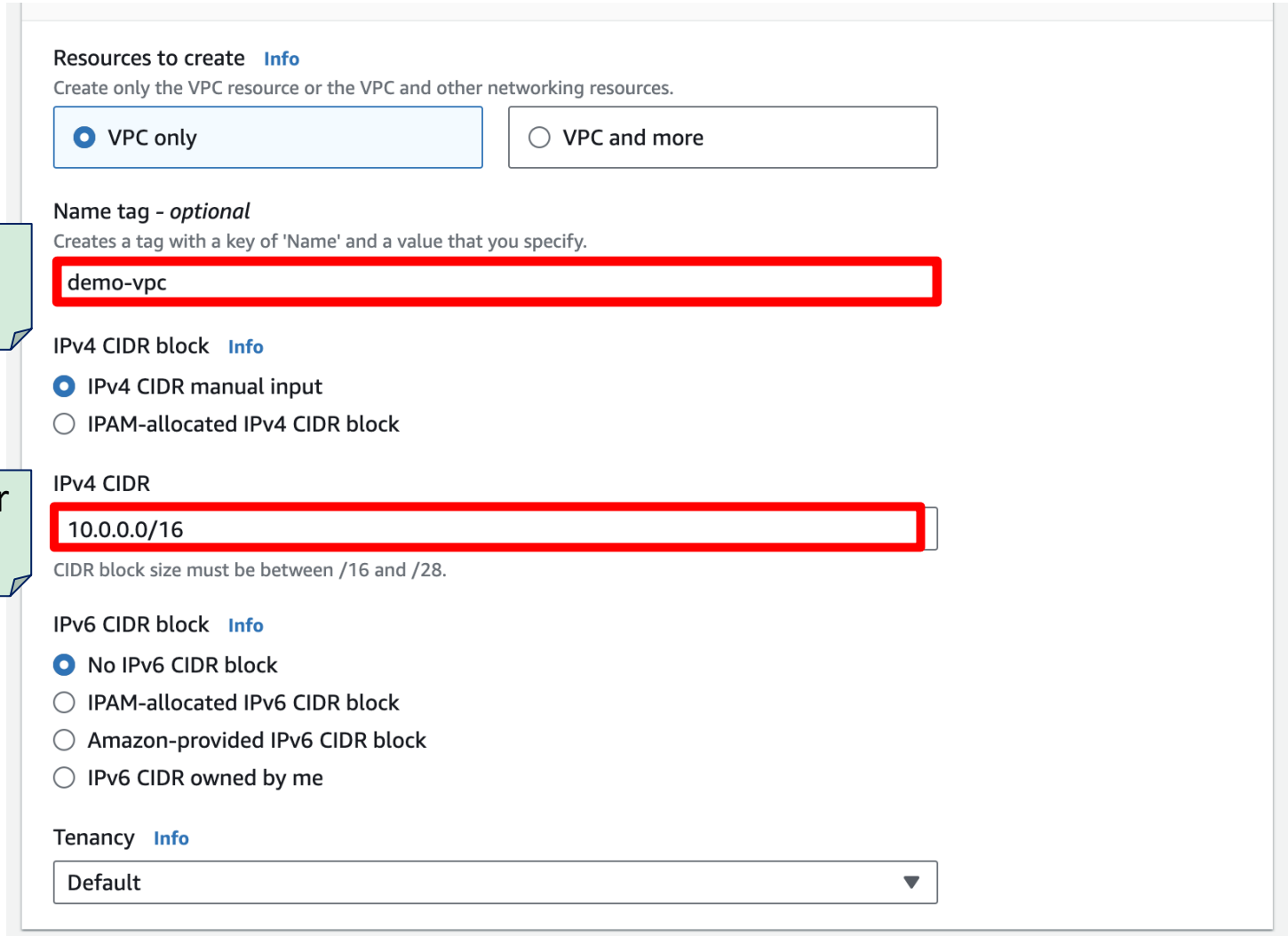
Select “*VPC only*”



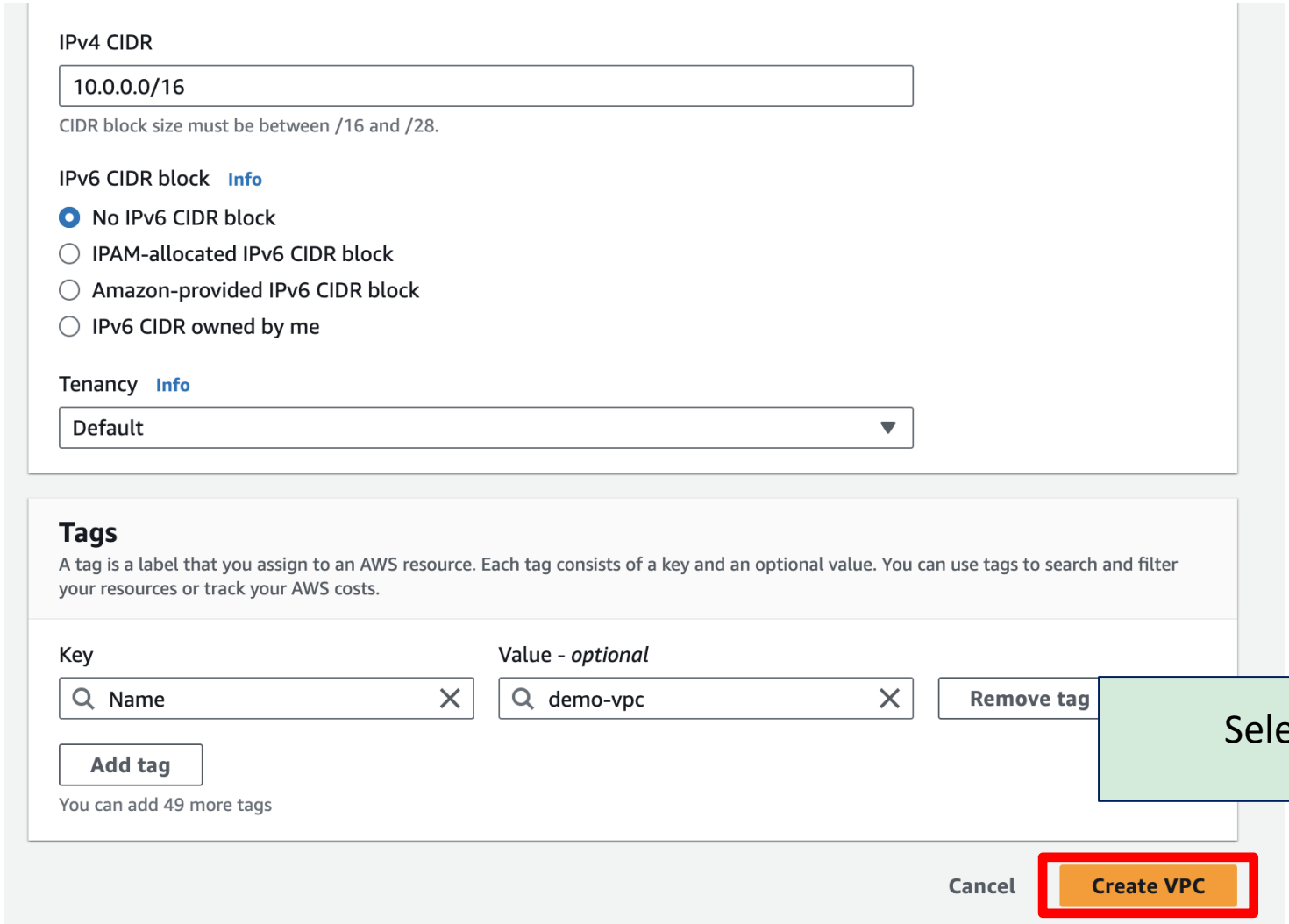
Short DEMO: Creating a VPC

1. Select "*Write the name of the VPC*"

2. Specify the CIDR block of our VPC

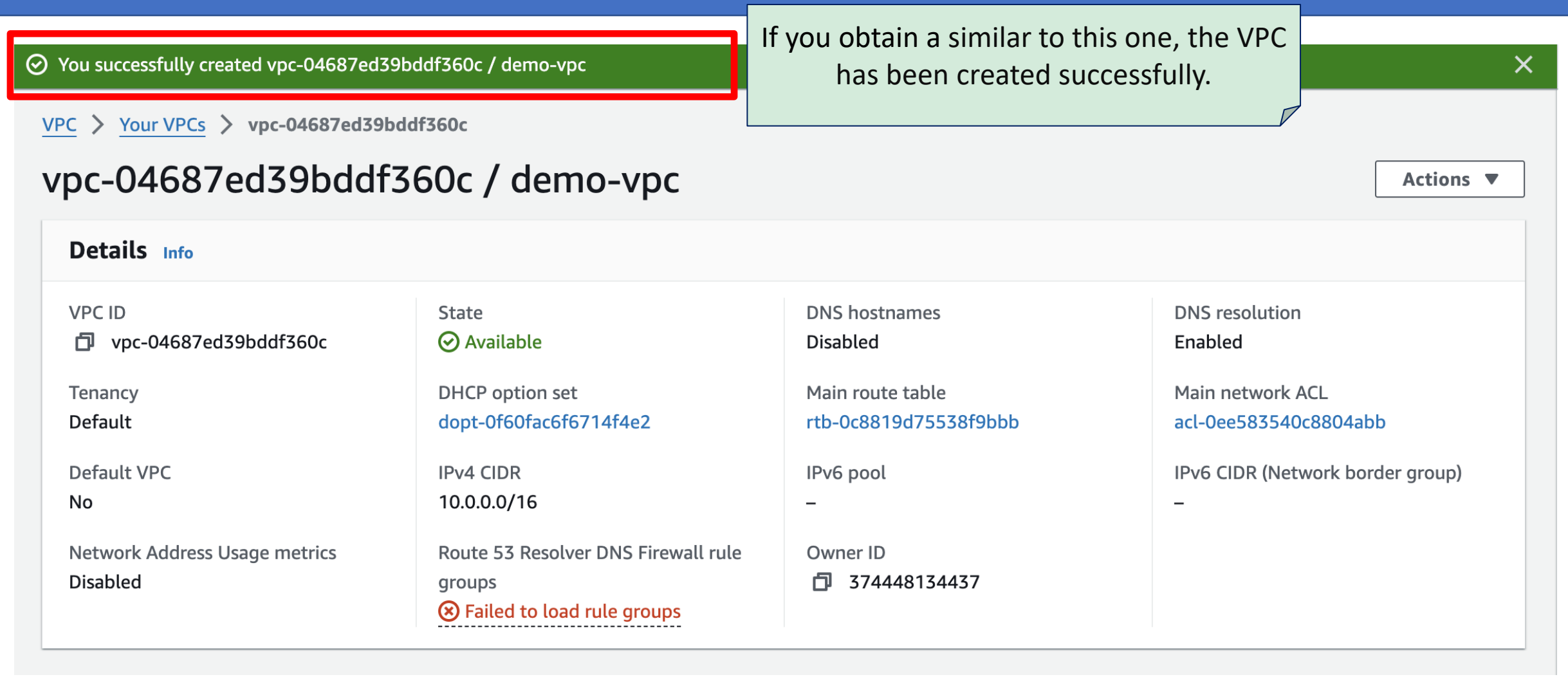


Short DEMO: Creating a VPC



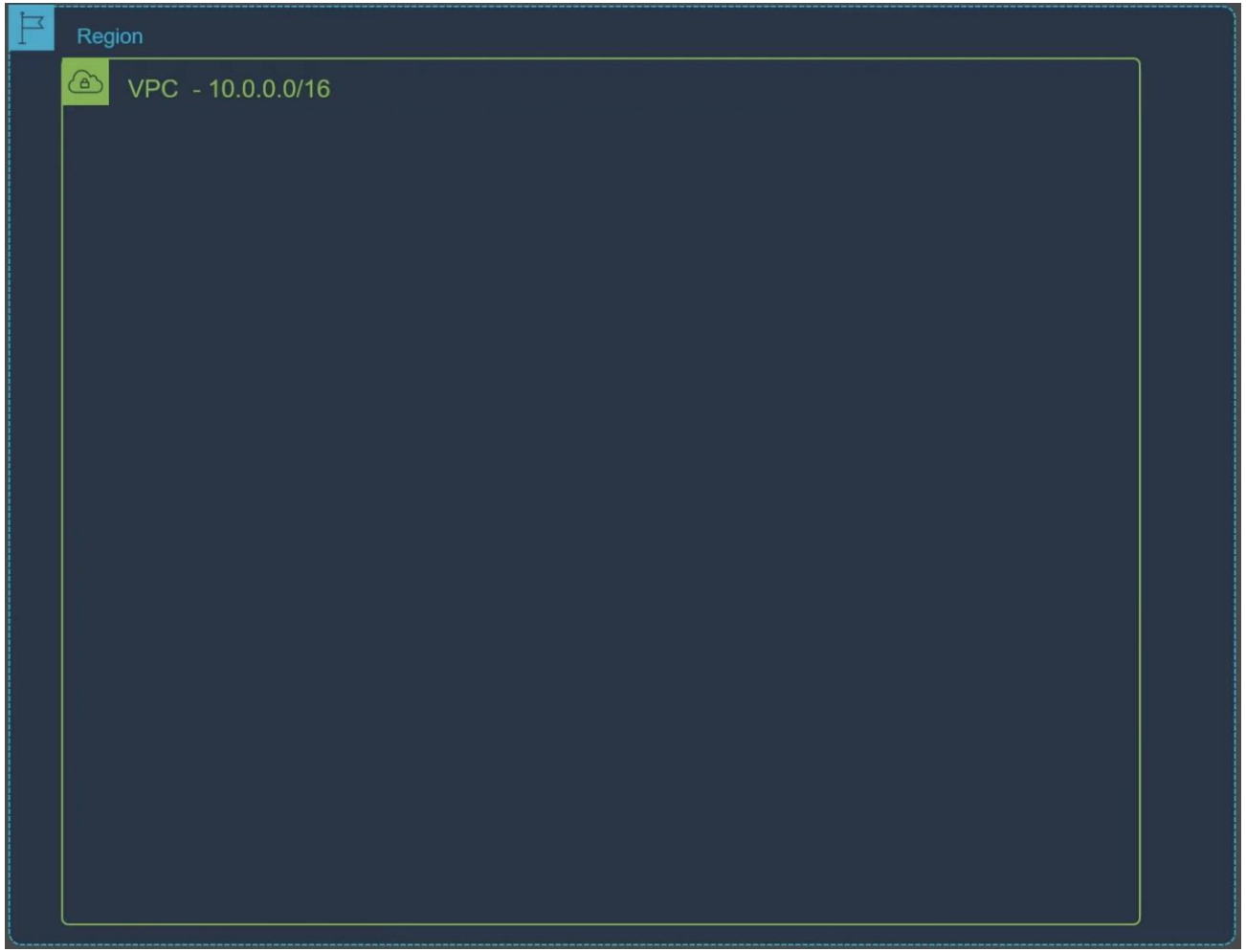
Select "*Create VPC*"

