Virtual Private Cloud (Amazon VPC)

The Amazon Virtual Private Cloud (Amazon VPC) is a custom-defined virtual network within the AWS Cloud. You can provision your own logically isolated section of AWS, similar to designing and implementing a separate independent network that would operate in an on premises data center.

Amazon VPC is the networking layer for Amazon Elastic Compute Cloud (Amazon EC2), and it allows you to build your own virtual network within AWS.

You will have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.

You can easily customize the network configuration for your Amazon Virtual Private Cloud.

For example, you can create a public-facing subnet for your webservers that has access to the Internet, and place your backend systems such as databases or application servers in a private-facing subnet with no Internet access.

VPCs also have a few limits set on them by default. For example, **you can have a maximum of five VPCs per region.** Each VPC can have a max of one Internet gateway as well as one virtual private gateway. Also, **each VPC has a limit of hosting a maximum of up to 200 subnets per VPC**. You can increase these limit by simply requesting AWS to do so.

An Amazon VPC consists of the following components:

- Subnets
- Route tables
- Dynamic Host Configuration Protocol (DHCP) option sets
- Security groups
- Network Access Control Lists (ACLs)

An Amazon VPC has the following optional components:

- Internet Gateways (IGWs)
- Elastic IP (EIP) addresses
- Elastic Network Interfaces (ENIs)
- Endpoints
- Peering
- Network Address Translation (NATs) instances and NAT gateways
- Virtual Private Gateway (VPG), Customer Gateways (CGWs), and Virtual Private Networks (VPNs)

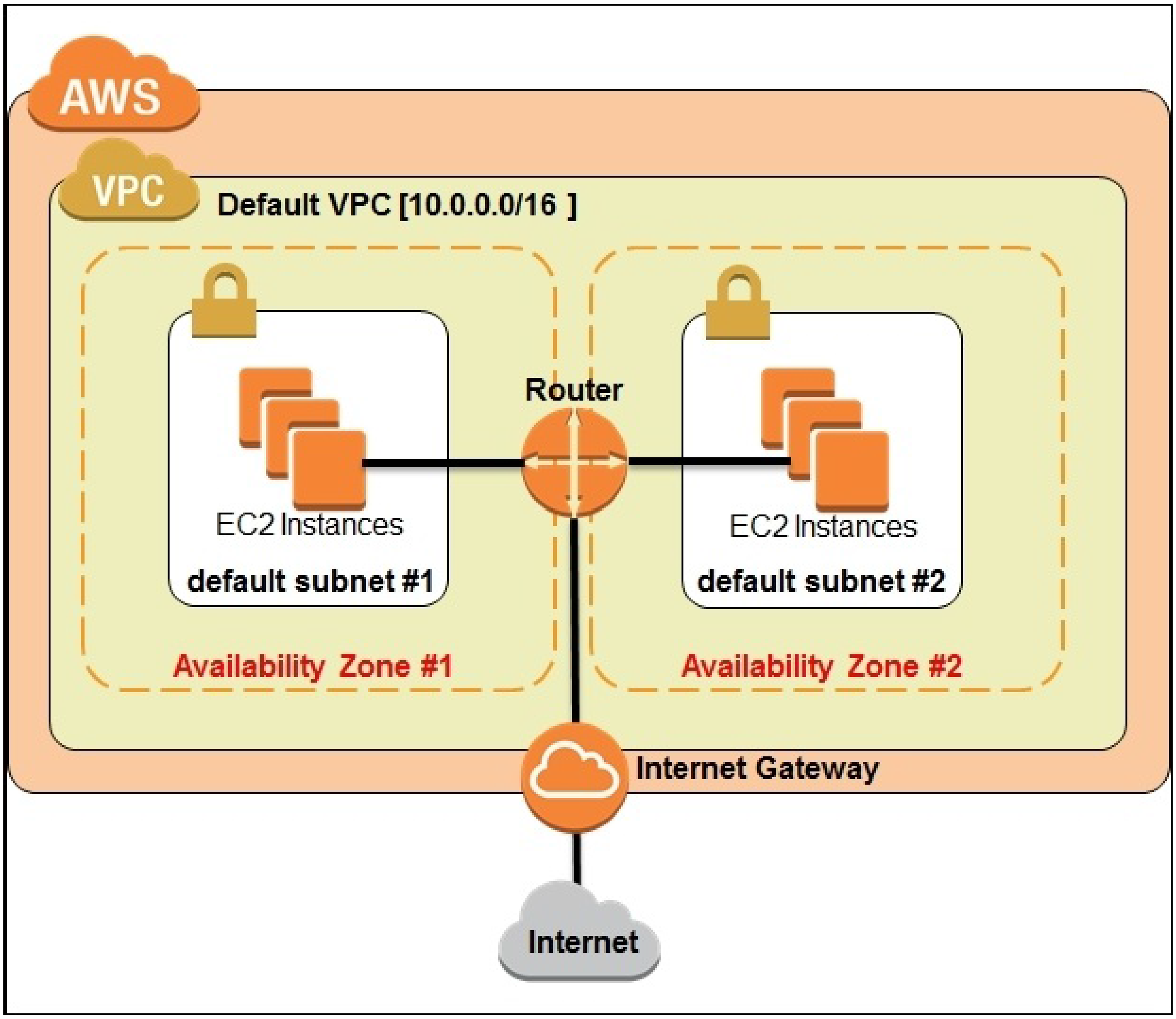
By default, AWS will create a VPC for you in your particular region the first time you sign up for the service. This is called as the default VPC. The default VPC comes preconfigured with the following set of configurations:

The default VPC is always created with a CIDR block of /16, which means it supports 65,536 IP addresses in it.

A default subnet is created in each AZ of your selected region. Instances launched in these default subnets have both a public and a private IP address by default as well.

An Internet Gateway is provided to the default VPC for instances to have Internet connectivity.

A few necessary route tables, security groups, and ACLs are also created by default that enable the instance traffic to pass through to the Internet. Refer to the following figure:



Classless Inter-Domain Routing (CIDR): When you create an Amazon VPC, you must specify the IPv4 address range by choosing a Classless Inter-Domain Routing (CIDR) block, such as 10.0.0.0/16. The address range of the Amazon VPC cannot be changed after the Amazon VPC is created. An Amazon VPC address range may be as large as /16 (65,536 available addresses) or as small as /28 (16 available addresses) and should not overlap any other network with which they are to be connected.

Subnets: A subnet is a segment of an Amazon VPC's IP address range where you can launch Amazon EC2 instances, Amazon Relational Database Service (Amazon RDS) databases, and other AWS resources.

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

- Remember that one subnet equals one Availability Zone. You can, however, have multiple subnets in one Availability Zone.

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

A **public subnet** is one in which the associated route table directs the subnet's traffic to the Amazon VPC's IGW.

A **private subnet** is one in which the associated route table does not direct the subnet's traffic to the Amazon VPC's IGW.

A **VPN-only subnet** is one in which the associated route table directs the subnet's traffic to the Amazon VPC's VPG and does not have a route to the IGW.

Route Tables:

A route table is a logical construct within an Amazon VPC that contains a set of rules (called routes) that are applied to the subnet and used to determine where network traffic is directed.

- You can modify route tables and add your own custom routes.
- You can also use route tables to specify which subnets are public (by directing Internet traffic to the IGW) and which subnets are private (by not having a route that directs traffic to the IGW).
- Each route table contains a default route called the local route, which enables communication within the Amazon VPC, and this route cannot be modified or removed.
- Additional routes can be added to direct traffic to exit the Amazon VPC via the IGW, the VPG, or the NAT instance.

You should remember the following points about route tables:

- Your VPC has an implicit router.
- Your VPC automatically comes with a main route table that you can modify.

- You can create additional custom route tables for your VPC.
- Each subnet must be associated with a route table, which controls the routing for the subnet. If you don't explicitly associate a subnet with a particular route table, the subnet uses the main route table.
- You can replace the main route table with a custom table that you've created so that each new subnet is automatically associated with it.

Internet Gateways:

An Internet Gateway (IGW) is a horizontally scaled, redundant, and highly available Amazon VPC component that allows communication between instances in your Amazon VPC and the Internet.

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

When an instance receives traffic from the Internet, the IGW translates the destination address (public IP address) to the instance's private IP address and forwards the traffic to the Amazon VPC.

You must do the following to create a public subnet with Internet access:

- Attach an IGW to your Amazon VPC.
- Create a subnet route table rule to send all non-local traffic (0.0.0.0/0) to the IGW.
- Configure your network ACLs and security group rules to allow relevant traffic to flow to and from your instance.

Elastic IP Addresses (EIP): An Elastic IP Addresses (EIP) is a static, public IP address in the pool for the region that you can allocate to your account (pull from the pool) and release (return to the pool).

AWS maintains a pool of public IP addresses in each region and makes them available for you to associate to resources within your Amazon VPCs.

- EIPs are specific to a region (that is, an EIP in one region cannot be assigned to an instance within an Amazon VPC in a different region).
- There is a one-to-one relationship between network interfaces and EIPs.
- You can move EIPs from one instance to another, either in the same Amazon VPC or a different Amazon VPC within the same region.
- EIPs remain associated with your AWS account until you explicitly release them.

- There are charges for EIPs allocated to your account, even when they are not associated with a resource.

Peering:

An Amazon VPC peering connection is a networking connection between two Amazon VPCs that enables instances in either Amazon VPC to communicate with each other as if they are within the same network. You can create an Amazon VPC peering connection between your own Amazon VPCs or with an Amazon VPC in another AWS account within a single region.

An Amazon VPC may have multiple peering connections, and peering is a one-to-one relationship between Amazon VPCs, meaning two Amazon VPCs cannot have two peering agreements between them.

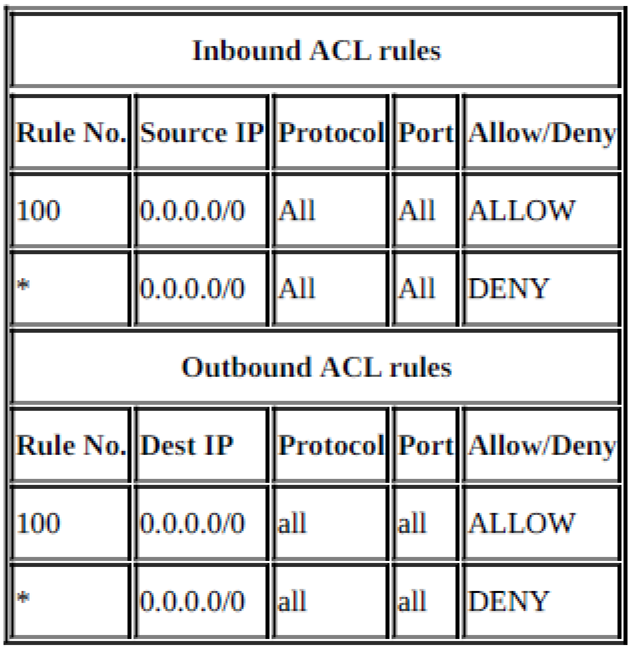
Peering connections are created through a request/accept protocol. The owner of the requesting Amazon VPC sends a request to peer to the owner of the peer Amazon VPC. If the peer Amazon VPC is within the same account, it is identified by its VPC ID. If the peer VPC is within a different account, it is identified by Account ID and VPC ID. The owner of the peer Amazon VPC has one week to accept or reject the request to peer with the requesting Amazon VPC before the peering request expires.

- You cannot create a peering connection between Amazon VPCs that have matching or overlapping CIDR blocks.
- You cannot create a peering connection between Amazon VPCs in different regions.
- Amazon VPC peering connections do not support transitive routing.
- You cannot have more than one peering connection between the same two Amazon VPCs at the same time.

Network Access Control Lists (ACLs):

A network access control list (ACL) is another layer of security that acts as a stateless firewall on a subnet level.

A network ACL is a numbered list of rules that AWS evaluates in order, starting with the lowest numbered rule, to determine whether traffic is allowed in or out of any subnet associated with the network ACL. Here is a small example of how ACL looks like.



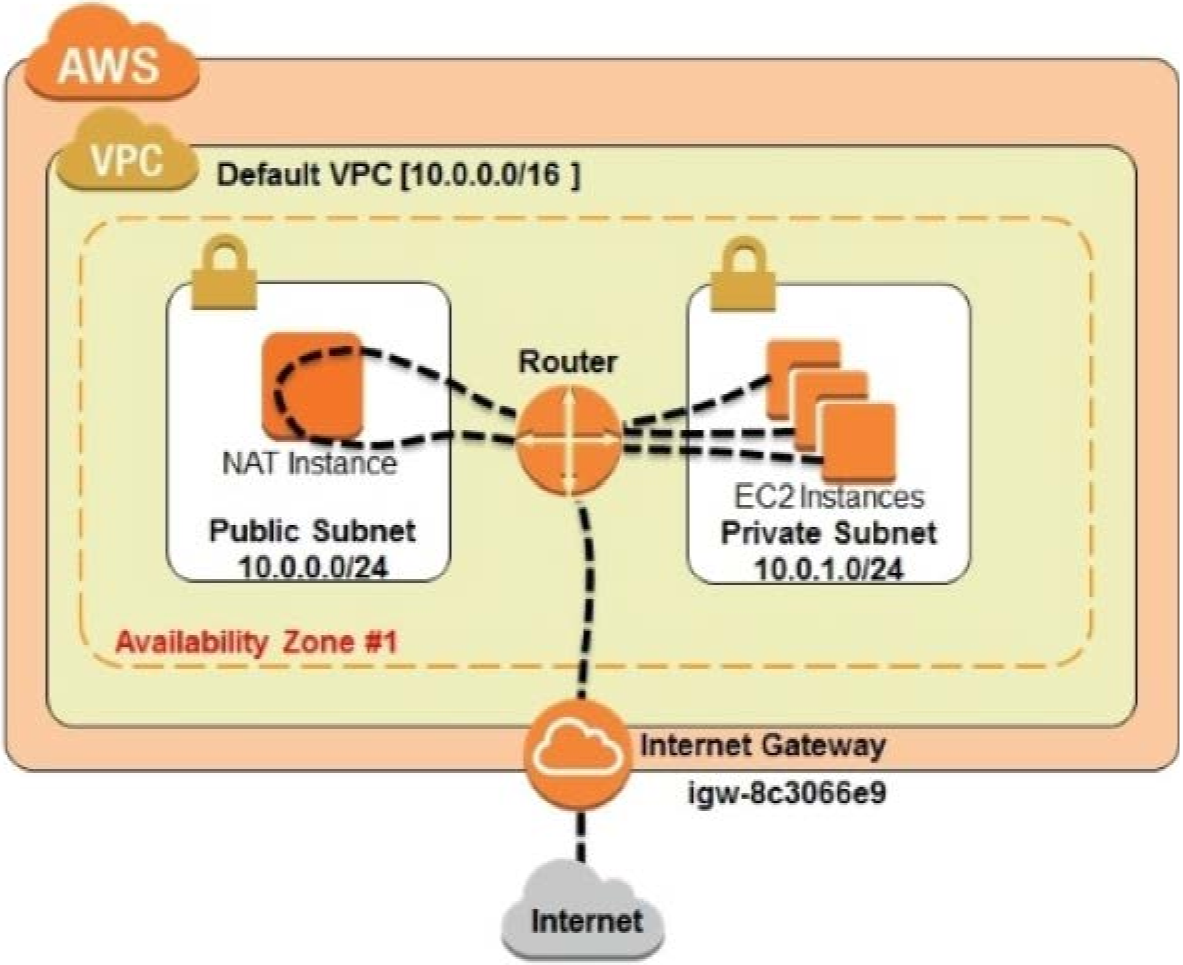
Inbound ACL rules				
Rule No.	**Source IP**	**Protocol**	**Port**	**Allow/Deny**
100	0.0.0.0/0	All	All	ALLOW
*	0.0.0.0/0	All	All	DENY
Outbound ACL rules				
Rule No.	**Dest IP**	**Protocol**	**Port**	**Allow/Deny**
100	0.0.0.0/0	all	all	ALLOW
*	0.0.0.0/0	all	all	DENY

When you create a custom network ACL, its initial configuration will deny all inbound and outbound traffic until you create rules that allow otherwise.

