

# 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

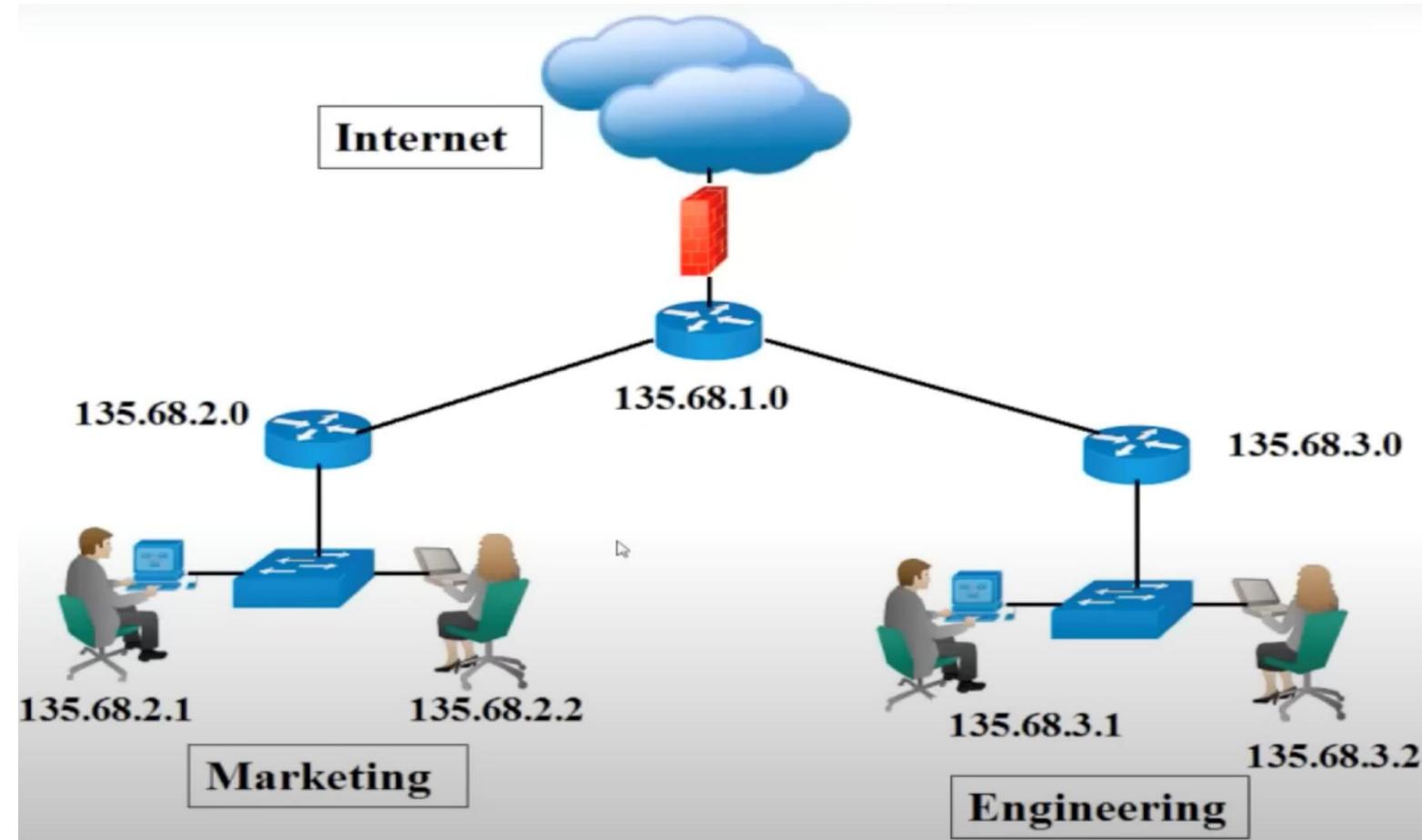


# Agenda

- **Introduction to networking**
- **AWS Global Infrastructure**
- **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.



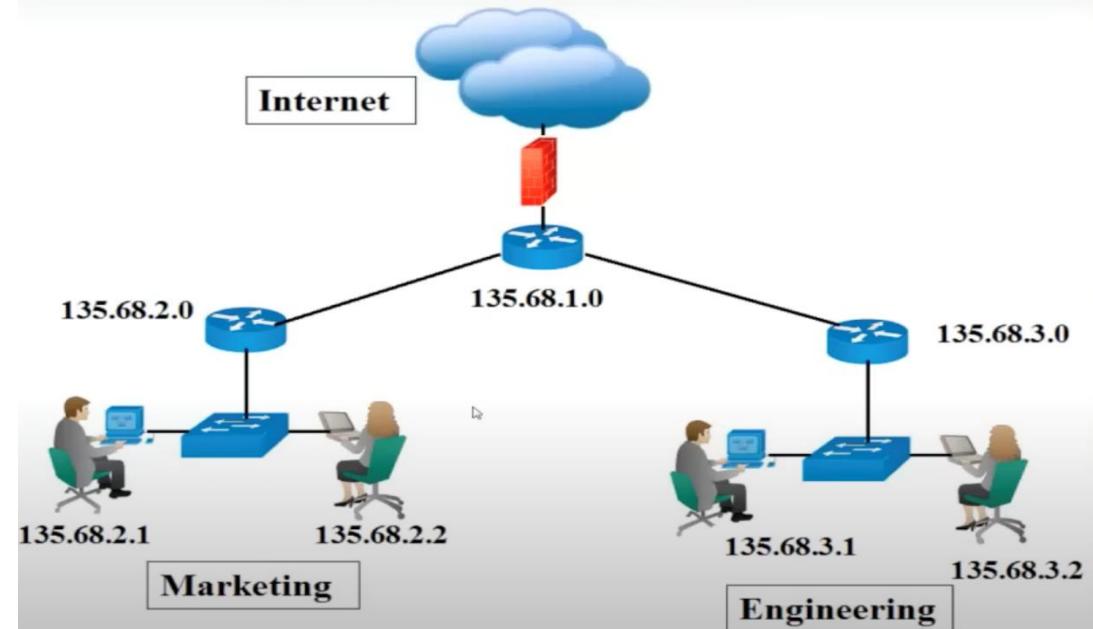
# Introduction to networking: Computer Network

How can we identify a specific machine within a subnet ?

# Introduction to networking: IP

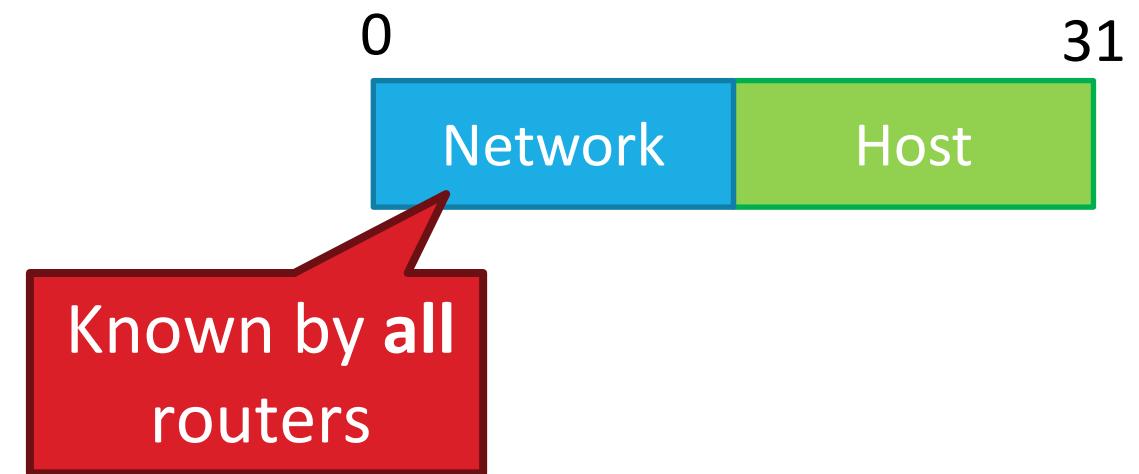
- Each machine in a network has a unique **Internet Protocol (IP) address** that identifies it.
- An **IP** address is a **numerical label** in decimal format. Machines convert that decimal number to a binary format.
- It consists of a series of four numbers separated by dots, for example, 192.168.21.76. 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 Address**.
- All the IP addresses of computers on the same network start with the same number to identify the network and the host (computer).

	0	8	16	24	31
Decimal	192	168	21	76	
Hex	C0	A8	15	4C	
Binary	11000000	10101000	00010101	01001100	



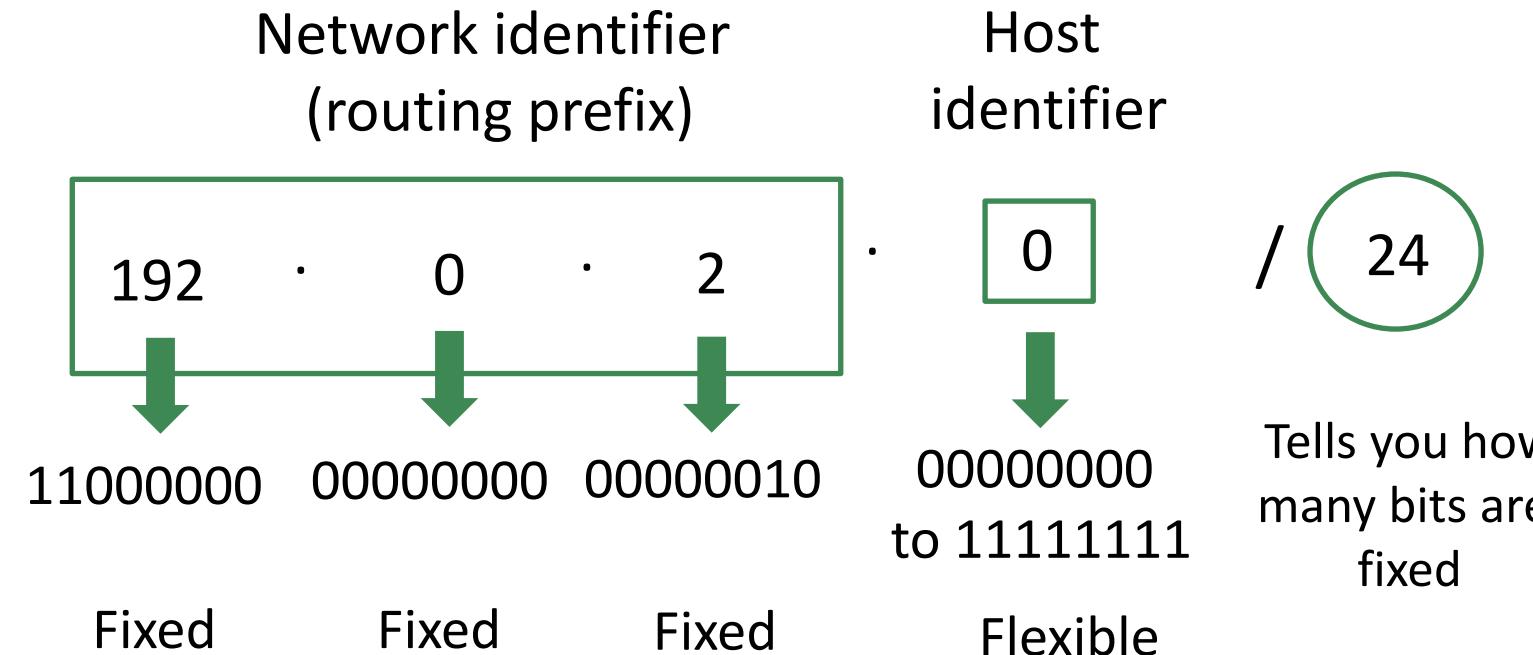
# Introduction to networking: IP

- 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.



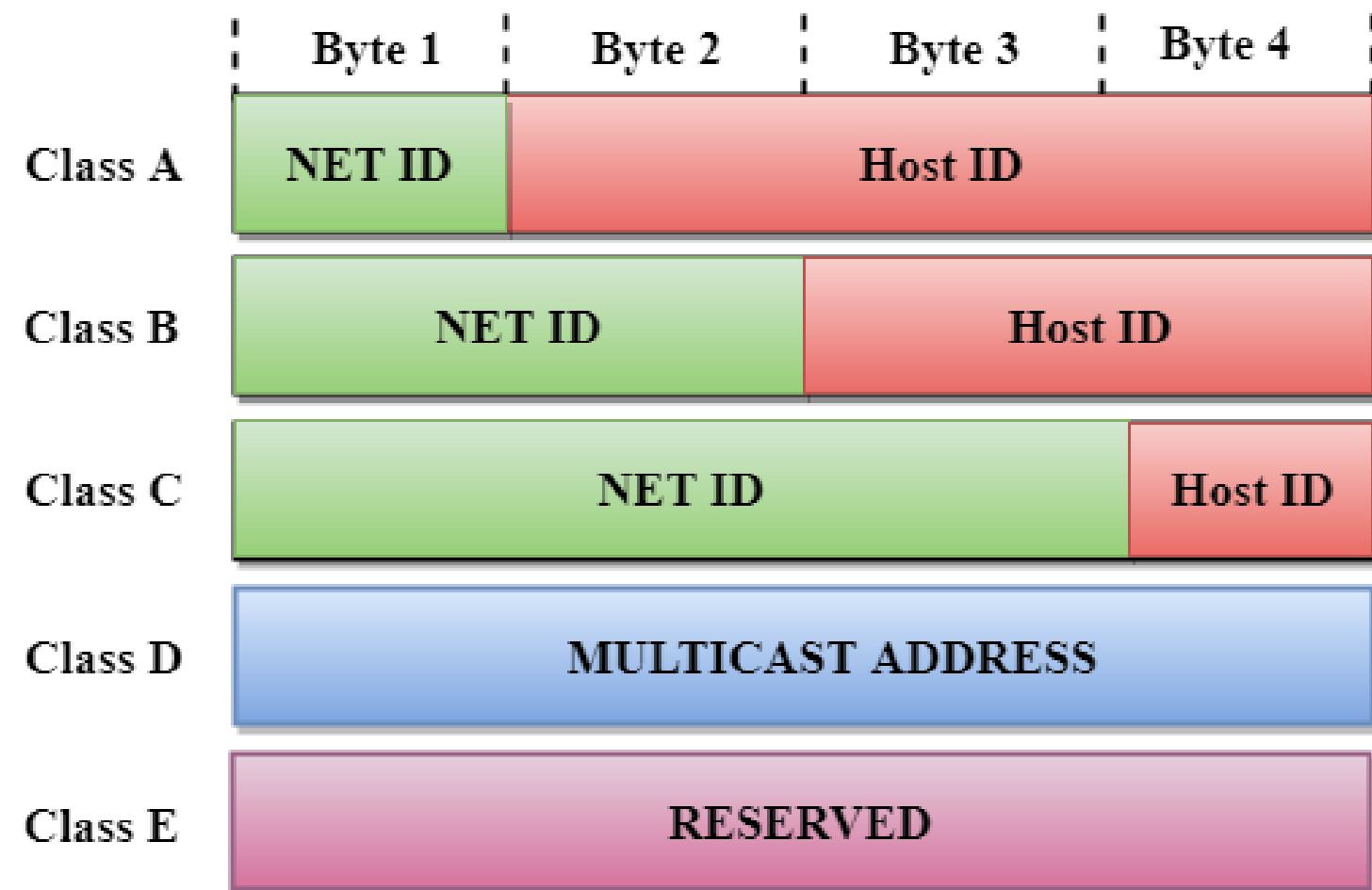
# Introduction to networking: Classless Inter-Domain Routing (CIDR)

- 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

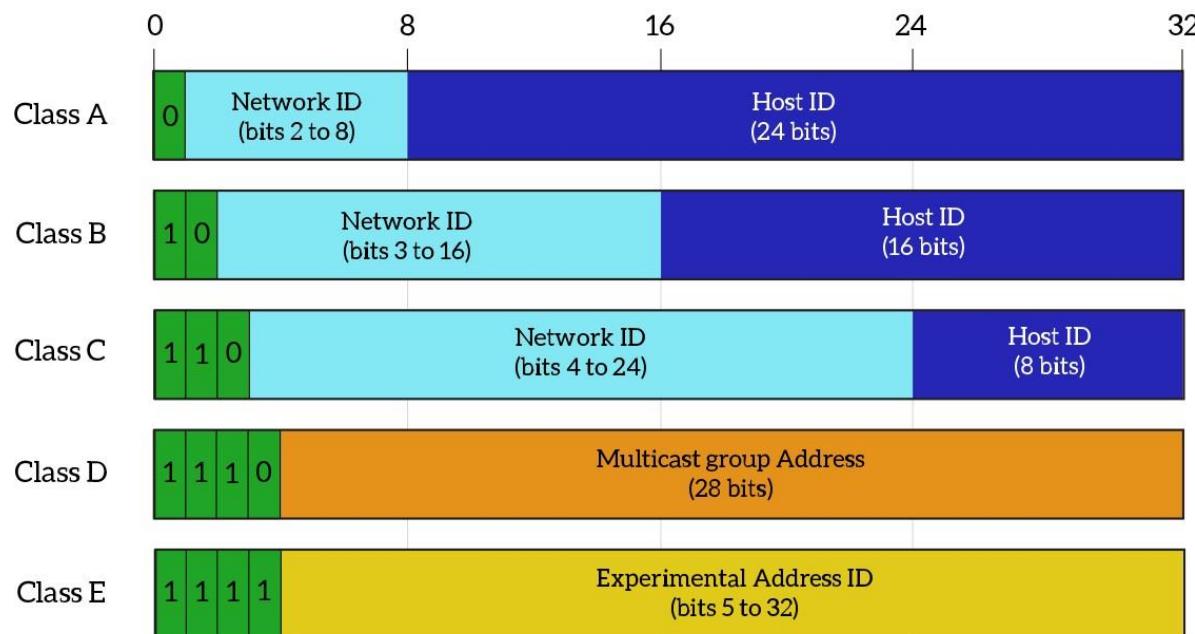


# Introduction to networking: IP Classes

- 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 medium 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