Short DEMO: Creating a VPC

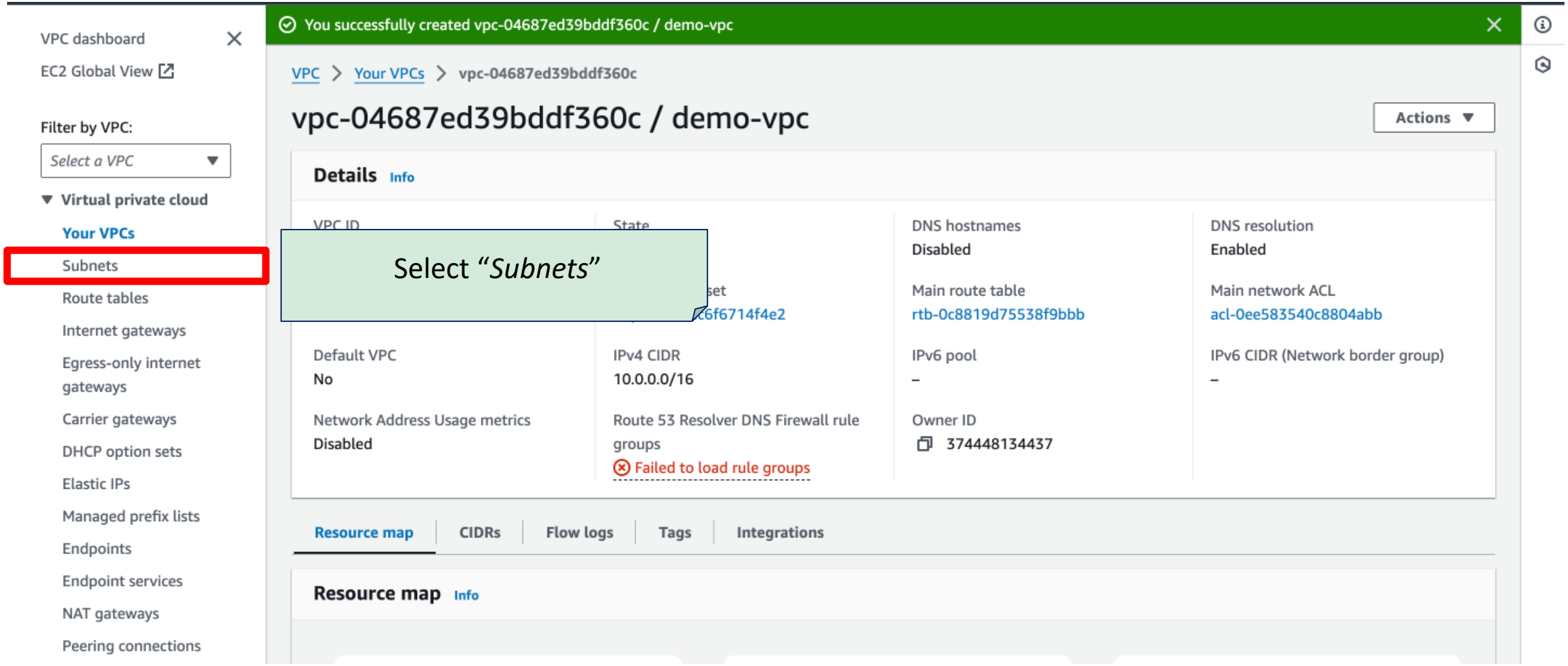


Short DEMO: Creating a VPC

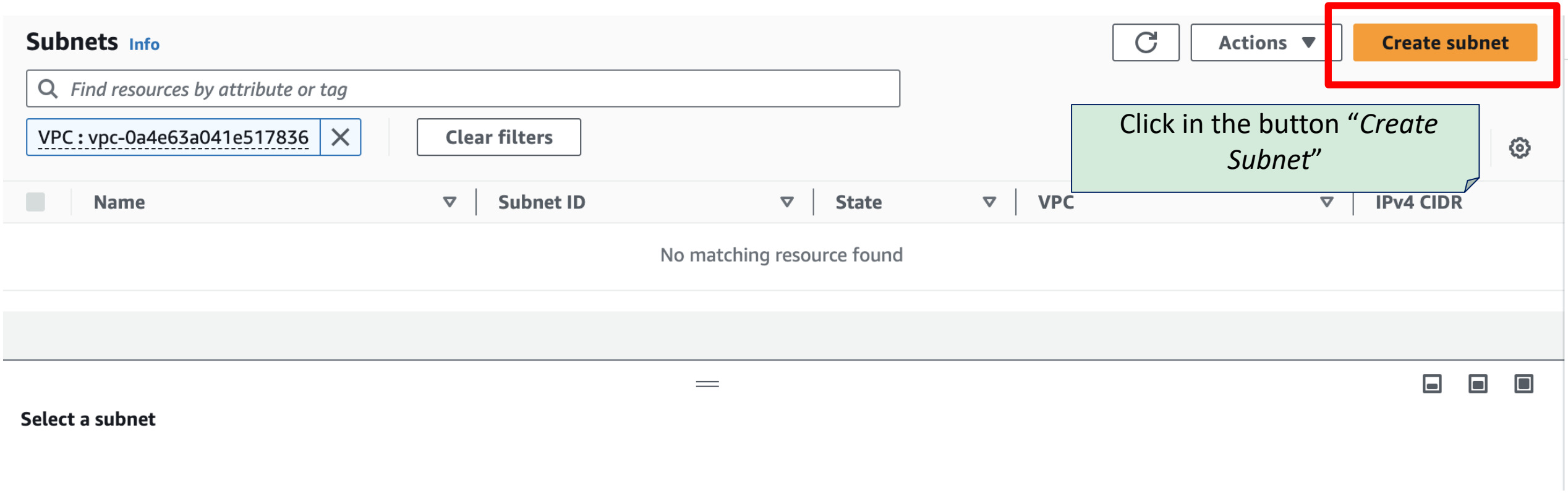
- At this point, we have created our VPC (similar to the figure).
- Note that virtual instances cannot be deployed directly in the VPC, but rather in a subnet of the VPC.
- Therefore, the next step is to create the subnets of the VPC. Remember that our VPC will containt 2 subnets: Private and Public



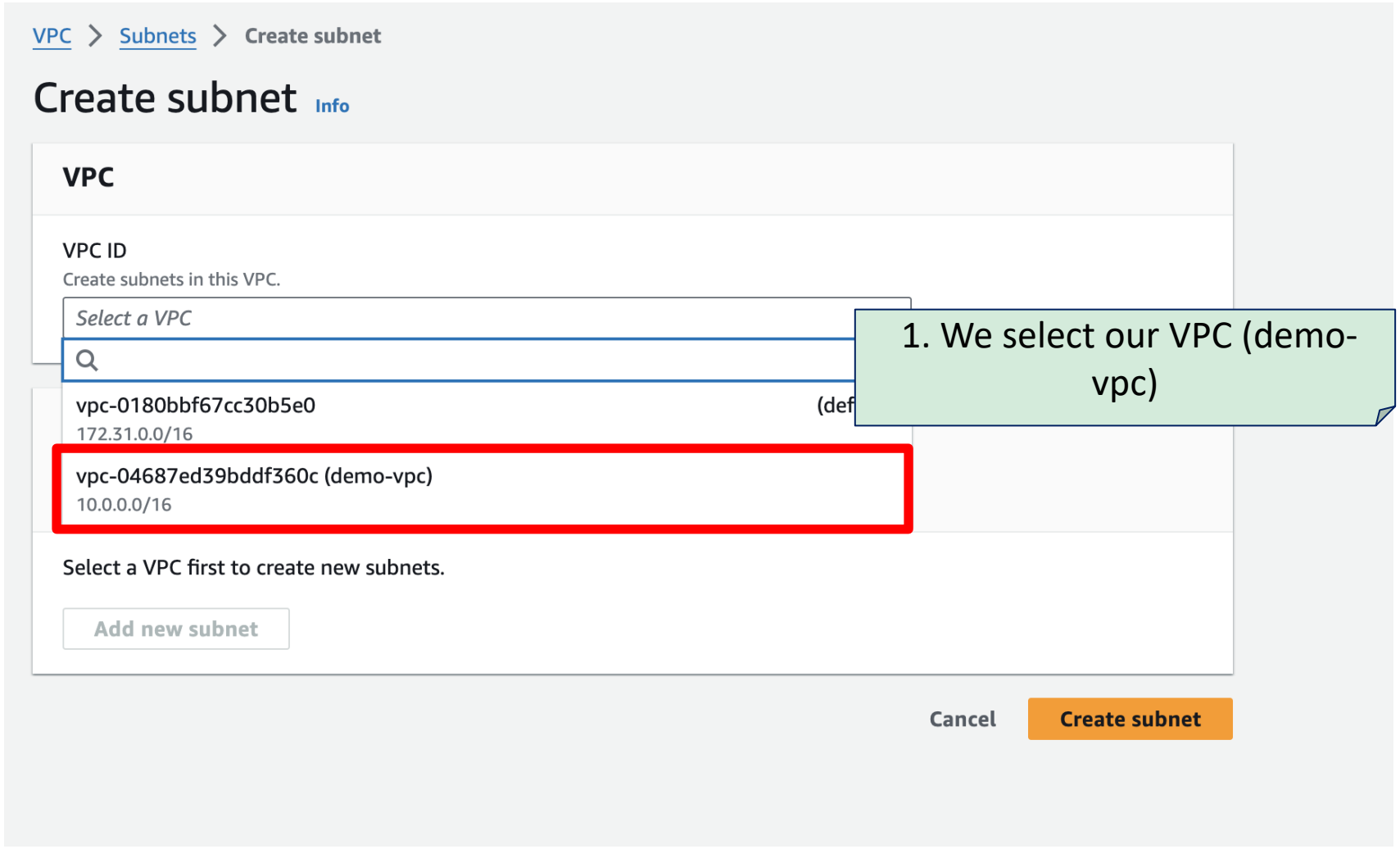
Short DEMO: Creating subnets within the VPC



Short DEMO: Creating subnets within the VPC



Short DEMO: Creating subnets within the VPC



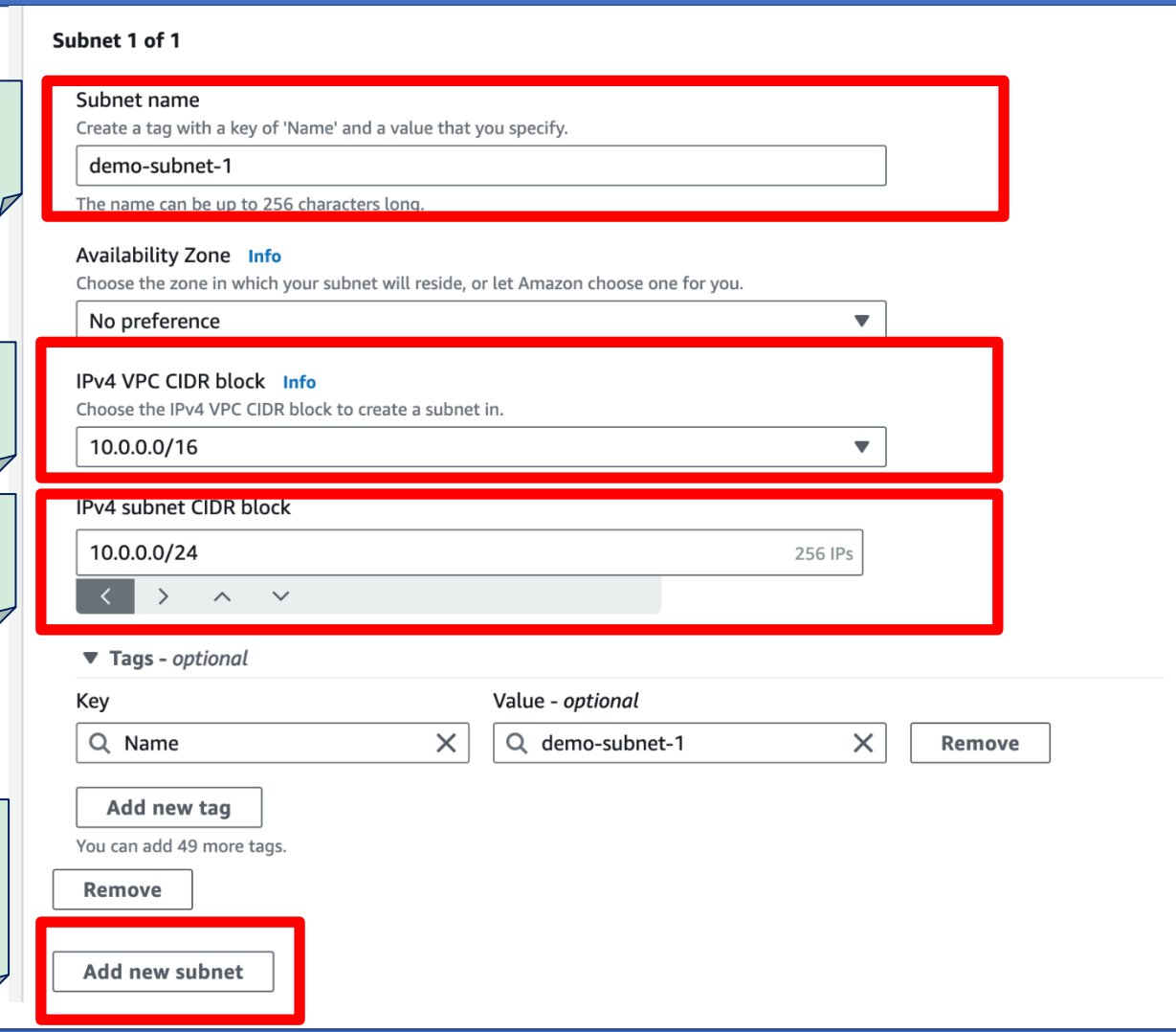
Short DEMO: Creating subnet 1 within the VPC