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

Network Address Translation (NAT) Instances and NAT Gateways

By default, any instance that you launch into a private subnet in an Amazon VPC is not able to communicate with the Internet through the IGW. AWS provides NAT instances and NAT gateways to allow instances deployed in private subnets to gain Internet access.



NAT Instance: A network address translation (NAT) instance is an Amazon Linux Amazon Machine Image (AMI) that is designed to accept traffic from instances within a private subnet, translate the source IP address to the public IP address of the NAT instance, and forward the traffic to the IGW.

NAT Instances allows in private subnets to send outbound Internet communication, but it prevents the instances from receiving inbound traffic initiated by someone on the Internet.

- Create a security group for the NAT with outbound rules that specify the needed Internet resources by port, protocol, and IP address.
- Launch an Amazon Linux NAT AMI as an instance in a public subnet and associate it with the NAT security group.
- Disable the Source/Destination Check attribute of the NAT.
- Configure the route table associated with a private subnet to direct Internet-bound traffic to the NAT instance (for example, i-1a2b3c4d).

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

- Allocate an EIP and associate it with the NAT gateway.
- Configure the route table associated with the private subnet to direct Internet-bound traffic to the NAT gateway.

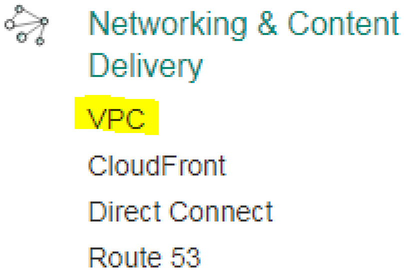
You can connect an existing data center to Amazon VPC using either hardware or software VPN connections, which will make Amazon VPC an extension of the data center. Amazon VPC offers two ways to connect a corporate network to a VPC: VPG and CGW.

A virtual private gateway: VPG is the virtual private network (VPN) concentrator on the AWS side of the VPN connection between the two networks.

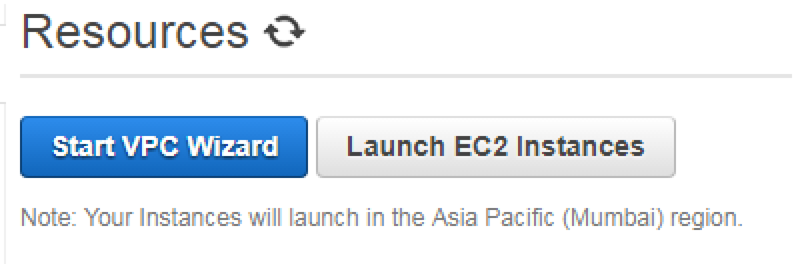
A customer gateway (CGW) represents a physical device or a software application on the customer's side of the VPN connection.

VPC deployment options:

1. You can find VPC under Network & Content Delivery category in AWS console. Select VPC.

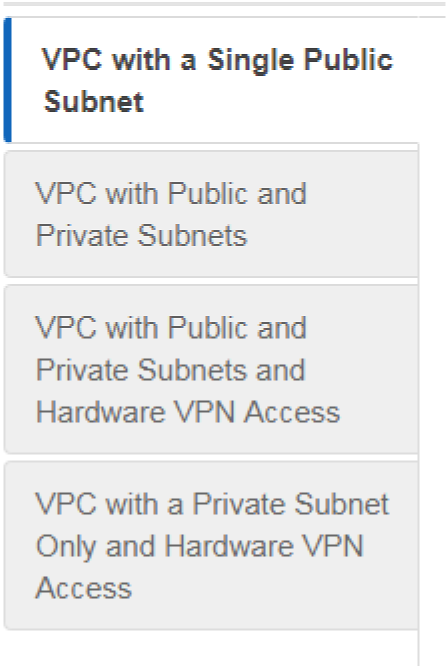


2. You can select the **Start VPC Wizard** option to get all the the VPC deployment methods.



Note: Your Instances will launch in the Asia Pacific (Mumbai) region.

3. We have 4 deployment models available currently with AWS VPC. Detailed description given below.


VPC with a Single Public Subnet
VPC with Public and Private Subnets
VPC with Public and Private Subnets and Hardware VPN Access
VPC with a Private Subnet Only and Hardware VPN Access

VPC with a single public subnet: This is by far the simplest of the four deployment scenarios. Using this scenario, we will get a **VPC will provision a single public subnet with a default Internet Gateway attached to it.** The subnet will also have a few simple and basic route tables, security groups, and network ACLs created. This type of deployment is ideal for small-scaled web applications or simple websites that don't require any separate application or subnet tiers.

VPC with public and private subnets (NAT): This is the most commonly used deployment scenario, this option will provide you with **a public subnet and a private subnet** as well. The public subnet will be connected to an Internet gateway and allow instances launched within it to have Internet connectivity, whereas the private subnet will not have any access to the outside world. This scenario will also provision a single NAT instance inside the public subnet using which your private subnet instances can connect with the outside world but not vice versa. Besides this, the wizard will also create and assign a route table to both the public and private subnets, each with the necessary routing information prefilled in them. This type of deployment is ideal for large-scale web applications and websites that leverage a mix of public facing (web servers) and non-public facing (database servers).

VPC with public and private subnets and hardware VPN access: This deployment scenario is very much similar to the VPC with public and private subnets, however, with one component added additionally, which is the Virtual Private Gateway. This Virtual Private Gateway connects to your on premise network's gateway using a standard VPN connection. This type of deployment is well suited for organizations that wish to extend their on premise datacenters and networks in to the public clouds while allowing their instances to communicate with the Internet.

VPC with a private subnet only and hardware VPN access: Unlike the previous deployment scenario, this scenario only provides you with a private subnet that can connect to your on premise datacenters using standard VPN connections. There is no Internet Gateway provided and thus your instances remain isolated from the Internet. This deployment scenario is ideal for cases where you wish to extend your on premise datacenters into the public cloud but do not wish your instances to have any communication with the outside world.

Here is a simple use case for creating Custom VPC

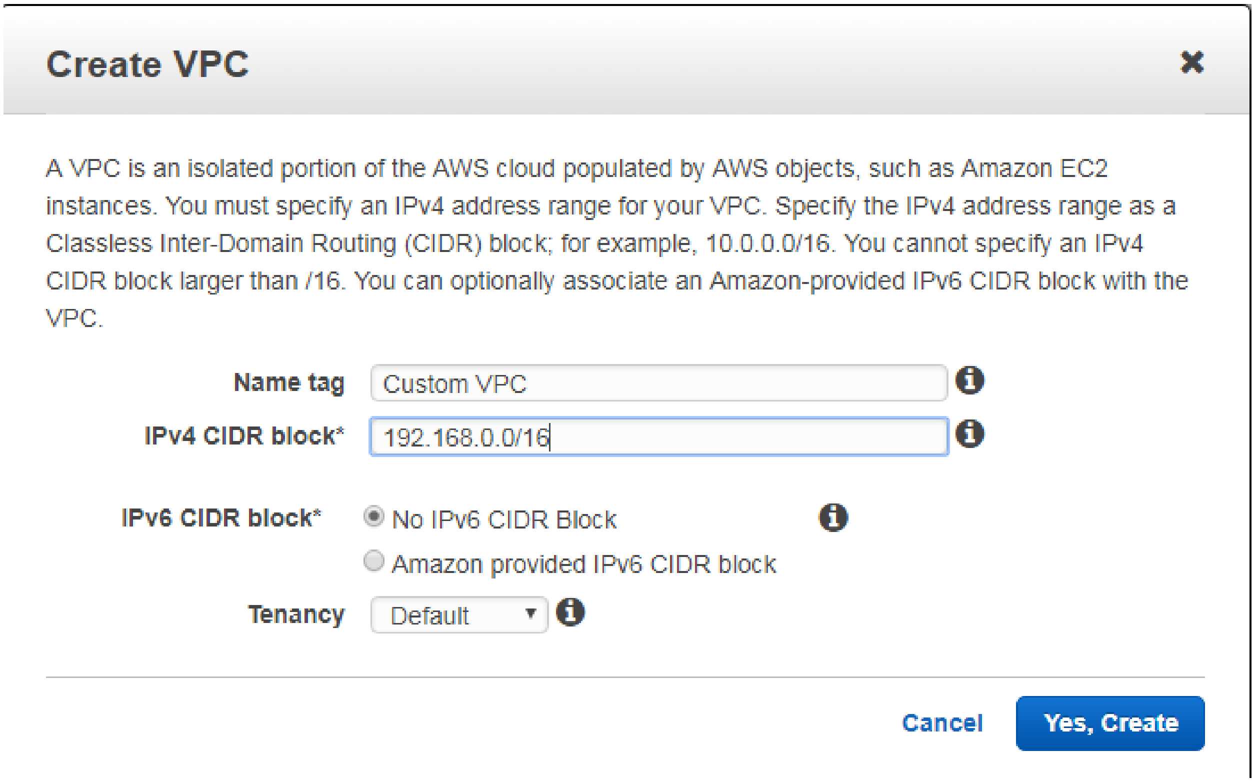
- Create a VPC (US-WEST-PROD-1 - 192.168.0.0/16) with separate secure environments for hosting the web servers and database servers.
- Only the web server environment (US-WEST-PROD-WEB - 192.168.1.0/24) should have direct Internet access.
- The database server environment (US-WEST-PROD-DB - 192.168.5.0/24) should be isolated from any direct access from the outside world.
- The database servers can have restricted Internet access only through a jump server (NAT Instance). The jump server needs to be a part of the web server environment.

You can follow the simple wizard, but to understand the flow clearly am going to create and configure each and every option manually. Here is the steps am going to perform.

- ➢ Creating a Custom VPC
- ➢ Creating Subnets under Custom VPC

- Creating IGW and associating with VPC
- Creating a Route table and performing subnet association
- Launching instance in Public subnet and private subnet

STEP 1: Creating a custom VPC



Create VPC

A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances. You must specify an IPv4 address range for your VPC. Specify the IPv4 address range as a Classless Inter-Domain Routing (CIDR) block; for example, 10.0.0.0/16. You cannot specify an IPv4 CIDR block larger than /16. You can optionally associate an Amazon-provided IPv6 CIDR block with the VPC.

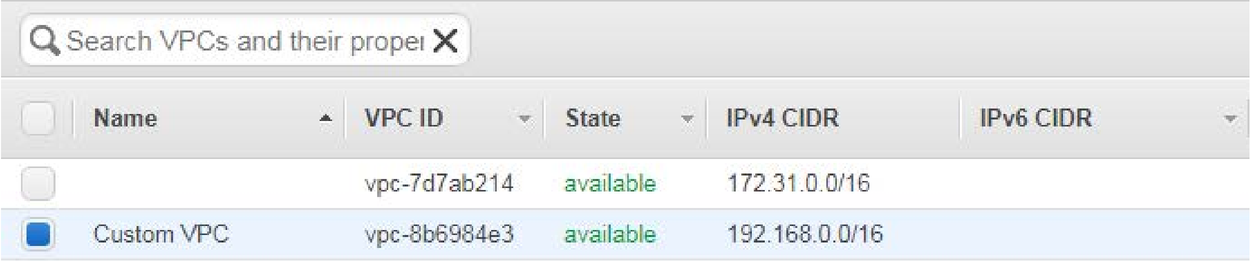
Name tag: Custom VPC

IPv4 CIDR block*: 192.168.0.0/16

IPv6 CIDR block*: ● No IPv6 CIDR Block ○ Amazon provided IPv6 CIDR block

Tenancy: Default

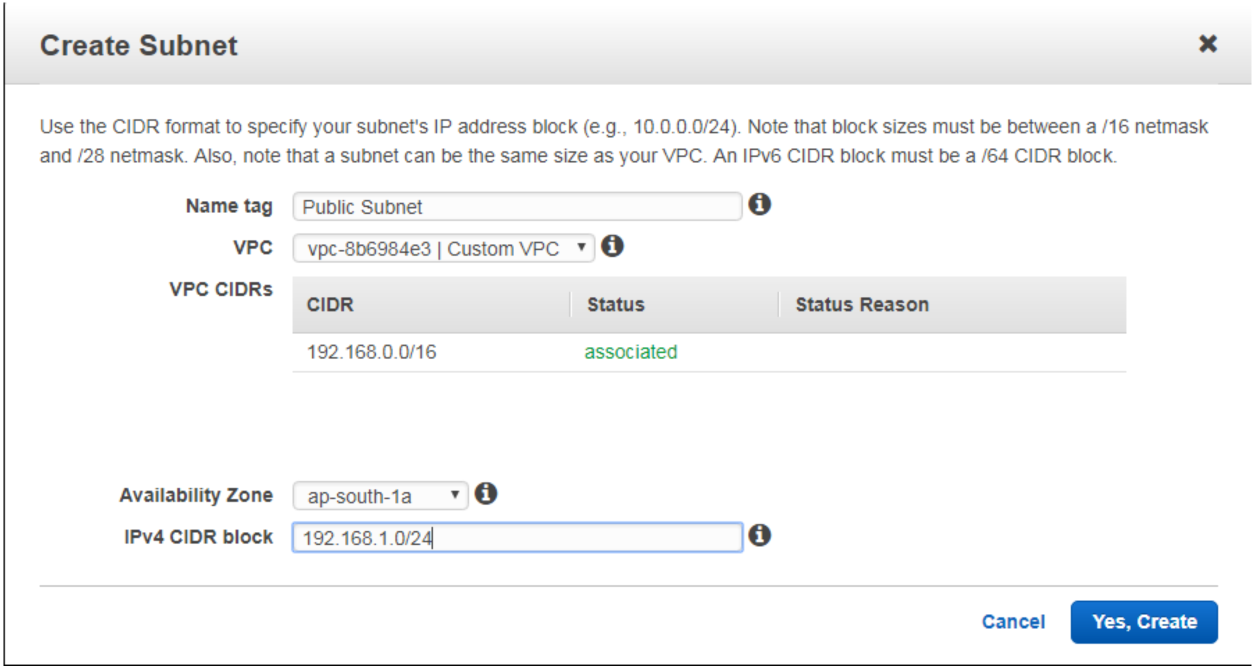
- As mentioned in above image, am creating a VPC with **CustomVPC** name and selecting CIDR block in Class C IP address range **192.168.0.0/16** (provide a /16 subnet will provide us 65,531 IP addresses to use) and selecting tenancy as **Default.**



□	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
□		vpc-7d7ab214	available	172.31.0.0/16	
☑	Custom VPC	vpc-8b6984e3	available	192.168.0.0/16	

STEP 2: Creating a subnets under custom VPC (One public and one private subnets)

- Navigating to Subnets option and selecting **"Creating Subnet"** and giving name as **"Public Subnet"** where I want to deploy my Internet Facing instances.
- Creating this Subnet under Custom VPC, Select that option and select the **ap-south-1a** Availability Zone , Given a CIDR block as 192.168.1.0/24 (all instances launched under ap-south-1a will get the same range Private IP addresses and we'll get 251 usable IP addresses) and click on Create. Remember again, one subnet is equal to one AZ.



