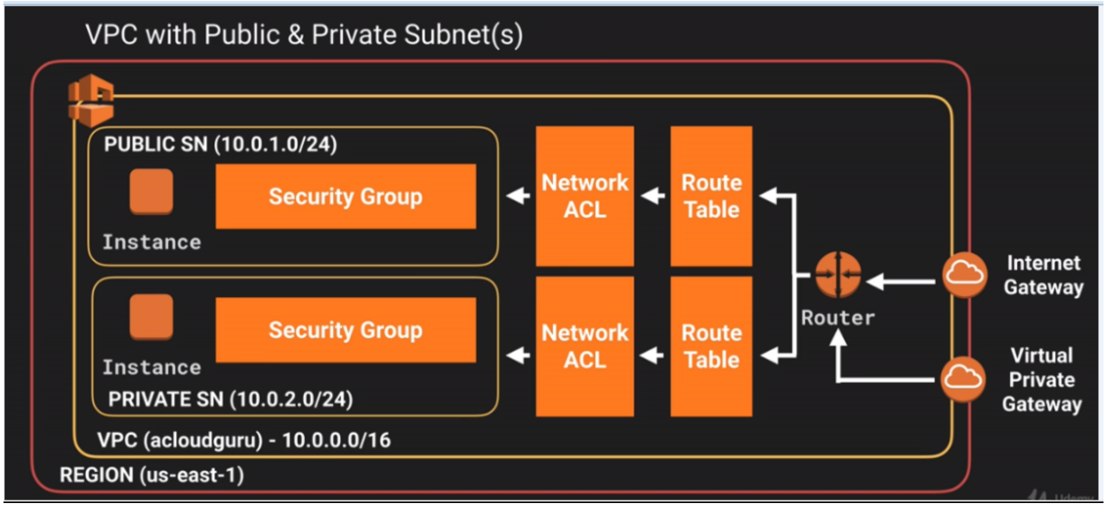
AWS- VPC concepts - (VPC, Internet Gateway Subnets, Route Tables ,& Nat Gateway)

VPC Concepts

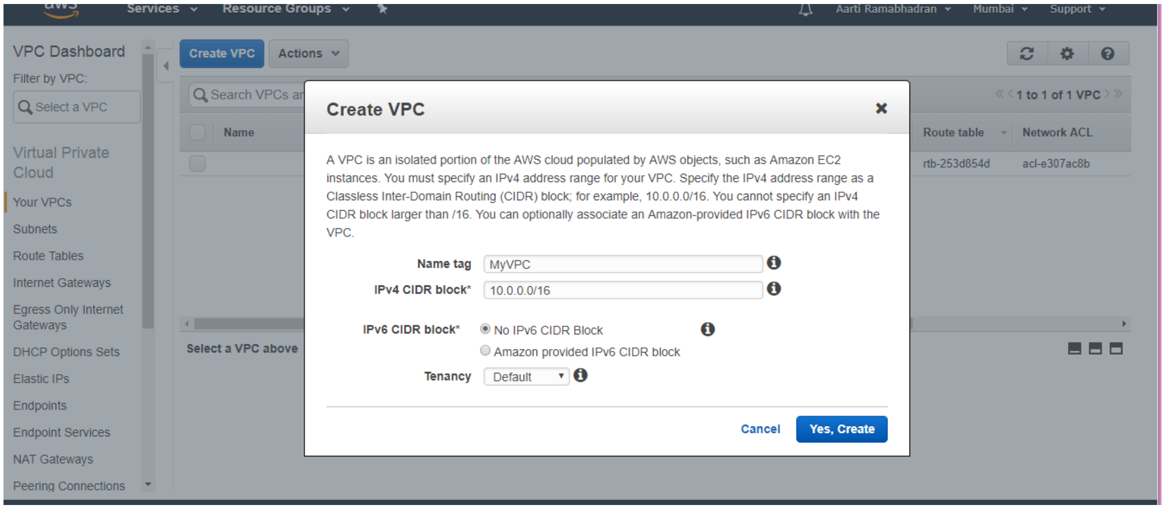
- It's a logically isolated section of AWS cloud, a virtual DC in cloud.
- It gives complete control over networking environment, including selection of own ip address, creating subnets configuring route tables and network gateways.
- Easily customize the network configuration.
- We create the public facing subnets for webservers that has access to the internet and place the backend systems such as databases and app servers in private facing subnet with no internet access.
- 1subnet=1AZ
- Security Groups are stateful and operate at instance level while NACLs are stateless and operate at subnet level.
- Default VPC is friendly and easy to deploy.
- All subnets in a default vpc have access to internet
- Each EC2 instances has both public and private Ip address
- We can have one VPC communicate with another VPC using a direct network connection having private IP address, its called VPC peering. VPC peering is based on star model, i.e. no transitive peering, one-to –one peering only.
- When we create a VPC the following are created by default:
    - Route Table
    - NACLs
    - Security Groups
- Amazon reserves 5 IP addresses
    - 0.0.0 : Network address.
    - 0.0.1 : Reserved by AWS for the VPC router.
    - 0.0.2 : Reserved by AWS DNS
    - 0.0.3 : Reserved by AWS for future use.
    - 0.0.255 : Network broadcast address.

VPC - Architecture Diagram



VPC Part 1 – Create a VPC , assisgn a public and private subnet and launch EC2 instances from them , make sure we are able to access internet from Public Facing EC2 instance.