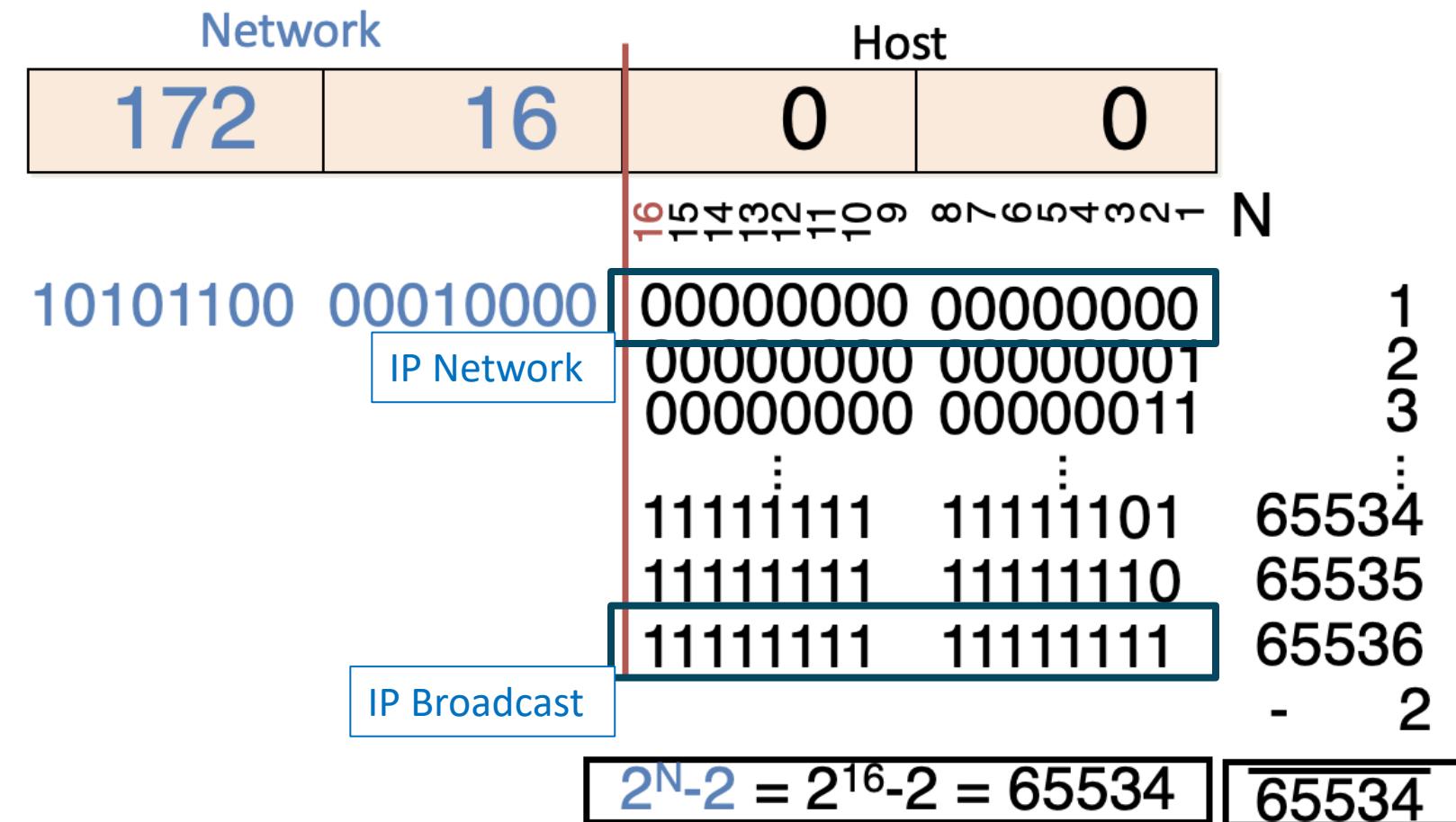


Address Class	RANGE	Default Subnet Mask
A	1.0.0.0 to 126.255.255.255	255.0.0.0
B	128.0.0.0 to 191.255.255.255	255.255.0.0
C	192.0.0.0 to 223.255.255.255	255.255.255.0
D	224.0.0.0 to 239.255.255.255	Reserved for Multicasting
E	240.0.0.0 to 254.255.255.255	Experimental

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 addresses

- 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.

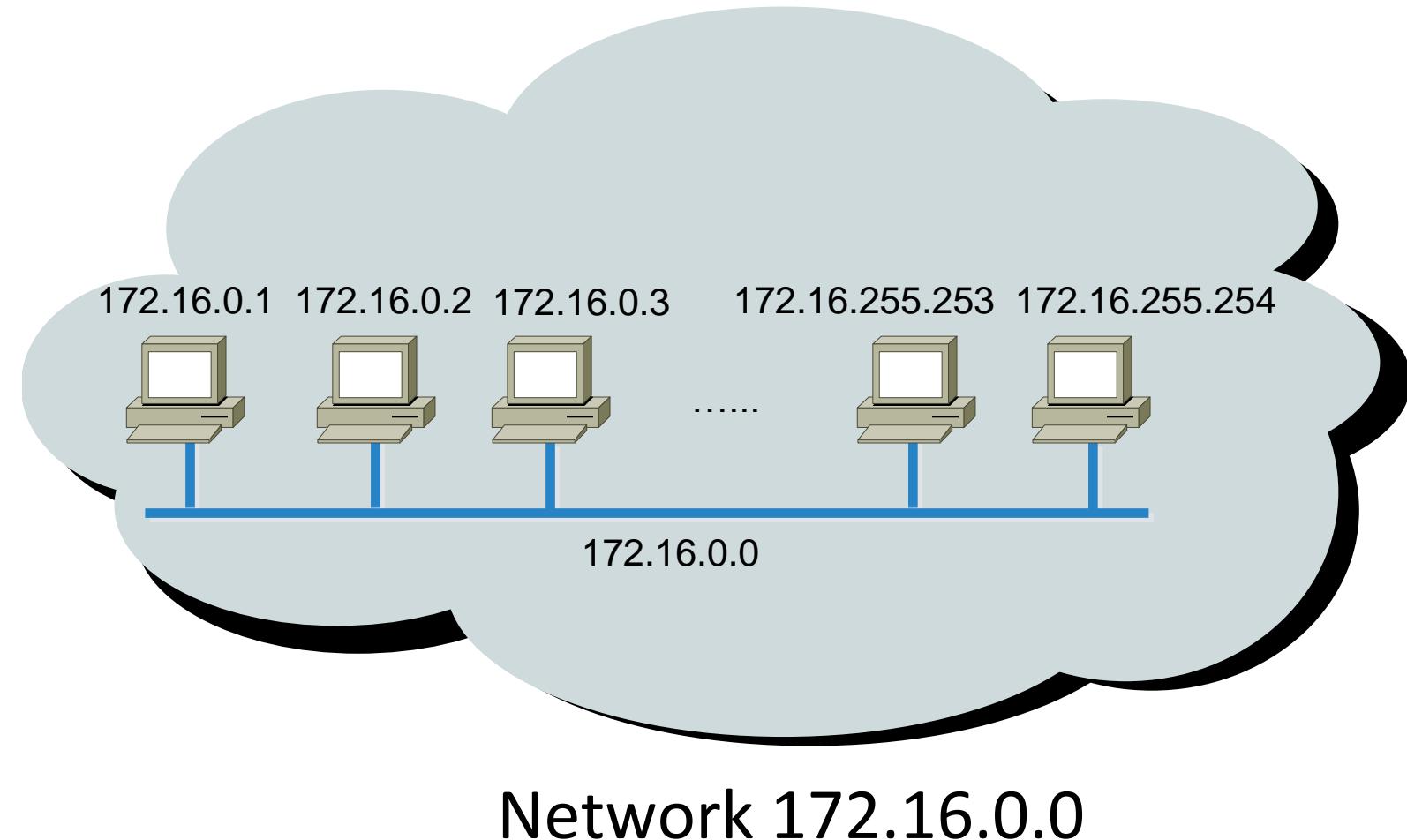


# Introduction to networking: Host addresses

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				

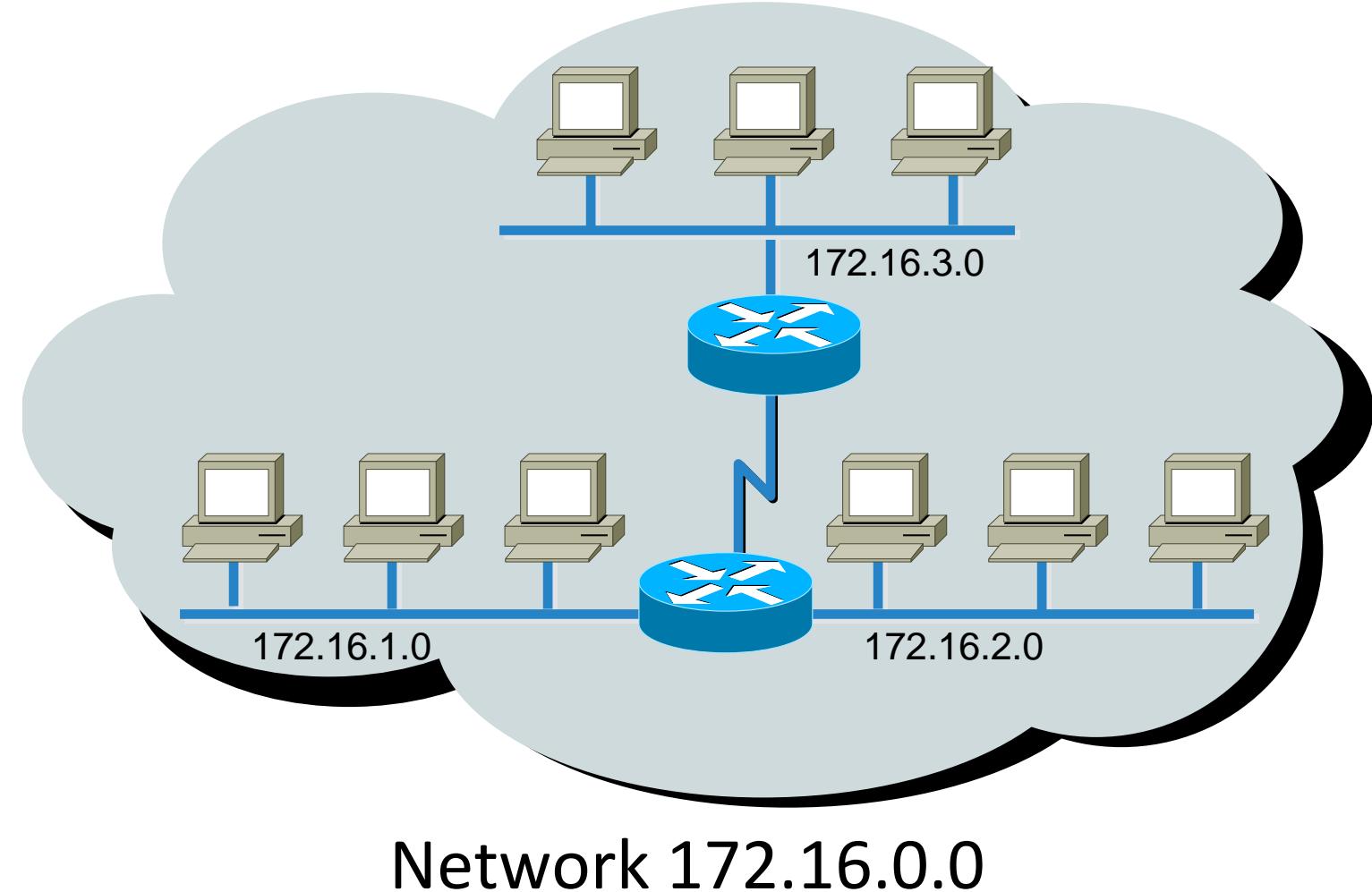
# Introduction to networking: Addressing without subnets

- 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).



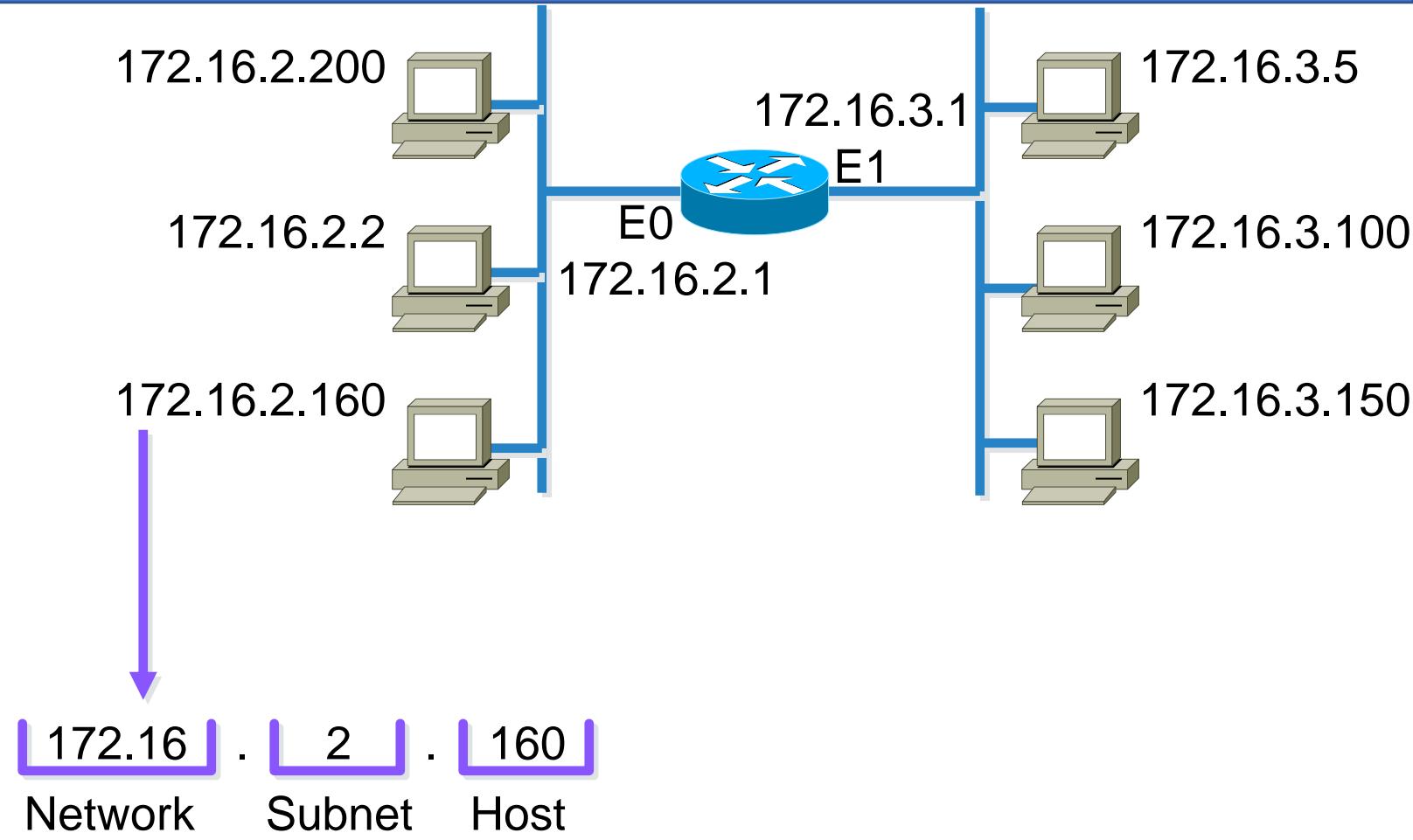
# Introduction to networking: Addressing with subnets

- 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



# 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

	Network		Host	
172.16.2.160	10101100	00010000	00000010	10100000
255.255.0.0	11111111	11111111	00000000	00000000
	10101100	00010000	00000000	00000000

Logical "And" Operation

Network Number	172	16	0	0
----------------	-----	----	---	---

# 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
255.255.255.0	11111111	11111111
	10101100	00010000

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

172.16.2.160

255.255.255.192

Network

Subnet

Host

10101100

00010000

00000010

10100000

11111111

11111111

11111111

11000000

10101100

00010000

00000010

10000000

128  
192  
224  
240  
248  
252  
254  
255

255  
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/>

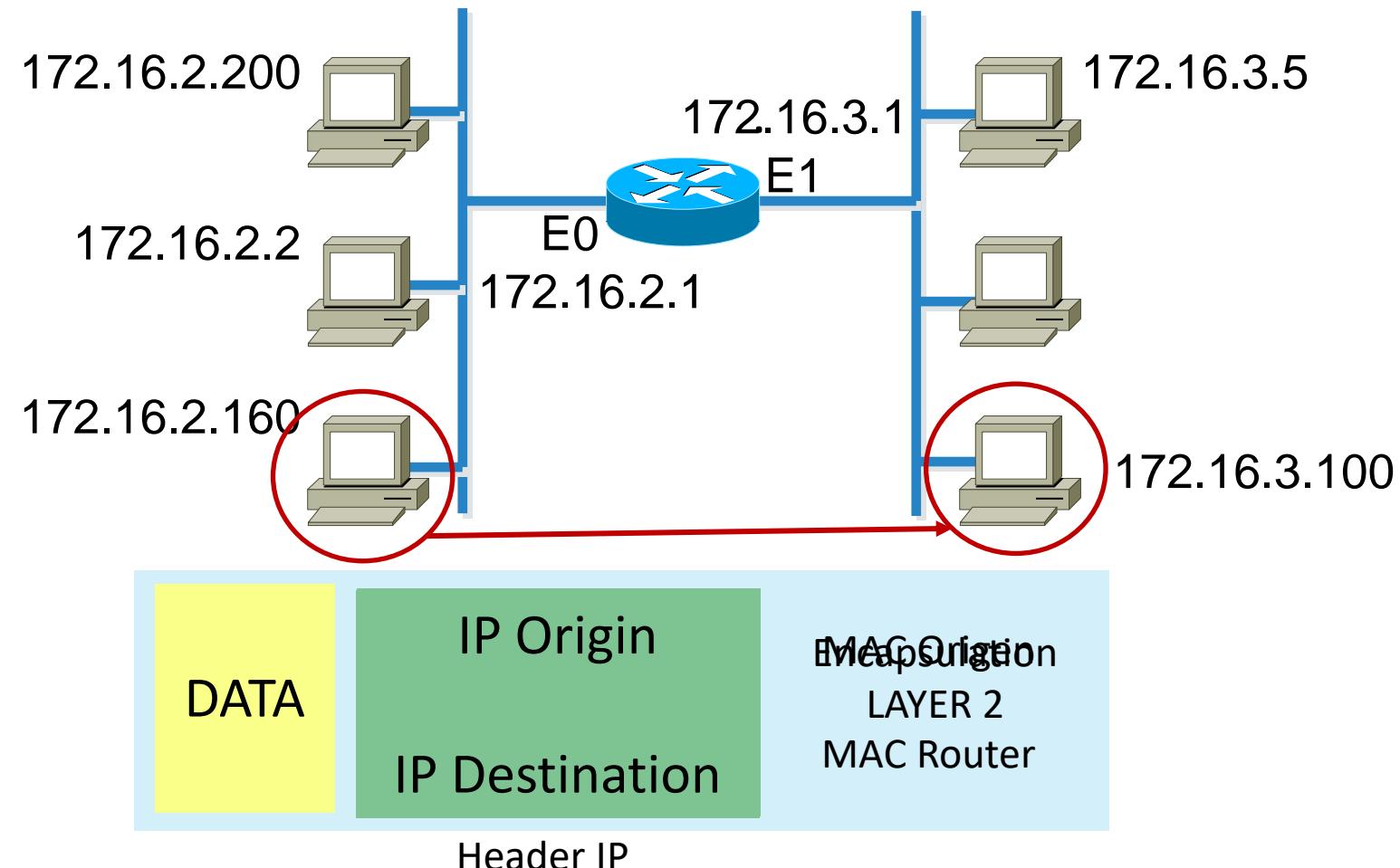
- **Extract network:**  
We apply an “and” logical operation between the IP and the Mask