Create Subnet

Use the CIDR format to specify your subnet's IP address block (e.g., 10.0.0.0/24). Note that block sizes must be between a /16 netmask and /28 netmask. Also, note that a subnet can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag: Public Subnet

VPC: vpc-8b6984e3 | Custom VPC

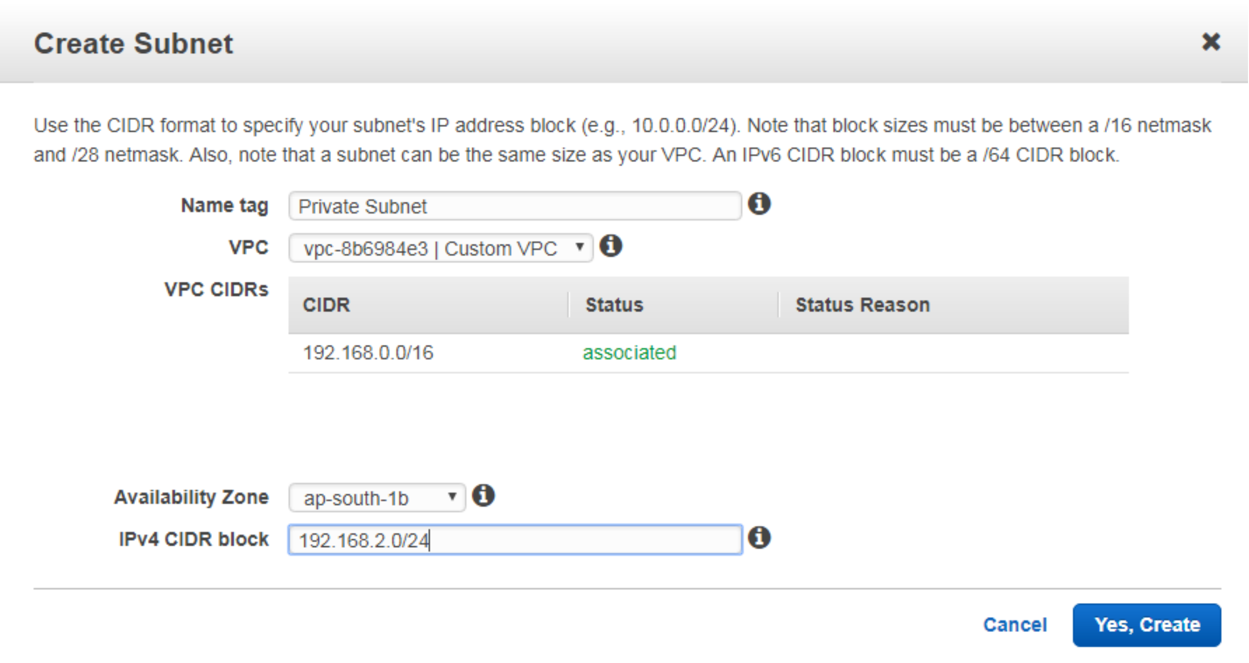
VPC CIDRs

CIDR	Status	Status Reason
192.168.0.0/16	associated	

Availability Zone: ap-south-1a

IPv4 CIDR block: 192.168.1.0/24

- Now creating another subnet and naming it as **"Private Subnet"** and want to deploy the instance which doesn't required internet faced.
- Creating this subnet under Custom VPC, and named as "Private Subnet" then provided CIDR as 192.168.2.0/24 and selecting Avaiablility Zone as **ap-south-1b** and click on Create option.



Create Subnet

Use the CIDR format to specify your subnet's IP address block (e.g., 10.0.0.0/24). Note that block sizes must be between a /16 netmask and /28 netmask. Also, note that a subnet can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag: Private Subnet

VPC: vpc-8b6984e3 | Custom VPC

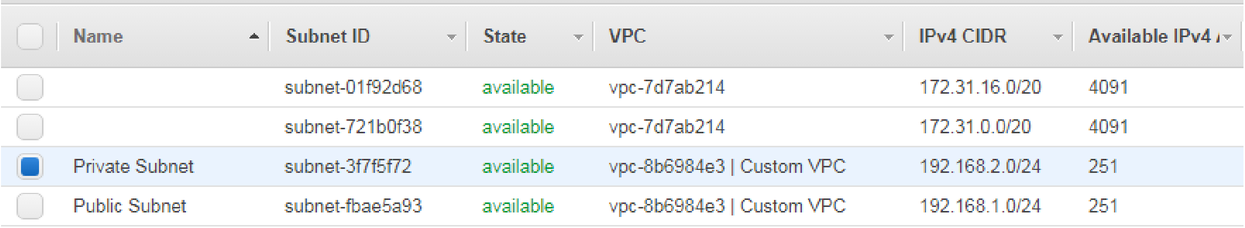
VPC CIDRs:

CIDR	Status	Status Reason
192.168.0.0/16	associated	

Availability Zone: ap-south-1b

IPv4 CIDR block: 192.168.2.0/24

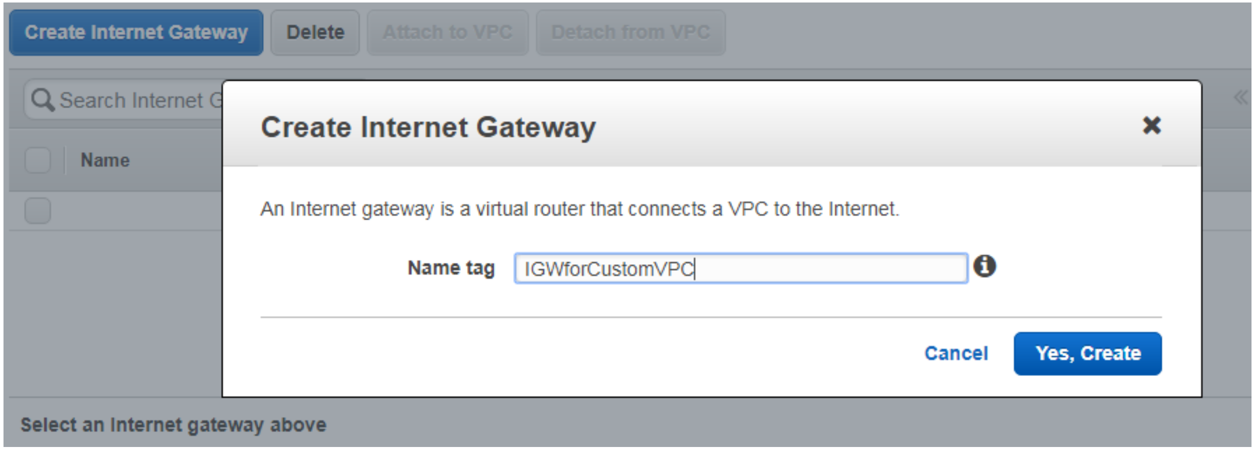
- This is how exactly subnet dashboard looks like now.



□	Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4
□		subnet-01f92d68	available	vpc-7d7ab214	172.31.16.0/20	4091
□		subnet-721b0f38	available	vpc-7d7ab214	172.31.0.0/20	4091
☑	Private Subnet	subnet-3f7f5f72	available	vpc-8b6984e3 | Custom VPC	192.168.2.0/24	251
□	Public Subnet	subnet-fbae5a93	available	vpc-8b6984e3 | Custom VPC	192.168.1.0/24	251

STEP 3: Creating an Internet gateway and Associating with Custom VPC.

- Navigate to internet Gateways from Navigation pane and Select **“Create Internet gateway**” option and provide a name for Internet Gateway.

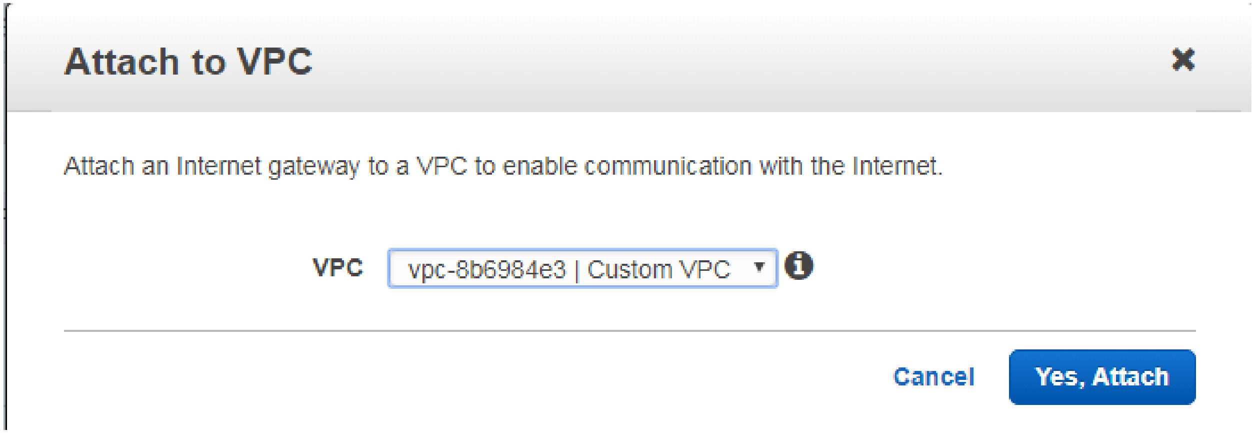


Create Internet Gateway

An Internet gateway is a virtual router that connects a VPC to the Internet.

Name tag: IGWforCustomVPC

- And select the "**Attach to VPC**" option and select the Custom VPC and click on **"Yes, Attach"** option.

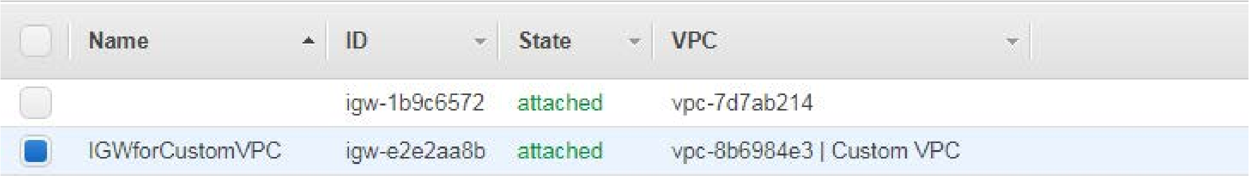


Attach to VPC

Attach an Internet gateway to a VPC to enable communication with the Internet.

VPC: vpc-8b6984e3 | Custom VPC

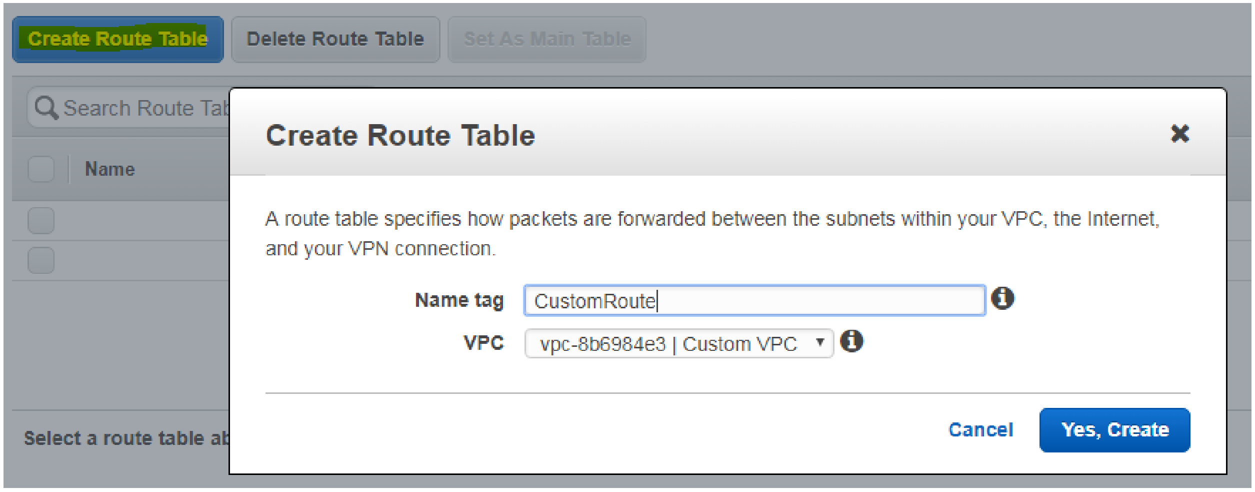
- This is how the IGW dashboard looks like after attaching it to custom VPC. Remember: One Internet gateway can be attached with only one VPC.



	Name	ID	State	VPC
□		igw-1b9c6572	attached	vpc-7d7ab214
☑	IGWforCustomVPC	igw-e2e2aa8b	attached	vpc-8b6984e3 | Custom VPC

STEP 4: Creating Route Table and Performing Subnet association.

- Till now we have created a Custom VPC, Private and Public subnets, Created internet gateway and associated that to our custom VPC. Now we need to allow the traffic to our newly created subnets through the internet gateway, for that we are going to create a Route Table.
- Select **"Create Route Table"** option and give a name tag and select the Custom VPC and click on **"Yes, Create"** option.