1. Create a VPC – Go to VPC ->create VPC and assign an IP address in the highest range /16, here we have put 10.0.0.0/16



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: MyVPC

IPv4 CIDR block*: 10.0.0.0/16

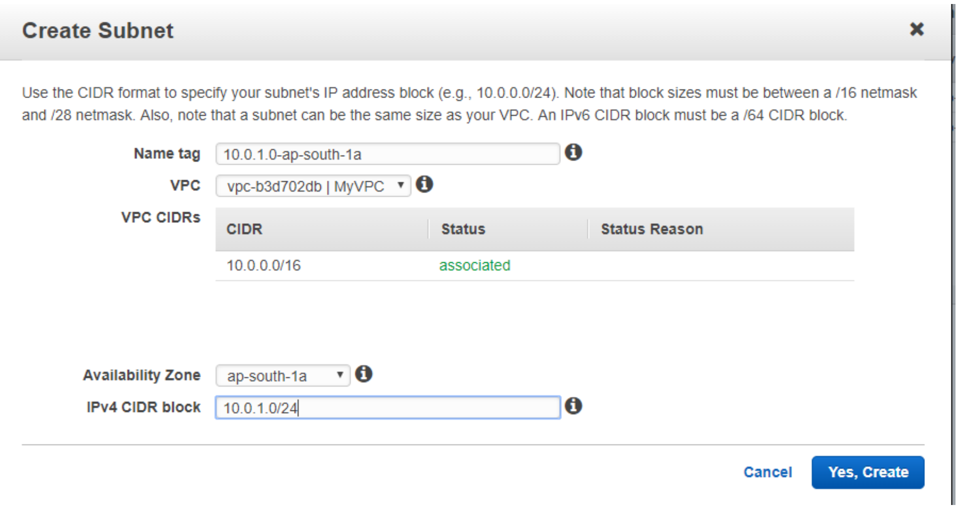
IPv6 CIDR block*: ● No IPv6 CIDR Block ○ Amazon provided IPv6 CIDR block

Tenancy: Default

2. Once we create our VPC, there will be a route table, security group and NACL already present.

3. Creating the subnets and assign IP address, make sure we have one subnet in one AZ as per diagram.

The below subnet has been created in AZ south -1a and will be used for public facing instance. Ip address assigned in range of 10.0.1.0/24. Lets create another subnet in other AZ for private instance and put them in IP range 10.0.2.0/24



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: 10.0.1.0-ap-south-1a

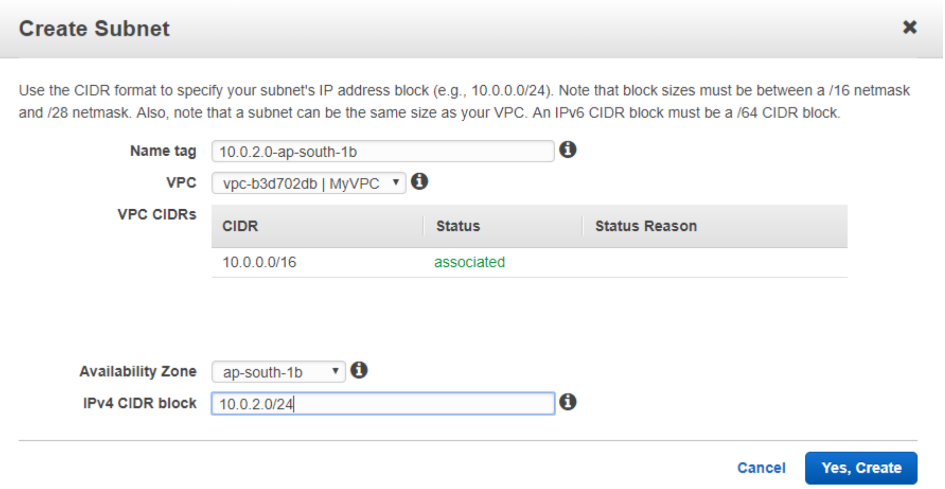
VPC: vpc-b3d702db | MyVPC

VPC CIDRs

CIDR	Status	Status Reason
10.0.0.0/16	associated	

Availability Zone: ap-south-1a

IPv4 CIDR block: 10.0.1.0/24



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: 10.0.2.0-ap-south-1b

VPC: vpc-b3d702db | MyVPC

VPC CIDRs

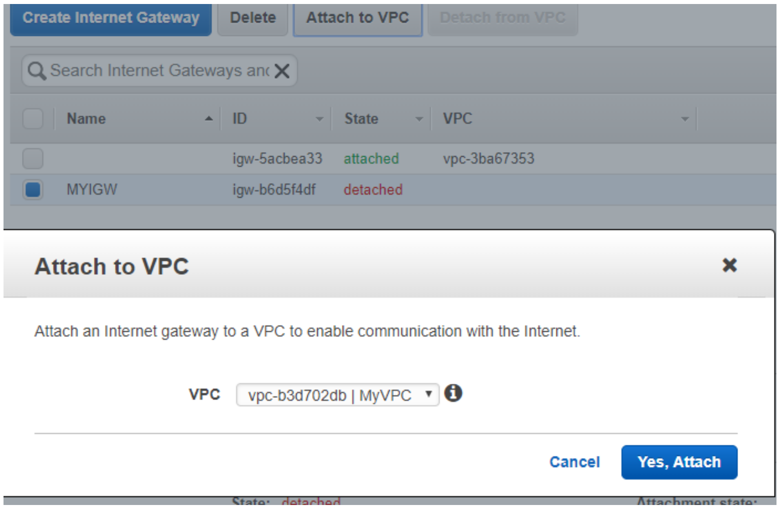
CIDR	Status	Status Reason
10.0.0.0/16	associated	

Availability Zone: ap-south-1b

IPv4 CIDR block: 10.0.2.0/24

Just one more thing we need to change the setting for public subnet as auto assign for public IP so that public IPs are automatically assigned Go to Subnets select the public subnet ->Subnet Action->Modify auto Assign IP settings and click enable.