	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

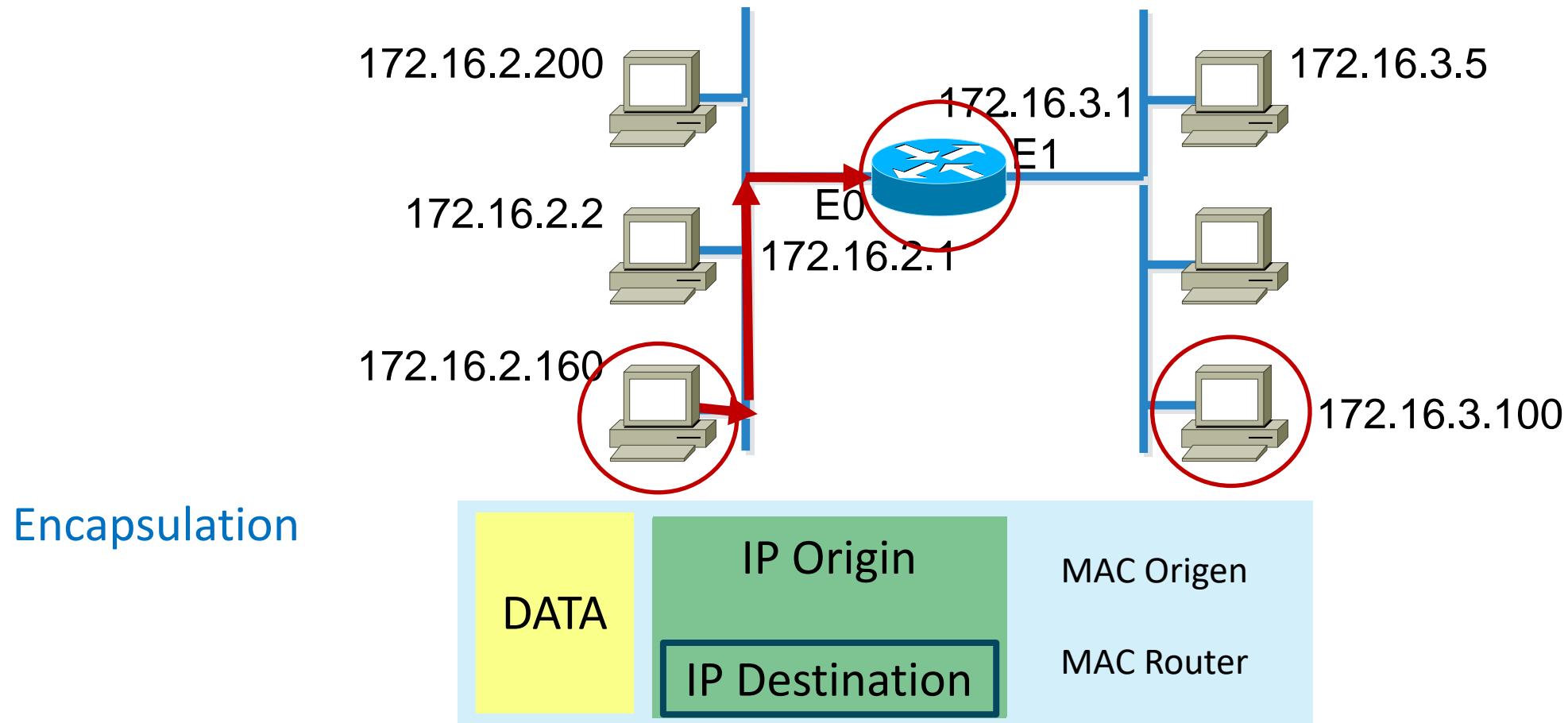
# Introduction to networking: Open Systems Interconnection (OSI) model

Layer	Number	Function	Protocol/Address
Application	7	Means for an application to access a computer network	HTTP(S), FTP, DHCP, LDAP
Presentation	6	<ul style="list-style-type: none"><li>Ensures that the application layer can read the data</li><li>Encryption</li></ul>	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)

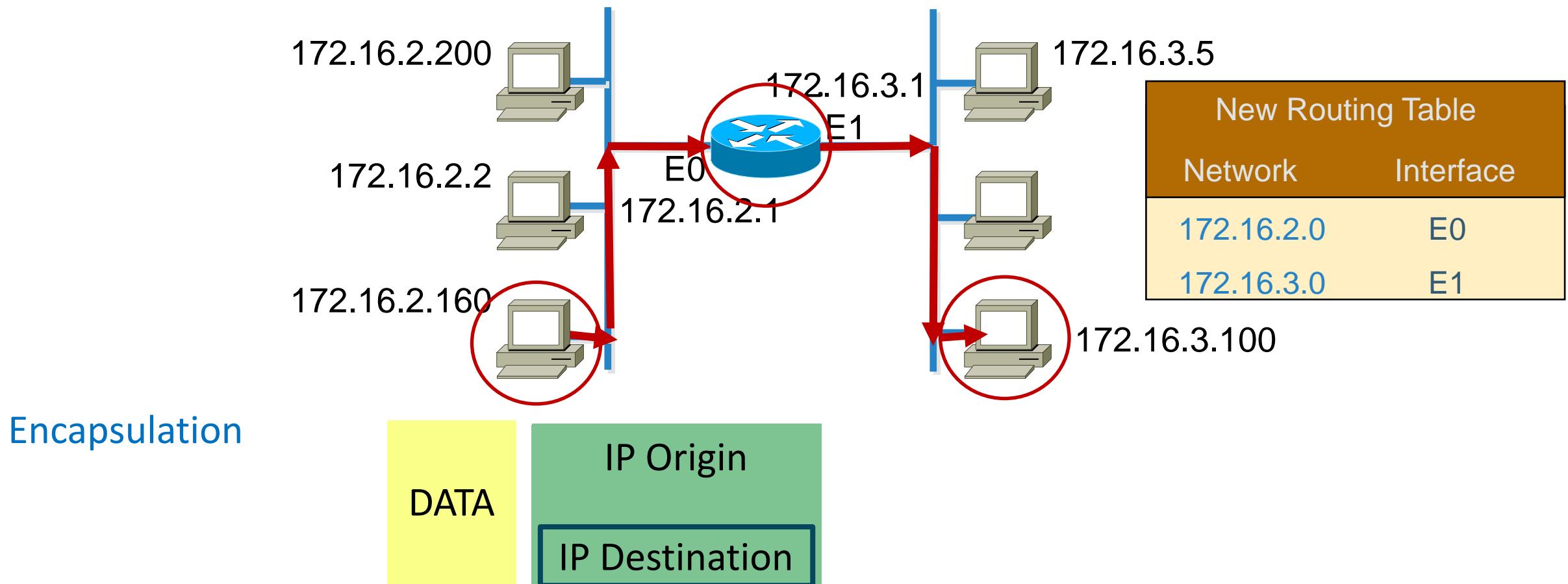
# Introduction to networking: Open Systems Interconnection (OSI) model



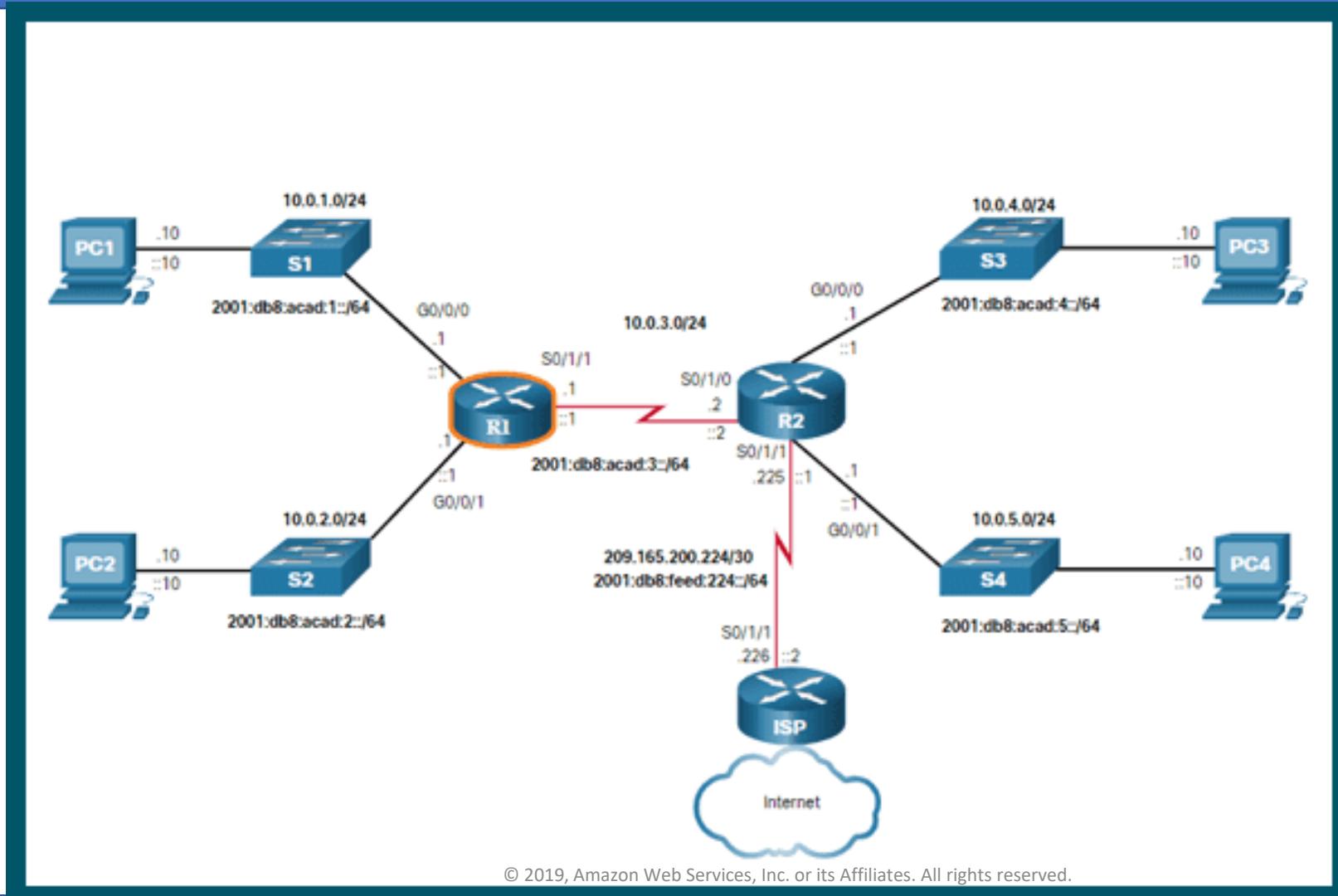
# Introduction to networking: Open Systems Interconnection (OSI) model



# Introduction to networking: Open Systems Interconnection (OSI) model



# Introduction to networking: Open Systems Interconnection (OSI) model



# AWS Global infrastructure

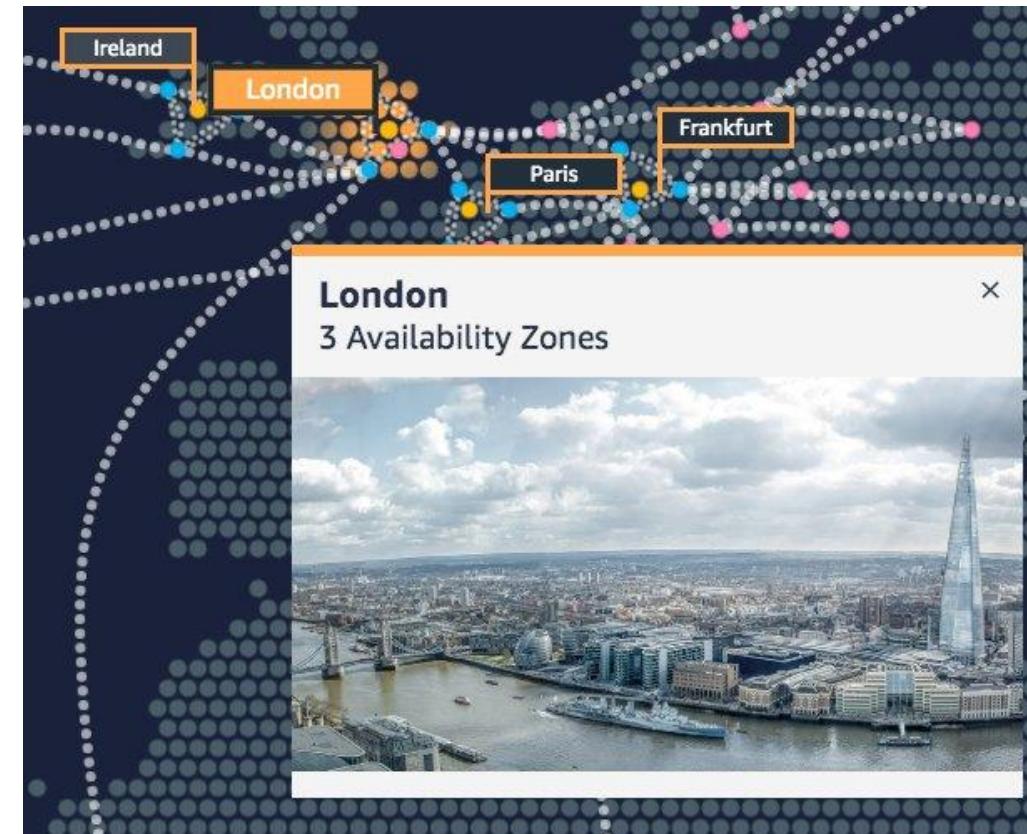
- The **AWS Global Infrastructure** is designed and built to deliver a **flexible, reliable, scalable, and secure** cloud computing environment with high-quality **global network performance**.
- This map from <https://infrastructure.aws> shows the current **AWS Regions** and more that are coming soon.



# AWS Global infrastructure

## Regions

- AWS [regions](#) are geographical areas.
- 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 controlled 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



Data governance, legal requirements



Proximity to customers (latency)



Services available within the Region

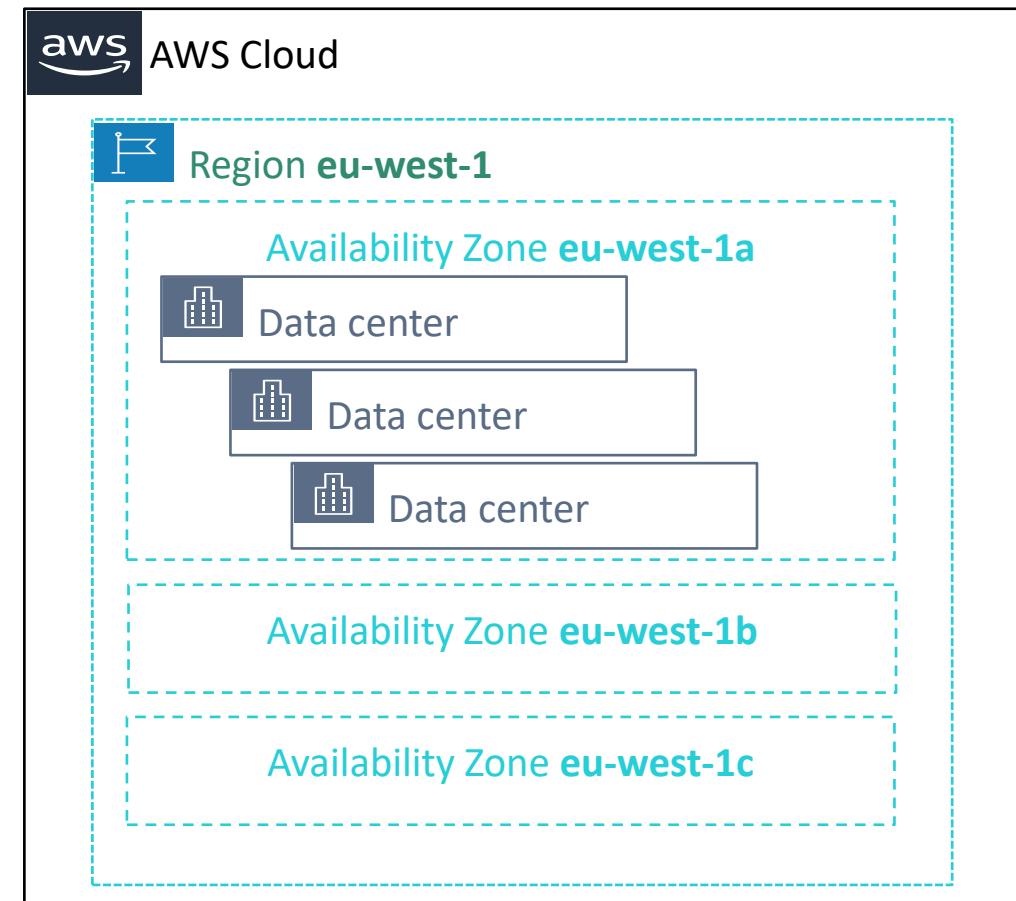


Costs (vary by Region)

# AWS Global infrastructure

## Availability Zones

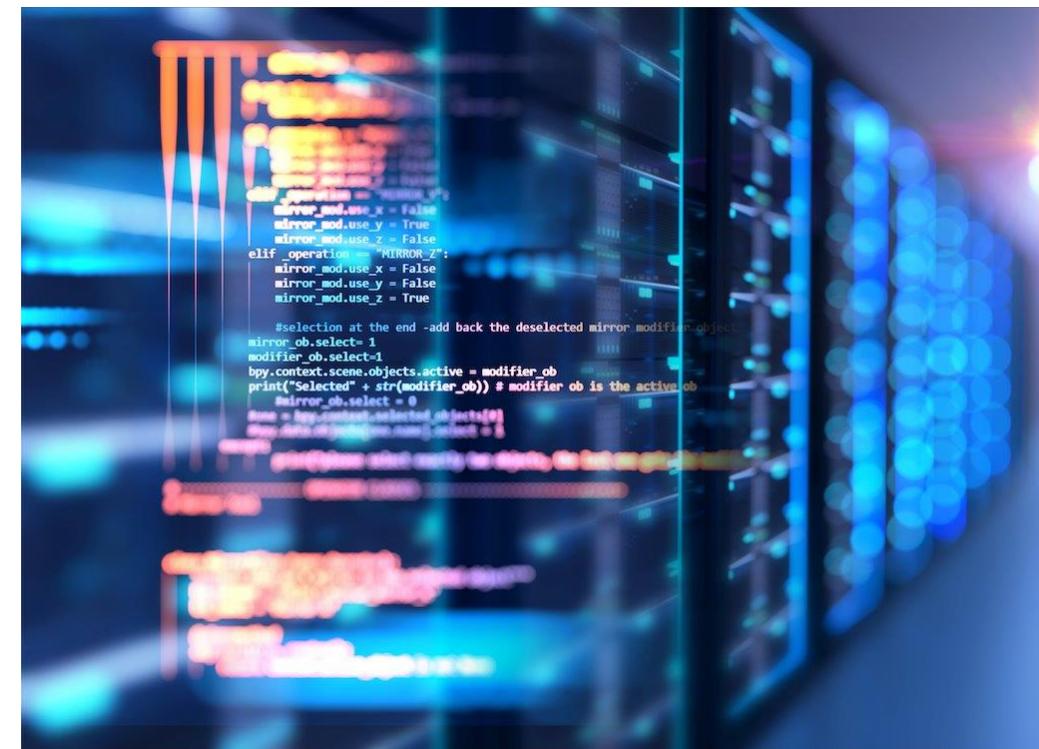
- Each Region has multiple Availability Zones.
- Each **Availability Zone** is a fully isolated partition of the AWS infrastructure.
  - Availability Zones consist of discrete **data centers**
  - They are designed for fault isolation
  - They are interconnected with other Availability Zones by using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