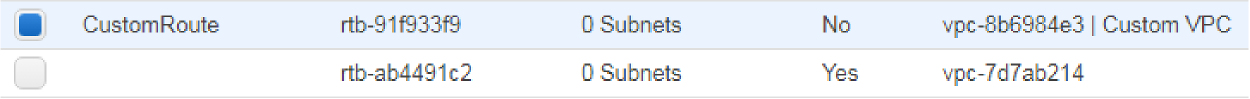
Create Route Table

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

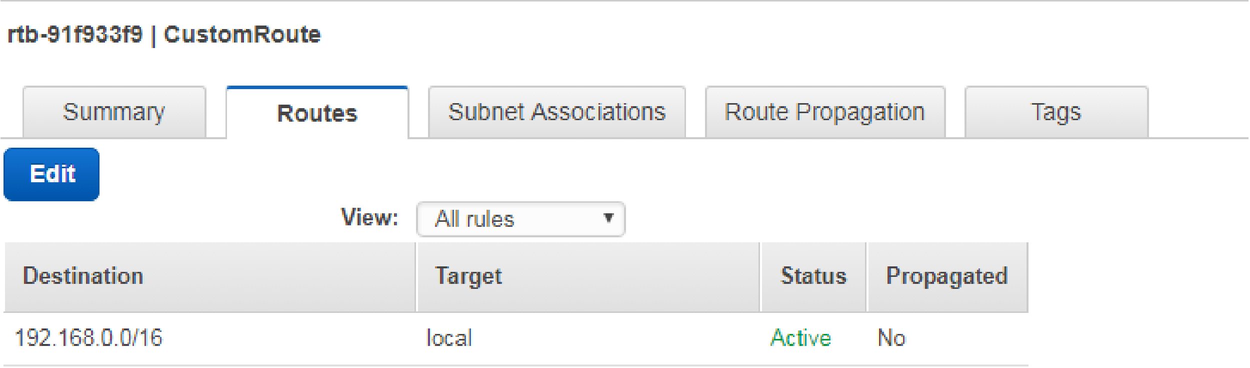
Name tag: CustomRoute

VPC: vpc-8b6984e3 | Custom VPC

- Newly created route is not enabled with any of the public routes through IGW, Select the newly created route table to choose Route option to verify this.



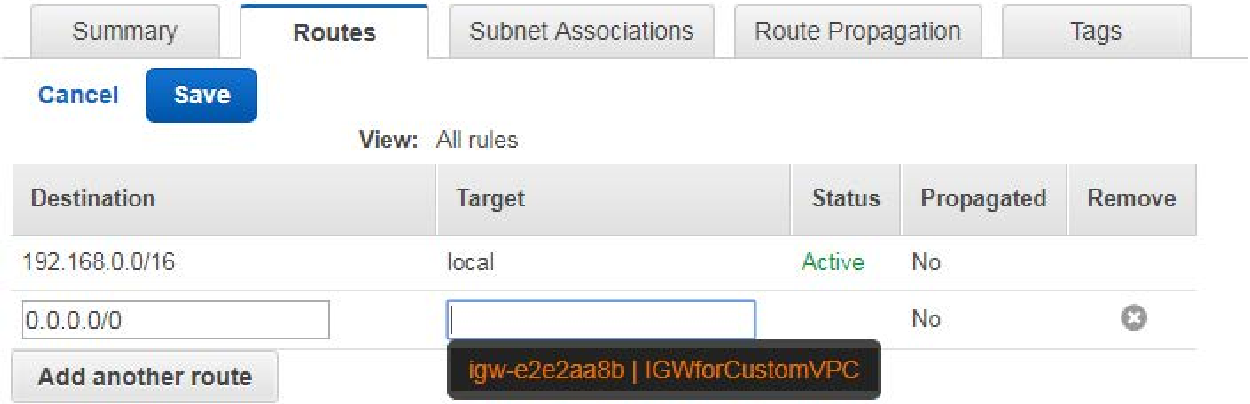
■	CustomRoute	rtb-91f933f9	0 Subnets	No	vpc-8b6984e3 | Custom VPC
□		rtb-ab4491c2	0 Subnets	Yes	vpc-7d7ab214



rtb-91f933f9 | CustomRoute

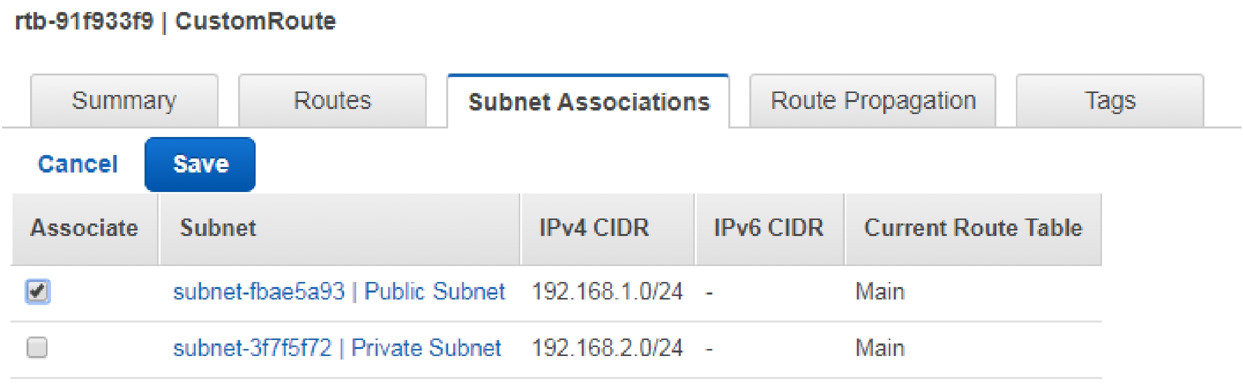
Destination	Target	Status	Propagated
192.168.0.0/16	local	Active	No

- Now we have to add a route by selecting edit option and select "**Add another Route"** option and enter **0.0.0.0/0** and when you click on Target automatically internet gateway will populate, choose **the populated IGW** and click on **save**.



Destination	Target	Status	Propagated	Remove
192.168.0.0/16	local	Active	No	
0.0.0.0/0			No	⊗

- Then select the **"Subnet Association"** ad click on **"Edit"** option and select the **"Public Subnet"** and click on save.



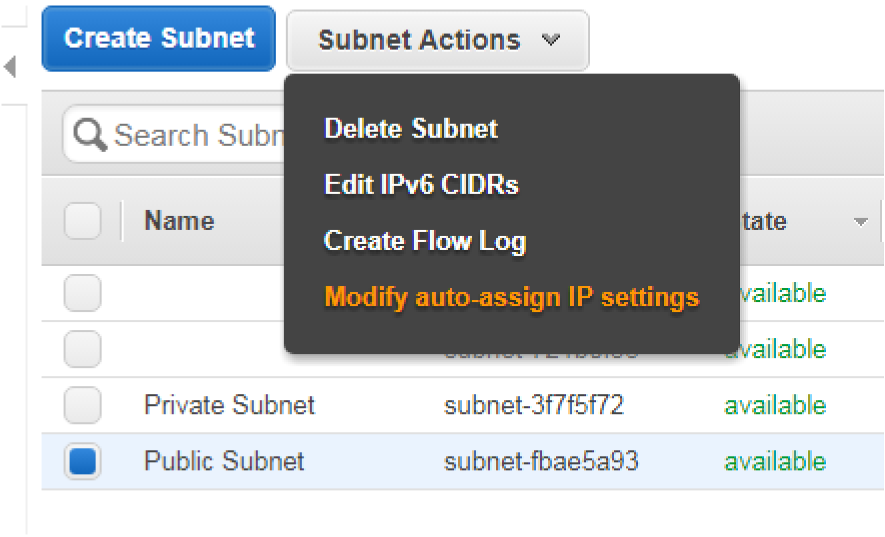
rtb-91f933f9 | CustomRoute

Associate	Subnet	IPv4 CIDR	IPv6 CIDR	Current Route Table
☑	subnet-fbae5a93 | Public Subnet	192.168.1.0/24	-	Main
□	subnet-3f7f5f72 | Private Subnet	192.168.2.0/24	-	Main

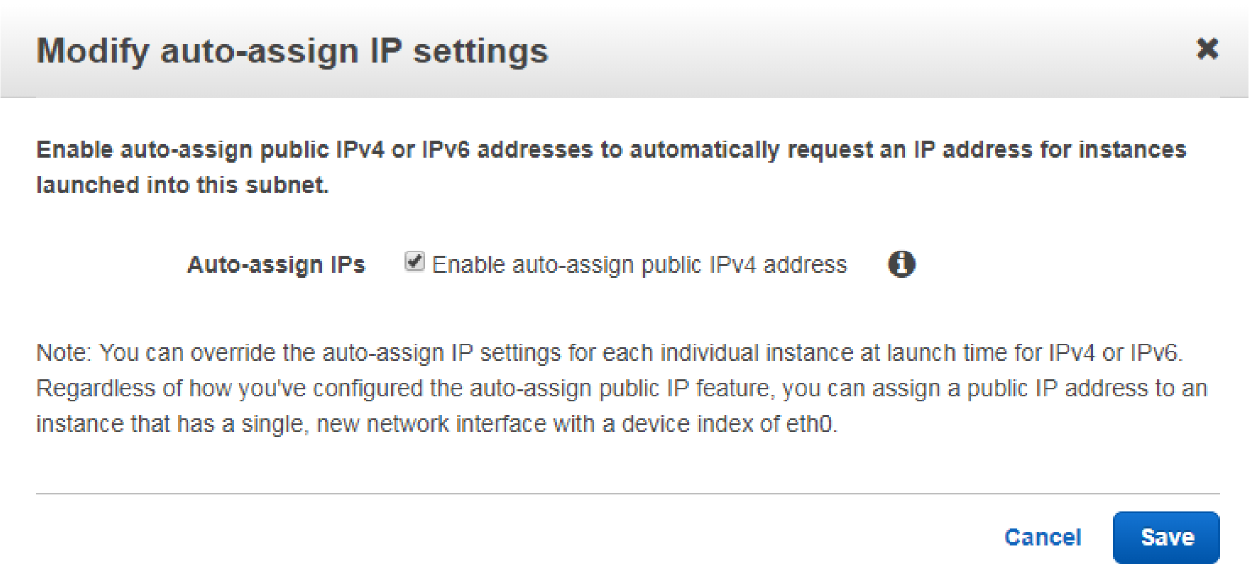
That's it our custom VPC is ready to deploy the resources. But we have one additional option.

STEP 5: Enabling Auto-assign IP Settings for Public Subnet (Optional Step).

You can enable auto assign public IP address option for Public Subnet instances, by editing the subnet settings. Navigate to Subnets dashboard and select the **"Public Subnet"** and choose the **"Subnet Actions"** and choose "**Modify auto-assign IP settings"**, select the checkbox and click on save.



	Name		tate
□			vailable
□			available
□	Private Subnet	subnet-3f7f5f72	available
☑	Public Subnet	subnet-fbae5a93	available



Modify auto-assign IP settings

Enable auto-assign public IPv4 or IPv6 addresses to automatically request an IP address for instances launched into this subnet.

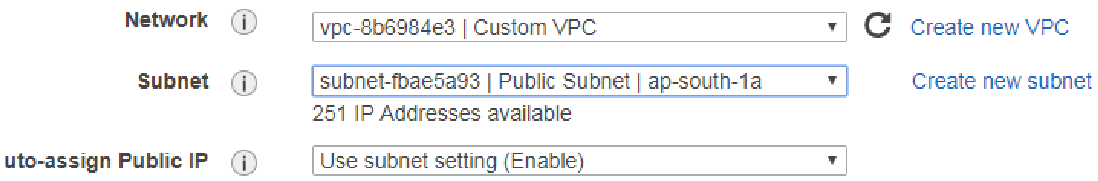
Auto-assign IPs ☑ Enable auto-assign public IPv4 address

Note: You can override the auto-assign IP settings for each individual instance at launch time for IPv4 or IPv6. Regardless of how you've configured the auto-assign public IP feature, you can assign a public IP address to an instance that has a single, new network interface with a device index of eth0.

- Now we will get public IP address for every instance when we are launching it under public subnet, we no need to select the option in instance launch wizard.

Now Launch Instances in newly created custom VPC and verify.

1. Launching an Instance in Custom VPC and selected to launch under "Public Subnet".

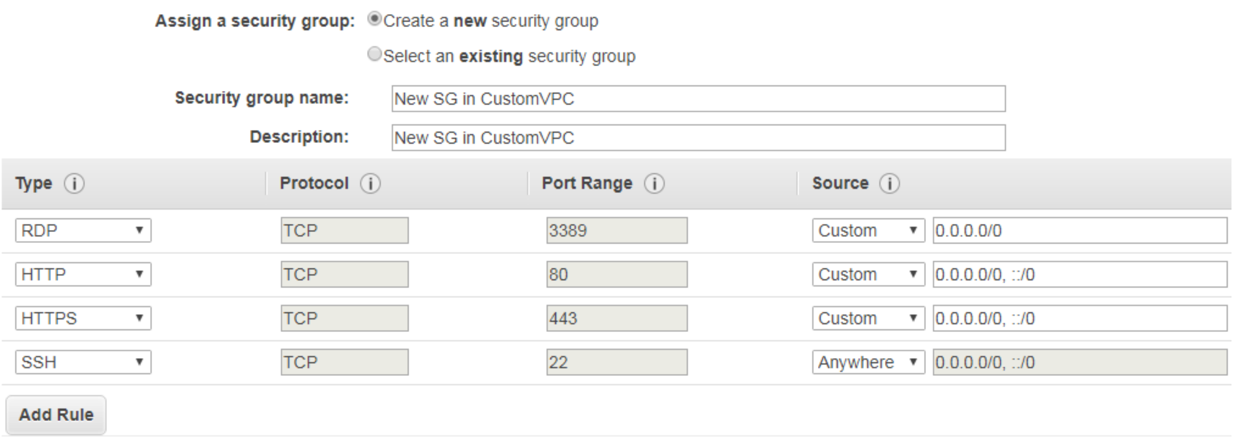


Network: vpc-8b6984e3 | Custom VPC

Subnet: subnet-fbae5a93 | Public Subnet | ap-south-1a
251 IP Addresses available

uto-assign Public IP: Use subnet setting (Enable)

2. As this is a first instance launching under Custom VPC, we have to create new security group and need to open required ports and protocols.