1. Write the name of our subnet

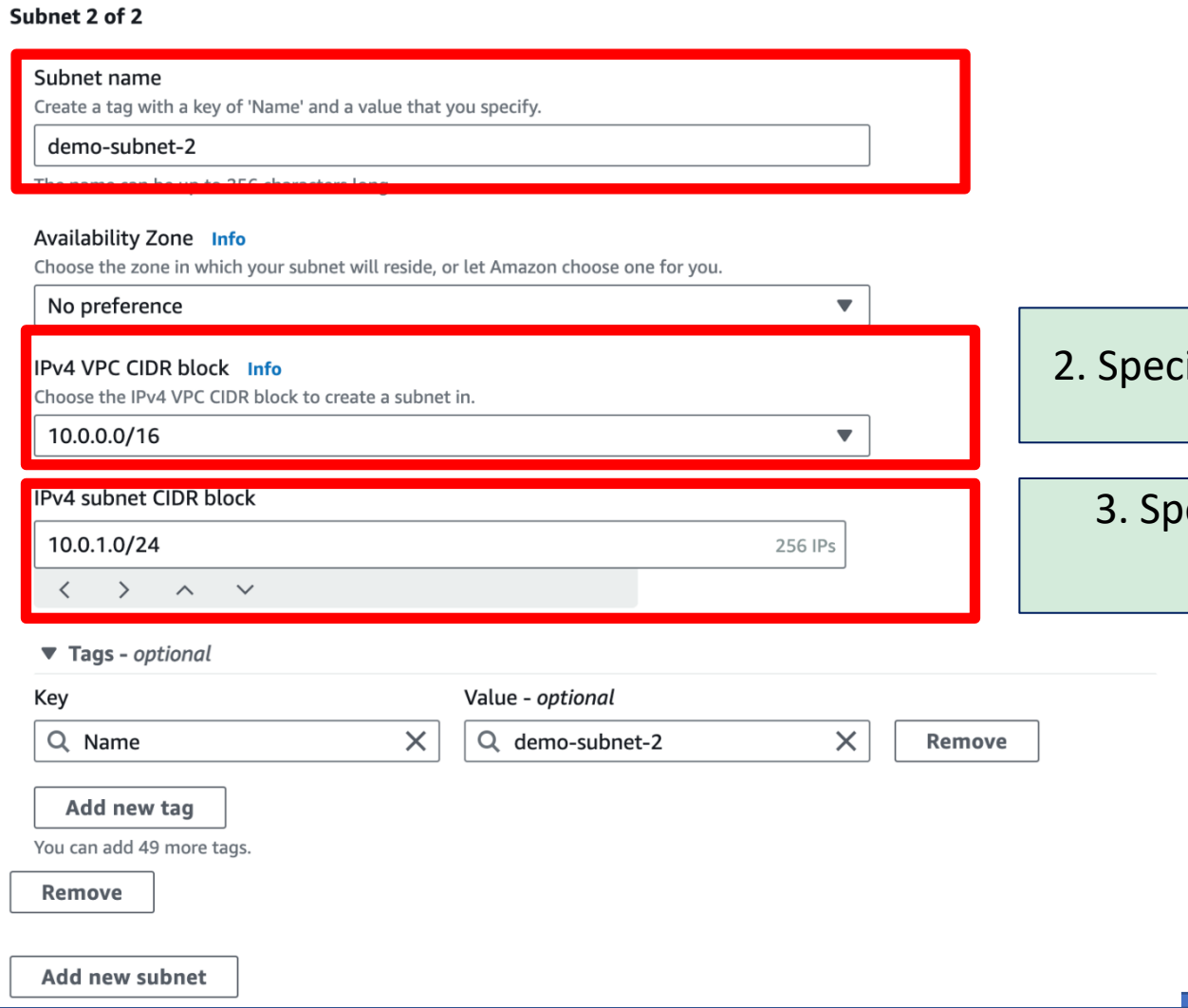
2. Specify the range of the VPC

3. Specify the range of the subnet

4. Select the button "*Add new subnet*" to créate the second subnet



Short DEMO: Creating subnet 2 within the VPC

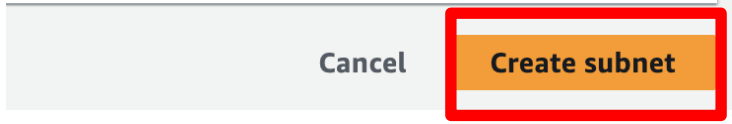


1. Write the name of our subnet

2. Specify the range of the VPC

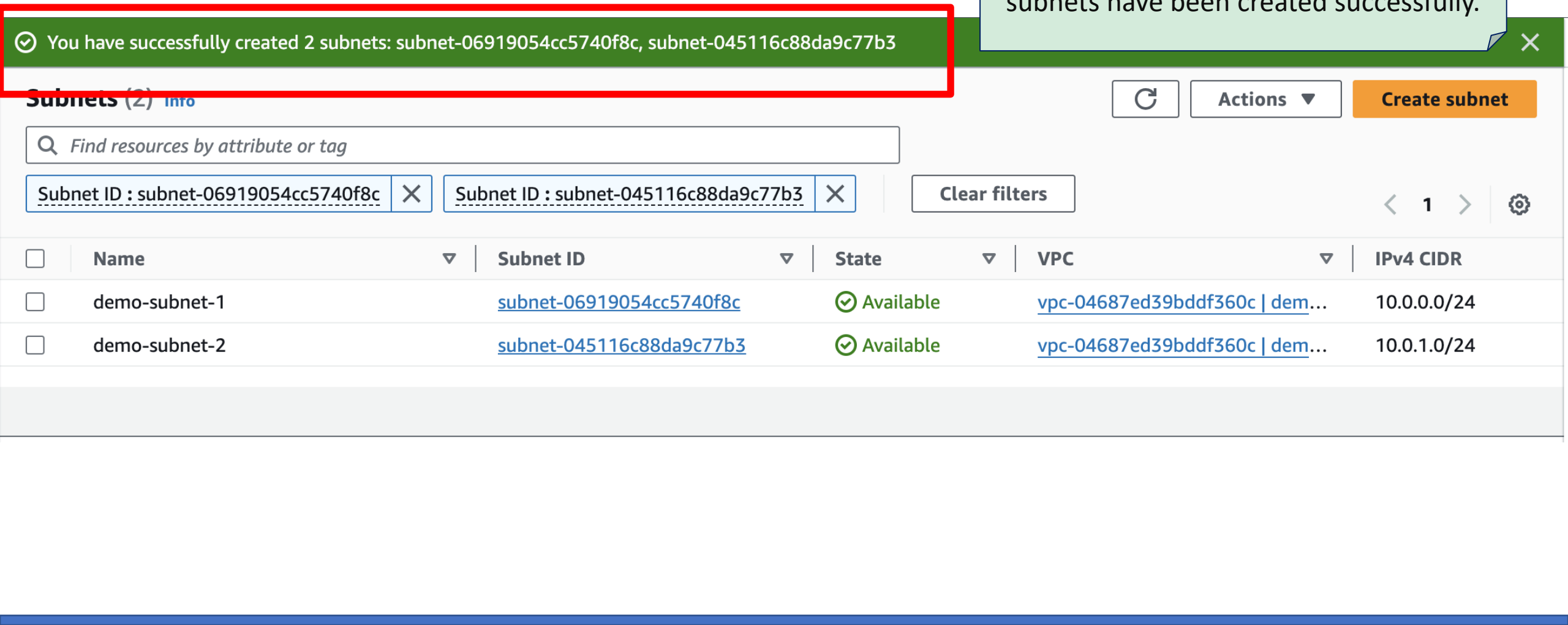
3. Specify the range of the subnet

4. Select the button "*create subnet*"



Short DEMO: Creating a subnet within the VPC

If you obtain a similar to this one, the subnets have been created successfully.

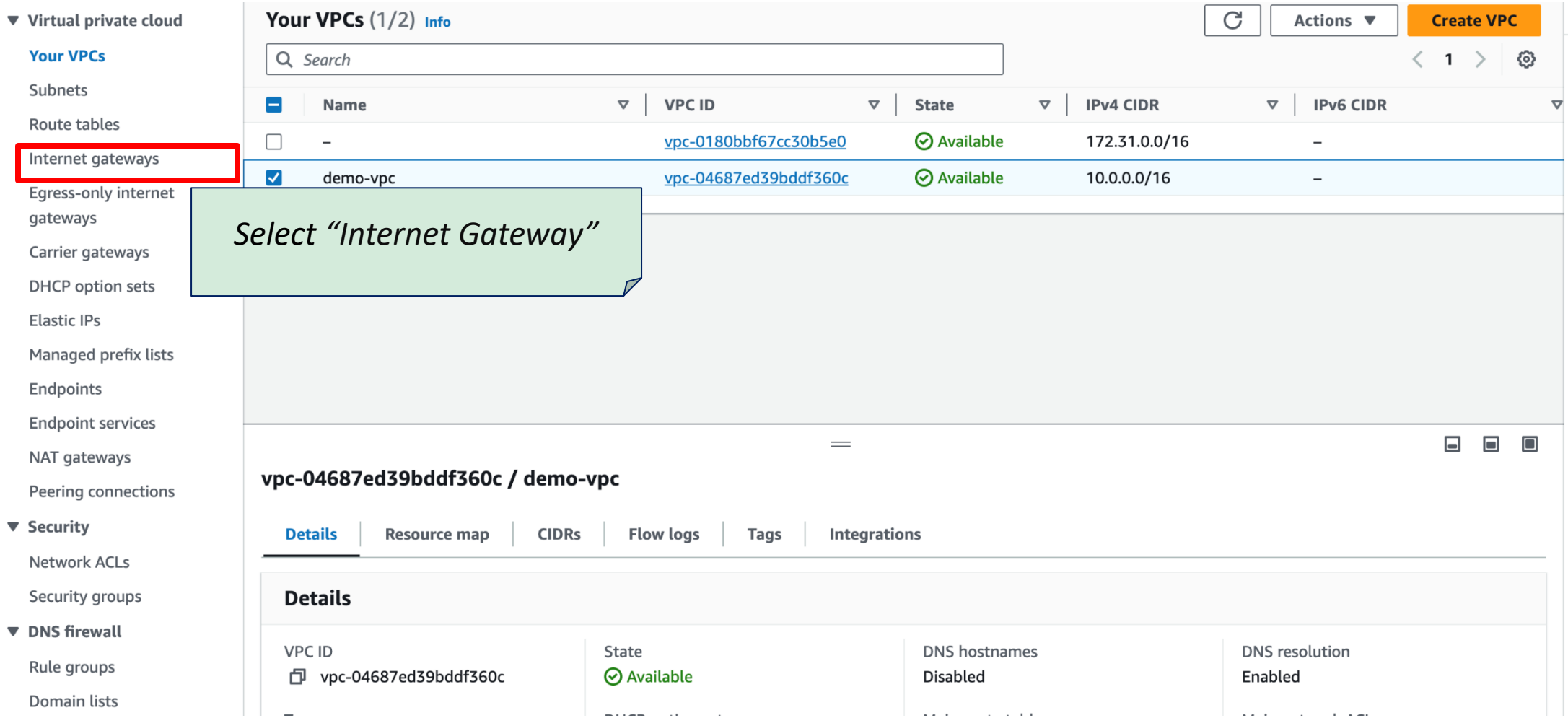


Short DEMO: Creating a subnet within the VPC

- At this point, we have created 2 private subnets (They have not conection to internet).

- Therefore, the next step is to create all the necessary components to provide access to the internet for subnet1.