# AWS Global infrastructure

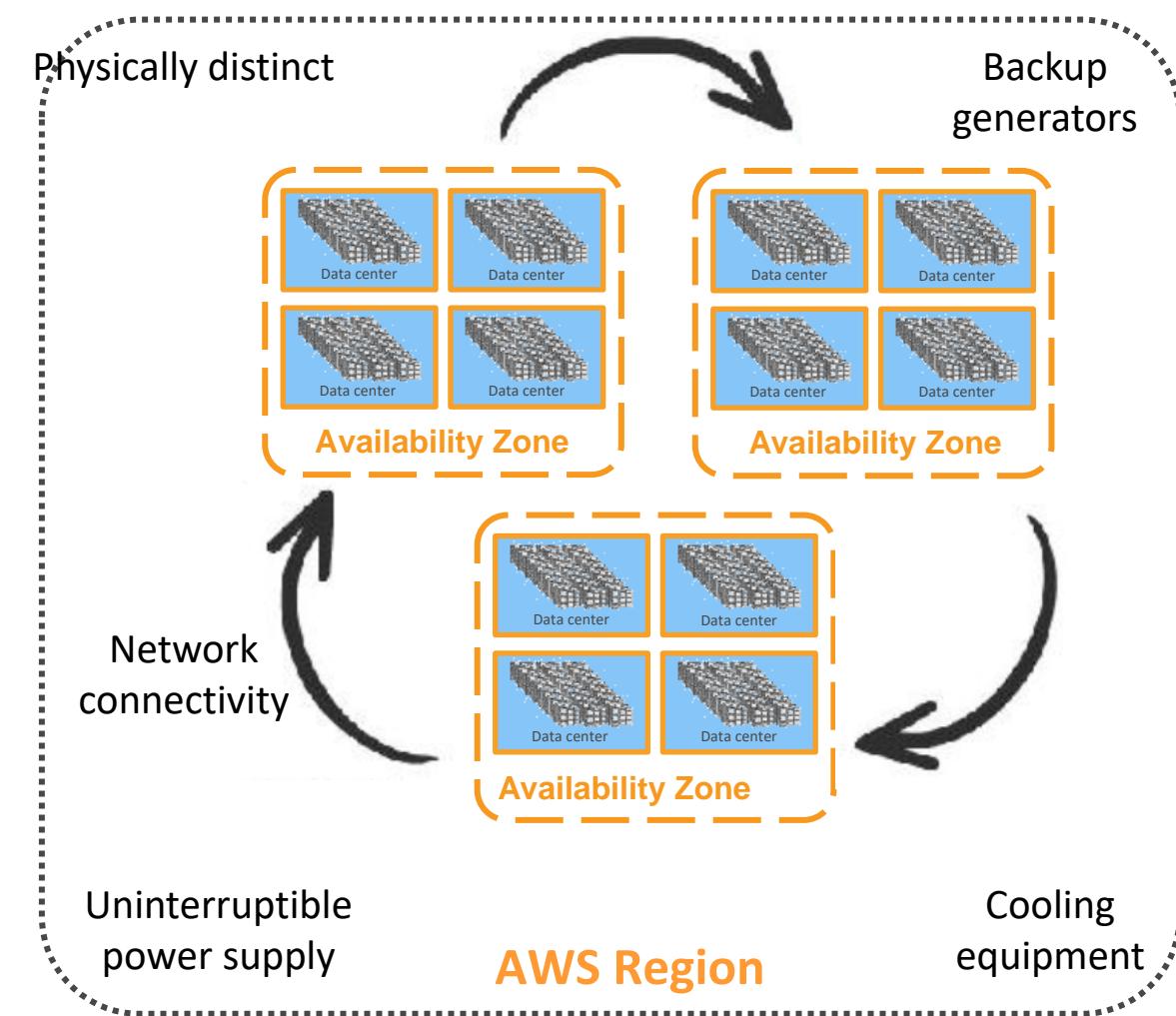
## AWS data centers

- 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



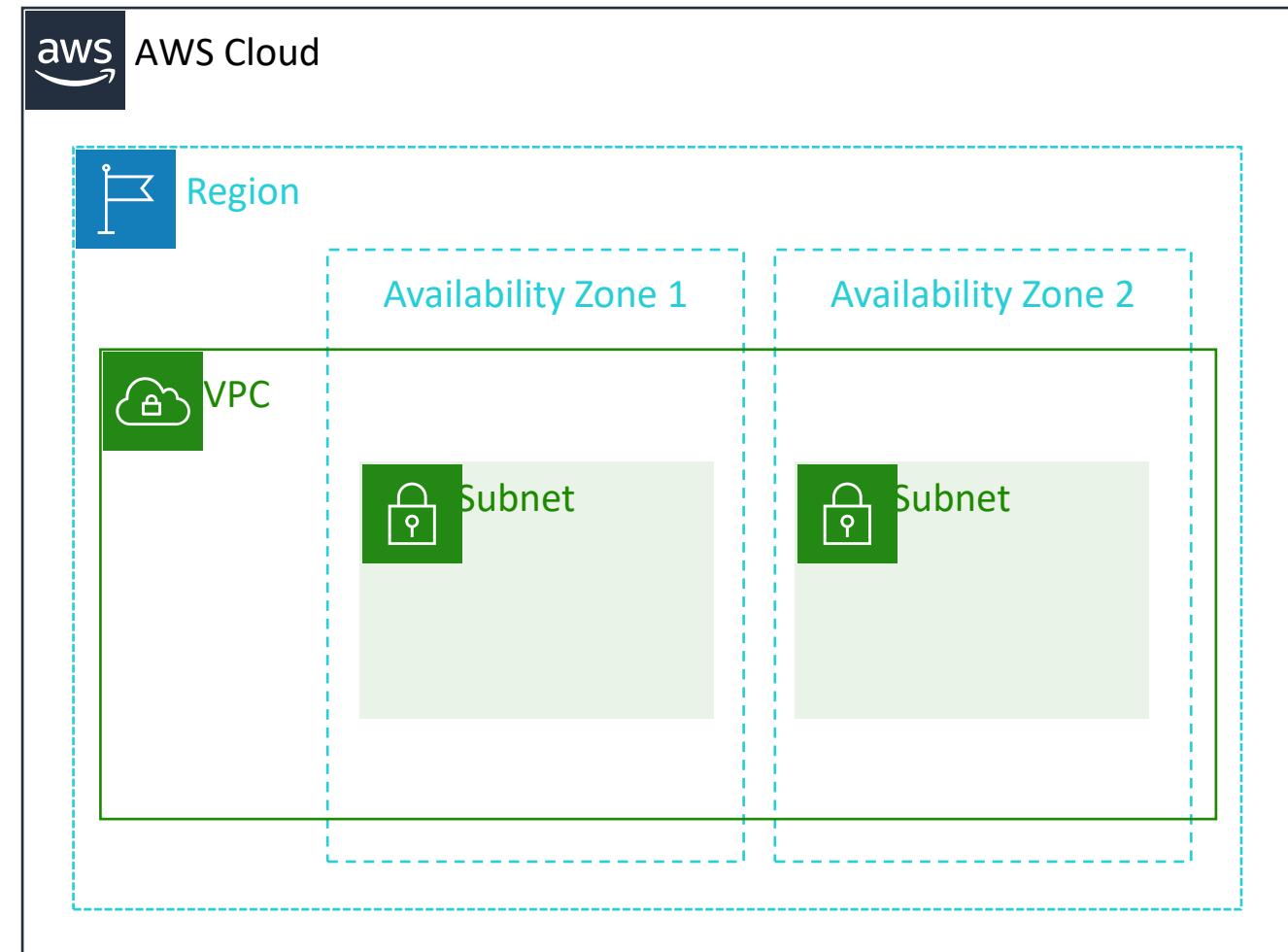
# Virtual Private Clouds (VPC)

- 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



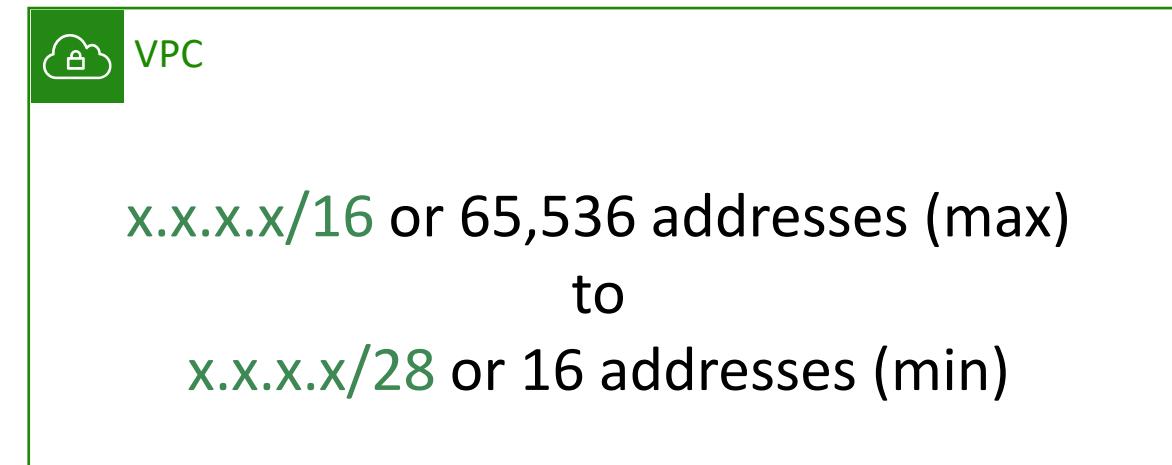
# VPCs and subnets

- **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

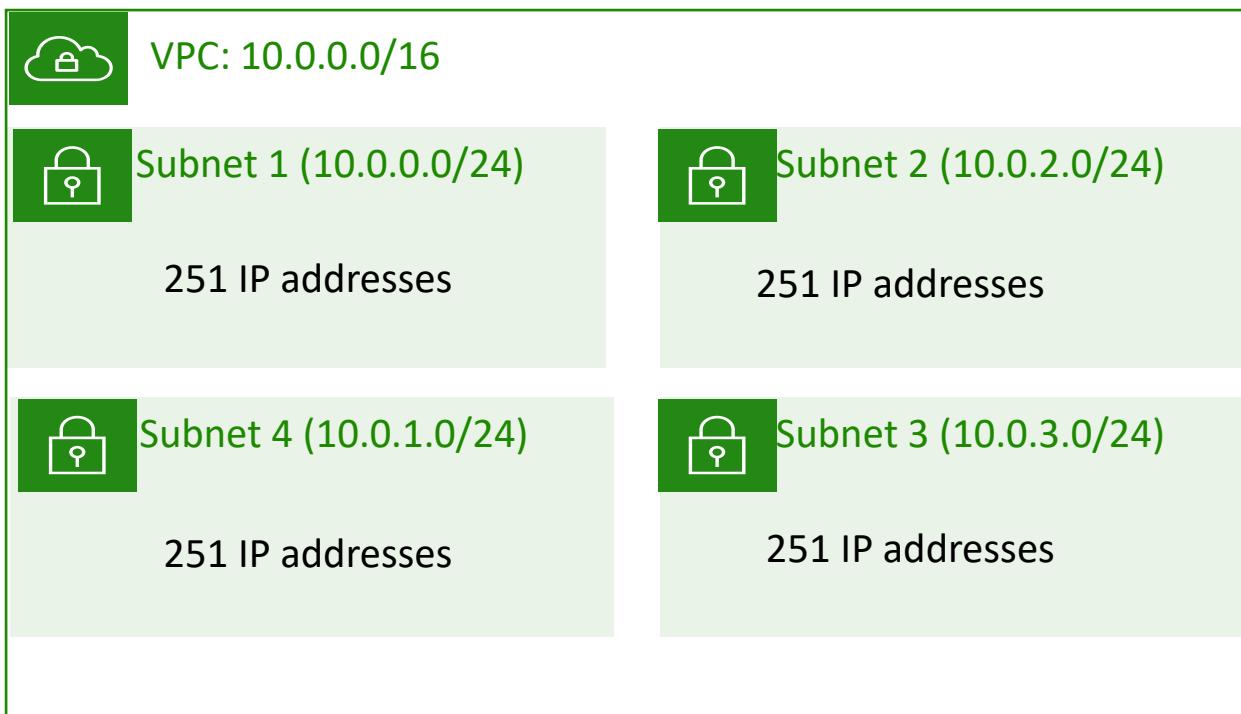
Subnet 2: **10.0.2.0/24** -> 256 addresses

Subnet 3: **10.0.3.0/24** -> 256 addresses

Subnet 4: **10.0.4.0/24** -> 256 addresses

# VPCs and subnets: Example

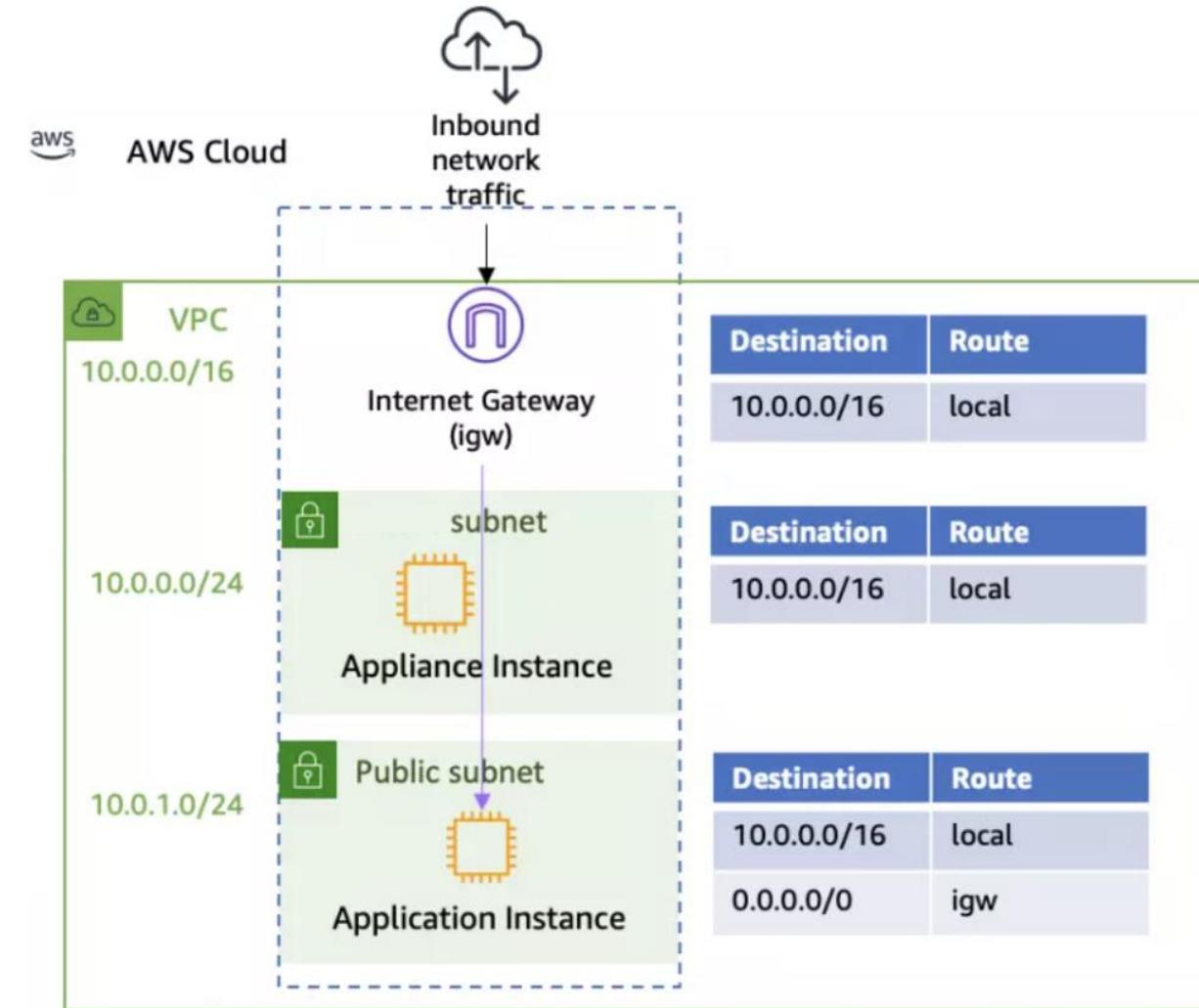
**Example:** A VPC with an IPv4 CIDR block of 10.0.0.0/16 has 65,536 total IP addresses. The VPC has four equal-sized subnets. Only 251 IP addresses are available for use by each subnet.



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.