4. As we need to create a public facing instance we have to create an Internet gateway through which our instances will communicate with internet. So let's create and Internet Gateway and attach to our VPC as per our diagram. We can have one Internet gateway per VPC, even if we create another gateway we wont be able to attach it to VPC



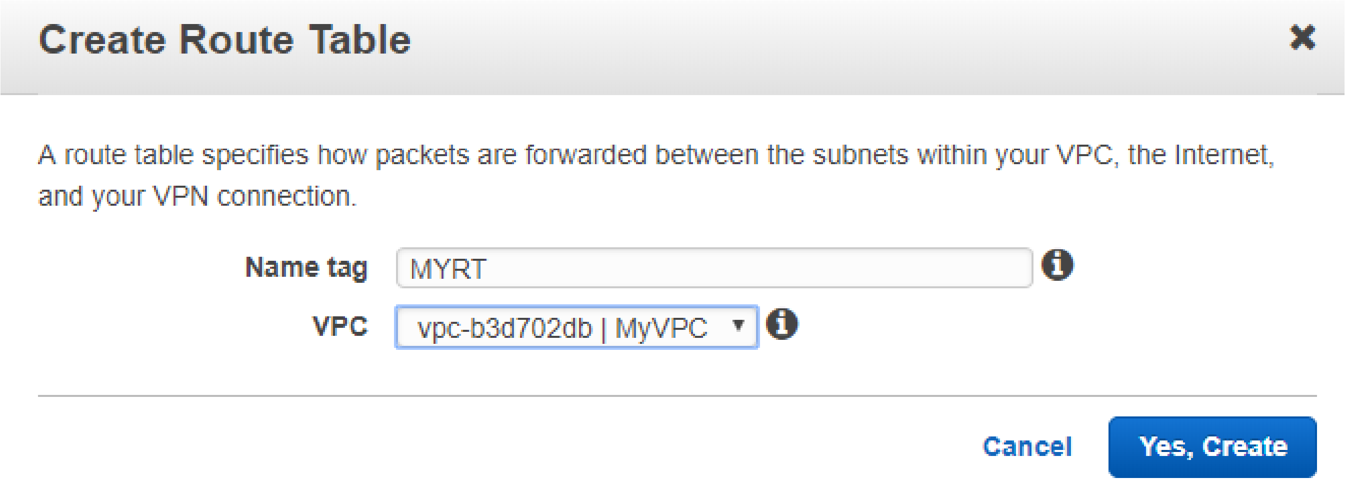
	Name	ID	State	VPC
□		igw-5acbea33	attached	vpc-3ba67353
☑	MYIGW	igw-b6d5f4df	detached	

Attach to VPC

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

VPC: vpc-b3d702db | MyVPC

5. Now we have to go to route table, by default there will be our default route table created for our VPC, we wont disturb that , we will create a new route table



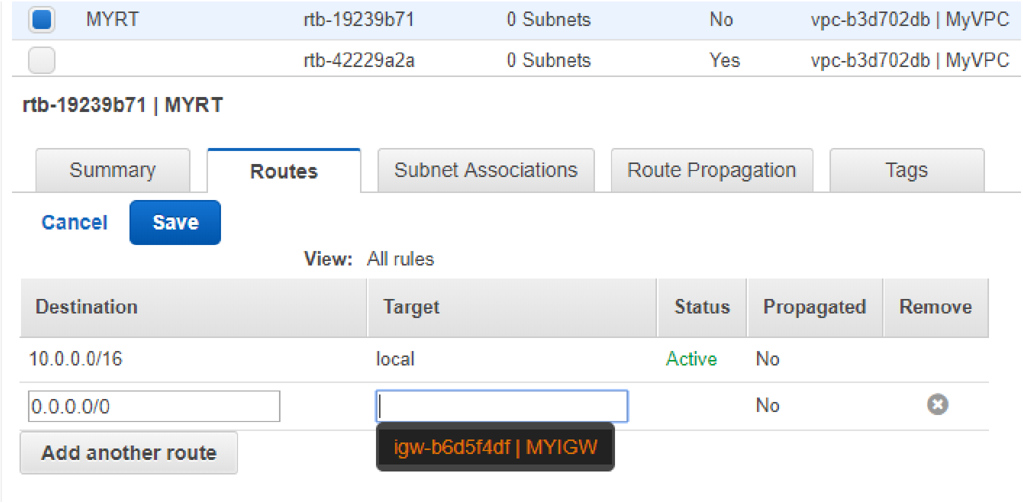
Create Route Table

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

Name tag: MYRT

VPC: vpc-b3d702db | MyVPC

Add a route to allow everything for public subnet through the Internet gateway created