Assign a security group: ● Create a **new** security group

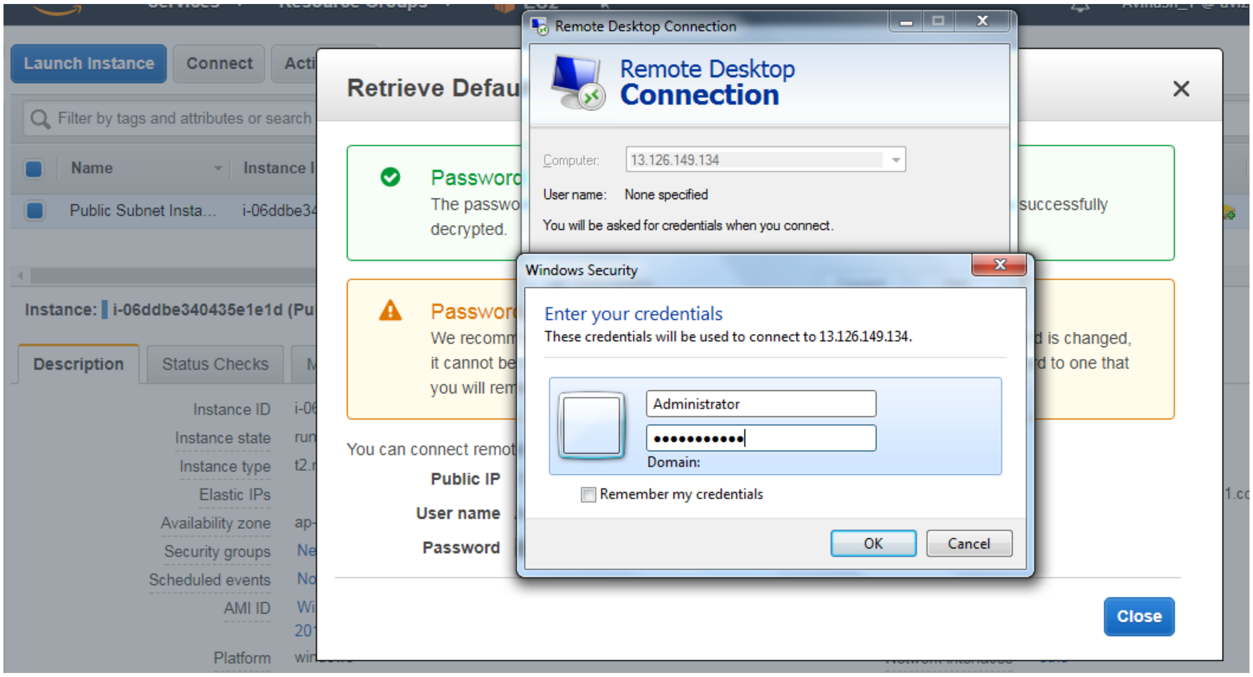
○ Select an **existing** security group

Security group name: New SG in CustomVPC

Description: New SG in CustomVPC

Type	Protocol	Port Range	Source	
RDP	TCP	3389	Custom	0.0.0.0/0
HTTP	TCP	80	Custom	0.0.0.0/0, ::/0
HTTPS	TCP	443	Custom	0.0.0.0/0, ::/0
SSH	TCP	22	Anywhere	0.0.0.0/0, ::/0

3. Now try to connect to the instance over the internet and verify the status as this is launched in Public Subnet, you can connect without any issues and you can browse the internet also in Instance.



Remote Desktop Connection

Computer: 13.126.149.134

User name: None specified

You will be asked for credentials when you connect.

Windows Security

Enter your credentials

These credentials will be used to connect to 13.126.149.134.

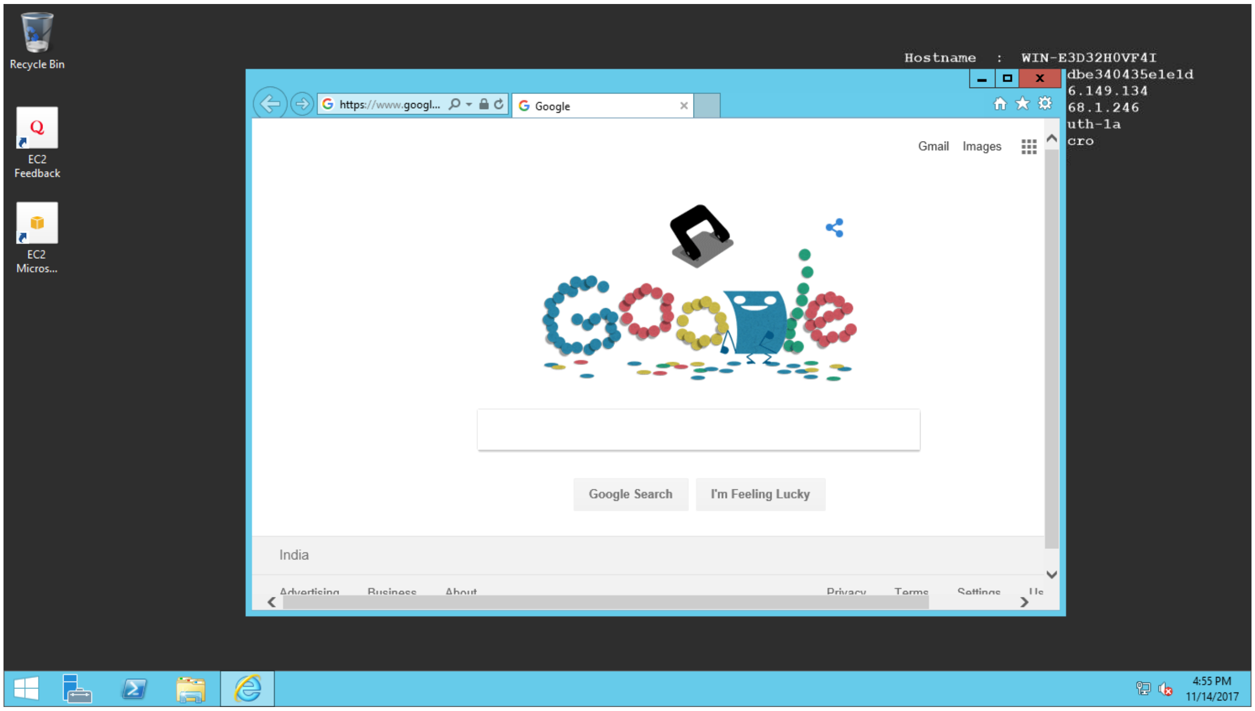
Administrator

•••••••••••

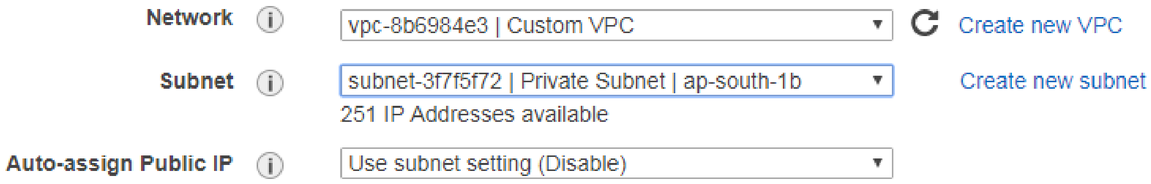
Domain:

□ Remember my credentials

And we have successfully connected to the Instance, That means this instance is internet-faced and we can access anywhere from the world.

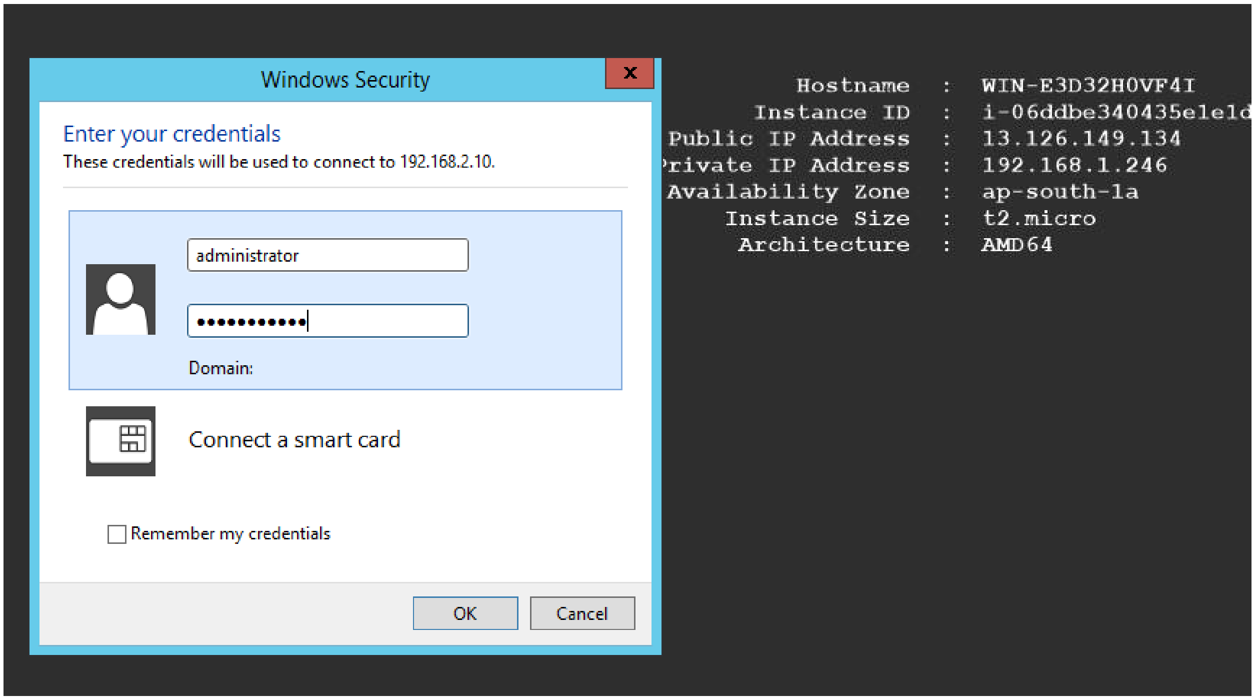


4. Now Launching another Instance in **"Custom VPC"** and selected to launch under **"Private Subnet"**.



Network	vpc-8b6984e3 | Custom VPC
Subnet	subnet-3f7f5f72 | Private Subnet | ap-south-1b 251 IP Addresses available
Auto-assign Public IP	Use subnet setting (Disable)

5. And try to connect to the Private Subnet launched instance. When you browse for Username and password for instance connectivity, you'll get a Private IP address and we cannot use this to connect to the Launched instance.
   a. But we can connect to the same instance from the Public Subnets launched Instance.
   b. Remember as this is a private subnet instance, we will not get Internet in the Private Subnet instances.



Enter your credentials

These credentials will be used to connect to 192.168.2.10.

administrator

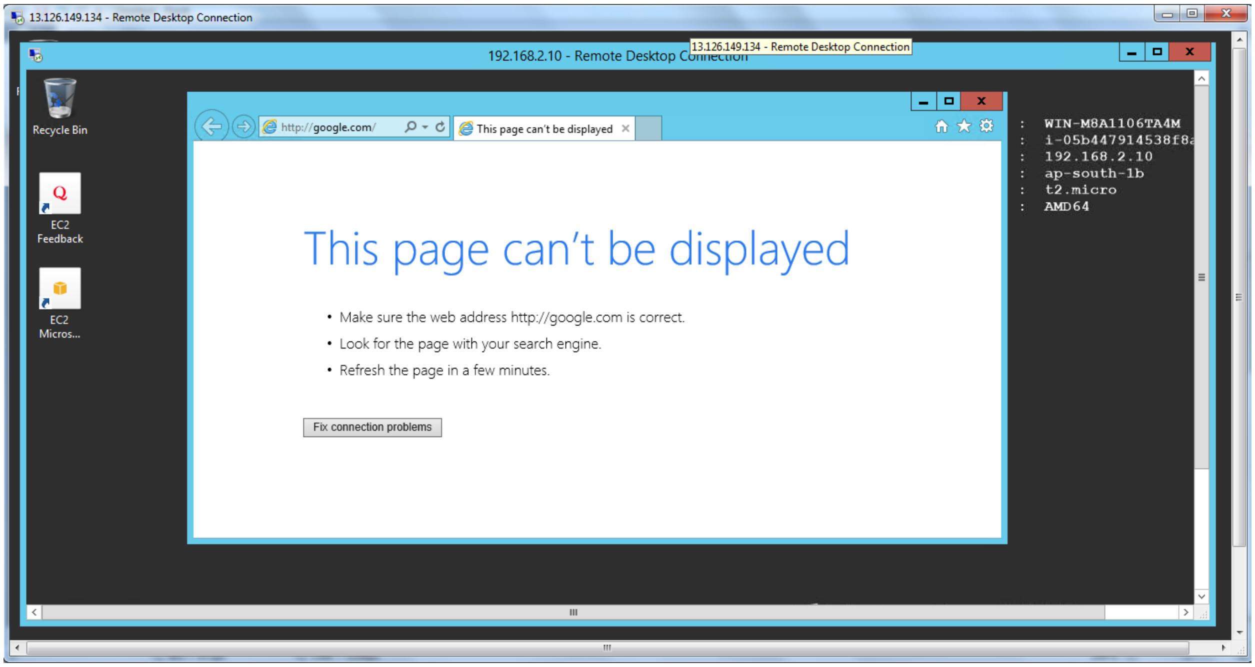
•••••••••••

Domain:

Connect a smart card

□ Remember my credentials

```
       Hostname  :  WIN-E3D32H0VF4I
    Instance ID  :  i-06ddbe340435e1e1d
Public IP Address  :  13.126.149.134
rivate IP Address  :  192.168.1.246
Availability Zone  :  ap-south-1a
   Instance Size  :  t2.micro
   Architecture  :  AMD64
```


13.126.149.134 - Remote Desktop Connection
192.168.2.10 - Remote Desktop Connection
13.126.149.134 - Remote Desktop Connection
Recycle Bin
EC2 Feedback
EC2 Micros...
http://google.com/
This page can't be displayed
Fix connection problems

This page can't be displayed

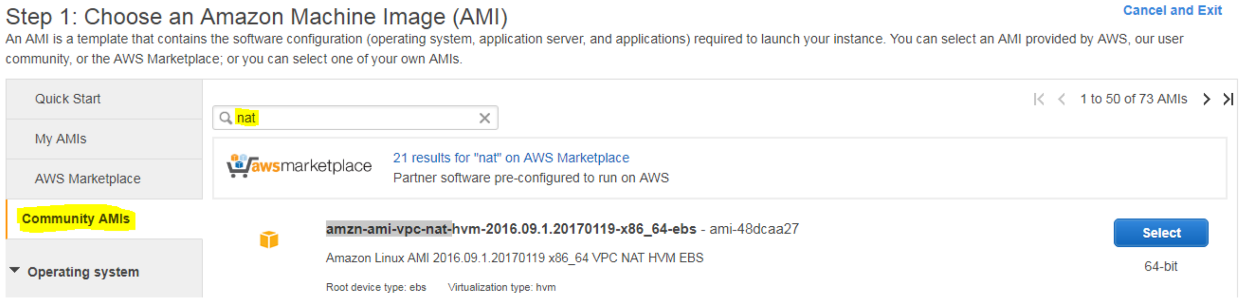
- Make sure the web address http://google.com is correct.
- Look for the page with your search engine.
- Refresh the page in a few minutes.

```
:  WIN-M8A1106TA4M
:  i-05b447914538f8a
:  192.168.2.10
:  ap-south-1b
:  t2.micro
:  AMD64
```

We have successfully connected to the Private Subnet instance from public Subnet instance, But We are not able to get internet connectivity in private subnet instance. TO get Internet in private Hosted instances we need to **launch a NAT Instance or NAT gateway.**

Launching NAT Instance:

- To launch NAT instance go to EC2 Dashboard and initiate an instance launch and Select **“Community AMI”** and Search for **“NAT”** as shown in below image and choose any of the instance.