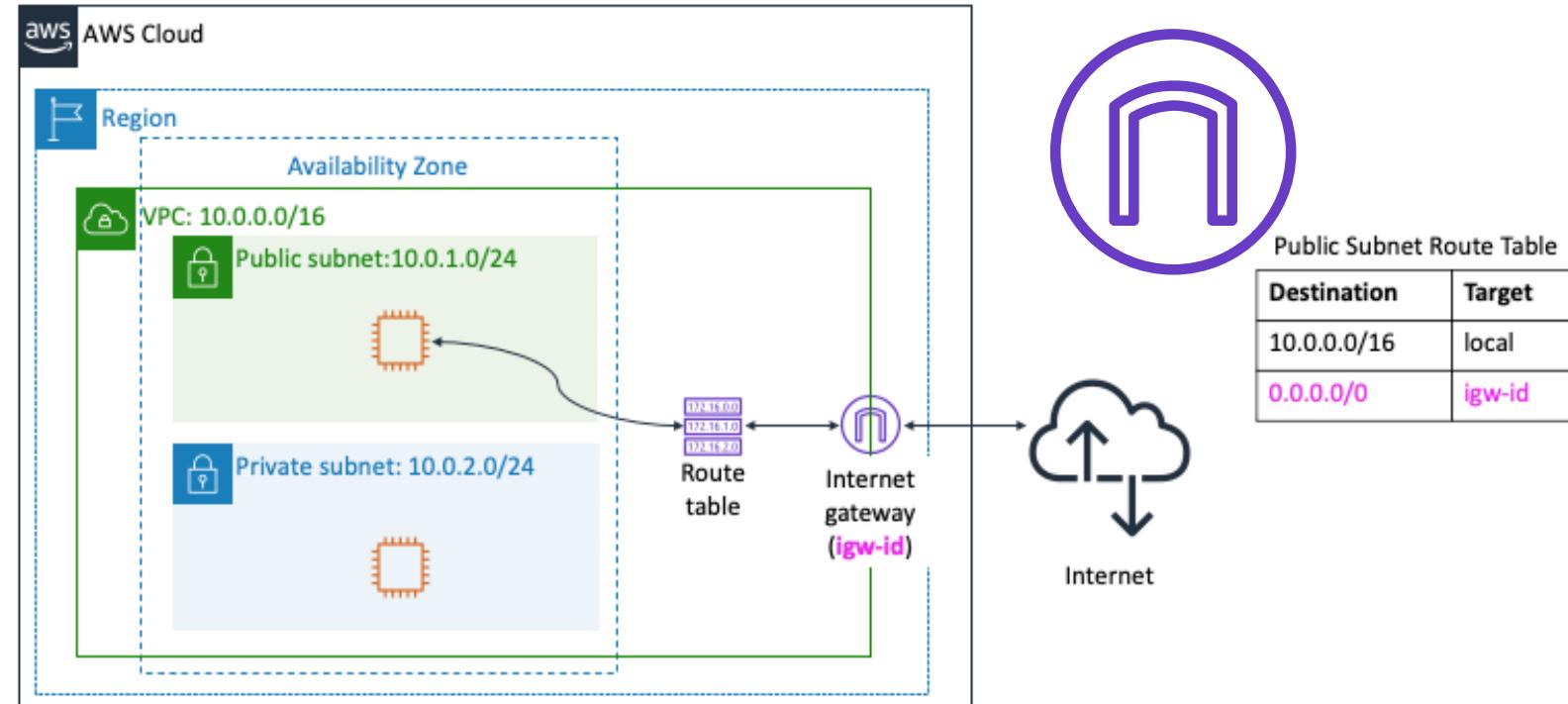


# Route tables and routes

- 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 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

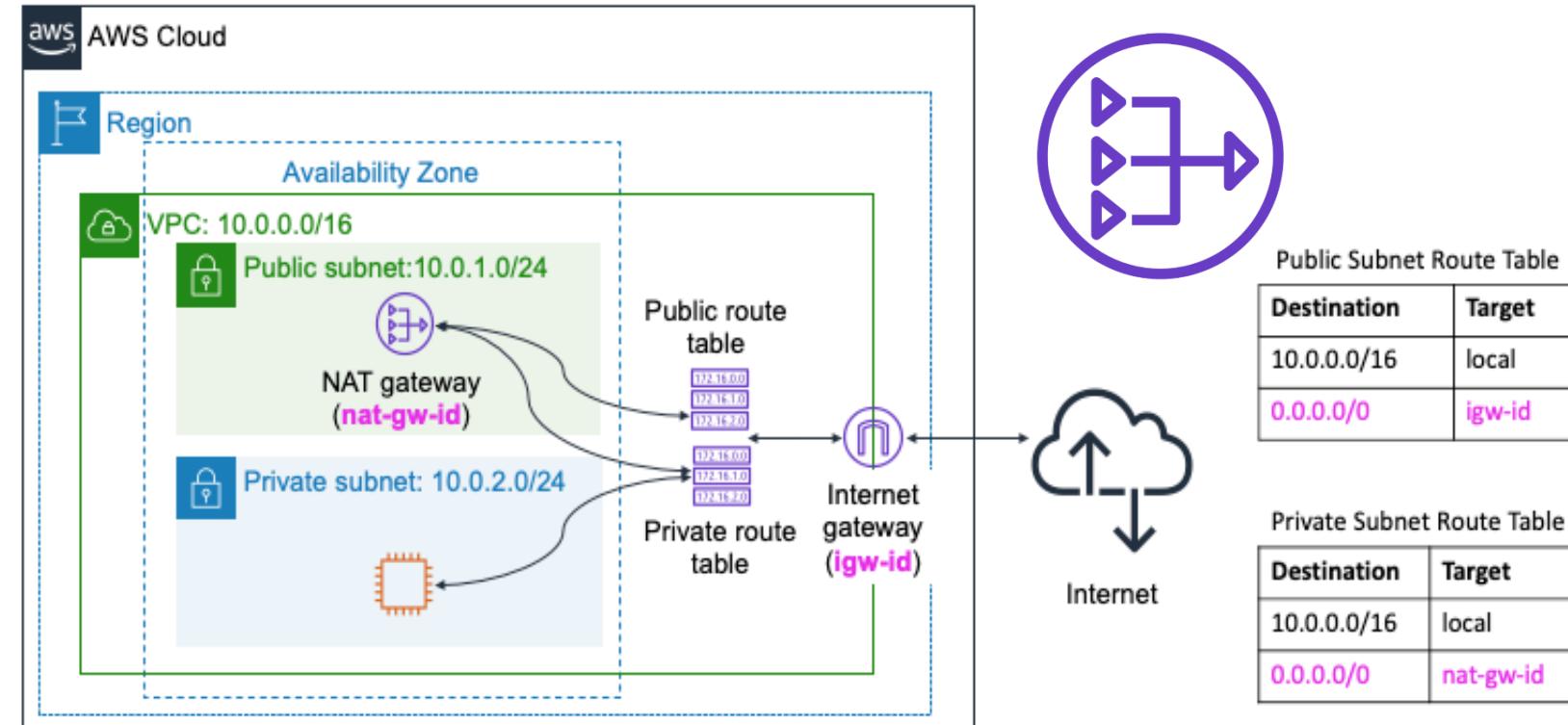
- An [internet gateway](#) is a scalable, redundant, and highly available VPC component that allows communication between instances in your VPC and internet.
- It is seen as a VPC level component.
- An internet gateway serves two purposes:
  - Provide a target in your VPC route tables for internet-routable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



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



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

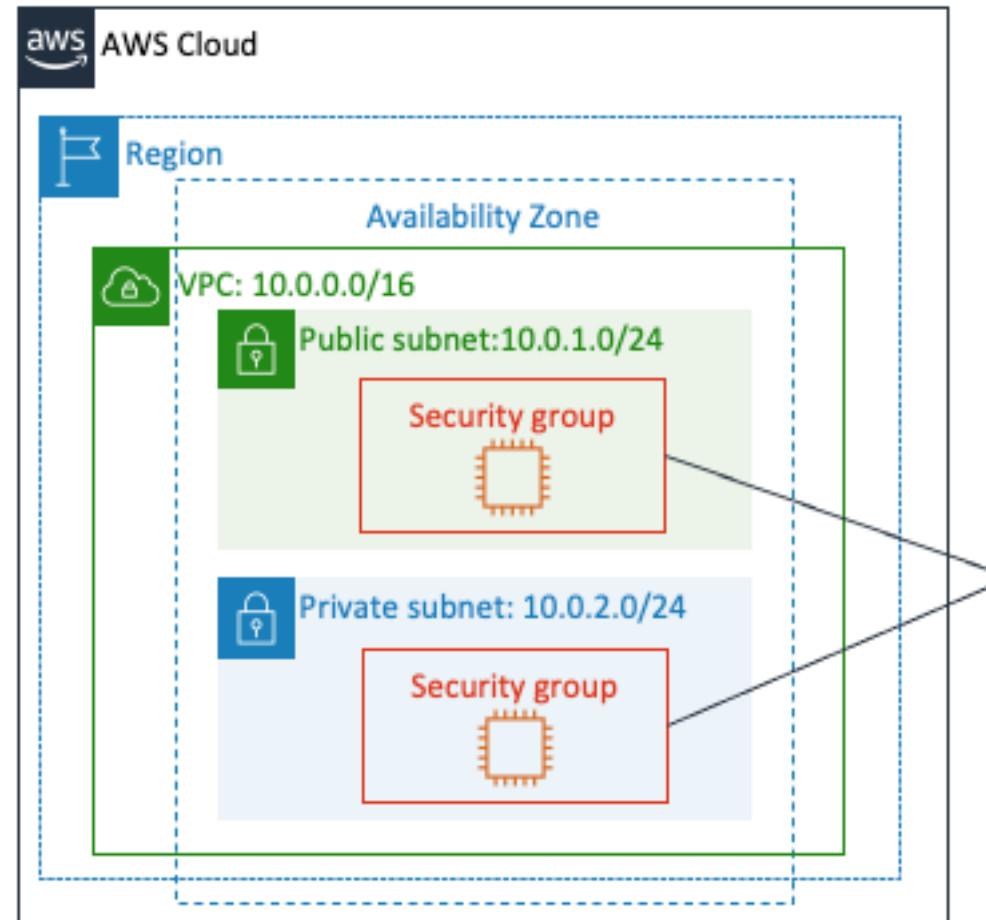
## Elastic IP

- 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 an Elastic IP.
- An Elastic IP is a public IPv4 you own as long as you do not delete it.
- You can attach it to one instance at a 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 act at the **instance level**.

# 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.

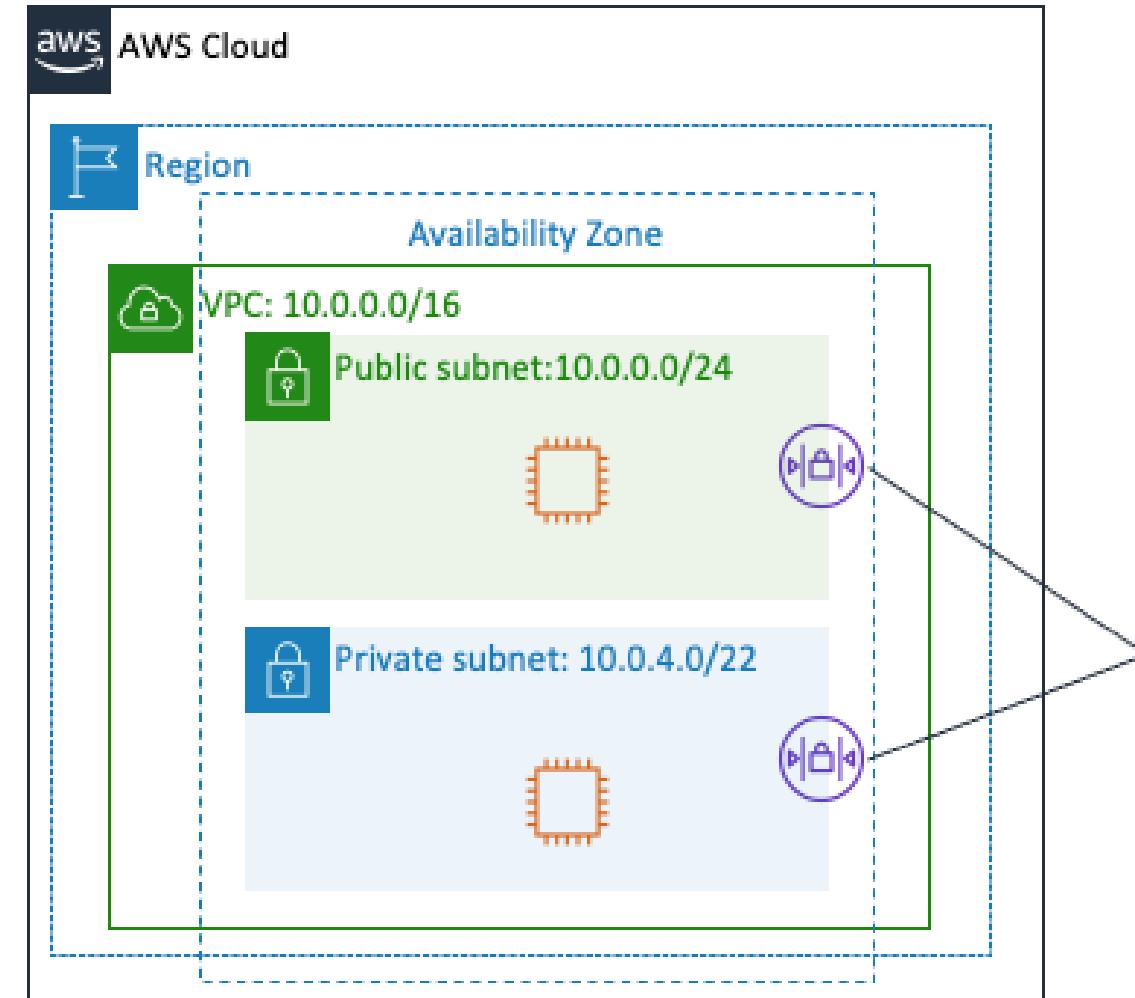
# Security Groups

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 act at **subnet level**.**

# 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

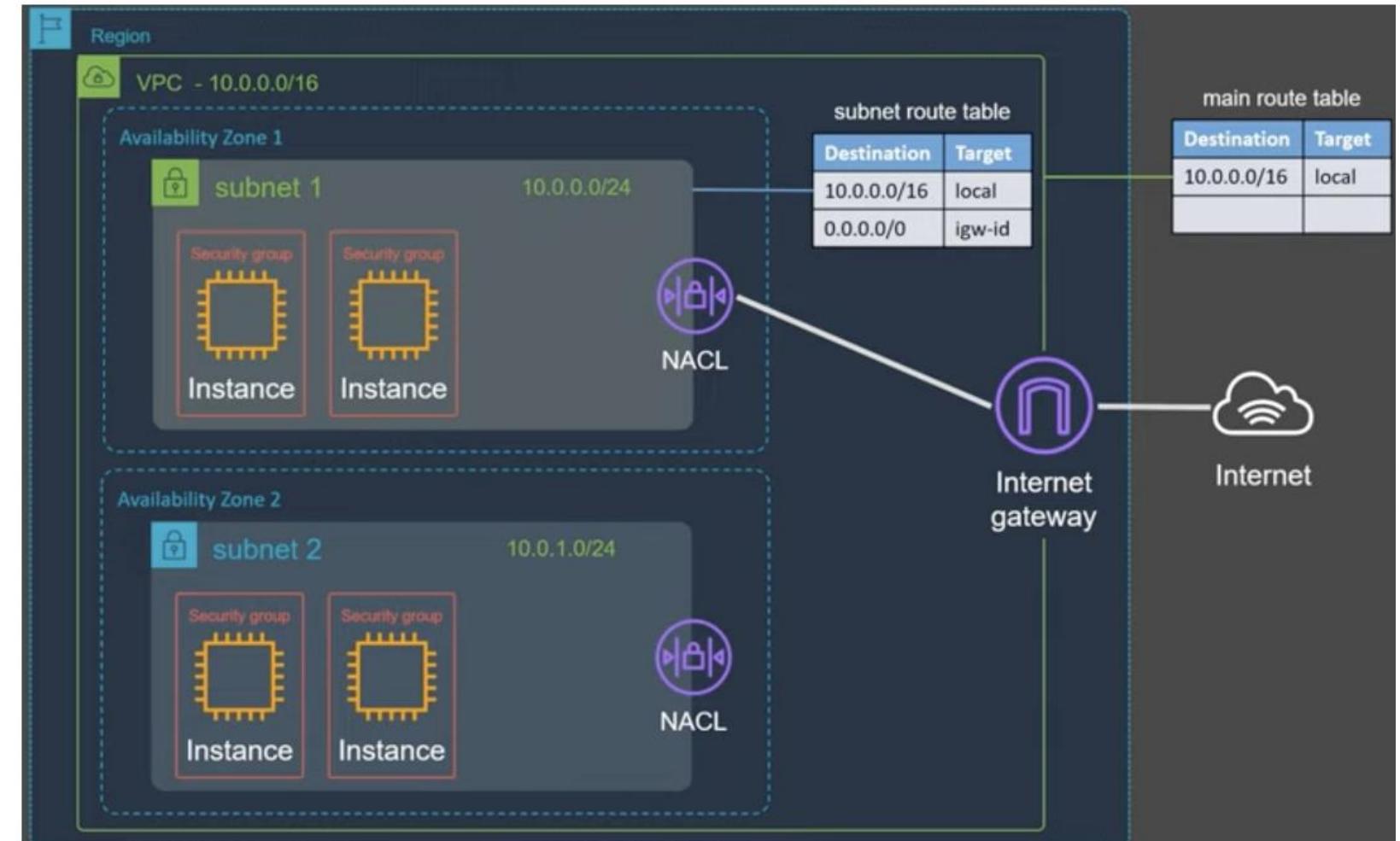
Attribute	Security Groups	Network ACLs
Scope	Instance level	Subnet level
Supported Rules	Allow rules only	Allow and deny rules
State	Stateful (return traffic is automatically allowed, regardless of rules)	Stateless (return traffic must be explicitly allowed by rules)
Order of Rules	All rules are evaluated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

1. Write "VPC"

A screenshot of the AWS search interface. The search bar at the top contains the text 'vpc'. Below the search bar, the results are displayed under the heading 'Search results for 'vpc''.

**Services** (12)

- VPC ★ Isolated Cloud Resources
- AWS Firewall Manager ★ Central management of firewall rules
- Detective ★ Investigate and Analyze potential security issues
- Managed Services ★ IT operations management for AWS

**Features** (57)

- Dashboard VPC feature
- Route 53 VPCs

See all 12 results ►

2. Select "VPC"

N. Virginia vclabs/user2802835

Default layout Create

Cost (USD)

# Short DEMO: Creating a VPC

Select the button "Create VPC"

The screenshot shows the AWS VPC service console. At the top, there's a navigation bar with the AWS logo, a search bar, and a "Create VPC" button which is highlighted with a red box. Below the navigation bar, there's a note: "Note: Your Instances will launch in the US East region." On the left, there's a sidebar titled "Filter by VPC:" with a dropdown menu set to "Select a VPC". Underneath, there's a list of options: Virtual private cloud, Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, Endpoint services, NAT gateways, and Peering connections. The main content area is titled "Resources by Region" and shows the following resource counts in the US East region: VPCs (1), Subnets (6), Route Tables (1), Internet Gateways (1), Egress-only Internet Gateways (0), NAT Gateways (0), VPC Peering Connections (0), Network ACLs (1), Security Groups (1), and Customer Gateways (0). To the right, there are sections for "Service Health" (with a link to "View complete service health"), "Settings" (with links to "Zones" and "Console Experiments"), "Additional Information" (with links to "VPC Documentation", "All VPC Resources", "Forums", and "Report an Issue"), "AWS Network Manager" (with a brief description and a link to "Get started with Network Manager"), and "Site-to-Site VPN Connect".

# Short DEMO: Creating a VPC

**Select “VPC only”**

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances. Mouse over a resource to highlight the related resources.

**Resources to create** [Info](#)  
Create only the VPC resources or the VPC and other networking resources.

**VPC only**  **VPC and more**

**Name tag auto-generation** [Info](#)  
Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.  
 **Auto-generate**  
project

**IPv4 CIDR block** [Info](#)  
Determine the starting IP and the size of your VPC using CIDR notation.  
10.0.0.0/16 65,536 IPs  
CIDR block size must be between /16 and /28.

**IPv6 CIDR block** [Info](#)  
 **No IPv6 CIDR block**  
 **Amazon-provided IPv6 CIDR block**

**Preview**

**VPC** [Show details](#)  
Your AWS virtual network  
project-vpc

**Subnets (4)**  
Subnets within this VPC

**us-east-1a**  
project-subnet-public1-us-east-1a  
project-subnet-private1-us-east-1a

**us-east-1b**  
project-subnet-public2-us-east-1b  
project-subnet-private2-us-east-1b

**Route tab**  
Route network

project-rtb-  
project-rtb-  
project-rtb-

# Short DEMO: Creating a VPC

1. Select “*Write the name of the VPC*”

## Resources to create [Info](#)

Create only the VPC resource or the VPC and other networking resources.

VPC only

VPC and more

## Name tag - *optional*

Creates a tag with a key of 'Name' and a value that you specify.

demo-vpc

## IPv4 CIDR block [Info](#)

IPv4 CIDR manual input

IPAM-allocated IPv4 CIDR block

## IPv4 CIDR

10.0.0.0/16

CIDR block size must be between /16 and /28.

## IPv6 CIDR block [Info](#)

No IPv6 CIDR block

IPAM-allocated IPv6 CIDR block

Amazon-provided IPv6 CIDR block

IPv6 CIDR owned by me

## Tenancy [Info](#)

Default

# Short DEMO: Creating a VPC

IPv4 CIDR

CIDR block size must be between /16 and /28.