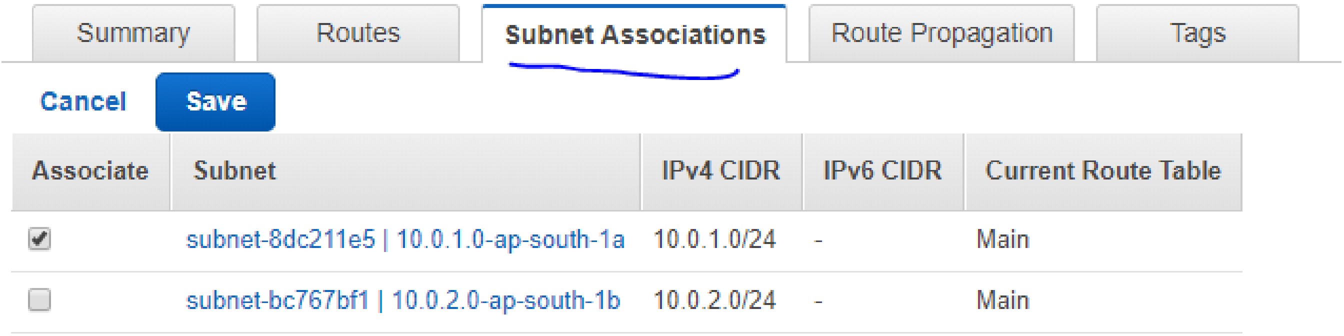
☑	MYRT	rtb-19239b71	0 Subnets	No	vpc-b3d702db | MyVPC
□		rtb-42229a2a	0 Subnets	Yes	vpc-b3d702db | MyVPC

rtb-19239b71 | MYRT

View: All rules

Destination	Target	Status	Propagated	Remove
10.0.0.0/16	local	Active	No	
0.0.0.0/0	igw-b6d5f4df | MYIGW		No	⊗

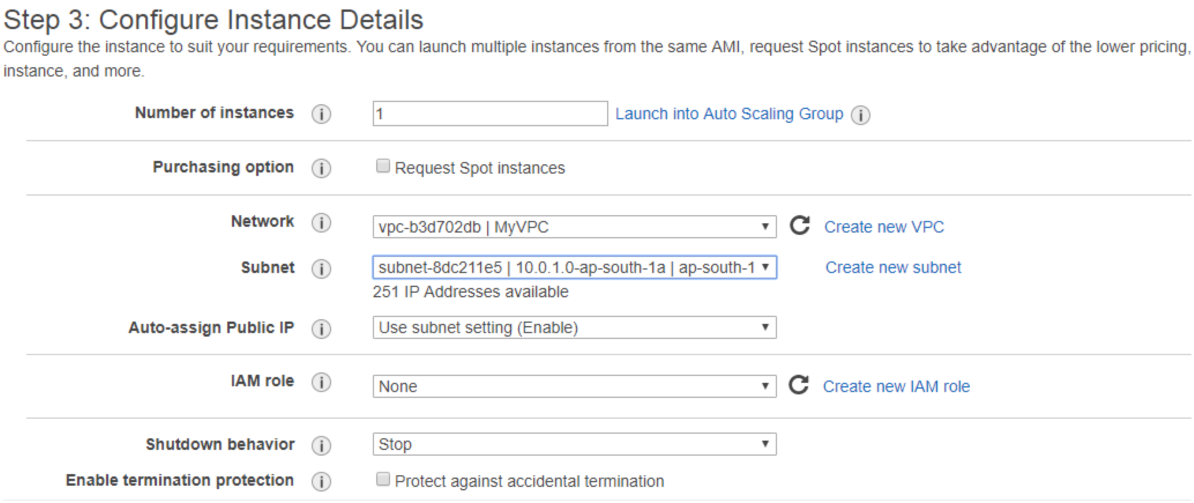
Associate Private Subnet in subnet associations.



Associate	Subnet	IPv4 CIDR	IPv6 CIDR	Current Route Table
☑	subnet-8dc211e5 | 10.0.1.0-ap-south-1a	10.0.1.0/24	-	Main
□	subnet-bc767bf1 | 10.0.2.0-ap-south-1b	10.0.2.0/24	-	Main

6. Now we have to launch the EC2 instances one in public and one in private and then we have to make sure we are able to access internet via the public facing EC2 Instance.

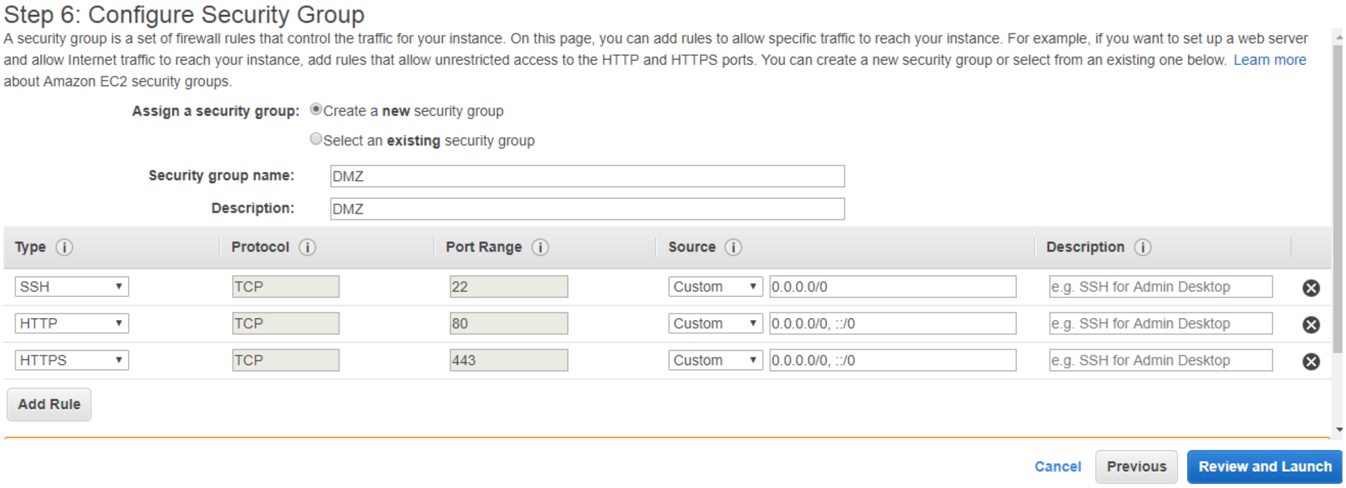
For public EC2 Instance the configuration and Security Group details to be chosen.


Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, instance, and more.

Number of instances	1 Launch into Auto Scaling Group
Purchasing option	□ Request Spot instances
Network	vpc-b3d702db | MyVPC ▾ Create new VPC
Subnet	subnet-8dc211e5 | 10.0.1.0-ap-south-1a | ap-south-1 ▾ Create new subnet 251 IP Addresses available
Auto-assign Public IP	Use subnet setting (Enable) ▾
IAM role	None ▾ Create new IAM role
Shutdown behavior	Stop ▾
Enable termination protection	□ Protect against accidental termination

Create a new security group and Launch


Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

Security group name: DMZ

Description: DMZ

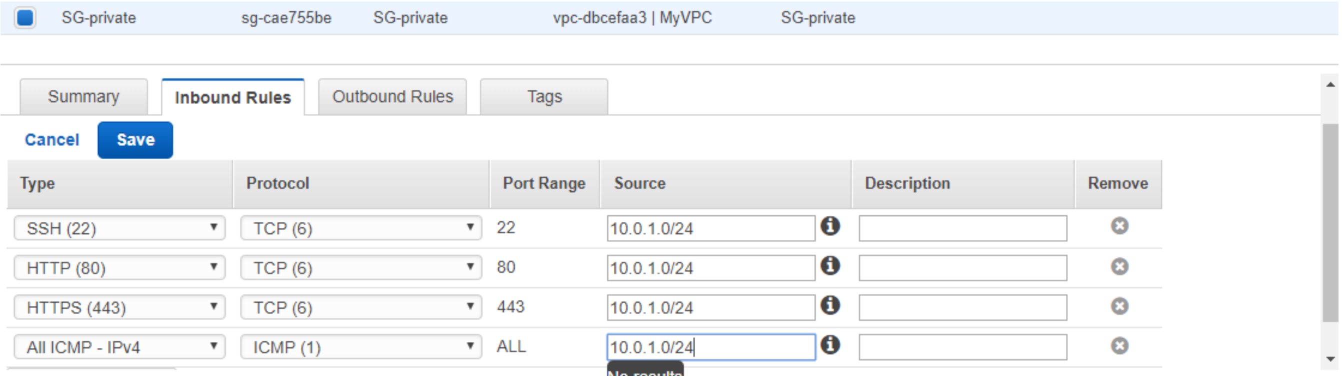
Type	Protocol	Port Range	Source		Description
SSH ▾	TCP	22	Custom ▾	0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP ▾	TCP	80	Custom ▾	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop
HTTPS ▾	TCP	443	Custom ▾	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

Now create another EC2 Instance and give it a default Security group. In configuration settings select you're VPC and the Subnet reserved for Private.

Now launch the Public EC2 and check for internet access by running Yum update

VPC Part 2 – The private EC2 Instance should be accessible from Public EC2 Instance and not your machine , and also try to access internet from EC2 private Instance and observe.

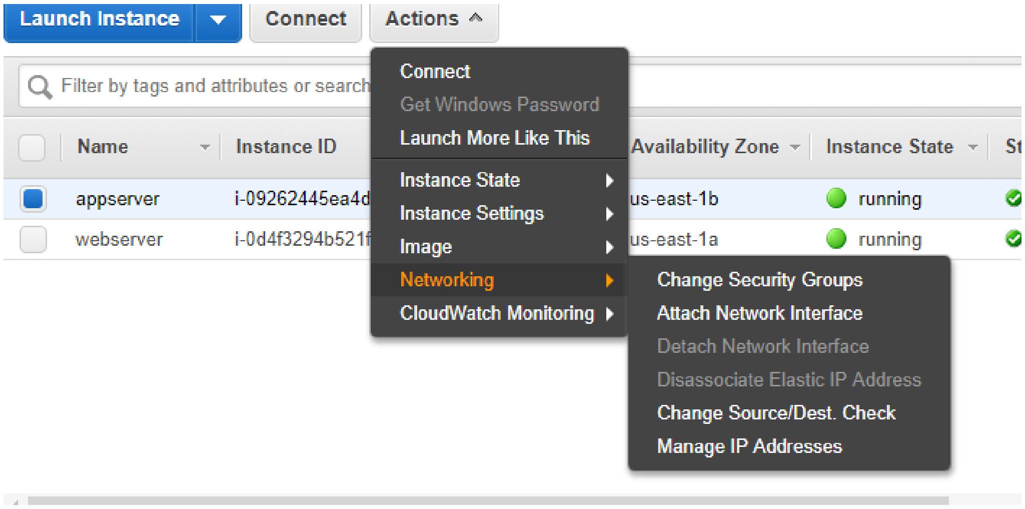
So Internet is accessible from public ec2 instance but we cannot ping the private EC2 instance. So we have to change the SG of the EC-private instance so that it allows the proper protocols from Public ec-instance