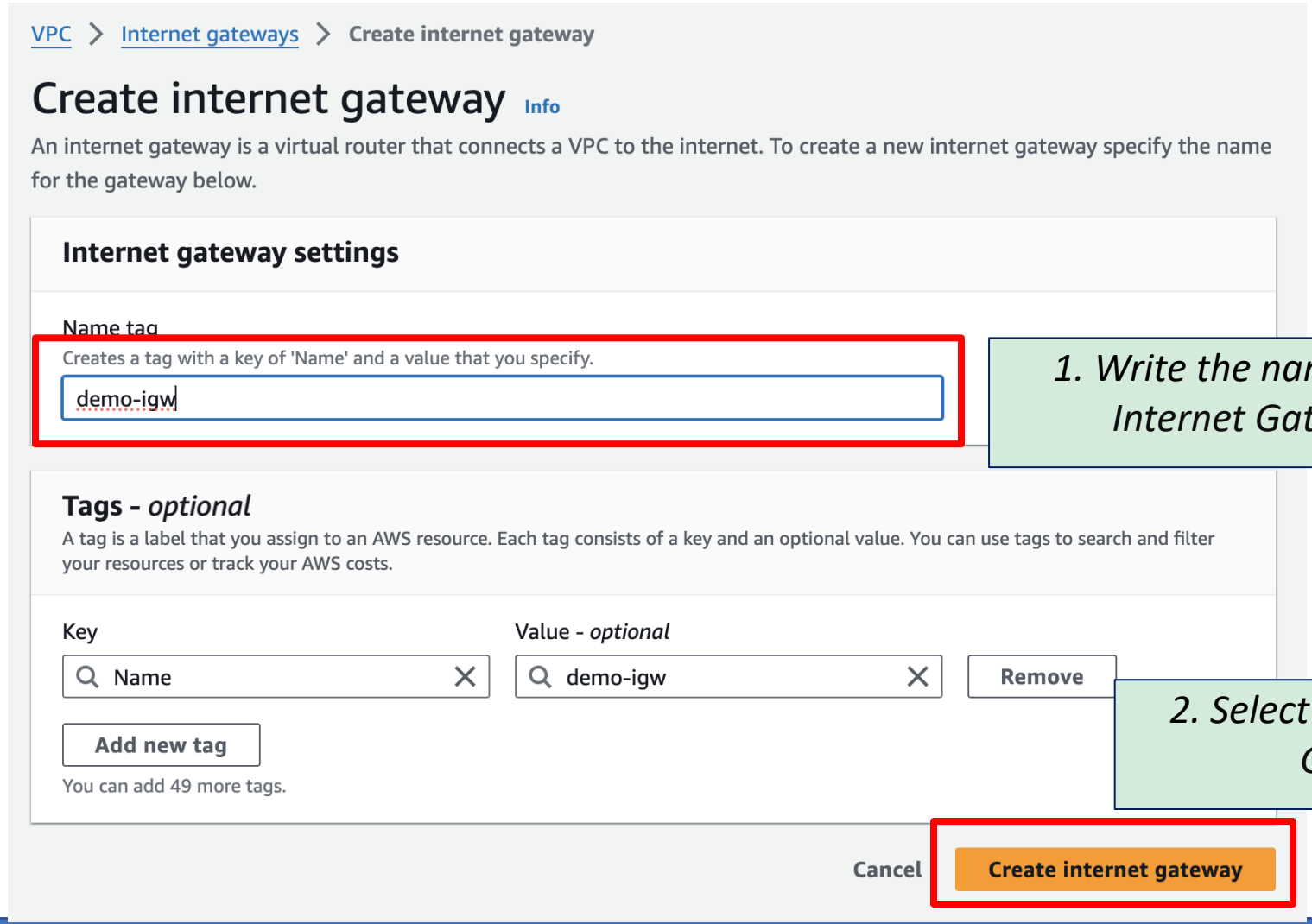
Short DEMO: Creating a Internet Gateway to acces to Internet



Short DEMO: Creating a Internet Gateway to acces to Internet

Select " Create Internet Gateway"

Short DEMO: Creating a Internet Gateway to acces to Internet

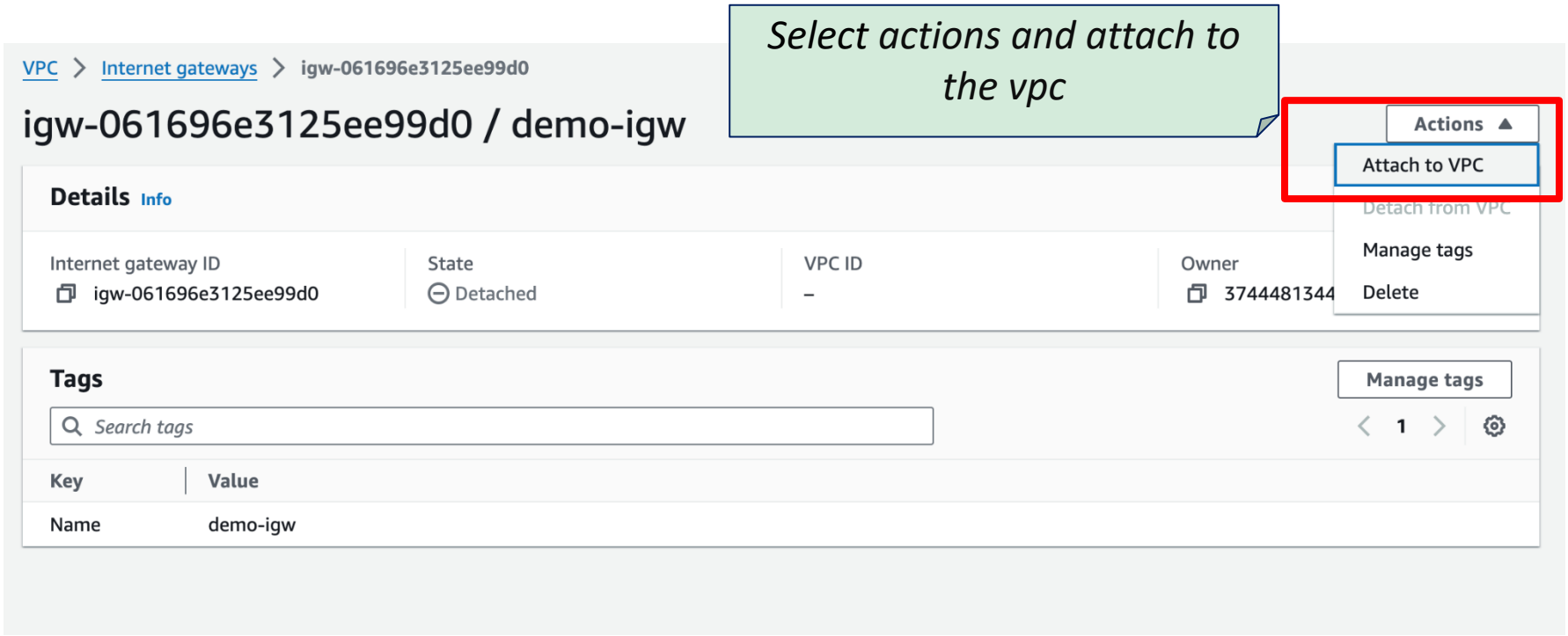


1. Write the name of the Internet Gateway

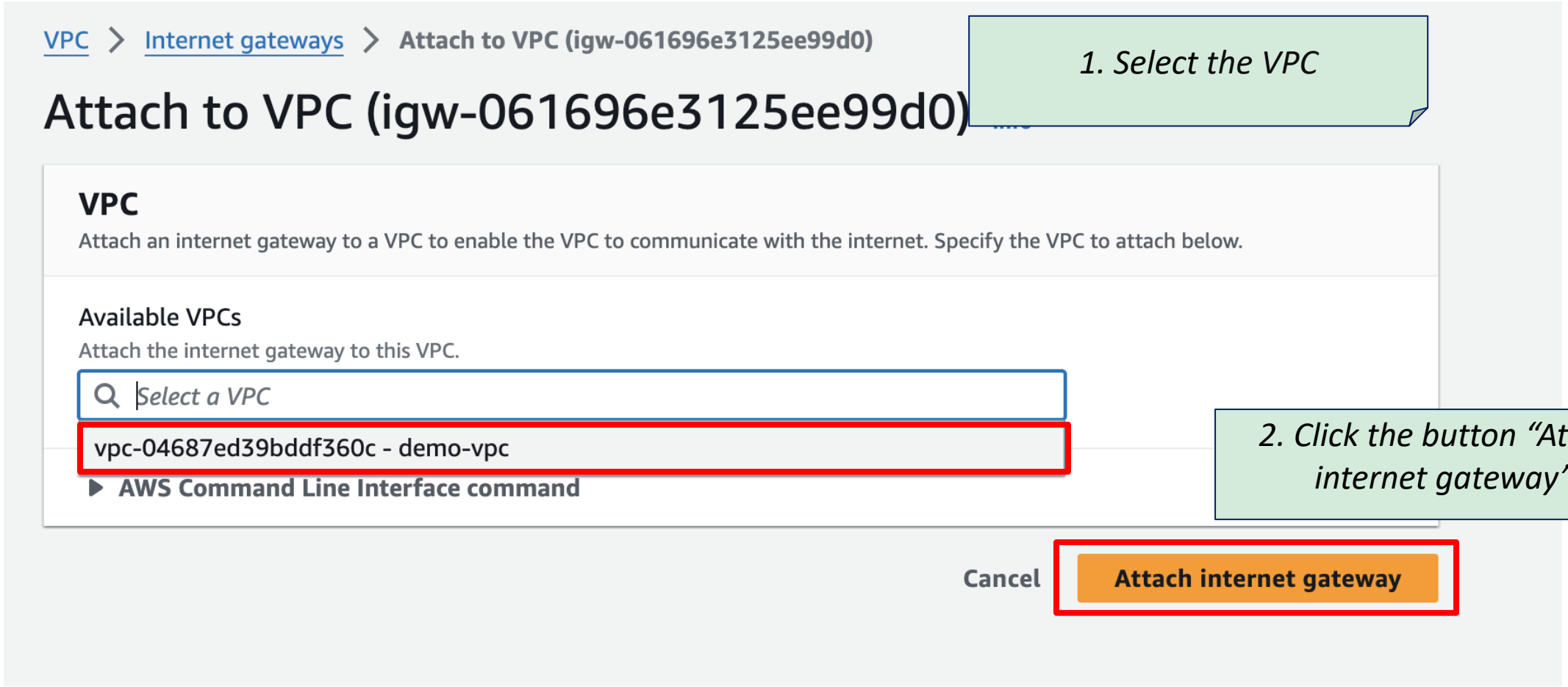
2. Select " Create Internet Gateway"

Short DEMO: Creating a Internet Gateway to acces to Internet

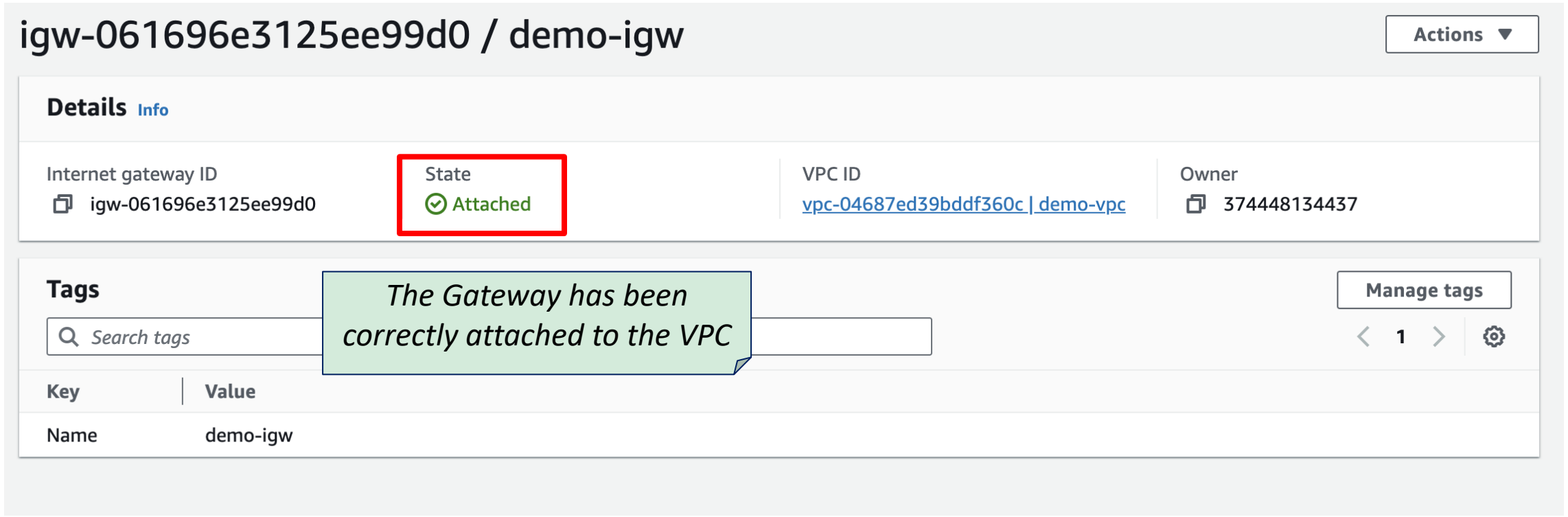
- Once the Internet Gateway has been created, it is necessary to attach it to our VPC.