IPv6 CIDR block [Info](#)

No IPv6 CIDR block  
 IPAM-allocated IPv6 CIDR block  
 Amazon-provided IPv6 CIDR block  
 IPv6 CIDR owned by me

Tenancy [Info](#)

**Tags**

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Name"/> <span>X</span>	<input type="text" value="demo-vpc"/> <span>X</span>
<span>Add tag</span>	<span>Remove tag</span>

You can add 49 more tags

Select “Create VPC”

Cancel Create VPC

# Short DEMO: Creating a VPC

You successfully created `vpc-04687ed39bddf360c / demo-vpc`

If you obtain a similar to this one, the VPC has been created successfully.

VPC > Your VPCs > `vpc-04687ed39bddf360c`

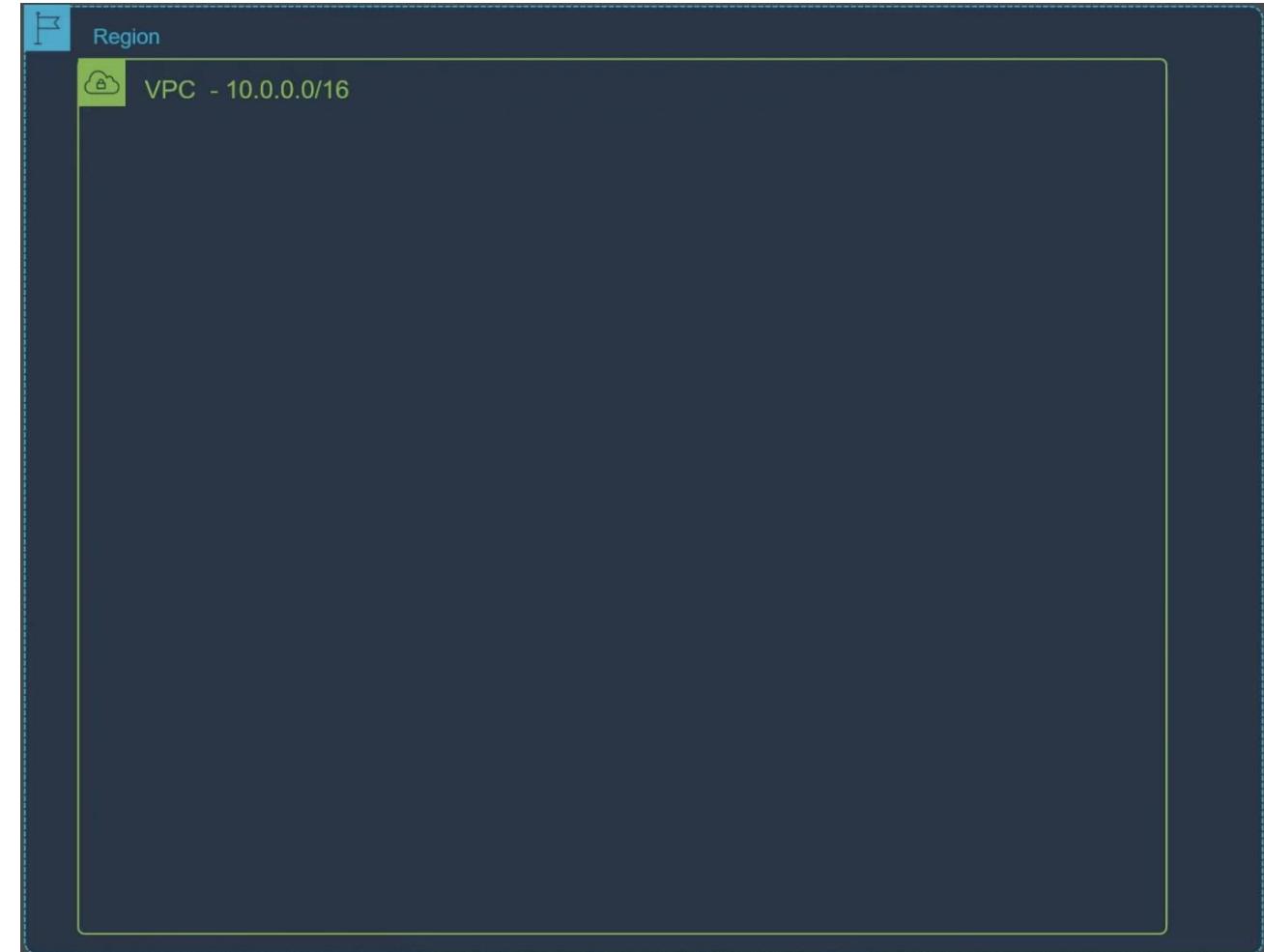
## `vpc-04687ed39bddf360c / demo-vpc`

Actions ▾

Details		Info	
VPC ID	<input type="checkbox"/> <code>vpc-04687ed39bddf360c</code>	State	<input checked="" type="checkbox"/> Available
Tenancy	Default	DHCP option set	<a href="#">dopt-0f60fac6f6714f4e2</a>
Default VPC	No	IPv4 CIDR	10.0.0.0/16
Network Address Usage metrics	Disabled	Route 53 Resolver DNS Firewall rule groups	<input checked="" type="checkbox"/> Failed to load rule groups
		DNS hostnames	Disabled
		Main route table	<a href="#">rtb-0c8819d75538f9bbb</a>
		IPv6 pool	-
		Owner ID	<input type="checkbox"/> 374448134437
		DNS resolution	Enabled
		Main network ACL	<a href="#">acl-0ee583540c8804abb</a>
		IPv6 CIDR (Network border group)	-

# 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 contain 2 subnets: Private and Public



# Short DEMO: Creating subnets within the VPC

VPC dashboard X

EC2 Global View X

Filter by VPC:

Virtual private cloud

Your VPCs

**Subnets**  

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

Endpoint services

NAT gateways

Peering connections

You successfully created **vpc-04687ed39bddf360c / demo-vpc** X

[VPC](#) > [Your VPCs](#) > **vpc-04687ed39bddf360c**

**vpc-04687ed39bddf360c / demo-vpc** Actions ▾

**Details** [Info](#)

VPC ID	State	
<b>vpc-04687ed39bddf360c</b>	<b>Set</b>	<b>6f6714f4e2</b>
Default VPC	IPv4 CIDR	10.0.0.0/16
No		
Network Address Usage metrics	Route 53 Resolver DNS Firewall rule groups	Route 53 Resolver DNS Firewall rule groups
Disabled		<span style="color: red;">✖ Failed to load rule groups</span>

DNS hostnames  
Disabled

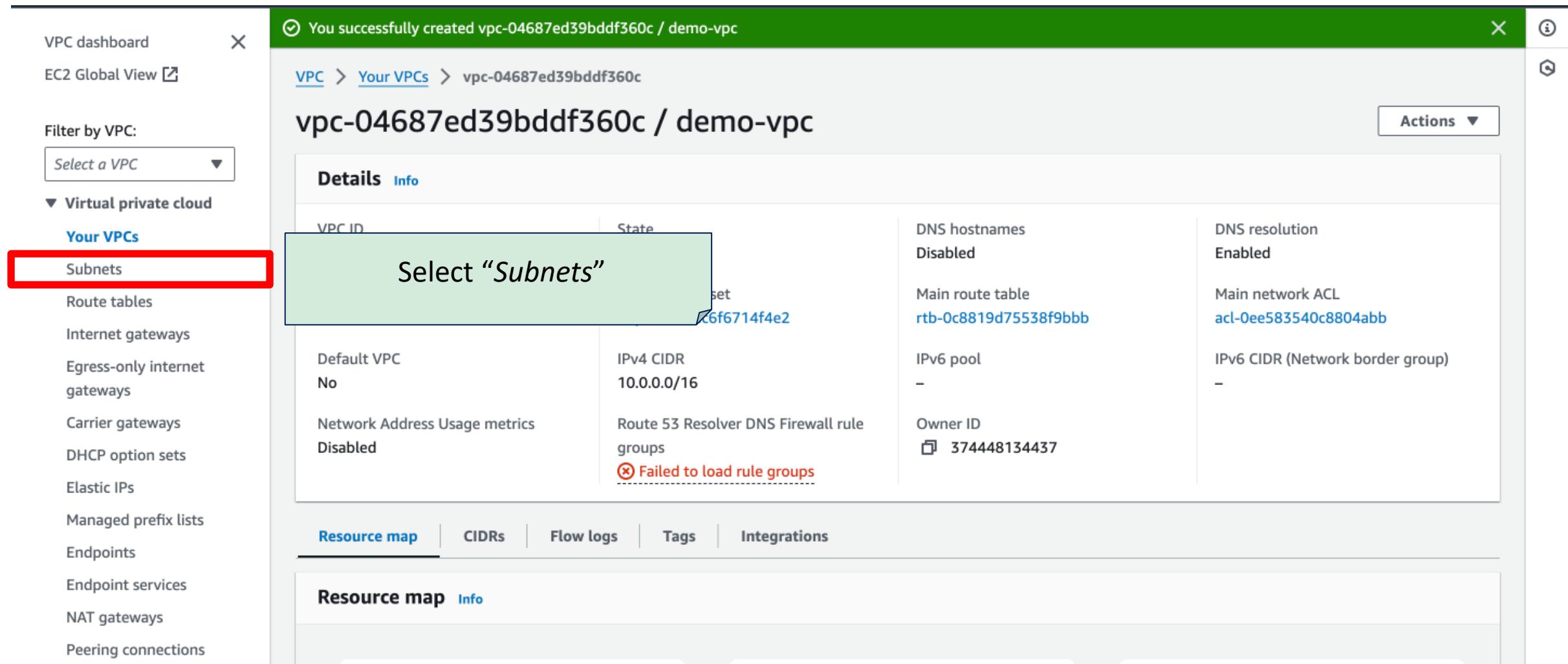
Main route table  
[rtb-0c8819d75538f9bbb](#)

IPv6 pool  
-

Owner ID  
[374448134437](#)

[Resource map](#) CIDRs Flow logs Tags Integrations

**Resource map** [Info](#)

A screenshot of the AWS VPC Dashboard. At the top, a green banner displays the message "You successfully created vpc-04687ed39bddf360c / demo-vpc". Below the banner, the breadcrumb navigation shows "VPC > Your VPCs > vpc-04687ed39bddf360c". The main title is "vpc-04687ed39bddf360c / demo-vpc". On the left, a sidebar menu is open under "Virtual private cloud", with "Subnets" highlighted and surrounded by a red box. The main content area shows the VPC details: VPC ID is "vpc-04687ed39bddf360c", State is "Set", Default VPC is "No", IPv4 CIDR is "10.0.0.0/16", and Network Address Usage metrics are "Disabled". A modal window titled "Select 'Subnets'" is overlaid on the details table. In the bottom right corner of the main content area, there is an error message: "✖ Failed to load rule groups". At the bottom, there are tabs for "Resource map", "CIDRs", "Flow logs", "Tags", and "Integrations", with "Resource map" being the active tab.

# Short DEMO: Creating subnets within the VPC

**Subnets** [Info](#)

Find resources by attribute or tag

VPC: vpc-0a4e63a041e517836 [X](#) [Clear filters](#)

<input type="checkbox"/>	Name	Subnet ID	State	VPC	IPv4 CIDR
No matching resource found					

Select a subnet

[C](#) [Actions ▾](#) [Create subnet](#)

Click in the button “Create Subnet”

# Short DEMO: Creating subnets within the VPC

VPC > Subnets > Create subnet

## Create subnet Info

**VPC**

VPC ID  
Create subnets in this VPC.

Select a VPC

Q

vpc-0180bbf67cc30b5e0	(def
172.31.0.0/16	
vpc-04687ed39bddf360c (demo-vpc)	
10.0.0.0/16	

1. We select our VPC (demo-vpc)

Select a VPC first to create new subnets.

Add new subnet

Cancel Create subnet

The screenshot shows the 'Create subnet' page in the AWS VPC console. At the top, the navigation path is 'VPC > Subnets > Create subnet'. Below it, the title 'Create subnet' has an 'Info' link. A large green callout box with the text '1. We select our VPC (demo-vpc)' is positioned over the 'Select a VPC' dropdown. The dropdown menu lists two VPCs: 'vpc-0180bbf67cc30b5e0' (with IP range 172.31.0.0/16) and 'vpc-04687ed39bddf360c (demo-vpc)' (with IP range 10.0.0.0/16). The second VPC is highlighted with a red border. Below the dropdown, a message says 'Select a VPC first to create new subnets.' At the bottom, there are 'Add new subnet' and 'Create subnet' buttons, with 'Create subnet' being orange.

# 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 create the second subnet

**Subnet 1 of 1**

**Subnet name**  
Create a tag with a key of 'Name' and a value that you specify.  
The name can be up to 256 characters long.

**Availability Zone** [Info](#)  
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

**IPv4 VPC CIDR block** [Info](#)  
Choose the IPv4 VPC CIDR block to create a subnet in.

**IPv4 subnet CIDR block**  
 256 IPs

**Tags - optional**

Key	Value - optional
<input type="text" value="Name"/> <input type="button" value="X"/>	<input type="text" value="demo-subnet-1"/> <input type="button" value="X"/> <input type="button" value="Remove"/>

You can add 49 more tags.

# Short DEMO: Creating subnet 2 within the VPC

## Subnet 2 of 2

### Subnet name

Create a tag with a key of 'Name' and a value that you specify.

demo-subnet-2

The name can be up to 256 characters long.

### Availability Zone [Info](#)

Choose the zone in which your subnet will reside, or let Amazon choose one for you.

No preference

### IPv4 VPC CIDR block [Info](#)

Choose the IPv4 VPC CIDR block to create a subnet in.

10.0.0.0/16

### IPv4 subnet CIDR block

10.0.1.0/24

256 IPs

< > ^ v

### ▼ Tags - optional

#### Key

Name

#### Value - optional

demo-subnet-2

[Remove](#)

[Add new tag](#)

You can add 49 more tags.

[Remove](#)

[Add new subnet](#)

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"

[Cancel](#)

[Create subnet](#)

# Short DEMO: Creating a subnet within the VPC

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

You have successfully created 2 subnets: subnet-06919054cc5740f8c, subnet-045116c88da9c77b3

Subnets (2) Info

Actions ▼  Create subnet

Find resources by attribute or tag

Subnet ID : subnet-06919054cc5740f8c  Subnet ID : subnet-045116c88da9c77b3  Clear filters < 1 >

<input type="checkbox"/>	Name	Subnet ID	State	VPC	IPv4 CIDR
<input type="checkbox"/>	demo-subnet-1	<a href="#">subnet-06919054cc5740f8c</a>	<input checked="" type="checkbox"/> Available	<a href="#">vpc-04687ed39bddf360c   dem...</a>	10.0.0.0/24
<input type="checkbox"/>	demo-subnet-2	<a href="#">subnet-045116c88da9c77b3</a>	<input checked="" type="checkbox"/> Available	<a href="#">vpc-04687ed39bddf360c   dem...</a>	10.0.1.0/24

# Short DEMO: Creating a subnet within the VPC

- At this point, we have created 2 private subnets (They have not connection 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

The screenshot shows the AWS VPC console interface. On the left, a navigation sidebar lists various VPC-related resources: Virtual private cloud (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, Endpoint services, NAT gateways, Peering connections), Security (Network ACLs, Security groups), and DNS firewall (Rule groups, Domain lists). The 'Internet gateways' item is highlighted with a red box and has a green callout bubble pointing to it with the text "Select 'Internet Gateway'". The main content area displays "Your VPCs (1/2) Info" with a search bar and a table. The table has columns: Name, VPC ID, State, IPv4 CIDR, and IPv6 CIDR. It shows two entries: one for a VPC named '-' with VPC ID [ypc-0180bbf67cc30b5e0](#), State Available, IPv4 CIDR 172.31.0.0/16, and IPv6 CIDR -, and another for a VPC named "demo-vpc" with VPC ID [ypc-04687ed39bddf360c](#), State Available, IPv4 CIDR 10.0.0.0/16, and IPv6 CIDR -. At the bottom of the main content area, there is a summary for the selected VPC: "vpc-04687ed39bddf360c / demo-vpc". Below this summary, there is a navigation bar with tabs: Details, Resource map, CIDRs, Flow logs, Tags, and Integrations. The "Details" tab is currently selected. The "Details" section contains four data rows: VPC ID ([vpc-04687ed39bddf360c](#)), State (Available), DNS hostnames (Disabled), and DNS resolution (Enabled).

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
-	<a href="#">ypc-0180bbf67cc30b5e0</a>	Available	172.31.0.0/16	-
demo-vpc	<a href="#">ypc-04687ed39bddf360c</a>	Available	10.0.0.0/16	-

vpc-04687ed39bddf360c / demo-vpc

Details    Resource map    CIDRs    Flow logs    Tags    Integrations

**Details**

VPC ID <a href="#">vpc-04687ed39bddf360c</a>	State Available	DNS hostnames Disabled	DNS resolution Enabled
---	--------------------	---------------------------	---------------------------

# Short DEMO: Creating a Internet Gateway to acces to Internet

Select “Create Internet Gateway”

Create internet gateway

<input type="checkbox"/>	Name	Internet gateway ID	State	VPC ID	Owner
<input type="checkbox"/>	-	<a href="#">igw-01e2a6829705ae3cd</a>	Attached	<a href="#">vpc-0180bbf67cc30b5e0</a>	37444

Select an internet gateway above

# Short DEMO: Creating a Internet Gateway to acces to Internet

VPC > [Internet gateways](#) > Create internet gateway

## Create internet gateway Info

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

### Internet gateway settings

#### Name tag

Creates a tag with a key of 'Name' and a value that you specify.