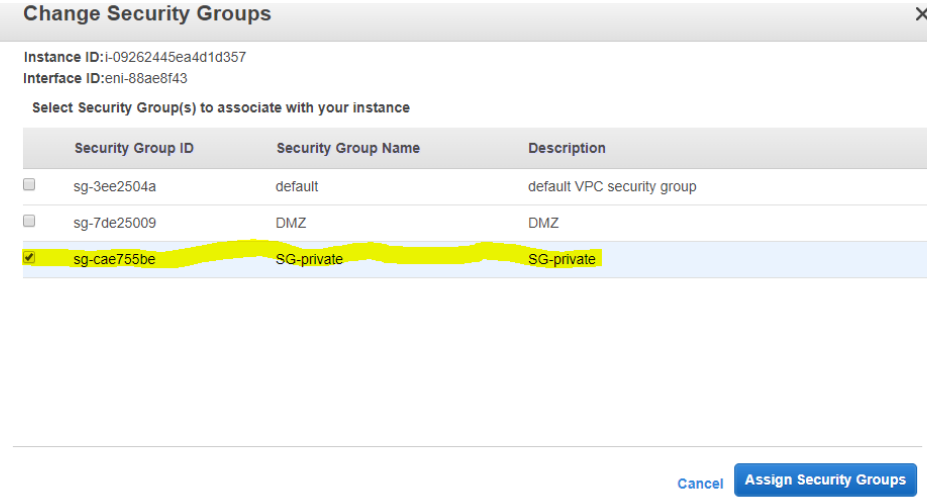
■ SG-private	sg-cae755be	SG-private	vpc-dbcefaa3 | MyVPC	SG-private

Type	Protocol	Port Range	Source	Description	Remove
SSH (22)	TCP (6)	22	10.0.1.0/24		⊗
HTTP (80)	TCP (6)	80	10.0.1.0/24		⊗
HTTPS (443)	TCP (6)	443	10.0.1.0/24		⊗
All ICMP - IPv4	ICMP (1)	ALL	10.0.1.0/24		⊗

Now associate the private EC2 instance with the new security Group created



	Name	Instance ID	Availability Zone	Instance State	St
■	appserver	i-09262445ea4d	us-east-1b	● running	✓
□	webserver	i-0d4f3294b521f	us-east-1a	● running	✓

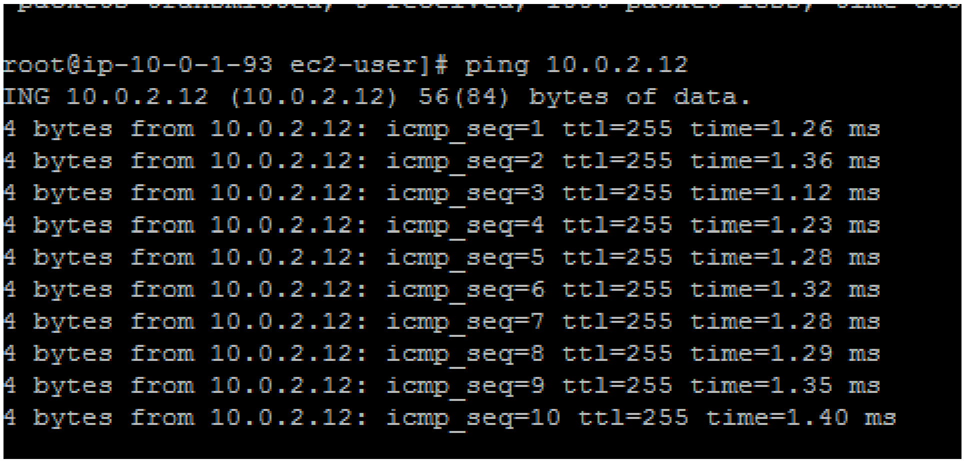


Instance ID:i-09262445ea4d1d357
Interface ID:eni-88ae8f43

Select Security Group(s) to associate with your instance

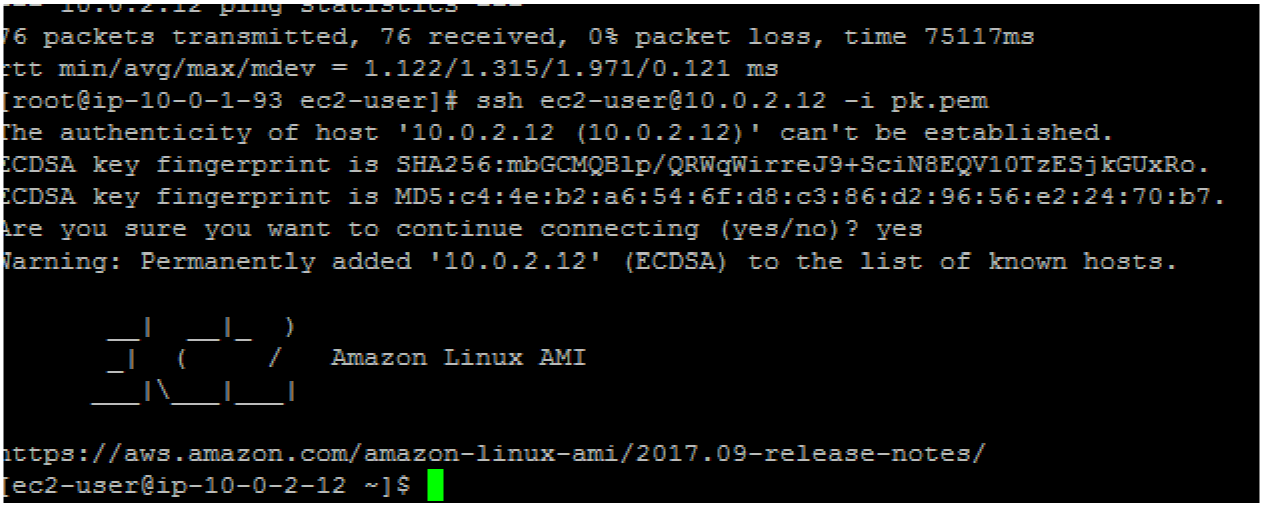
	Security Group ID	Security Group Name	Description
□	sg-3ee2504a	default	default VPC security group
□	sg-7de25009	DMZ	DMZ
☑	sg-cae755be	SG-private	SG-private