Step 1: Choose an Amazon Machine Image (AMI)

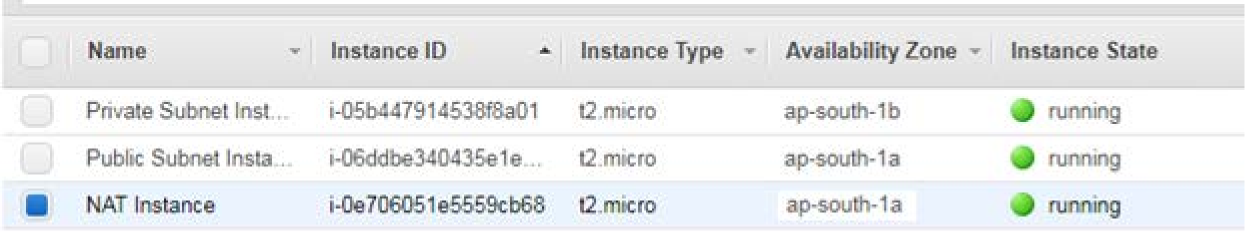
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

nat

21 results for "nat" on AWS Marketplace
Partner software pre-configured to run on AWS

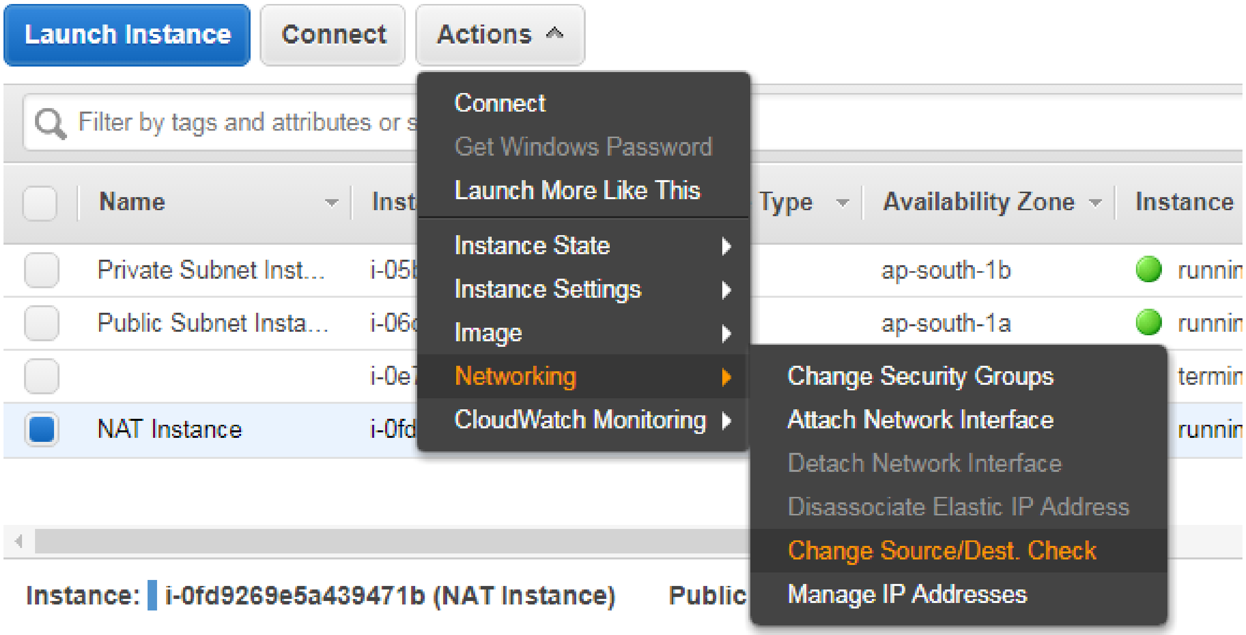
amzn-ami-vpc-nat-hvm-2016.09.1.20170119-x86_64-ebs - ami-48dcaa27
Amazon Linux AMI 2016.09.1.20170119 x86_64 VPC NAT HVM EBS
Root device type: ebs Virtualization type: hvm

- Select one of the instances from the listed instances, and choose NAT instance with t2.micro and follow the instance launch wizard same as a regular instance.
  Note: The amount of traffic that NAT instances supports, depends on the instance size. If you are bottlenecking, increase the instance configuration.
  Note: Make sure your NAT instance security group is opened with Http and Https.
  Note: NAT Instance must be launched in **Custom VPC’s Public Subnet.**

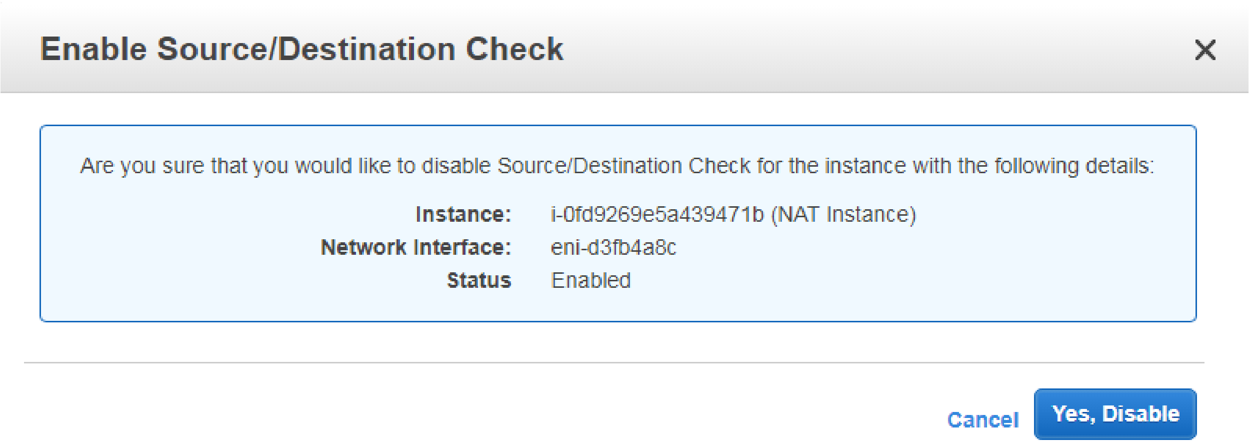


	Name	Instance ID	Instance Type	Availability Zone	Instance State
□	Private Subnet Inst...	i-05b447914538f8a01	t2.micro	ap-south-1b	● running
□	Public Subnet Insta...	i-06ddbe340435e1e...	t2.micro	ap-south-1a	● running
■	NAT Instance	i-0e706051e5559cb68	t2.micro	ap-south-1a	● running

- We need to disable Source/Destination check for NAT instance.
  Each EC2 instance performs source/destination checks by default. This means that the instance must be the source or destination of any traffic it sends or receives. However, a NAT instance must be able to send and receive traffic when the source or destination is not itself. Therefore, you must disable source/destination checks on the NAT instance.
- To disable source/destination check, Select the NAT Instance, Goto Actions, Networking and choose **“Change Source/Destination Check”** and select **“Yes, Disable”.**



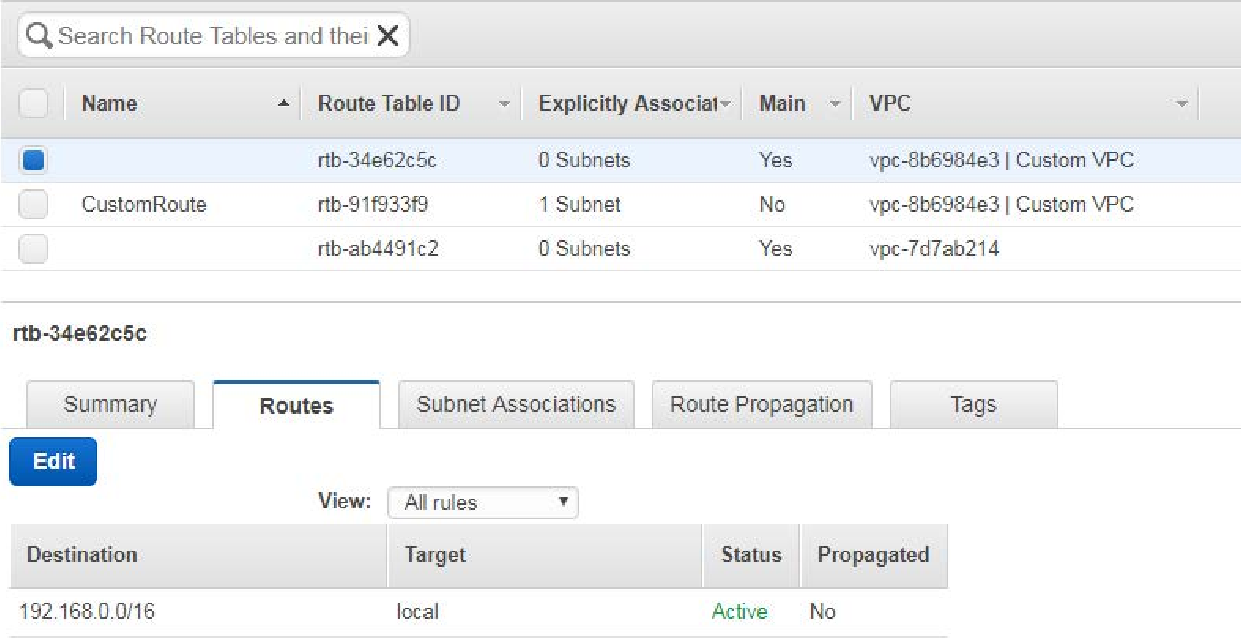
Instance: i-0fd9269e5a439471b (NAT Instance)



Are you sure that you would like to disable Source/Destination Check for the instance with the following details:

Instance: i-0fd9269e5a439471b (NAT Instance)
Network Interface: eni-d3fb4a8c
Status Enabled

- Now we have to edit **"Custom VPCs Main Route table"** and need to add a route through the NAT Instance, then the private subnet instances will get the internet connectivity.



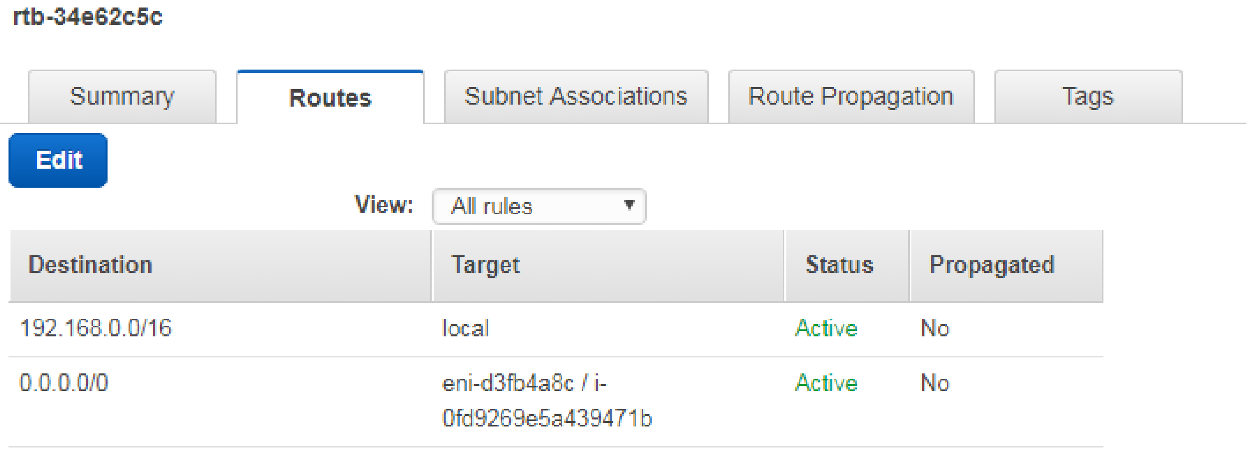
□	Name	Route Table ID	Explicitly Associat	Main	VPC
☑		rtb-34e62c5c	0 Subnets	Yes	vpc-8b6984e3 | Custom VPC
□	CustomRoute	rtb-91f933f9	1 Subnet	No	vpc-8b6984e3 | Custom VPC
□		rtb-ab4491c2	0 Subnets	Yes	vpc-7d7ab214

rtb-34e62c5c

View: All rules

Destination	Target	Status	Propagated
192.168.0.0/16	local	Active	No

- Select the Edit option and enter the Destination as **0.0.0.0/0** and select the target as **NAT Instance**.


rtb-34e62c5c

View: All rules

Destination	Target	Status	Propagated
192.168.0.0/16	local	Active	No
0.0.0.0/0	eni-d3fb4a8c / i-0fd9269e5a439471b	Active	No

- Now we will get the internet for our Private subnet instances through the NAT instances. And here is the output.

13.127.13.134 - Remote Desktop Connection
192.168.2.10 - Remote Desktop Connection
13.127.13.134 - Remote Desktop Connection
EC2 Feedback
EC2 Micros...
https://aws.amazon...
Amazon Web Services (AW...
Menu
aws
English
My Account
Create an AWS Account
Create A Free Account
View AWS Free Tier Details »
R2

Start Building on AWS Today

Whether you're looking for compute power, database storage, content delivery or other functionality, AWS has the services to help you build sophisticated applications with increased flexibility, scalability and reliability.

Broad & Deep Platform
AWS has more than 90 services and is continually launching new features

Customer Success
Explore how millions of active customers every month are innovating with

Pace of Innovation
The AWS Cloud platform expands daily. Take a look at what we launched this

Global Infrastructure
AWS operates 43 Availability Zones within 16 geographic Regions

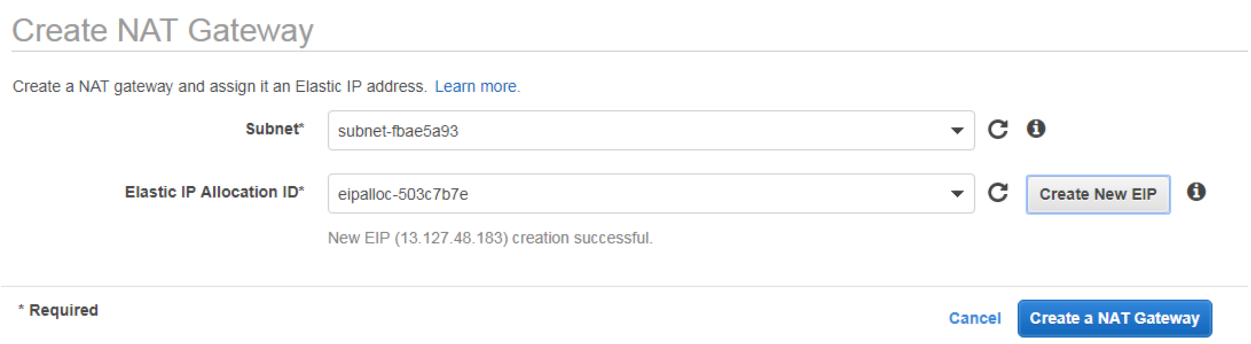
NAT GATEWAYS: Instead of NAT Instances, we can use NAT Gateways. We have lot of advantages with NAT gateways compare to NAT instances. Make sure you terminate the NAT Instance before performing the NAT Gateways, we don't required two resources to provide internet to Private subnet.