1. Write the name of the Internet Gateway

#### Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

##### Key

##### Value - optional

RemoveAdd new tag

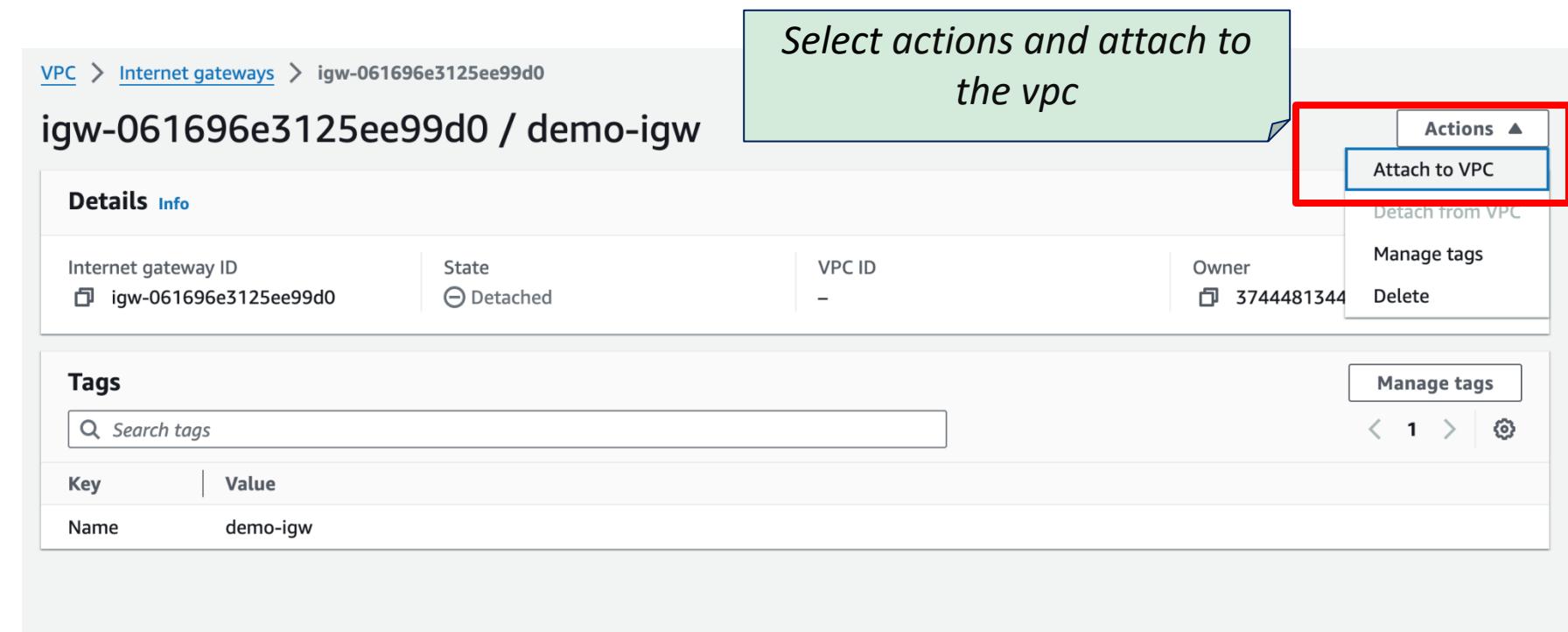
You can add 49 more tags.

2. Select "Create Internet Gateway"

CancelCreate 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

VPC > Internet gateways > Attach to VPC (igw-061696e3125ee99d0)

## Attach to VPC (igw-061696e3125ee99d0)

**VPC**  
Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

**Available VPCs**  
Attach the internet gateway to this VPC.

Select a VPC

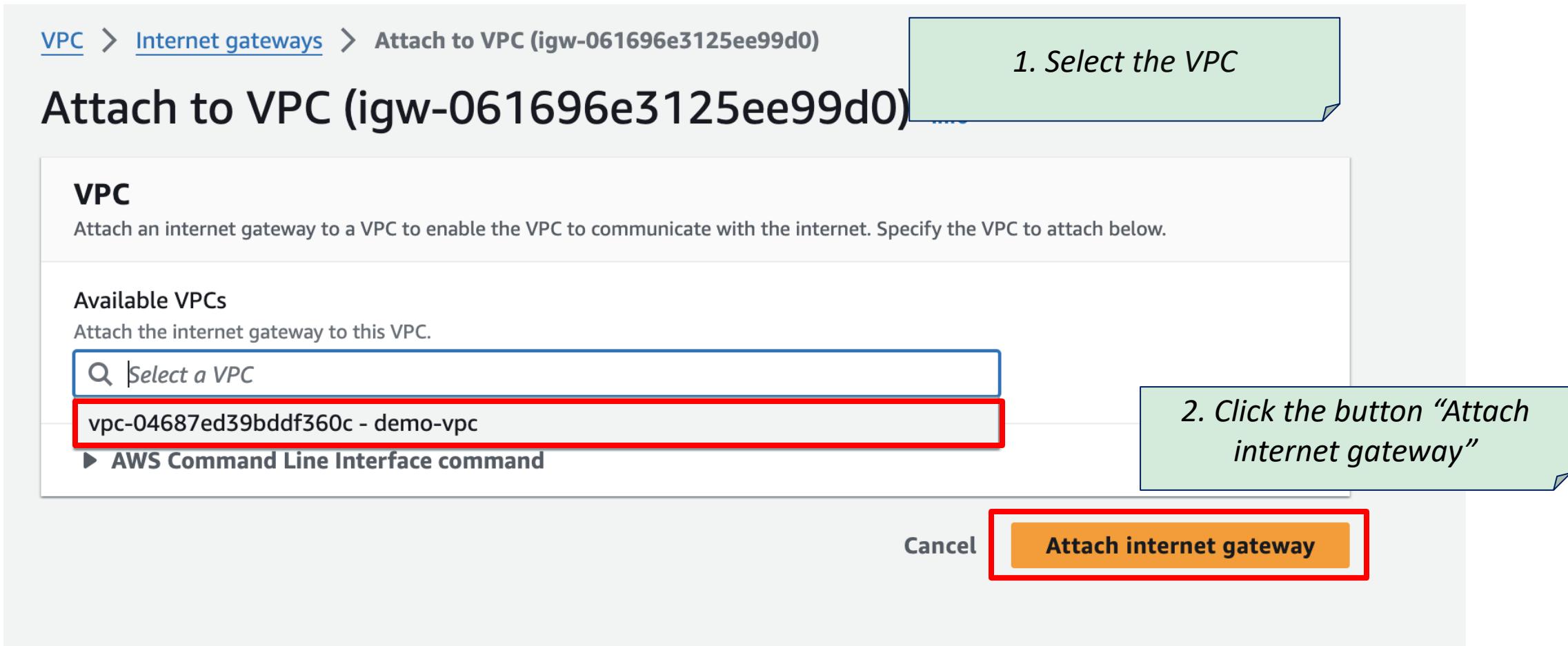
vpc-04687ed39bddf360c - demo-vpc

AWS Command Line Interface command

1. Select the VPC

2. Click the button “Attach internet gateway”

Cancel **Attach internet gateway**



# Short DEMO: Creating a Internet Gateway to acces to Internet

igw-061696e3125ee99d0 / demo-igw Actions ▾

Details <small>Info</small>			
Internet gateway ID	<input type="checkbox"/> igw-061696e3125ee99d0	State	<input checked="" type="checkbox"/> Attached
VPC ID	<a href="#">vpc-04687ed39bddf360c   demo-vpc</a>		
Owner	<input type="checkbox"/> 374448134437		

**Tags** Manage tags < 1 >

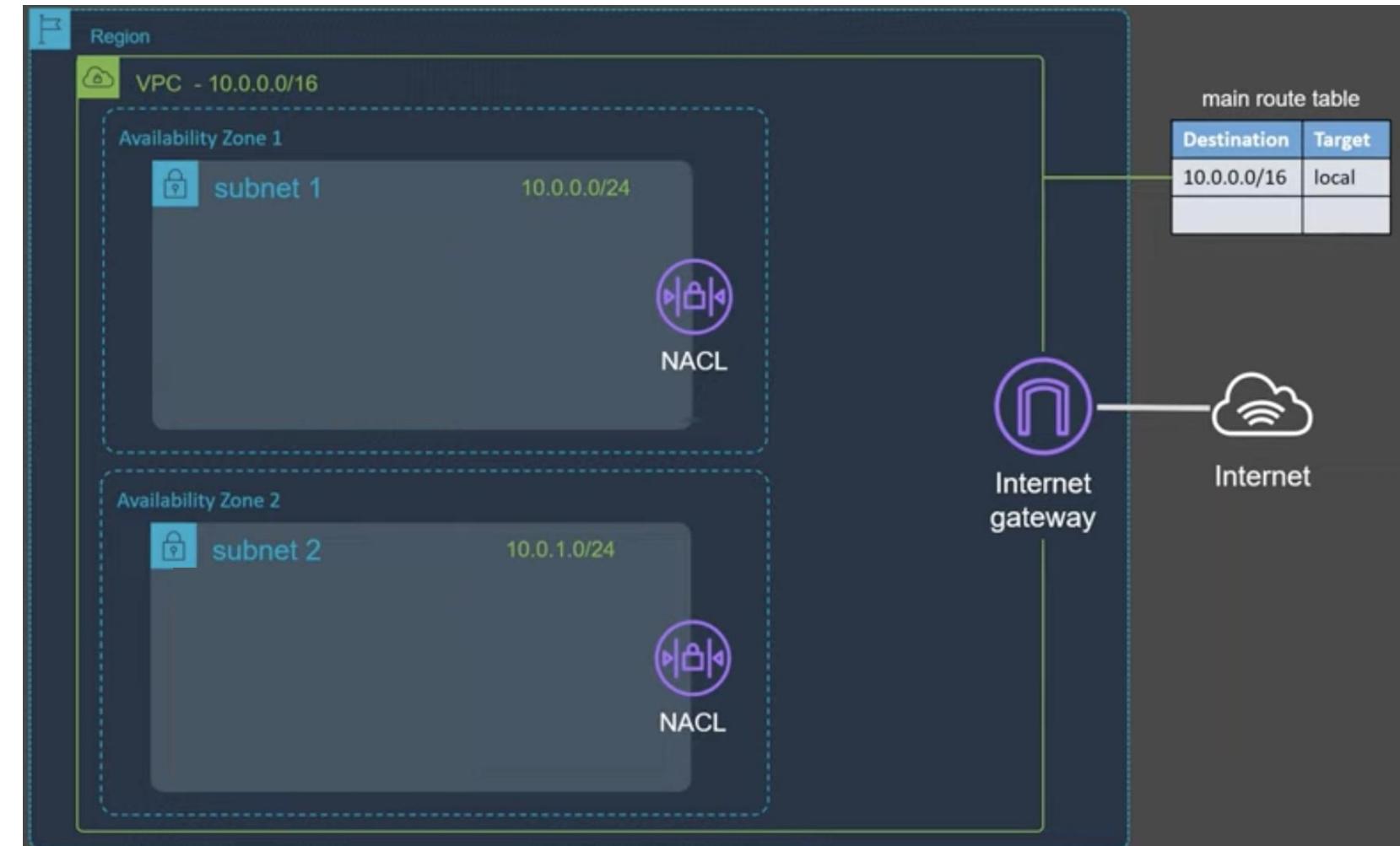
Search tags

Key	Value
Name	demo-igw

*The Gateway has been correctly attached to the VPC*

# 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 connection routes



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

VPC dashboard X

EC2 Global View [ ]

Filter by VPC:  
Select a VPC ▼

Virtual private cloud

Your VPCs

Subnets

Route tables Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

Endpoint services

**Your VPCs (1/2) Info**

C Actions ▾ Create VPC

Search

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
-	<a href="#">vpc-0180bbf67cc30b5e0</a>	<span style="color: green;">Available</span>	172.31.0.0/16	-
<input checked="" type="checkbox"/> demo-vpc	<a href="#">vpc-04687ed39bddf360c</a>	<span style="color: green;">Available</span>	10.0.0.0/16	-

Select “Route tables”

vpc-04687ed39bddf360c / demo-vpc

Details Resource map CIDRs Flow logs Tags Integrations

Details

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

The screenshot shows the AWS VPC dashboard with the following interface elements:

- Left sidebar:** Includes links for "VPC dashboard", "EC2 Global View", "Filter by VPC" (with a dropdown menu), and a "Virtual private cloud" section containing "Your VPCs", "Subnets", "Route tables" (which is currently selected and highlighted in blue), "Internet gateways", "Egress-only internet gateways", "Carrier gateways", "DHCP option sets", "Elastic IPs", "Managed prefix lists", "Endpoints", and "Endpoint services".
- Main content area:** A table titled "Route tables (2) Info" showing two existing route tables:

Name	Route table ID	Explicit subnet associati...	Edge associa...
-	<a href="#">rtb-086a13b1b207a9ba4</a>	-	-
-	<a href="#">rtb-0c8819d75538f9bbb</a>	-	-
- Action bar:** Contains a "Create route table" button, which is highlighted with a red box and a callout bubble saying "Select ‘Create route table’".

# 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"**

**Create route table** Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

**Route table settings**

**Name - optional**  
Create a tag with a key of 'Name' and a value that you specify.

**VPC**  
The VPC to use for this route table.

**Tags**  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Name"/> <input type="button" value="X"/>	<input type="text" value="demo-rt"/> <input type="button" value="X"/> <input type="button" value="Remove"/>
<input type="button" value="Add new tag"/>	

You can add 49 more tags.

# 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.

Route table rtb-0344d9f1a1bac33a2 | demo-rt was created successfully.

VPC > Route tables > rtb-0344d9f1a1bac33a2

rtb-0344d9f1a1bac33a2 / demo-rt

Actions ▾

**Details** Info

Route table ID rtb-0344d9f1a1bac33a2	Main No	Explicit subnet associations -	Edge associations -
VPC vpc-04687ed39bddf360c   demo-vpc	Owner ID 374448134437		

Select “Edit routes”

Routes Subnet associations Edge associations Route propagation Tags

Both Edit routes

Filter routes

Routes (1)

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

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

VPC > Route tables > rtb-0344d9f1a1bac33a2 > Edit routes

## Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	<input checked="" type="checkbox"/> Active	No

**Add route** *Select “Add route”*

Cancel Preview Save changes

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

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	<input checked="" type="checkbox"/> Active	No
<input type="text" value="0.0.0.0/0"/> X	<input type="text" value="local"/> X		
	<input type="text" value="Internet Gateway"/> X		No
	<input type="text" value="igw-061696e3125ee99d0"/> X		

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

Cancel Preview **Save changes**

# 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.

VPC > Route tables > rtb-0344d9f1a1bac33a2

## rtb-0344d9f1a1bac33a2 / demo-rt

Actions ▾

Details		Info	Explicit subnet associations	Edge associations
Route table ID	<input type="checkbox"/> rtb-0344d9f1a1bac33a2	Main	<input type="checkbox"/> No	-
VPC	vpc-04687ed39bddf360c   demo-vpc	Owner ID	<input type="checkbox"/> 374448134437	-

**Routes** | Subnet associations | Edge associations | Route propagation | Tags

Routes (2)		Both	Edit routes
<input type="text"/> Filter routes		< 1 >	⚙️
Destination	Target	Status	Propagated
0.0.0.0/0	<a href="#">igw-061696e3125ee99d0</a>	<input checked="" type="checkbox"/> Active	No
10.0.0.0/16	local	<input checked="" type="checkbox"/> Active	No

# Short DEMO: Associating the Route Table with the subnet 1

The screenshot shows the AWS VPC Subnets page. On the left, there's a sidebar with 'VPC dashboard' and 'EC2 Global View'. Below that, under 'Virtual private cloud', are sections for 'Your VPCs' (with a red box around 'Subnets') and 'Route tables', 'Internet gateways', 'Egress-only internet gateways', 'Carrier gateways', 'DHCP option sets', 'Elastic IPs', and 'Managed prefix lists'. The main content area has a header 'Subnets Info' with a search bar, a filter for 'VPC : vpc-0a4e63a041e517836', and buttons for 'Actions' and 'Create subnet'. A green callout box with the text 'We select “subnets”' points to the 'Subnets' section in the sidebar. The main table has columns for Name, Subnet ID, State, VPC, and IPv4 CIDR. A message 'No matching resource found' is displayed. At the bottom, there's a 'Select a subnet' button.

VPC dashboard

EC2 Global View

Filter by VPC:

Select a VPC

▼ Virtual private cloud

Your VPCs

**Subnets**

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

**Subnets** Info

Find resources by attribute or tag

VPC : vpc-0a4e63a041e517836

Clear filters

C Actions ▾ Create subnet

No matching resource found

We select “subnets”

Select a subnet

# Short DEMO: Associating the Route Table with the subnet 1

VPC dashboard X

EC2 Global View

Filter by VPC:

▼ Virtual private cloud

Your VPCs

**Subnets**

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

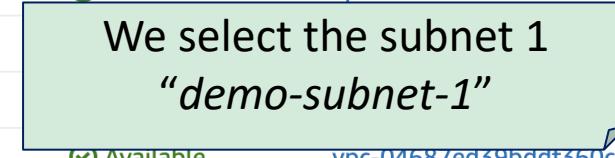
Elastic IPs

Managed prefix lists

...

**Subnets (8) Info**

Actions

<input type="checkbox"/>	Name	Subnet ID	State	VPC	IPv4 CIDR
<input type="checkbox"/>	-	<a href="#">subnet-01ff0a9d6ac4d4e5c</a>	Available	<a href="#">vpc-0180bbf67cc30b5e0</a>	172.31.0.0/20
<input type="checkbox"/>	-	<a href="#">subnet-04206d82c7a2fb3a2</a>	Available	<a href="#">vpc-0180bbf67cc30b5e0</a>	172.31.80.0/20
<input type="checkbox"/>	-	<a href="#">subnet-0c705d08575108f82</a>	Available	<a href="#">vpc-0180bbf67cc30b5e0</a>	172.31.16.0/20
<input type="checkbox"/>	-	<a href="#">subnet-0792390533a56a801</a>	Available	<a href="#">vpc-0180bbf67cc30b5e0</a>	172.31.32.0/20
<input type="checkbox"/>	-	<a href="#">subnet-0f8ff5901f50124f3</a>	Available	<a href="#">vpc-0180bbf67cc30b5e0</a>	172.31.64.0/20
<input type="checkbox"/>	-	<a href="#">subnet-016b329e1d9af6c75</a>			172.31.48.0/20
<input type="checkbox"/>	<b>demo-subnet-1</b>	<a href="#">subnet-06919054cc5740f8c</a>		Available	 We select the subnet 1 “ <b>demo-subnet-1</b> ”
<input type="checkbox"/>	demo-subnet-2	<a href="#">subnet-045116c88da9c77b3</a>			dem... 10.0.0.0/24

# Short DEMO: Associating the Route Table with the subnet 1

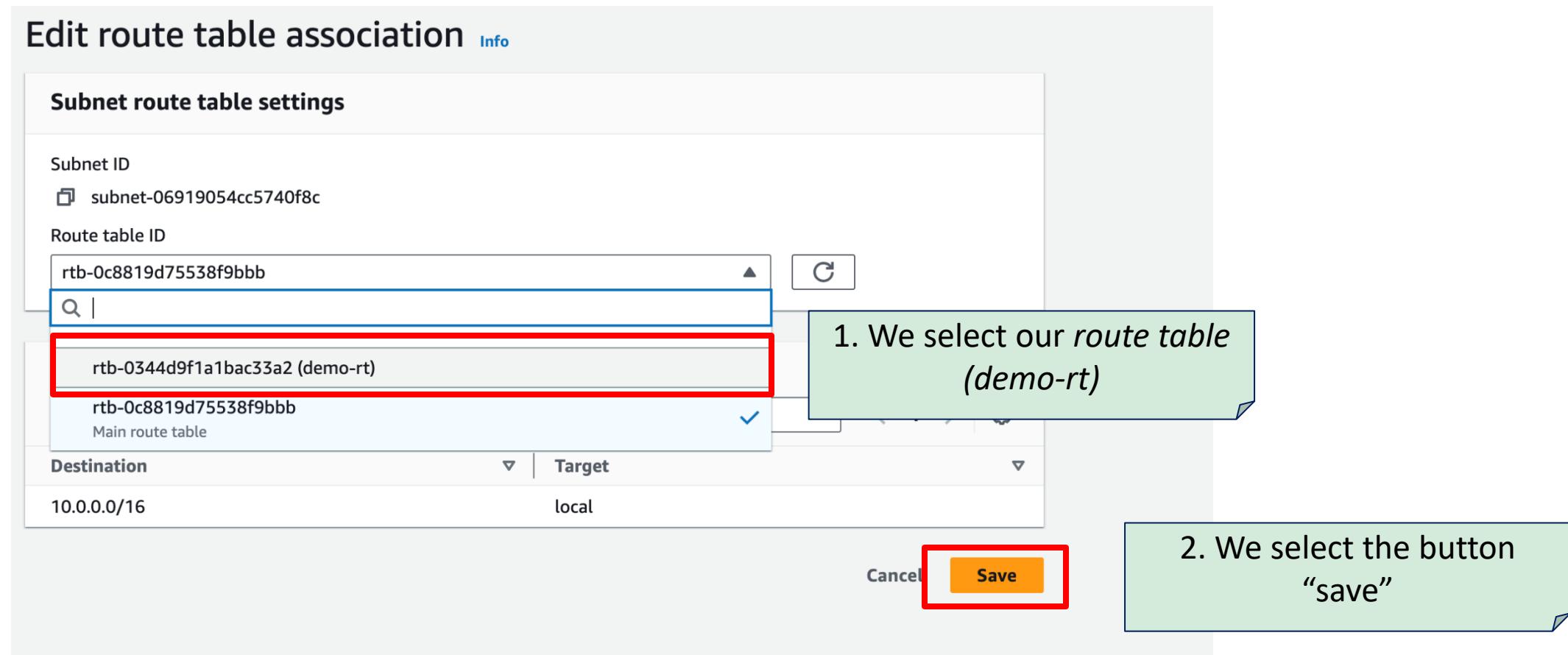
The screenshot shows the AWS Route Table configuration page. At the top, there are tabs for "Flow logs" and "Route table". The "Route table" tab is highlighted with a red box and contains the text "1. We select ‘Route Table’". Below the tabs are sections for "Sharing" and "Tags".