Short DEMO: Creating a Internet Gateway to acces to Internet

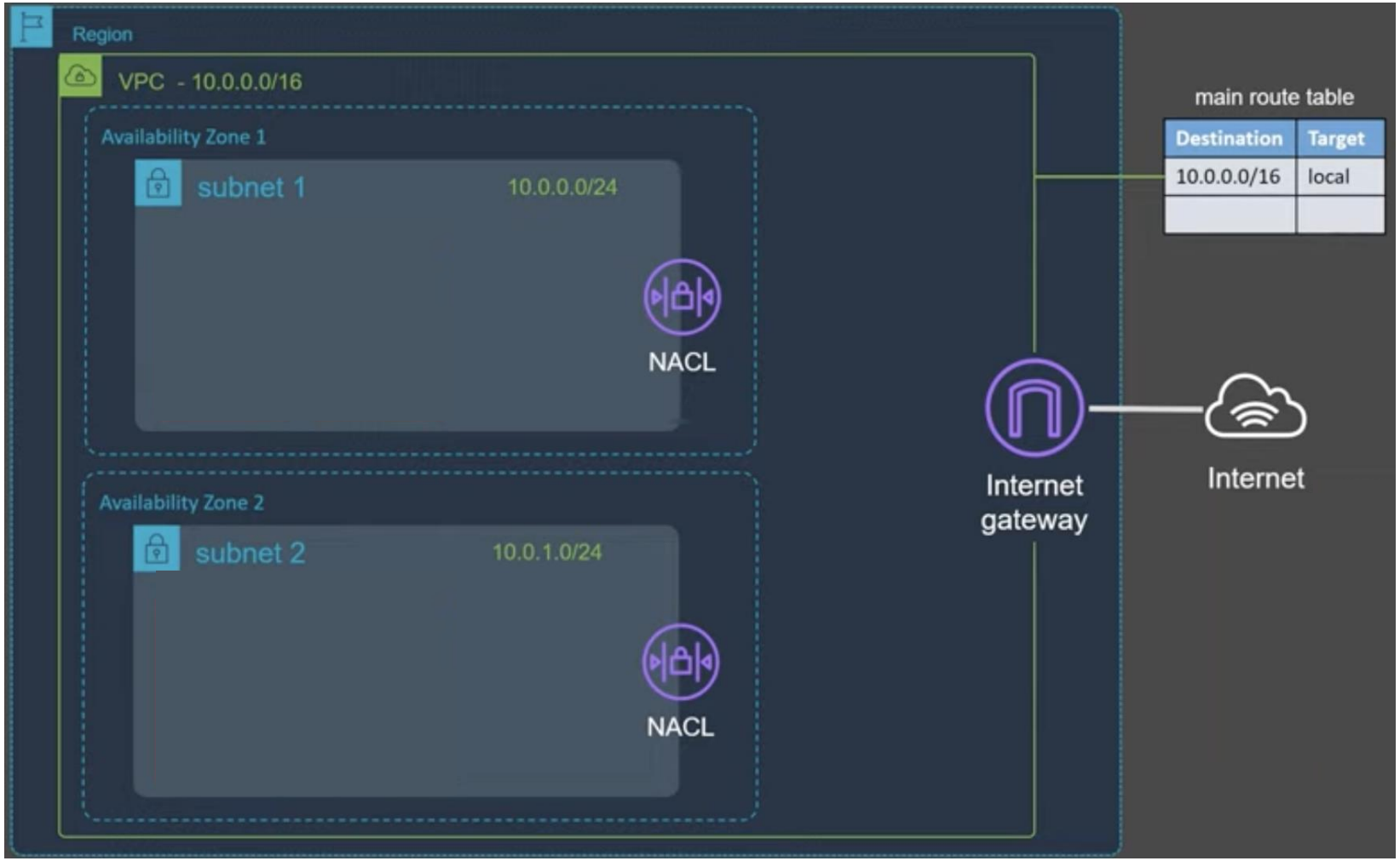


Short DEMO: Creating a Internet Gateway to acces to Internet

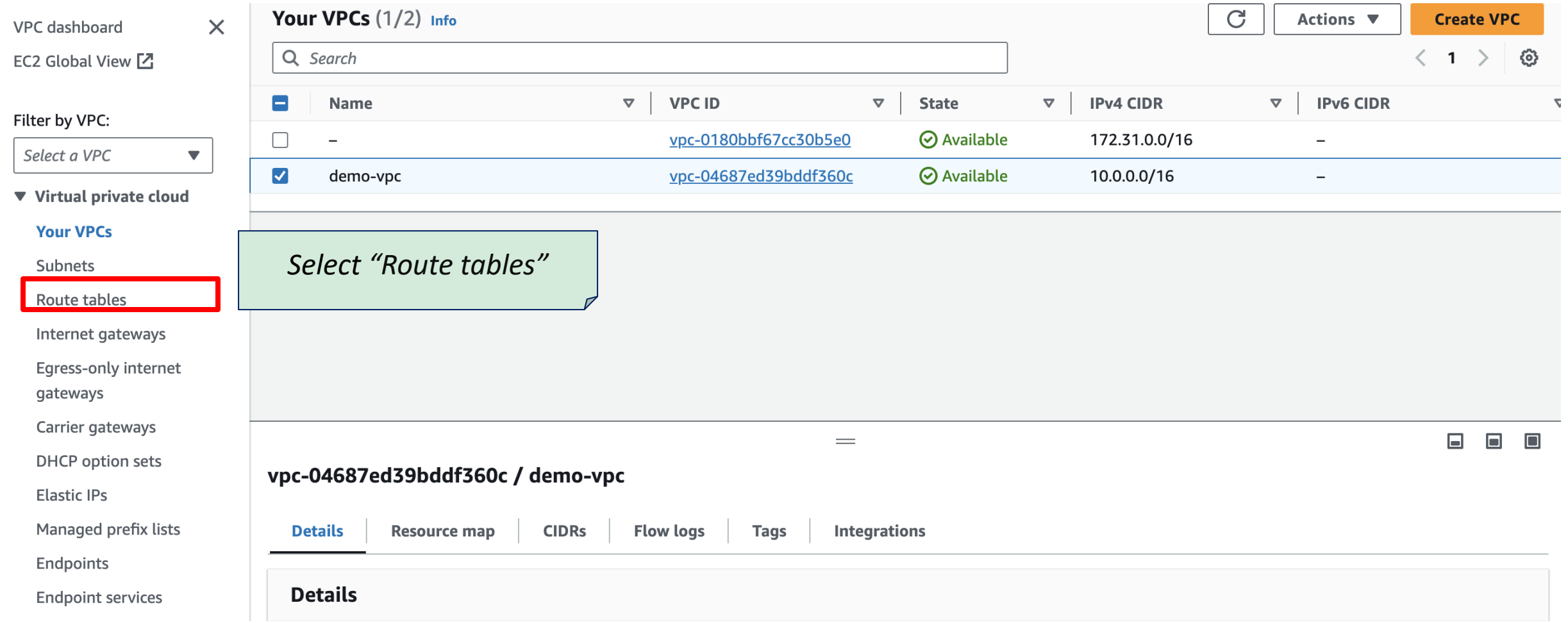


Short DEMO: Creating a Internet Gateway to acces to Internet

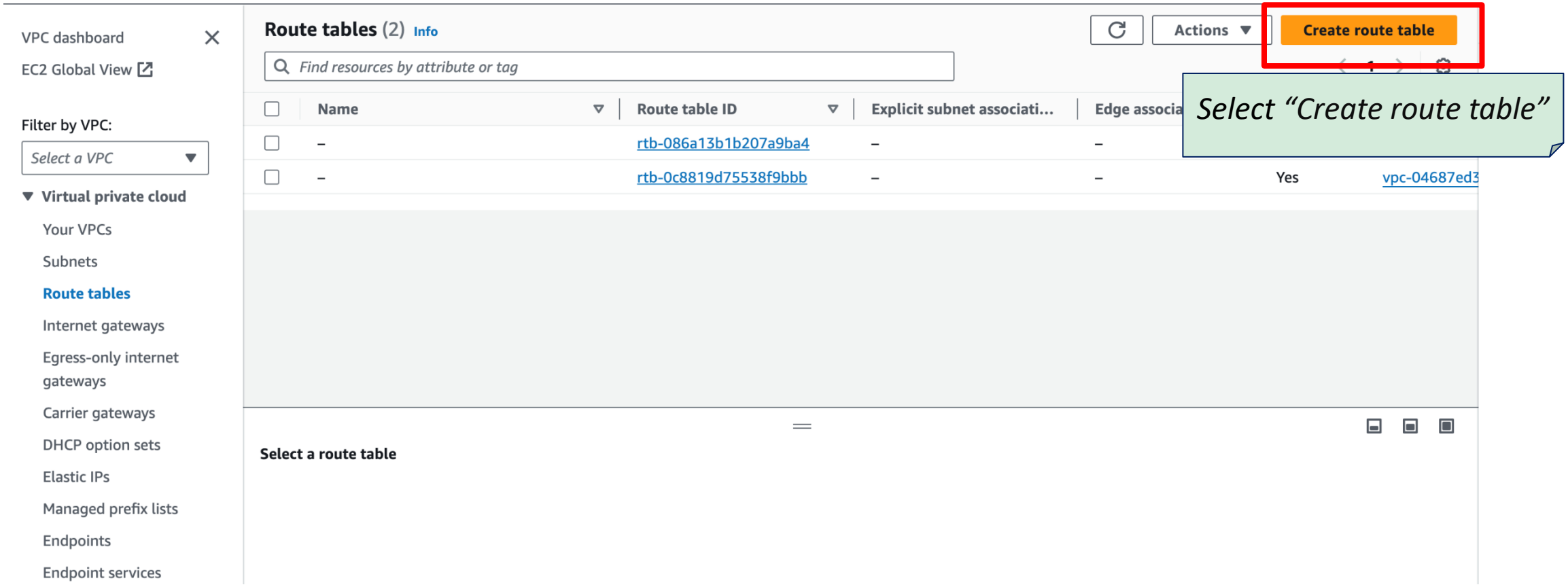
- So far, our VPC has the following structure.
- Subnets still have not access to internet. The routes to link the subnet1 with the Gateway are missing.
- It is necessary to create the routes to connect the Gateway with subnet1.
- The next step is to create a Route Table to carry out the connnection routes



Short DEMO: Creating a Route Table to link the Subnet1 with the Gateway to acces to Internet



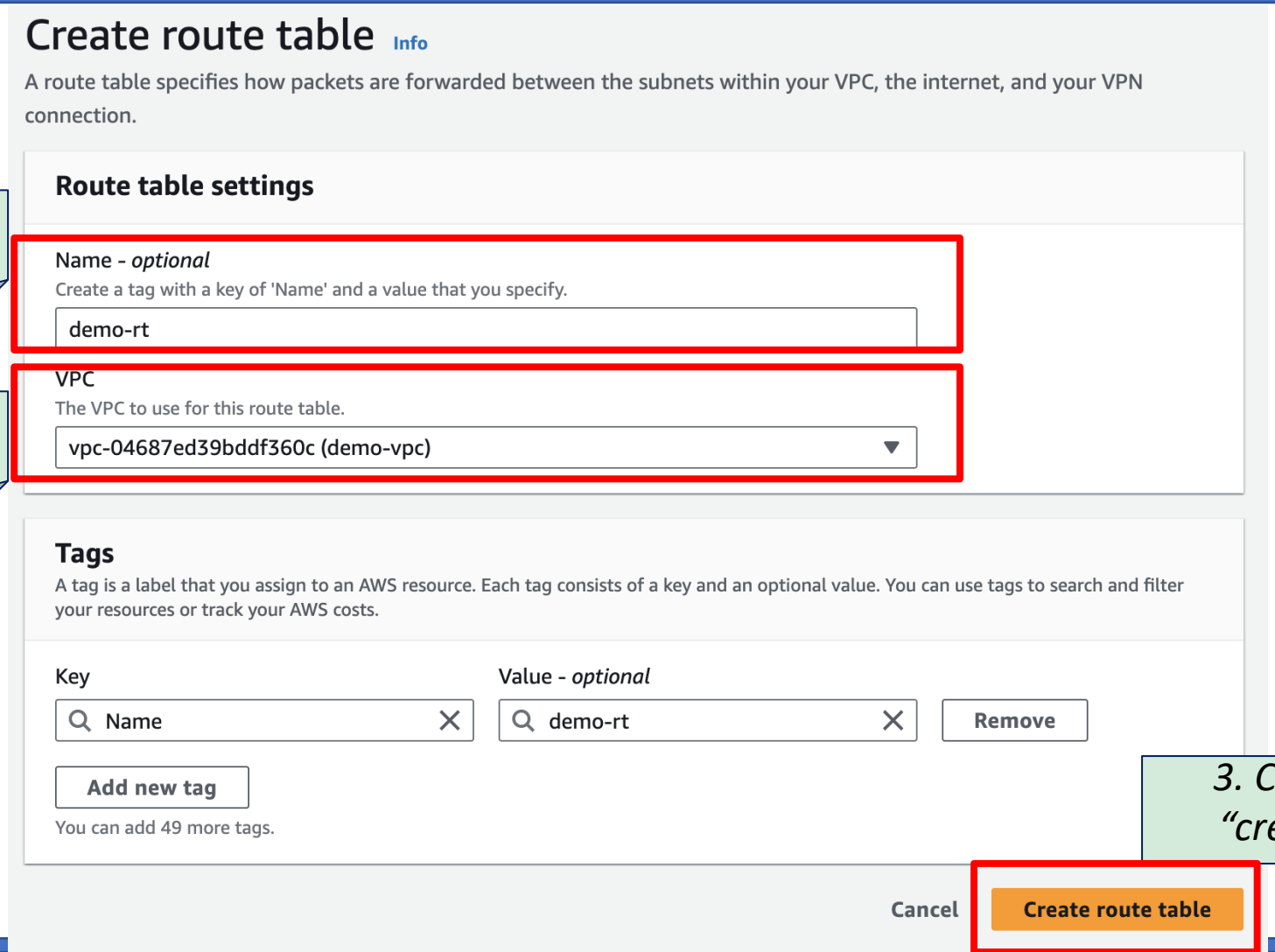
Short DEMO: Creating a Route Table to link the Subnet1 with the Gateway to acces to Internet



Short DEMO: Creating a Route Table to link the Subnet1 with the Gateway to acces to Internet