So now if we login to Public Instance and ping the private instance, it pings successfully



```
root@ip-10-0-1-93 ec2-user]# ping 10.0.2.12
ING 10.0.2.12 (10.0.2.12) 56(84) bytes of data.
4 bytes from 10.0.2.12: icmp_seq=1 ttl=255 time=1.26 ms
4 bytes from 10.0.2.12: icmp_seq=2 ttl=255 time=1.36 ms
4 bytes from 10.0.2.12: icmp_seq=3 ttl=255 time=1.12 ms
4 bytes from 10.0.2.12: icmp_seq=4 ttl=255 time=1.23 ms
4 bytes from 10.0.2.12: icmp_seq=5 ttl=255 time=1.28 ms
4 bytes from 10.0.2.12: icmp_seq=6 ttl=255 time=1.32 ms
4 bytes from 10.0.2.12: icmp_seq=7 ttl=255 time=1.28 ms
4 bytes from 10.0.2.12: icmp_seq=8 ttl=255 time=1.29 ms
4 bytes from 10.0.2.12: icmp_seq=9 ttl=255 time=1.35 ms
4 bytes from 10.0.2.12: icmp_seq=10 ttl=255 time=1.40 ms
```

Now we can also take ssh from ec public to ec private as we had allowed all the protocols in the SG assigned.

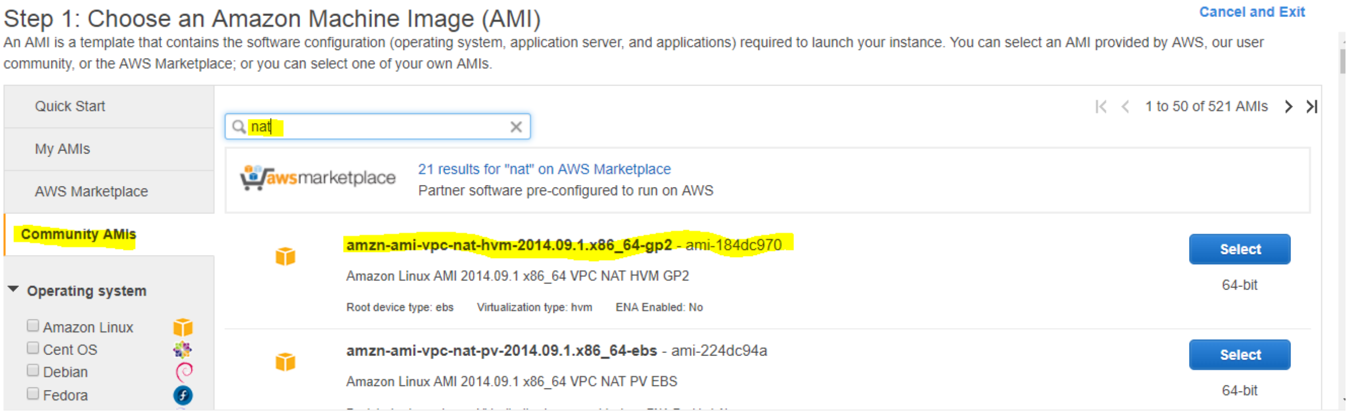


```
-- 10.0.2.12 ping statistics ---
76 packets transmitted, 76 received, 0% packet loss, time 75117ms
rtt min/avg/max/mdev = 1.122/1.315/1.971/0.121 ms
[root@ip-10-0-1-93 ec2-user]# ssh ec2-user@10.0.2.12 -i pk.pem
The authenticity of host '10.0.2.12 (10.0.2.12)' can't be established.
ECDSA key fingerprint is SHA256:mbGCMQBlp/QRWqWirreJ9+SciN8EQV10TzESjkGUxRo.
ECDSA key fingerprint is MD5:c4:4e:b2:a6:54:6f:d8:c3:86:d2:96:56:e2:24:70:b7.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.12' (ECDSA) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2017.09-release-notes/
[ec2-user@ip-10-0-2-12 ~]$ 
```

<u>VPC Part 3 – The private EC2 Instance should be able to access Internet, there are two ways:</u>

1. Using NAT Instance
2. Using NAT Gateway – widely used

Nat Instance: Create a Nat instance by choosing options given below and make sure you assign it to Public Subnet



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.