Here is some advantages listed:

- ➢ Preferred for the enterprise/Production level
- ➢ Scale automatically up to 10 Gbps
- ➢ Not associated with security groups
- ➢ Automatically assigned a public ip address (EIP)
- ➢ You have to update route tables to take effect.
- ➢ No O.S so No need to patch
- ➢ No Instance so No need to disable Source/Destination Checks

Steps to create NAT gateways:

- Select NAT Gateways option from VPC Navigation Pane. And click on **"Create NAT Gateway"** option.
- As same as NAT instance, we have to create the NAT Gateway also in **Public Subnet of Custom VPC**.
- If you have any Elastic IP without associating to any of the resource, we can use the same here, if you don't have select the **Create New EIP** option and click on **Create a NAT Gateway**.



Create NAT Gateway

Create a NAT gateway and assign it an Elastic IP address. Learn more.

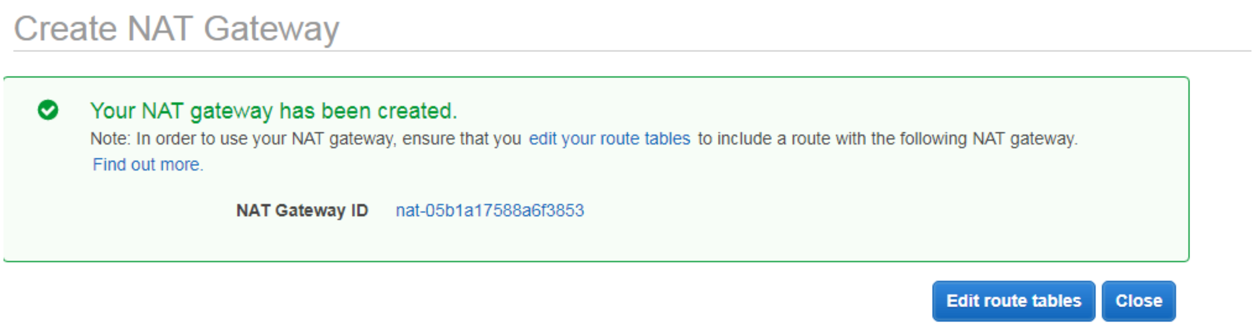
Subnet*: subnet-fbae5a93

Elastic IP Allocation ID*: eipalloc-503c7b7e

New EIP (13.127.48.183) creation successful.

* Required

- And we have to edit the Route table as same as NAT instance process. Select the Custom VPCs Main Route table and open the Destination **0.0.0.0/0** and target as **NAT Gateway.**

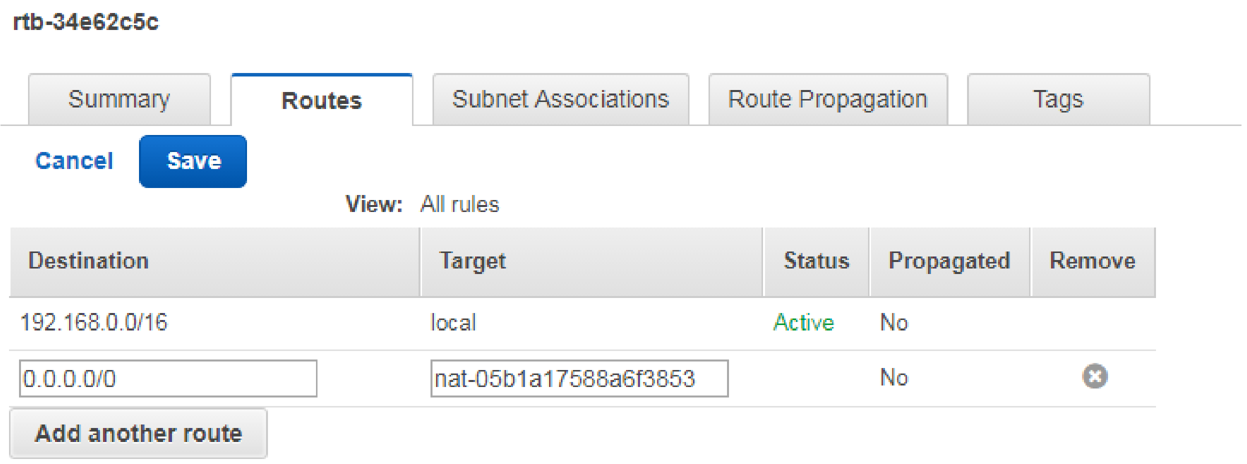


Create NAT Gateway

Your NAT gateway has been created.

Note: In order to use your NAT gateway, ensure that you edit your route tables to include a route with the following NAT gateway. Find out more.

NAT Gateway ID nat-05b1a17588a6f3853

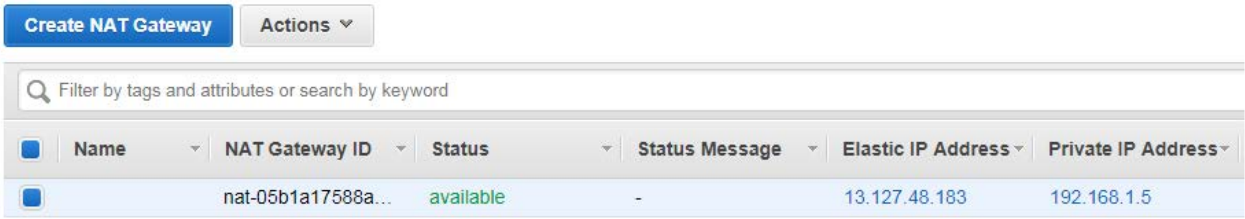


rtb-34e62c5c

View: All rules

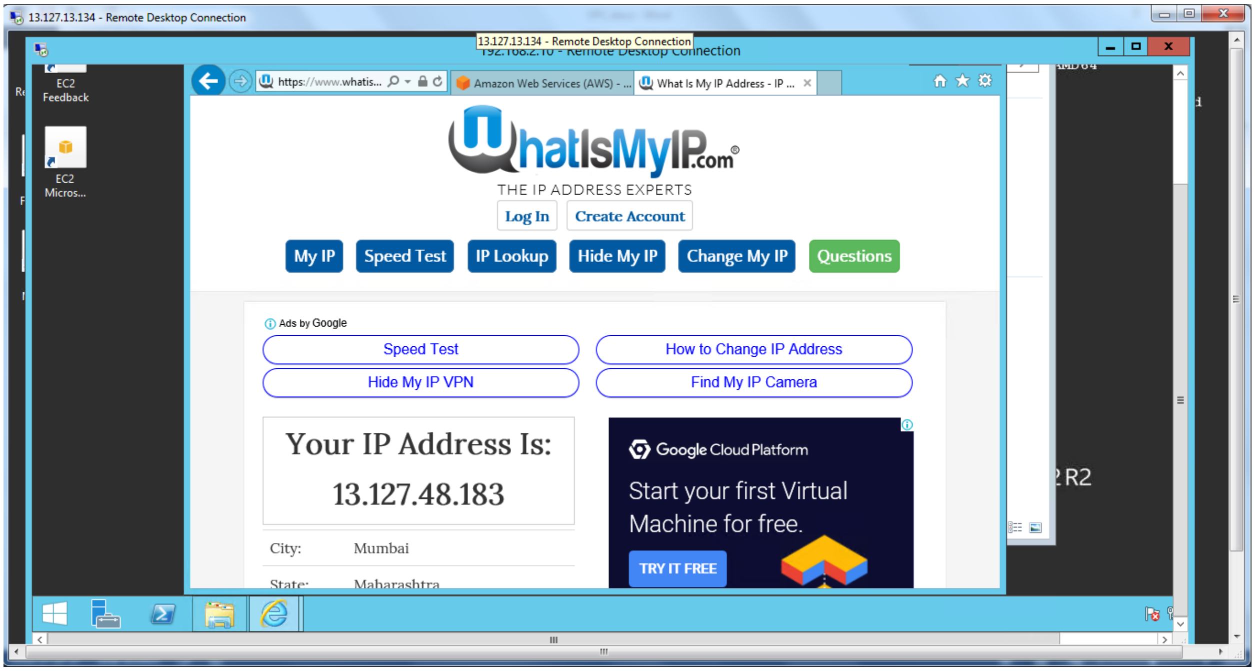
Destination	Target	Status	Propagated	Remove
192.168.0.0/16	local	Active	No	
0.0.0.0/0	nat-05b1a17588a6f3853		No	⊗

- Here is the NAT Gateway information after creation.



Name	NAT Gateway ID	Status	Status Message	Elastic IP Address	Private IP Address
	nat-05b1a17588a...	available	-	13.127.48.183	192.168.1.5

- Now go to private subnet instance and verify the internet connectivity. You will able to browse the internet and try to look for the public Ip information from the private subnet instance you'll get the NAT gateway's IP Address, That means we are getting internet through NAT Gateway to the Private subnet instance.



Your IP Address Is:
13.127.48.183

City: Mumbai

State: Maharashtra

Network Access Control Lists (ACLs)

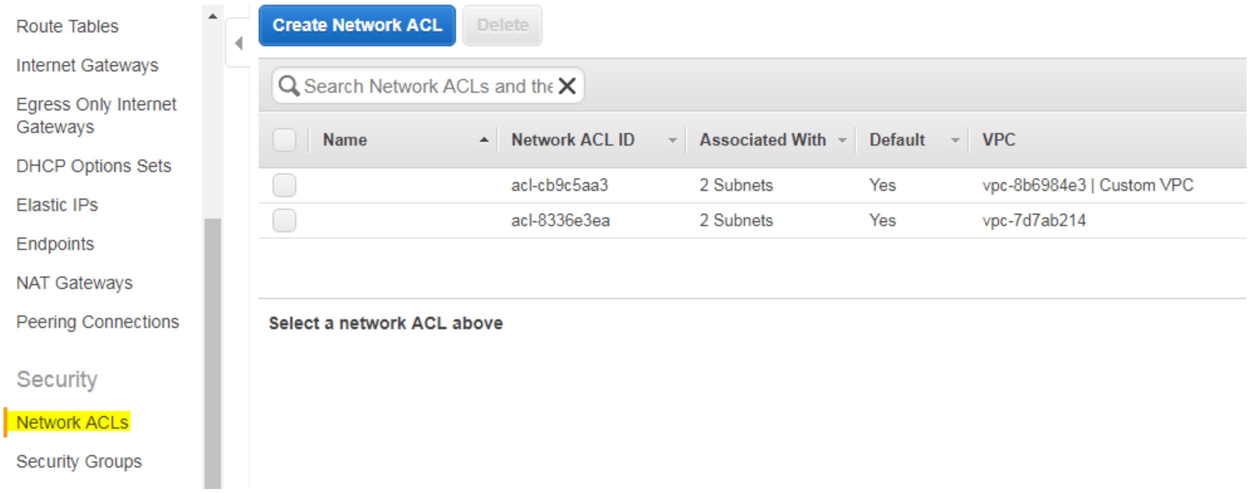
A network access control list (ACL) is another layer of security that acts as a stateless firewall on a subnet level. A network ACL is a numbered list of rules that AWS evaluates in order, starting with the lowest numbered rule, to determine whether traffic is allowed in or out of any subnet associated with the network ACL.

Every subnet must be associated with a network ACL.

Security Groups Vs Network ACLs

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

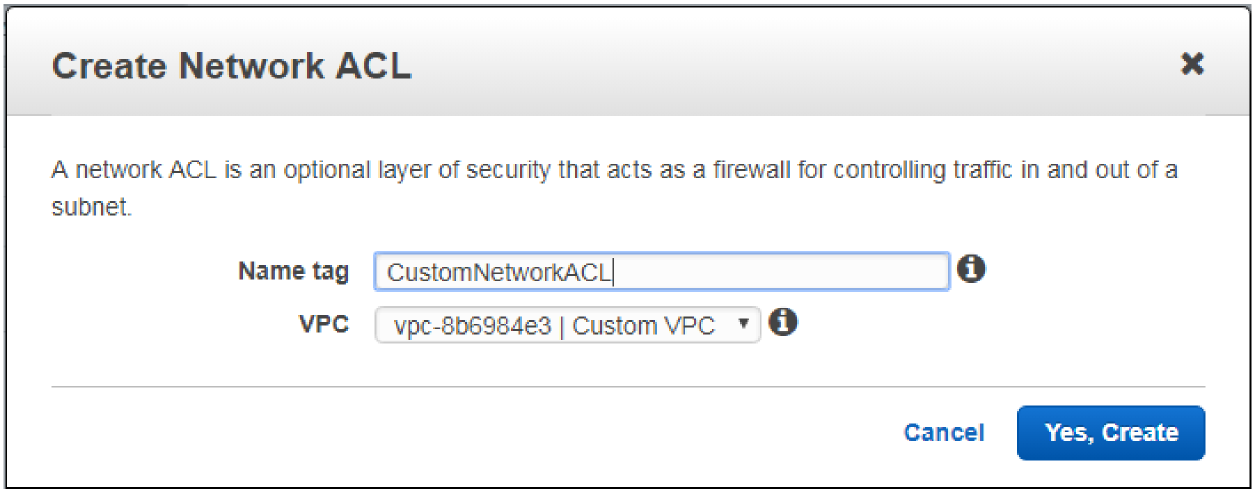
- ➢ Navigate to the "Network ACLs" under "Security" option and choose "Create Network ACL" option.



☐	Name	Network ACL ID	Associated With	Default	VPC
☐		acl-cb9c5aa3	2 Subnets	Yes	vpc-8b6984e3 | Custom VPC
☐		acl-8336e3ea	2 Subnets	Yes	vpc-7d7ab214

Select a network ACL above

- ➢ Give a name for the newly creating Network ACL and Create this under Custom VPC.



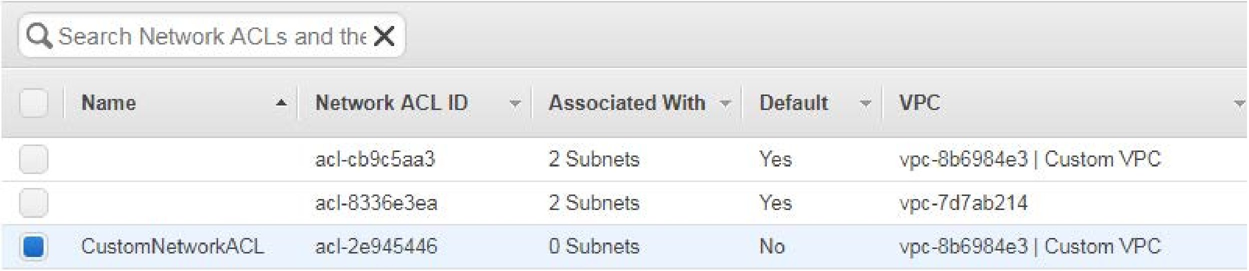
Create Network ACL

A network ACL is an optional layer of security that acts as a firewall for controlling traffic in and out of a subnet.