1. We write the name of the route table

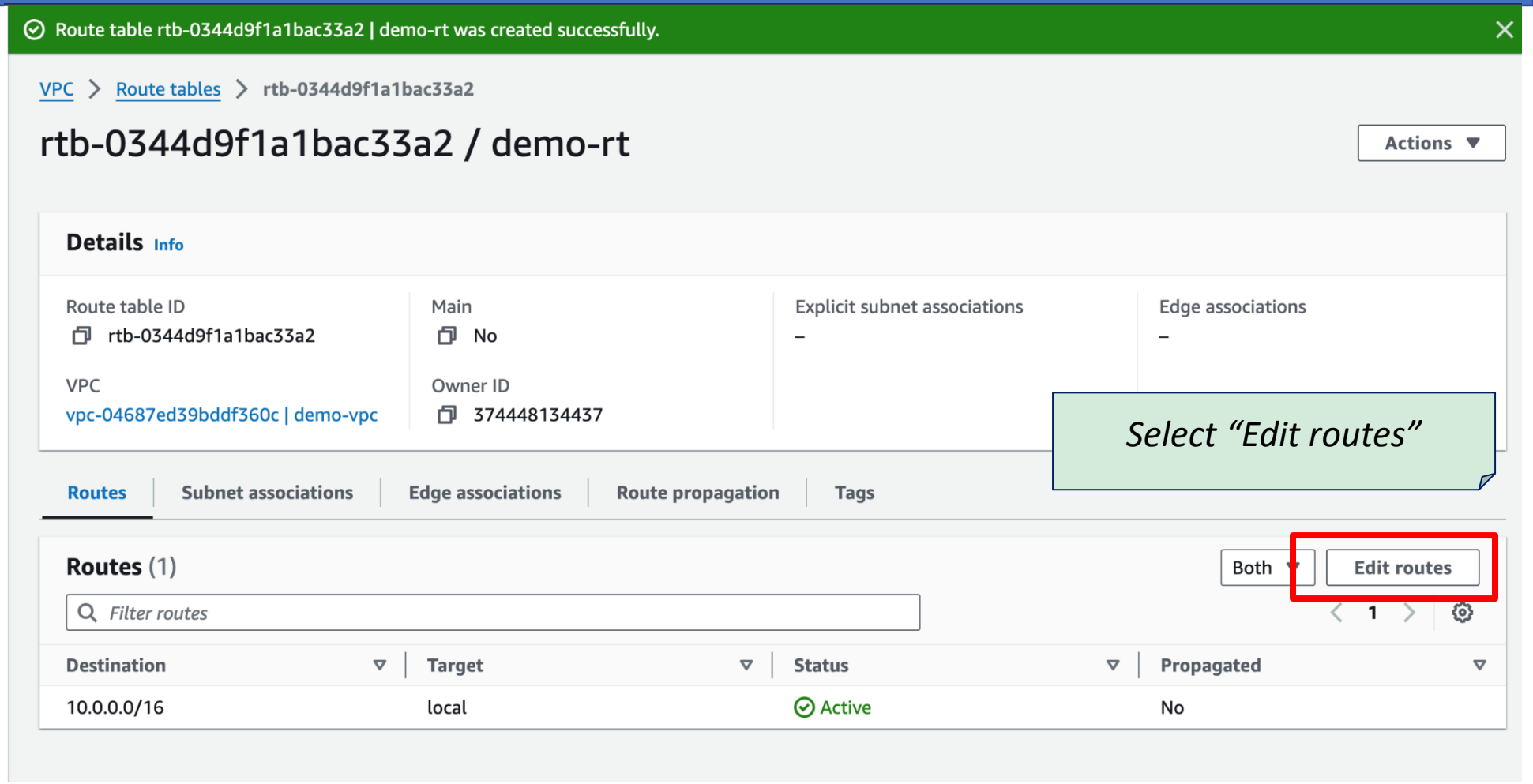
2. We select our VPC



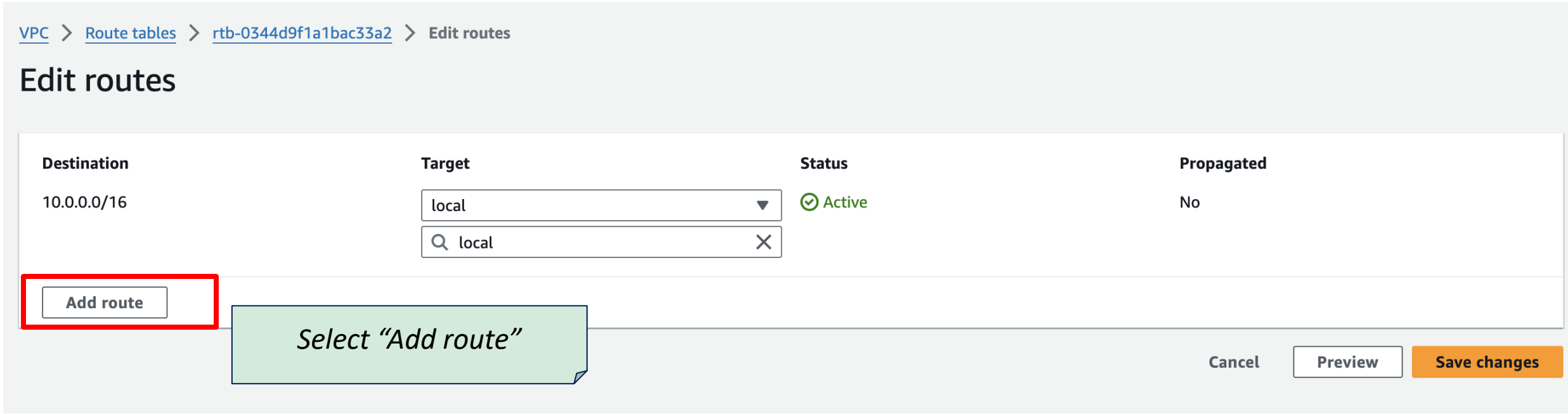
3. Click in the button "create route table"

Short DEMO: Creating a Route Table to link the Subnet1 with the Gateway to acces to Internet

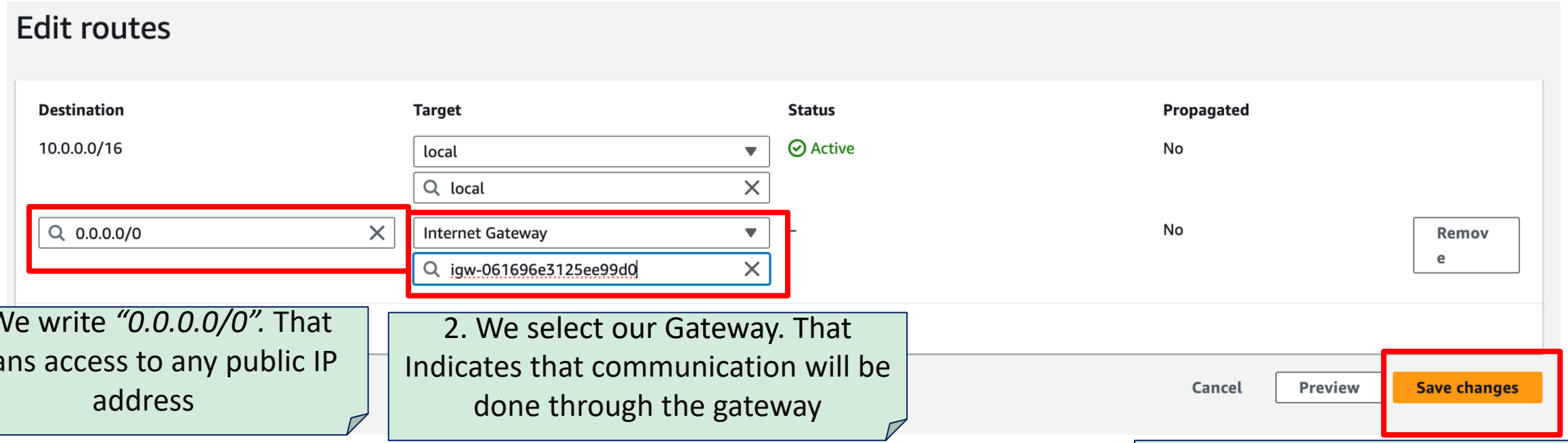
- We edit the route table to add a new entry that redirects traffic to the gateway.



Short DEMO: Creating a Route Table to link the Subnet1 with the Gateway to acces to Internet



Short DEMO: Creating a Route Table to link the Subnet1 with the Gateway to acces to Internet



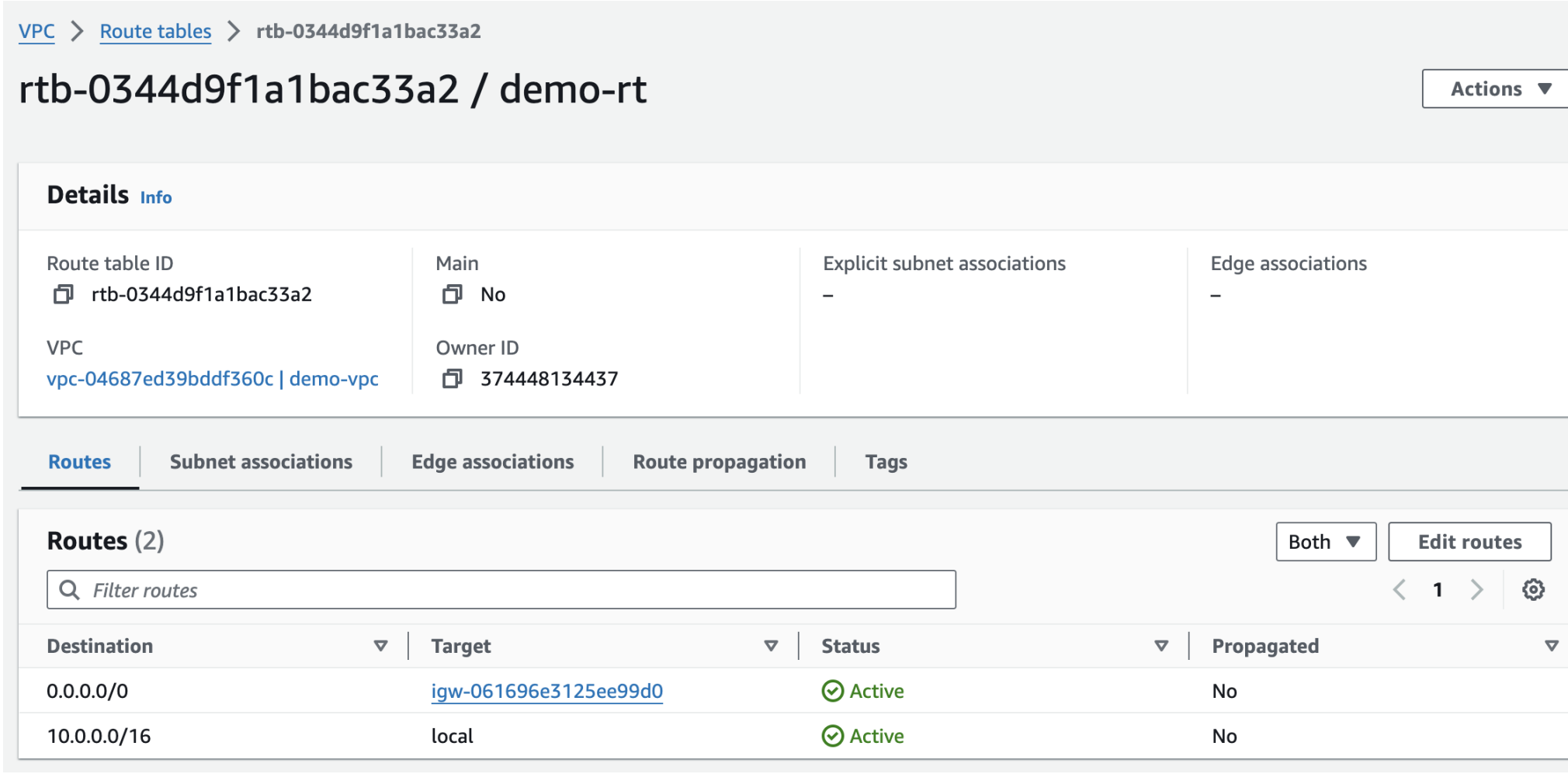
1. We write *"0.0.0.0/0".* That means access to any public IP address

2. We select our Gateway. That Indicates that communication will be done through the gateway

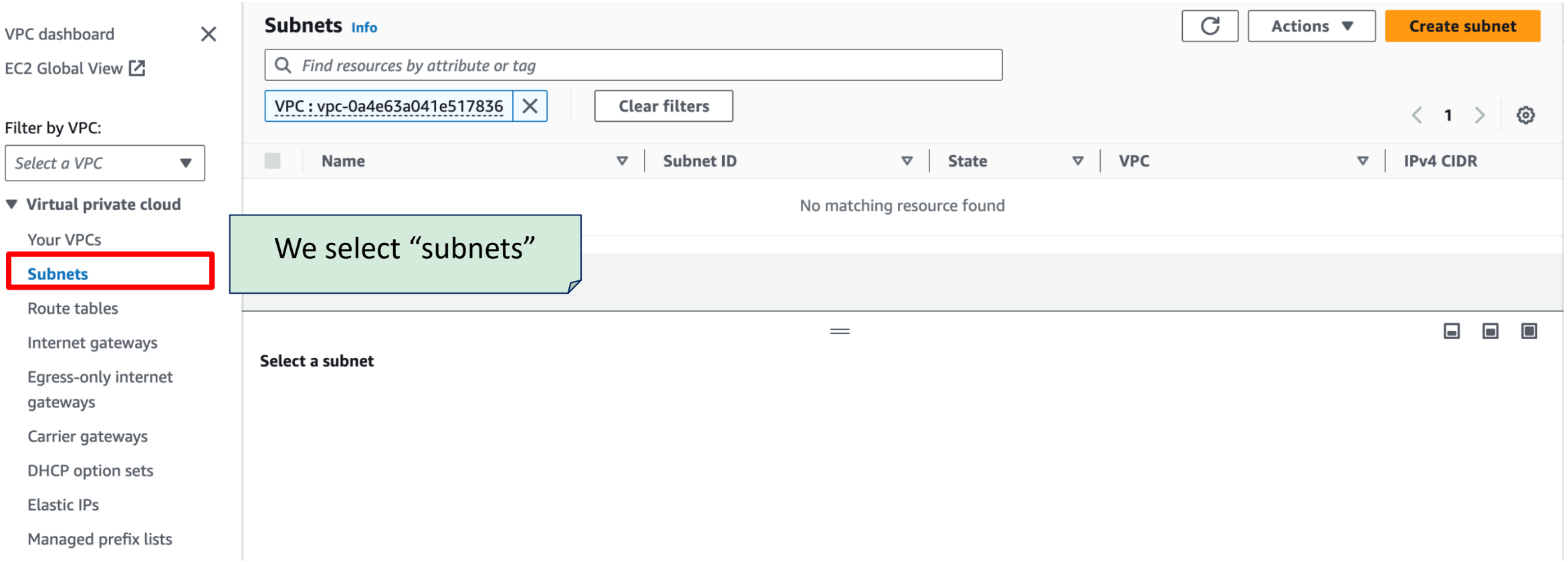
3. We select or Gateway to add to indicate that