The main area displays the "Route table: rtb-0c8819d75538f9bbb" and a table titled "Routes (1)". The table has columns for "Destination" and "Target". One row shows "10.0.0.0/16" in the Destination column and "local" in the Target column. A search bar labeled "Filter routes" and a pagination section with "1" are also visible.

A red box highlights the "Edit route table association" button in the top right corner of the main content area, with the text "2. We select ‘Edit route table association’" placed above it.

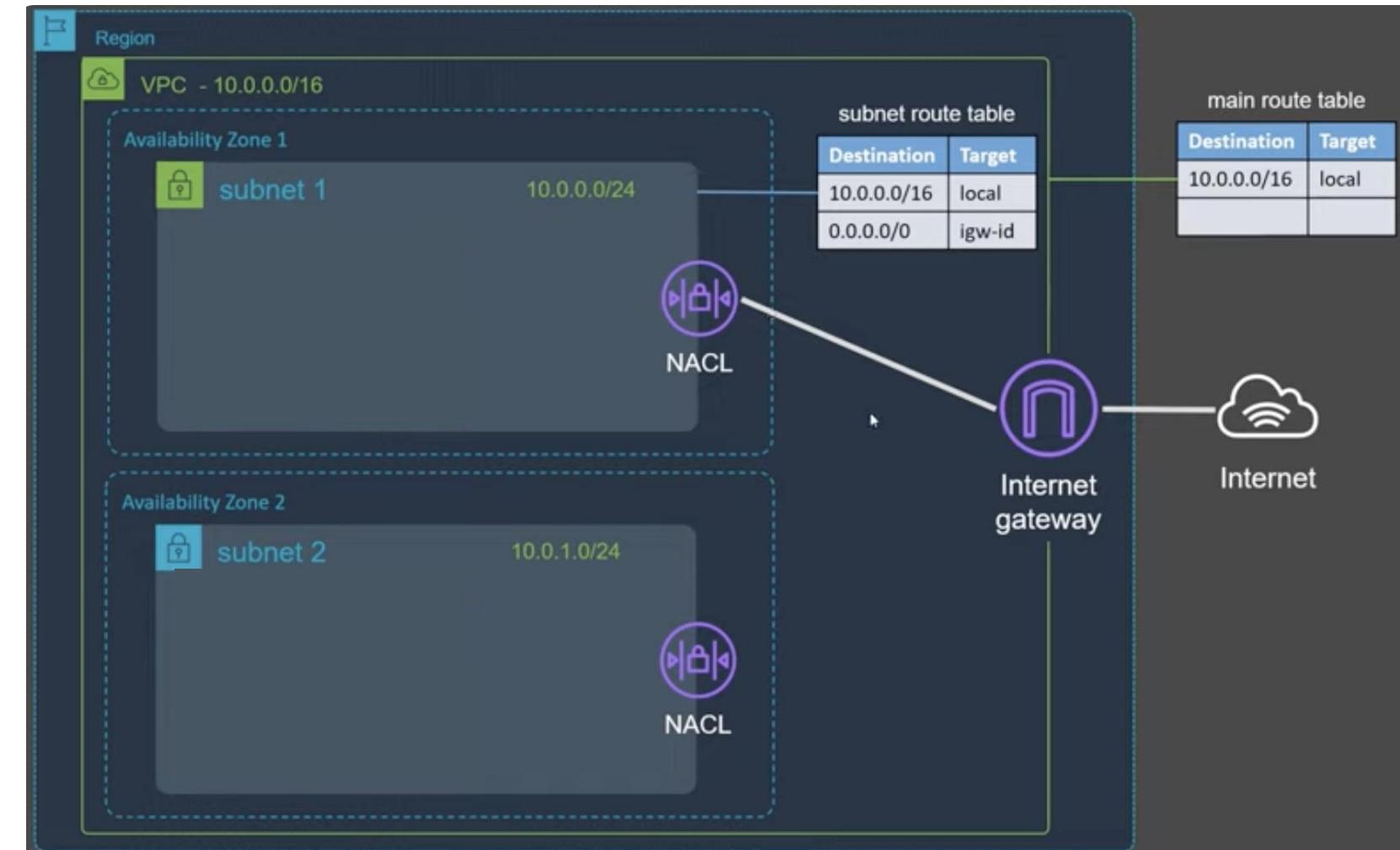
Destination	Target
10.0.0.0/16	local

# Short DEMO: Associating the Route Table with the subnet 1



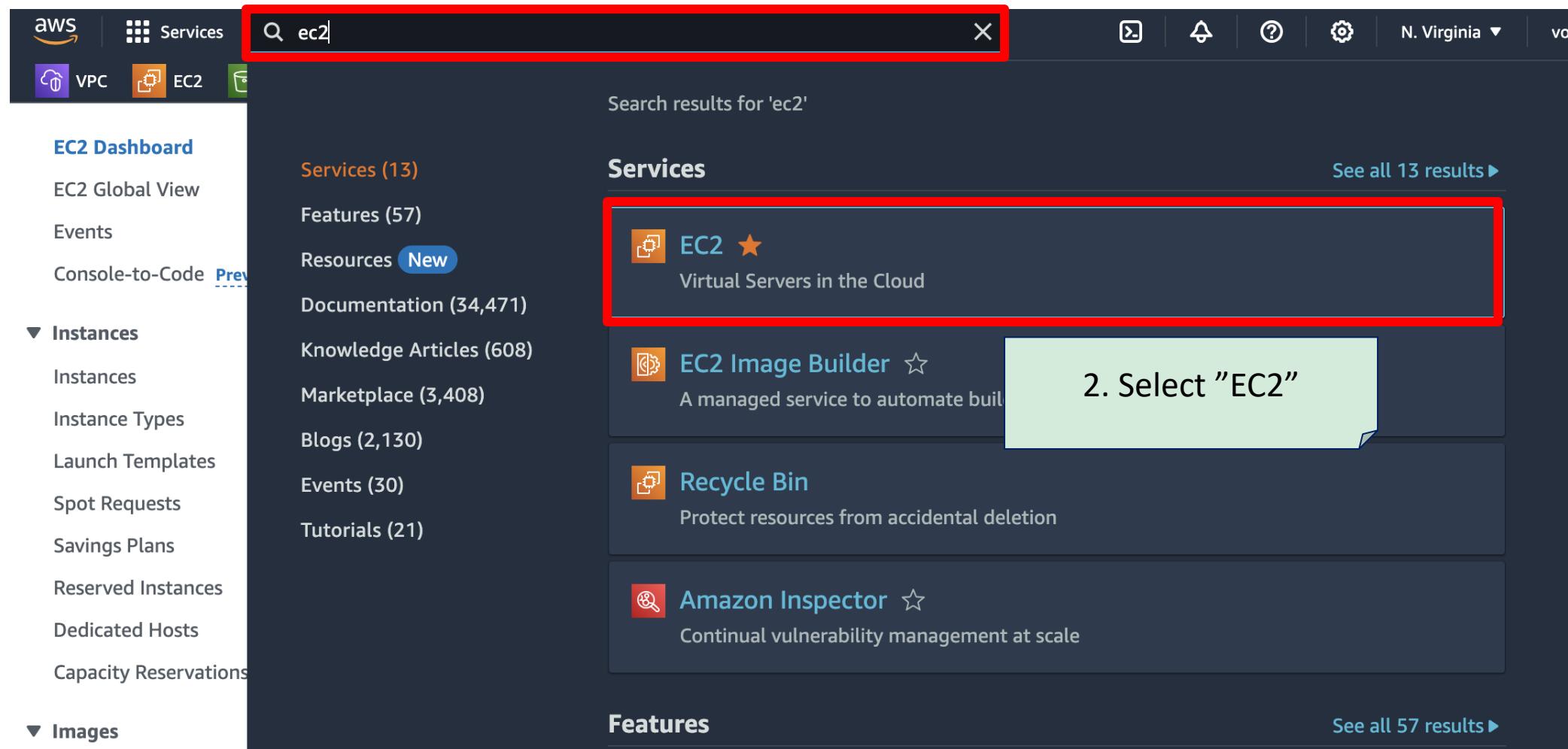
# Short DEMO: Associating the Route Table with the subnet 1

- 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.



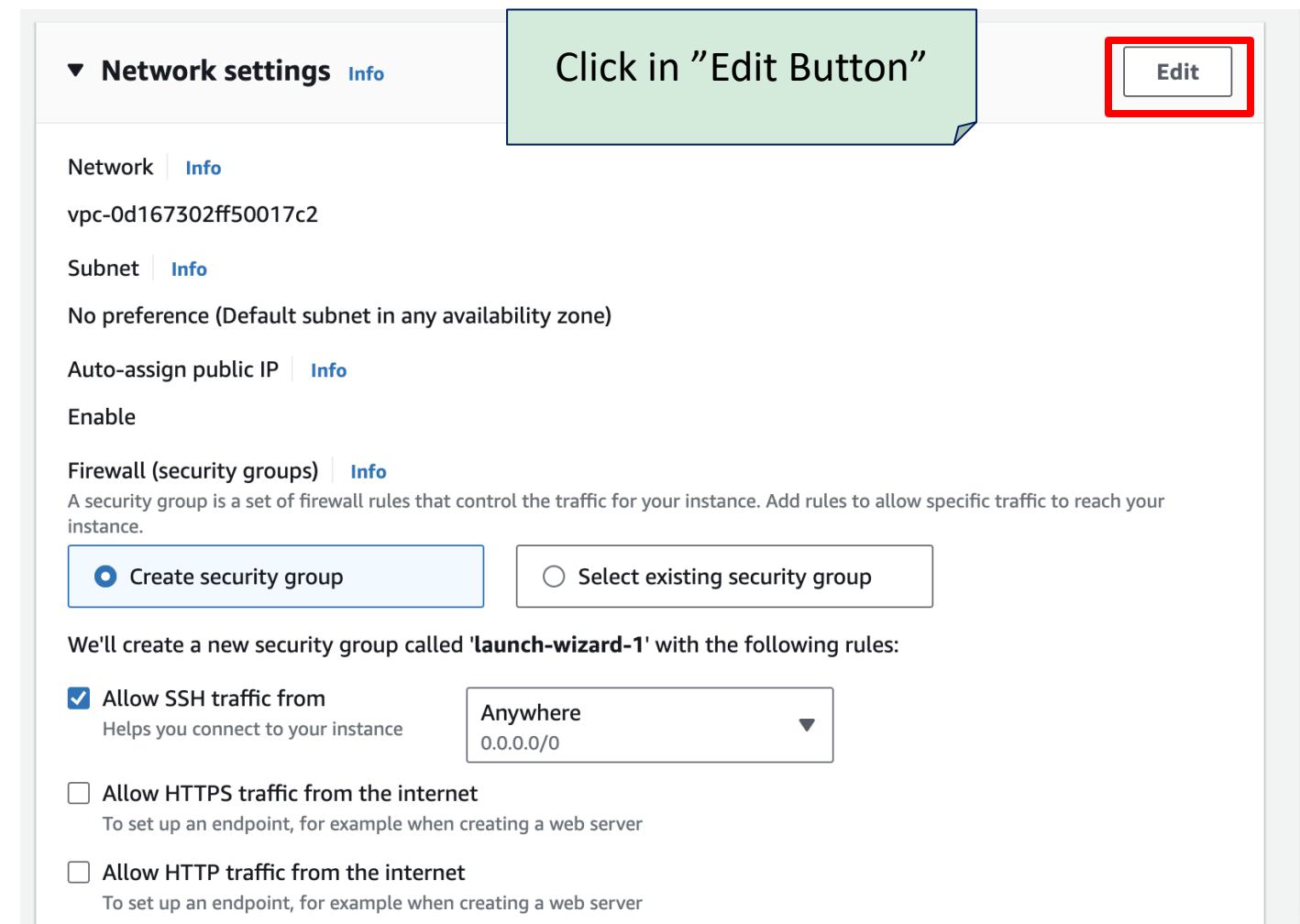
# Short DEMO: Deploying Instances in the Subnet1 of our VPC

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

The screenshot shows the AWS EC2 Dashboard. At the top, there's a navigation bar with the AWS logo, a 'Services' dropdown, a search bar containing 'Search', a '[Option+S]' keybinding, and several icons for notifications, help, and settings. Below the navigation bar, there are tabs for VPC, EC2 (which is selected), and S3. On the left, a sidebar titled 'EC2 Dashboard' lists various EC2 services: EC2 Global View, Events, Console-to-Code (with a 'Preview' link), Instances (expanded), Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations (with a 'New' link), Images (expanded), AMIs, and AMI Catalog. In the main content area, there's a summary section with counts for Instances (running), Auto Scaling Groups, Dedicated Hosts, Elastic IPs, Instances, Key pairs, Load balancers, Placement groups, Security groups, Snapshots, and Volumes. Below this, a large green callout box contains the text 'Select “Launch Instance”'. A red rectangle highlights the 'Launch instance' button, which is orange with white text. Below it is a 'Migrate a server' button. A note at the bottom states: 'Note: Your instances will launch in the US East (N. Virginia) Region'. To the right, there are sections for 'Regions' (set to US East (N. Virginia)) and 'Zones' (with columns for Zone name and Zone ID). The bottom of the page has a footer with links for 'Amazon Machine Images', 'AWS Lambda', 'AWS Lambda Functions', and 'AWS Lambda Metrics'.

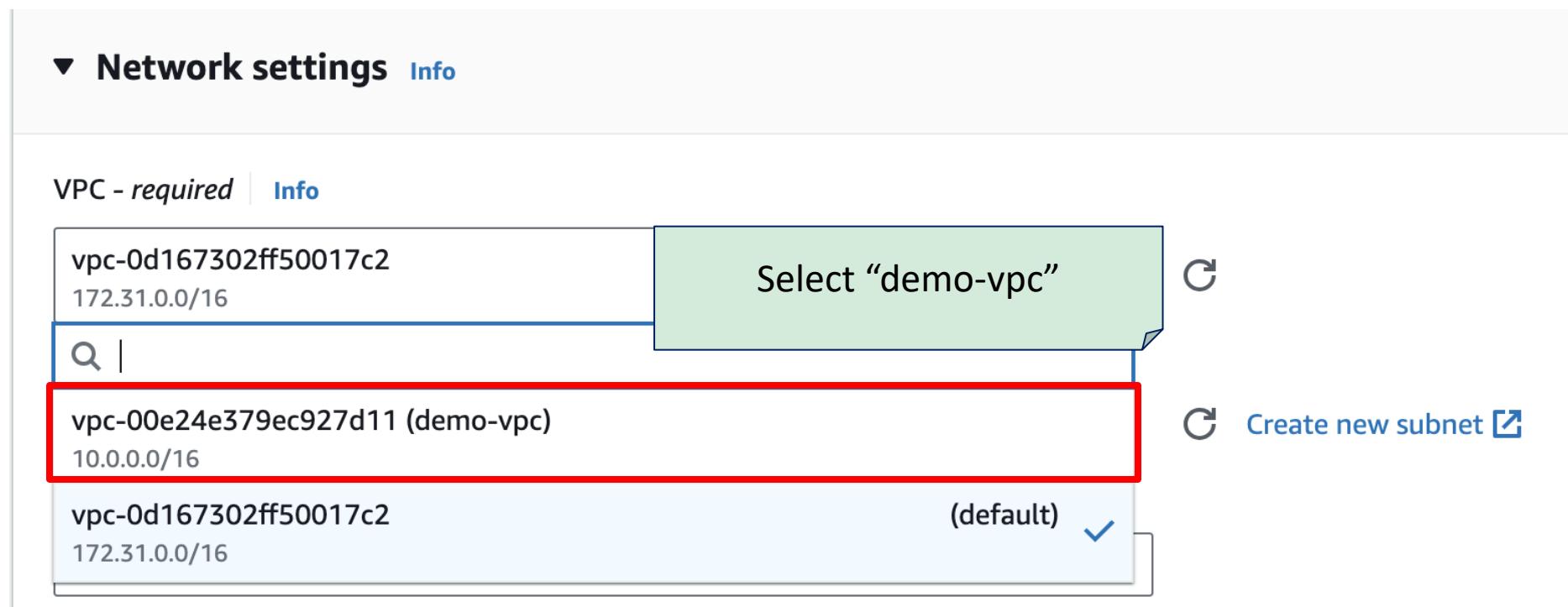
# Short DEMO: Deploying Instances in the Subnet1 of our VPC

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

The screenshot shows the 'Network settings' section of a CloudFormation stack configuration. At the top, it says 'VPC - required' with a dropdown menu showing 'vpc-00e24e379ec927d11 (demo-vpc) 10.0.0.0/16'. Below that is a 'Subnet' table with three rows. The first row is highlighted with a green background and contains the text 'Select "demo-subnet-1"'. The second row has a red border around its entire content and contains the text 'demo-subnet-1' in the 'Subnet' column. The third row is partially visible. On the right side of the table, there is a 'Create new subnet' button. The overall interface is clean and modern, typical of AWS's web-based management tools.

Subnet	
subnet-07ef8cebd9ae615d9	Select "demo-subnet-1"
subnet-07ef8cebd9ae615d9	demo-subnet-1
subnet-07309bf6712ab8f5e	demo-subnet-2

VPC: vpc-00e24e379ec927d11 Owner: 471112729721 Availability Zone: us-east-1d IP addresses available: 251 CIDR: 10.0.0.0/24)

VPC: vpc-00e24e379ec927d11 Owner: 471112729721 Availability Zone: us-east-1d IP addresses available: 251 CIDR: 10.0.0.0/24)

VPC: vpc-00e24e379ec927d11 Owner: 471112729721 Availability Zone: us-east-1d IP addresses available: 251 CIDR: 10.0.1.0/24)