Name tag: CustomNetworkACL

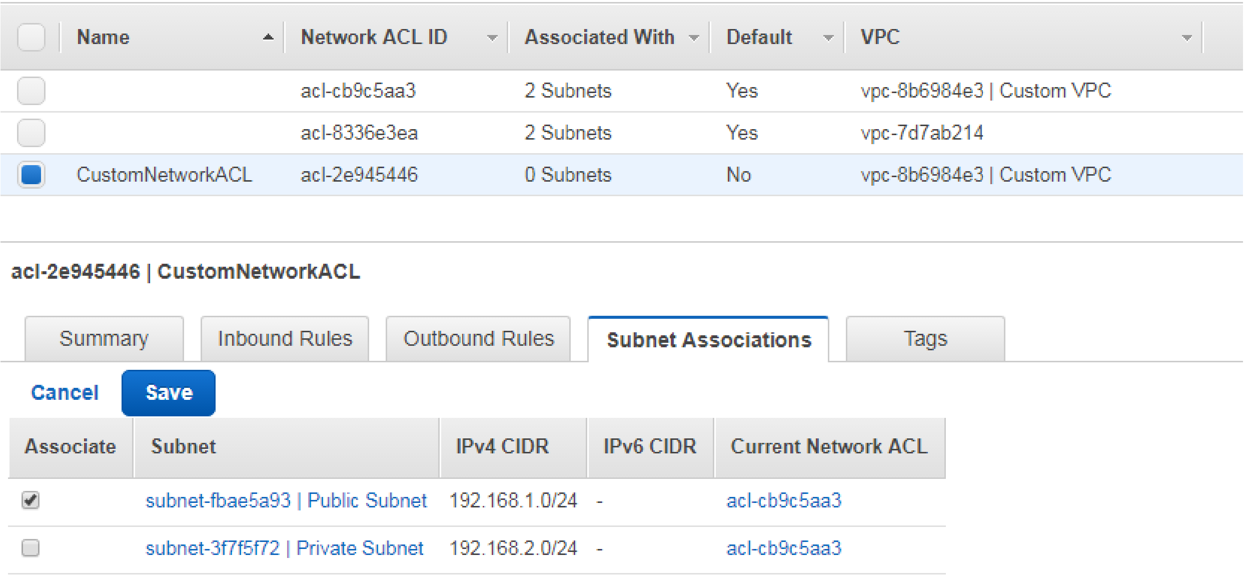
VPC: vpc-8b6984e3 | Custom VPC

- ➢ Newly Created NACL will not have any Subnets Associated with it.



	Name	Network ACL ID	Associated With	Default	VPC
□		acl-cb9c5aa3	2 Subnets	Yes	vpc-8b6984e3 | Custom VPC
□		acl-8336e3ea	2 Subnets	Yes	vpc-7d7ab214
■	CustomNetworkACL	acl-2e945446	0 Subnets	No	vpc-8b6984e3 | Custom VPC

- ➢ To Associate a subnet Select the “Subnet Association” and choose the subnet you want to associate under the “Custom Network ACL”.

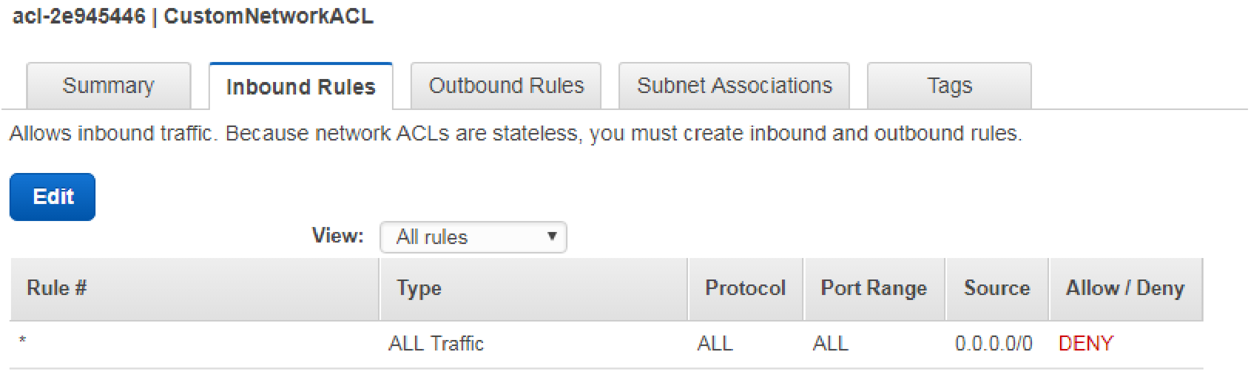


	Name	Network ACL ID	Associated With	Default	VPC
□		acl-cb9c5aa3	2 Subnets	Yes	vpc-8b6984e3 | Custom VPC
□		acl-8336e3ea	2 Subnets	Yes	vpc-7d7ab214
■	CustomNetworkACL	acl-2e945446	0 Subnets	No	vpc-8b6984e3 | Custom VPC

acl-2e945446 | CustomNetworkACL

Associate	Subnet	IPv4 CIDR	IPv6 CIDR	Current Network ACL
☑	subnet-fbae5a93 | Public Subnet	192.168.1.0/24	-	acl-cb9c5aa3
□	subnet-3f7f5f72 | Private Subnet	192.168.2.0/24	-	acl-cb9c5aa3

- By Default, all the Inbound and outbound traffic will be set to Deny mode.



acl-2e945446 | CustomNetworkACL

Allows inbound traffic. Because network ACLs are stateless, you must create inbound and outbound rules.

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

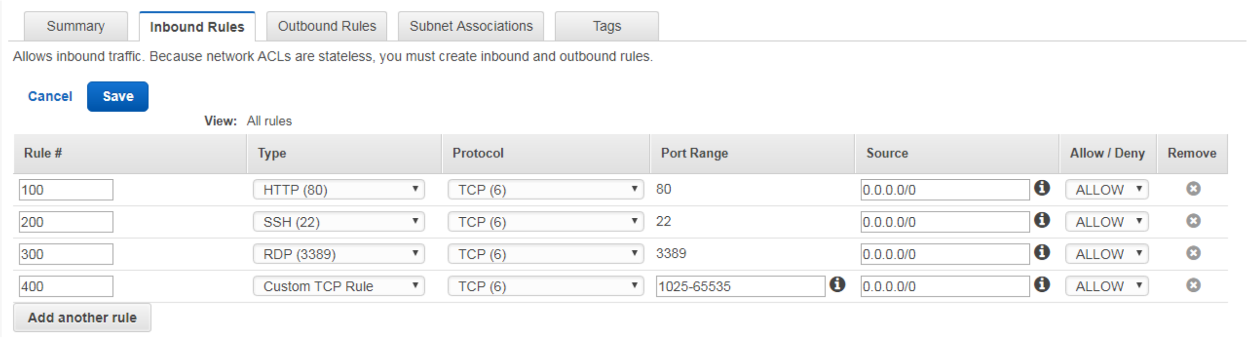
- Here we have to Edit and add the required Protocol and Port Range and Source same as Security groups.
  The following are the parts of a network ACL rule:

  Rule number: Rules are evaluated starting with the lowest numbered rule. As soon as a rule matches traffic, it's applied regardless of any higher-numbered rule that may contradict it.

  Protocol: You can specify any protocol that has a standard protocol number. For more information, see Protocol Numbers. If you specify ICMP as the protocol, you can specify any or all of the ICMP types and codes.
  [Inbound rules only] The source of the traffic (CIDR range) and the destination (listening) port or port range.
  [Outbound rules only] The destination for the traffic (CIDR range) and the destination port or port range.
  Choice of **ALLOW or DENY** for the specified traffic.

- And AWS will suggest to create the rules increments of 100.
- If you want to use this Network ACL with Elastic Load balancers, open the Ephemeral ports in inbound and outbound.

  Ephemerals port range varies depending on the client's operating system.
  Many Linux kernels use ports 32768-61000.
  Elastic Load Balancing use ports 1024-65535.
  Windows Server 2008 and later versions use ports 49152-65535.
  A NAT gateway uses ports 1024-65535.

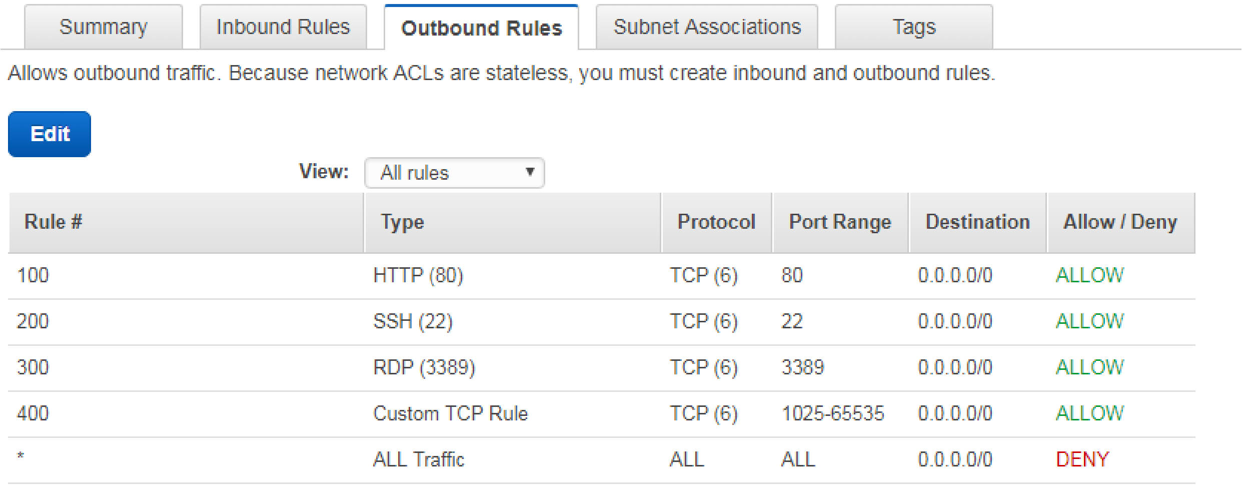


Allows inbound traffic. Because network ACLs are stateless, you must create inbound and outbound rules.

View: All rules

Rule #	Type	Protocol	Port Range	Source	Allow / Deny	Remove
100	HTTP (80)	TCP (6)	80	0.0.0.0/0	ALLOW	⊗
200	SSH (22)	TCP (6)	22	0.0.0.0/0	ALLOW	⊗
300	RDP (3389)	TCP (6)	3389	0.0.0.0/0	ALLOW	⊗
400	Custom TCP Rule	TCP (6)	1025-65535	0.0.0.0/0	ALLOW	⊗

- ➢ Perform the same for Outbound Rules also, as the Network ACLs are Stateless.



Allows outbound traffic. Because network ACLs are stateless, you must create inbound and outbound rules.

View: All rules

Rule #	Type	Protocol	Port Range	Destination	Allow / Deny
100	HTTP (80)	TCP (6)	80	0.0.0.0/0	ALLOW
200	SSH (22)	TCP (6)	22	0.0.0.0/0	ALLOW
300	RDP (3389)	TCP (6)	3389	0.0.0.0/0	ALLOW
400	Custom TCP Rule	TCP (6)	1025-65535	0.0.0.0/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

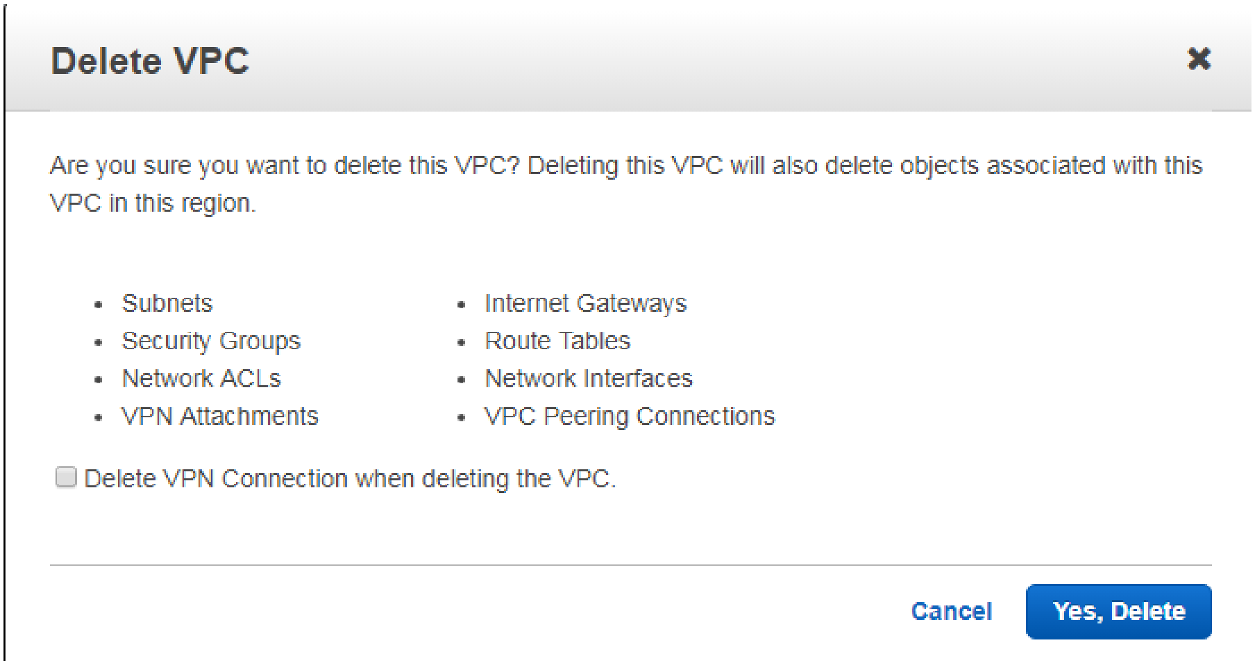
- ➢ We have Deny option also here with Network ACLs. We can create another rule for same Protocol and we can set it to Allow/Deny based on our requirement. **Lowest Rule will takes the Highest Priority.**

VPC Peering

- ➢ Allows you to connect one VPC with another via a direct network route using private IP addresses.
- ➢ Instances behave as if they were on the same private network
- ➢ You can peer VPC's with other AWS accounts as well as with other VPCs in the same account.
- ➢ Peering is in a star configuration, ie 1 central VPC peers with 4 others. NO TRANSITIVE PEERING!!!

VPC Cleanup:

When you delete the VPC, Automatically all the resources attached to the VPC also deletes. As mentioned below image, Subnets, Security groups, Network ACLs, interent Gateways, Route tables etc will delete along with VPC.



Delete VPC

Are you sure you want to delete this VPC? Deleting this VPC will also delete objects associated with this VPC in this region.

- Subnets
- Security Groups
- Network ACLs
- VPN Attachments
- Internet Gateways
- Route Tables
- Network Interfaces
- VPC Peering Connections

□ Delete VPN Connection when deleting the VPC.