Short DEMO: Associating the Route Table with the subnet 1

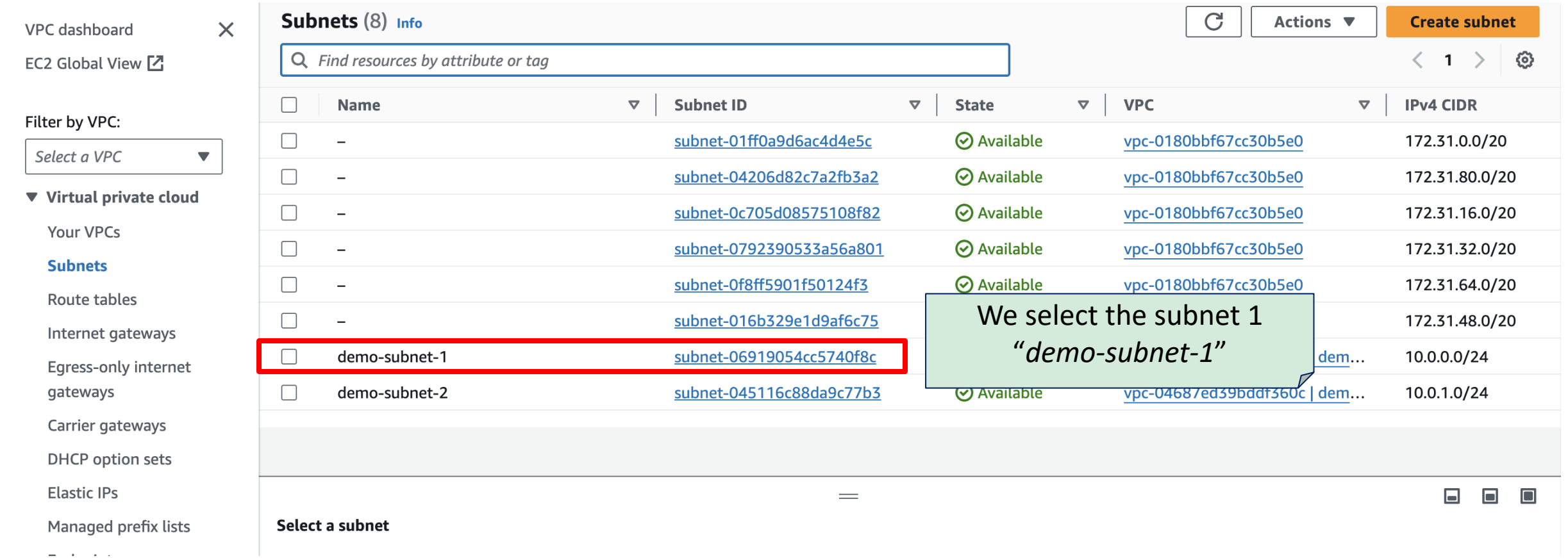
- Now, we must associate the routing table with the subnet that we want to have access to the Internet. In our case it will be subnet 1.



Short DEMO: Associating the Route Table with the subnet 1



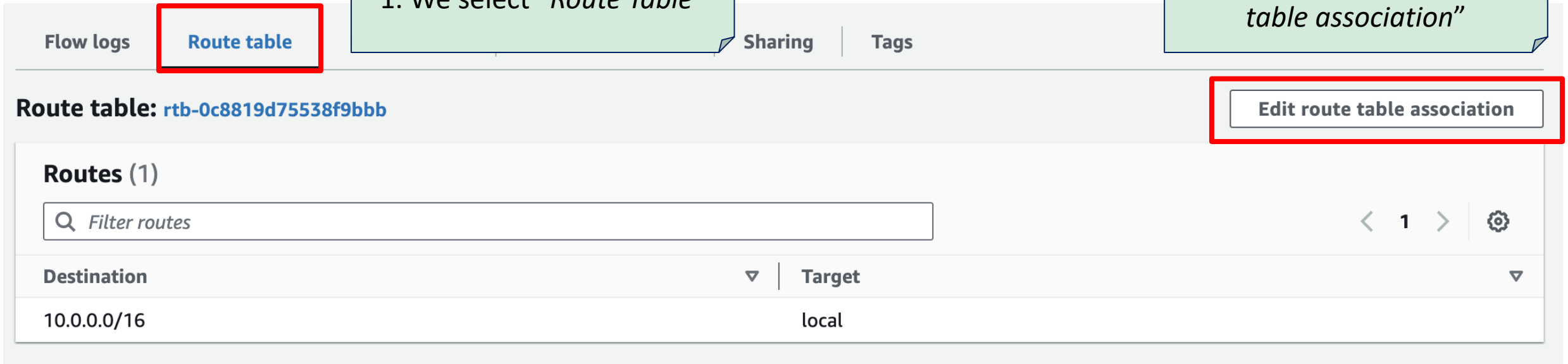
Short DEMO: Associating the Route Table with the subnet 1



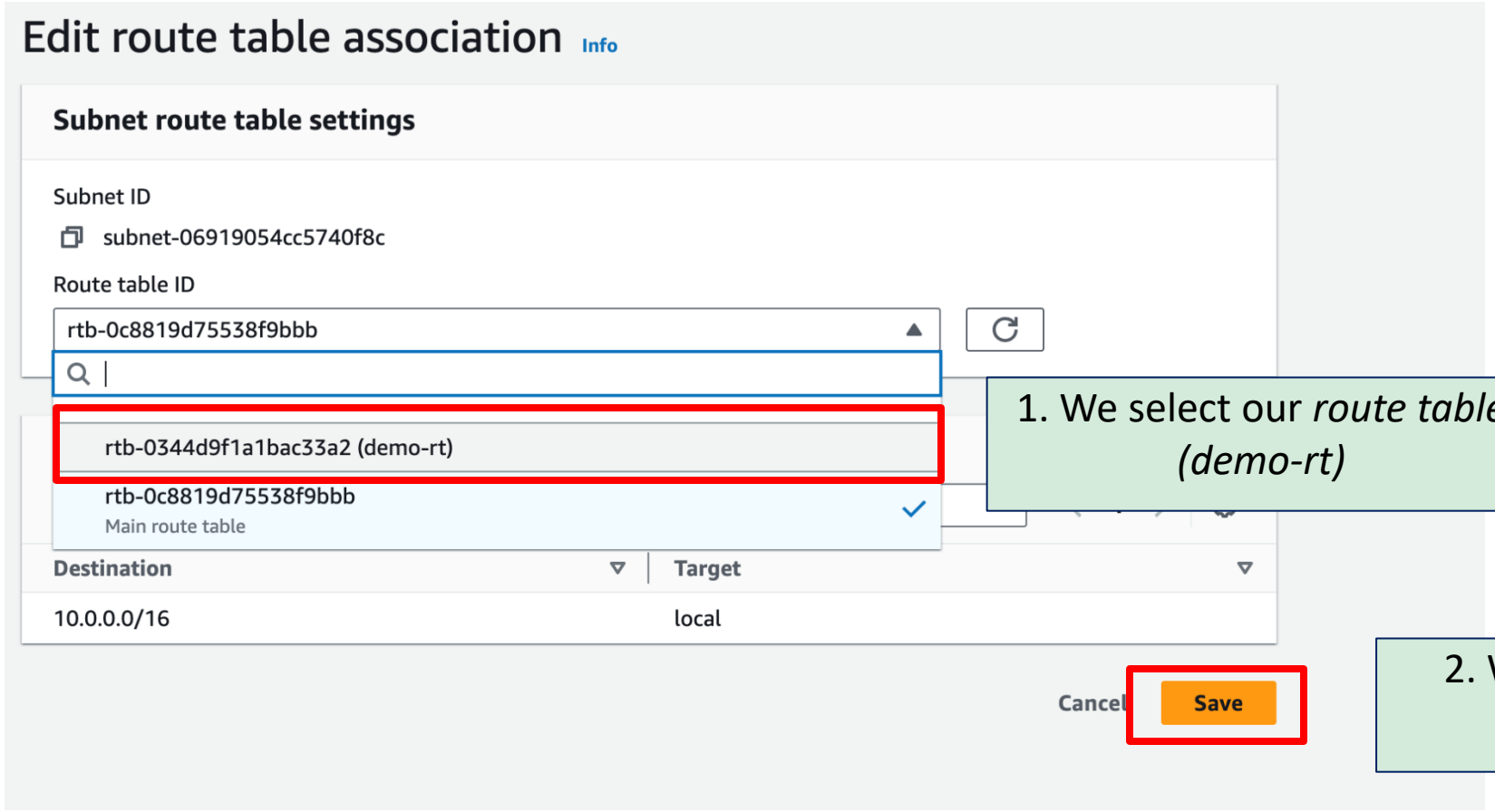
Short DEMO: Associating the Route Table with the subnet 1

1. We select "*Route Table*"

2. We select "*Edit route table association*"



Short DEMO: Associating the Route Table with the subnet 1

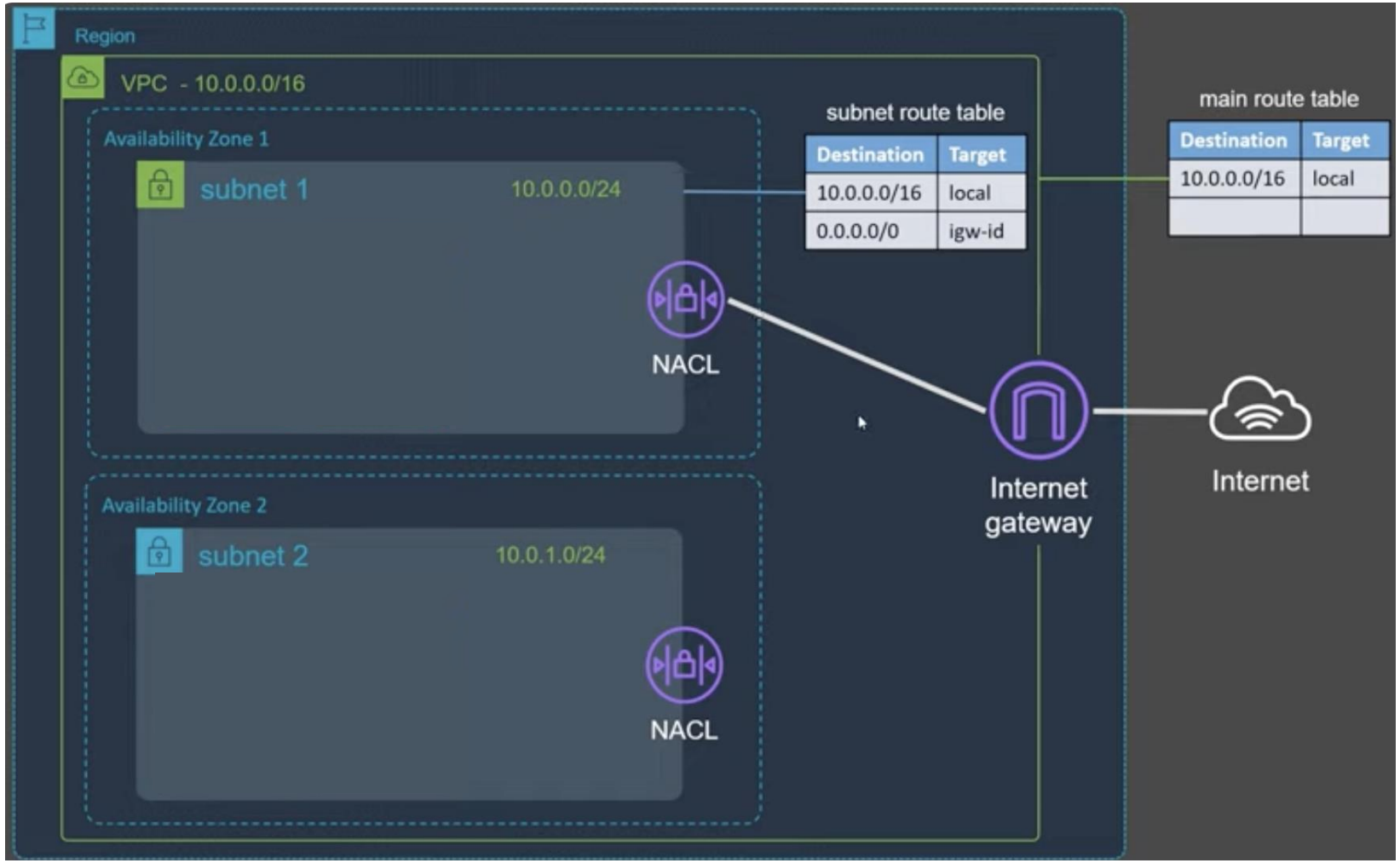


1. We select our *route table (demo-rt)*

2. We select the button “save”

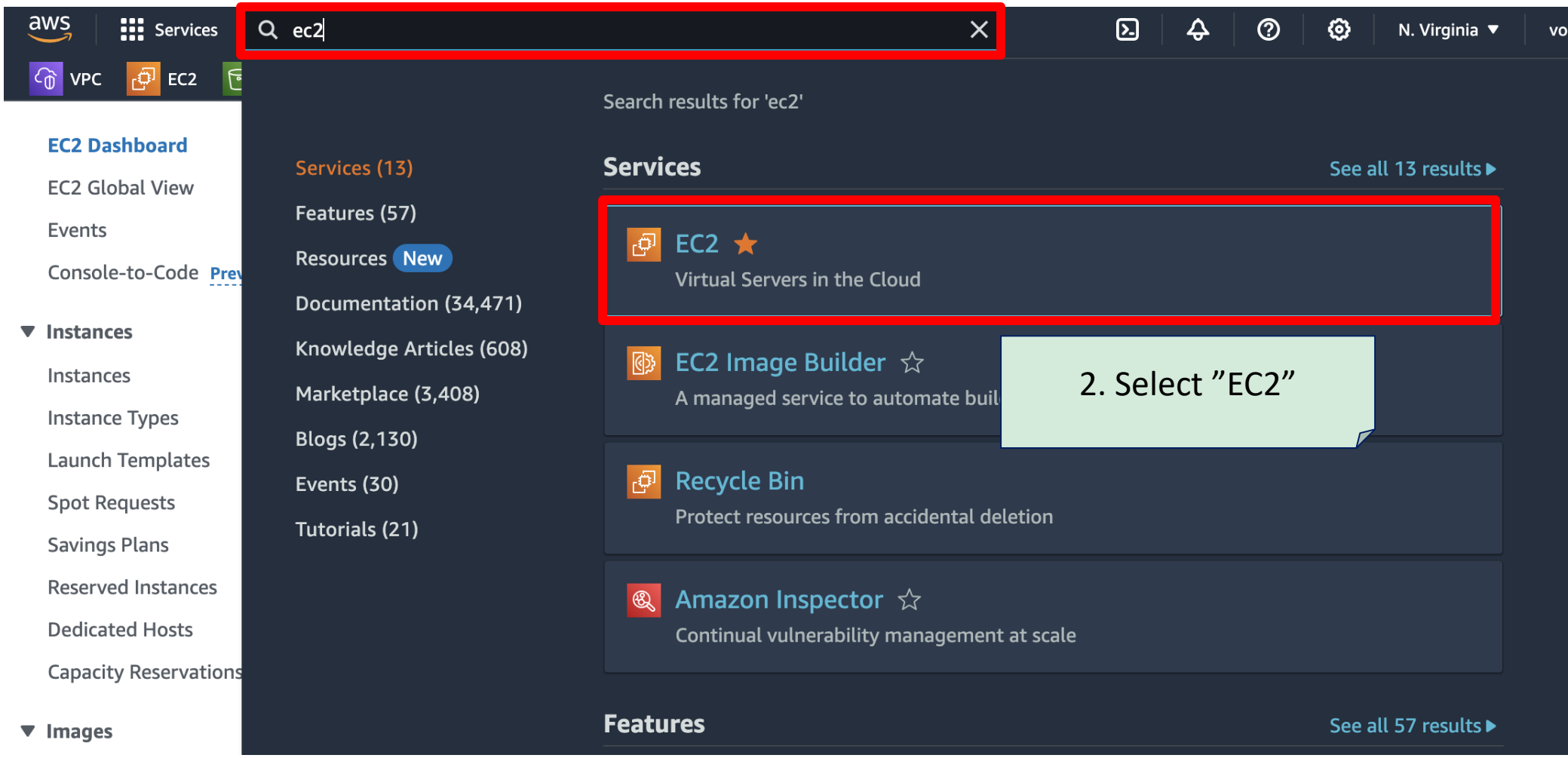
Short DEMO: Associating the Route Table with the subnet 1

- Our VPC has this structure.
- Subnet 1 has internet access.
- Now it is possible to deploy instances in Subnet 1, which will be able to connect to the internet.



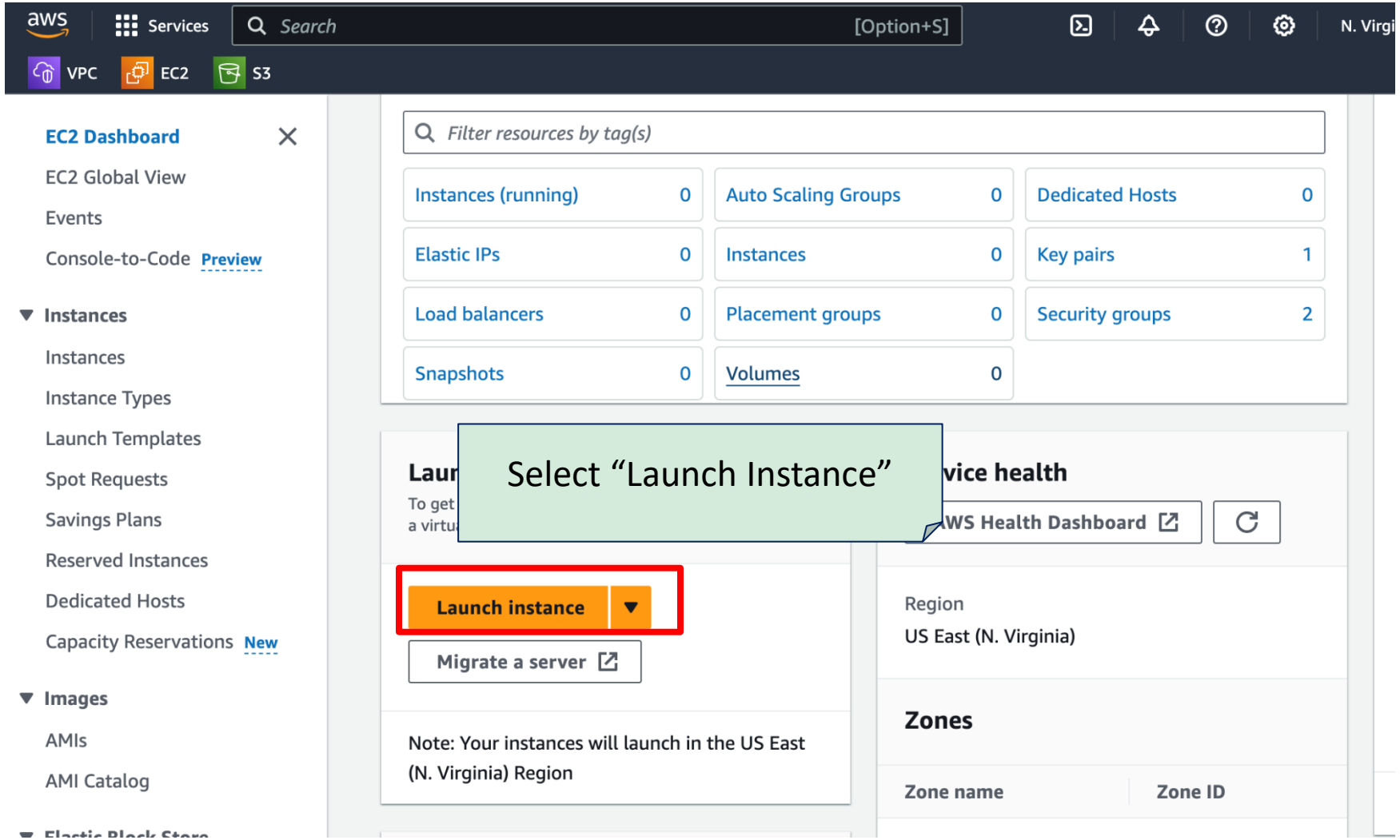
Short DEMO: Deploying Instances in the Subnet1 of our VPC

- In the search bar, we write the EC2 to deploy computing instances.



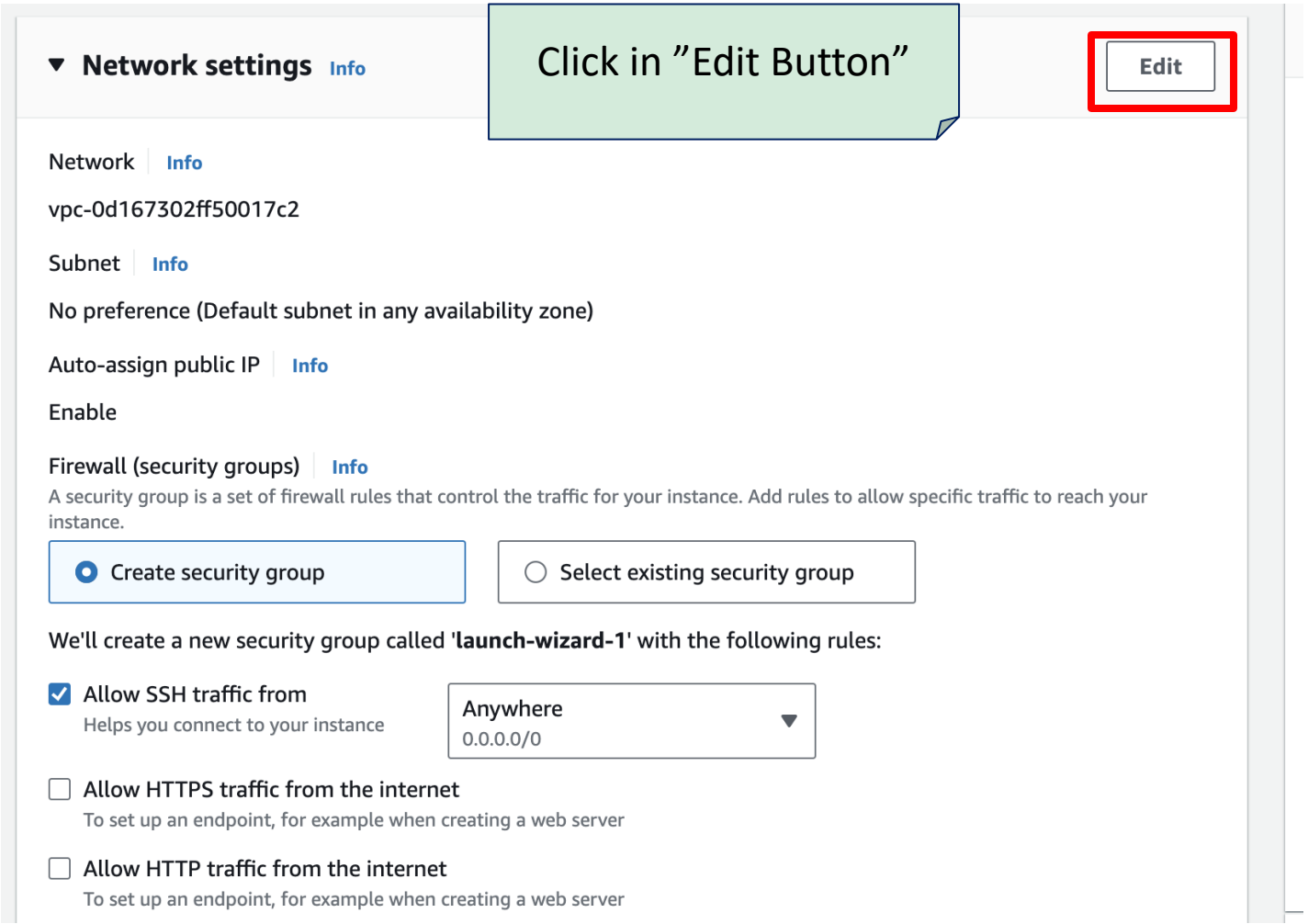
Short DEMO: Deploying Instances in the Subnet1 of our VPC

- We select "Launch Instance" in order to configure and deploy a compute instance.



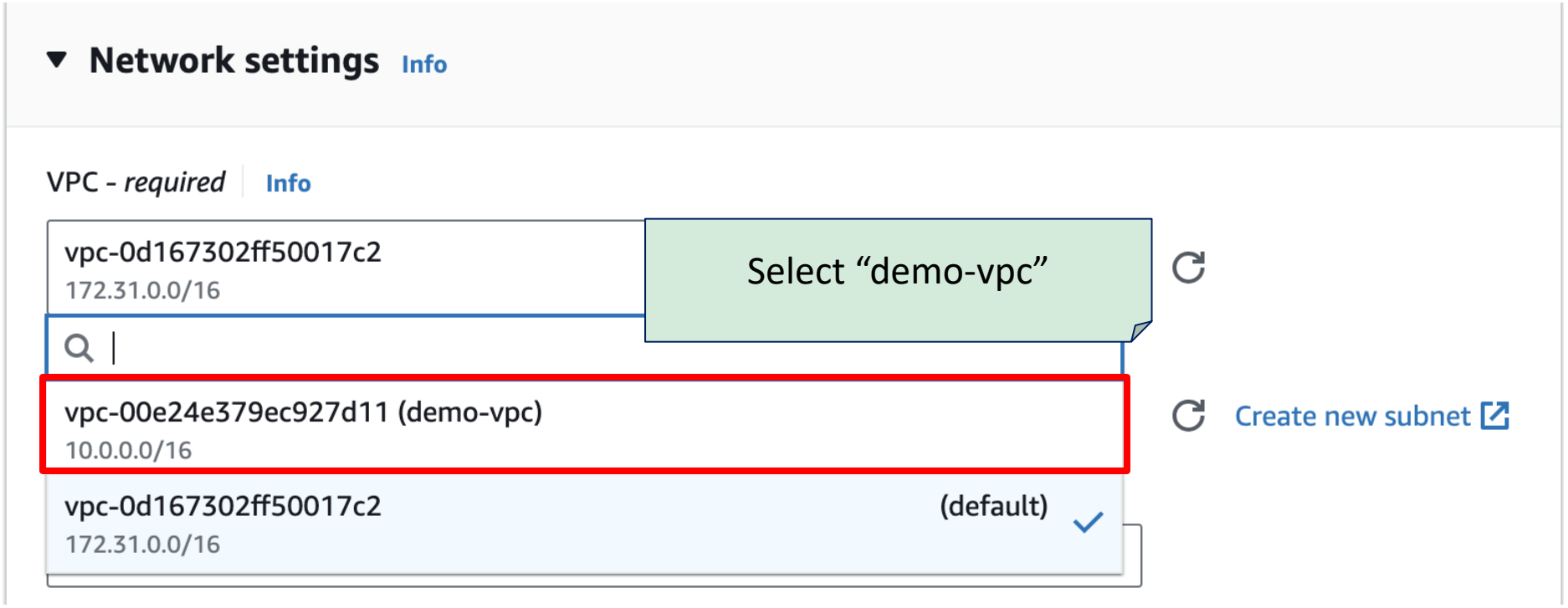
Short DEMO: Deploying Instances in the Subnet1 of our VPC

- In the “Network Configuration” menu we select "Edit" to select the VPC and the subnet where we want to deploy our instance.



Short DEMO: Deploying Instances in the Subnet1 of our VPC

- We select the VPC where we want to deploy the instance.



Short DEMO: Deploying Instances in the Subnet1 of our VPC

- We select the the subnet of the VPC where we want to deploy the instance.