Search: nat

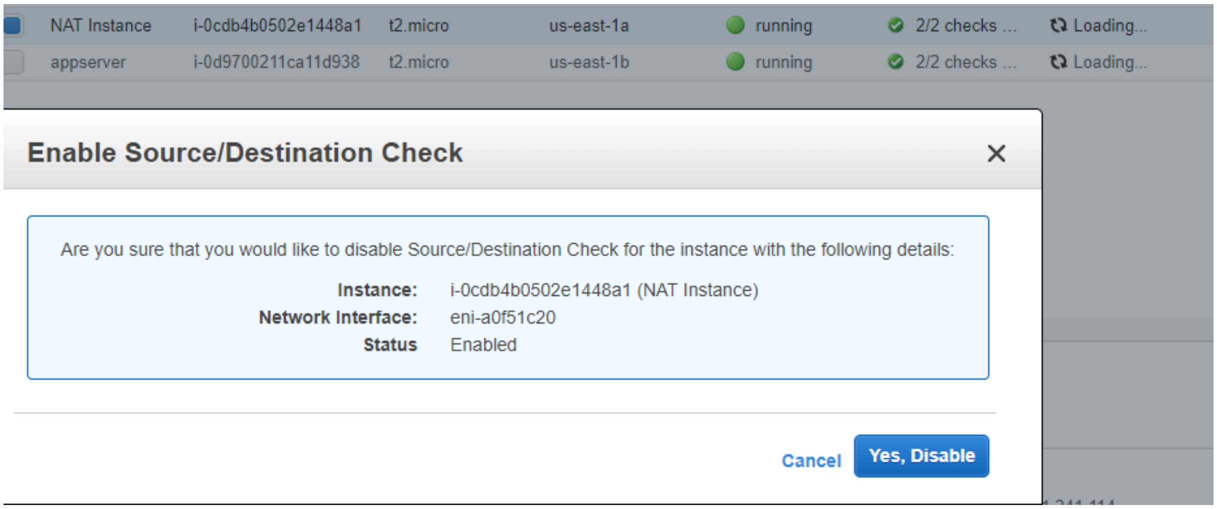
1 to 50 of 521 AMIs

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

amzn-ami-vpc-nat-hvm-2014.09.1.x86_64-gp2 - ami-184dc970
Amazon Linux AMI 2014.09.1 x86_64 VPC NAT HVM GP2
Root device type: ebs   Virtualization type: hvm   ENA Enabled: No
64-bit

amzn-ami-vpc-nat-pv-2014.09.1.x86_64-ebs - ami-224dc94a
Amazon Linux AMI 2014.09.1 x86_64 VPC NAT PV EBS
64-bit

Then we have to go to the created NAT Instance and disable Source and Destination Checks (**Select the NAT instance, choose Actions, select Networking, and then select Change Source/Dest. Check**) 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.



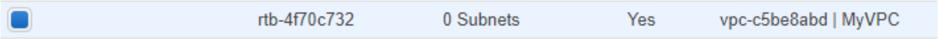
■	NAT Instance	i-0cdb4b0502e1448a1	t2.micro	us-east-1a	● running	✓ 2/2 checks ...	Loading...
□	appserver	i-0d9700211ca11d938	t2.micro	us-east-1b	● running	✓ 2/2 checks ...	Loading...

Enable Source/Destination Check ✕

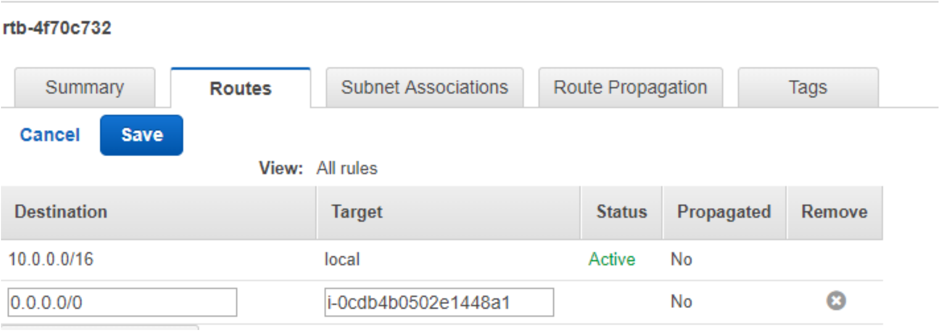
Are you sure that you would like to disable Source/Destination Check for the instance with the following details:

Instance:	i-0cdb4b0502e1448a1 (NAT Instance)
Network Interface:	eni-a0f51c20
Status	Enabled

Now add a route in the route table associated with private subnet for internet access through NAT Instance. Then try accessing internet from EC2 private and it will update.



■	rtb-4f70c732	0 Subnets	Yes	vpc-c5be8abd | MyVPC



rtb-4f70c732

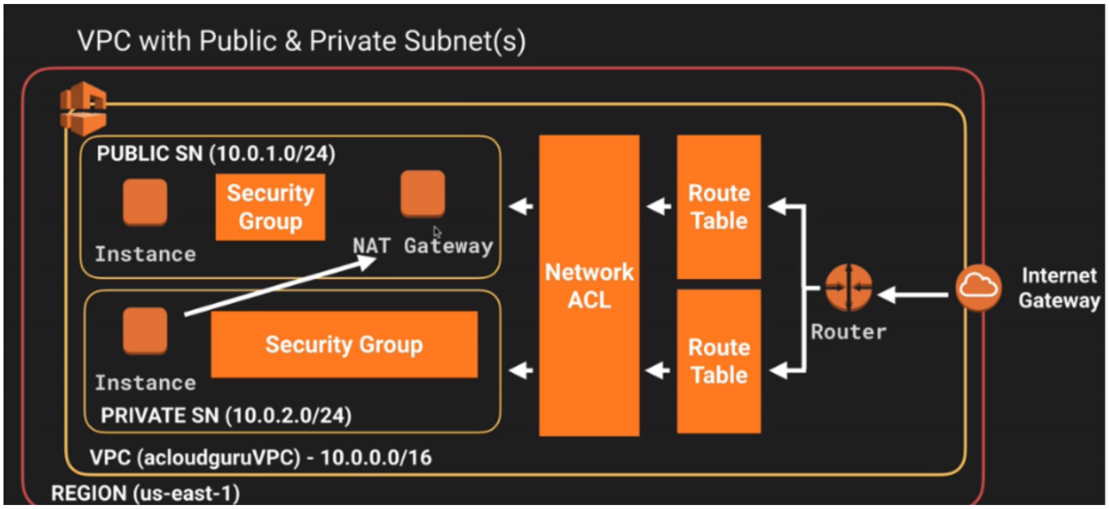
View: All rules

Destination	Target	Status	Propagated	Remove
10.0.0.0/16	local	Active	No	
0.0.0.0/0	i-0cdb4b0502e1448a1		No	⊗

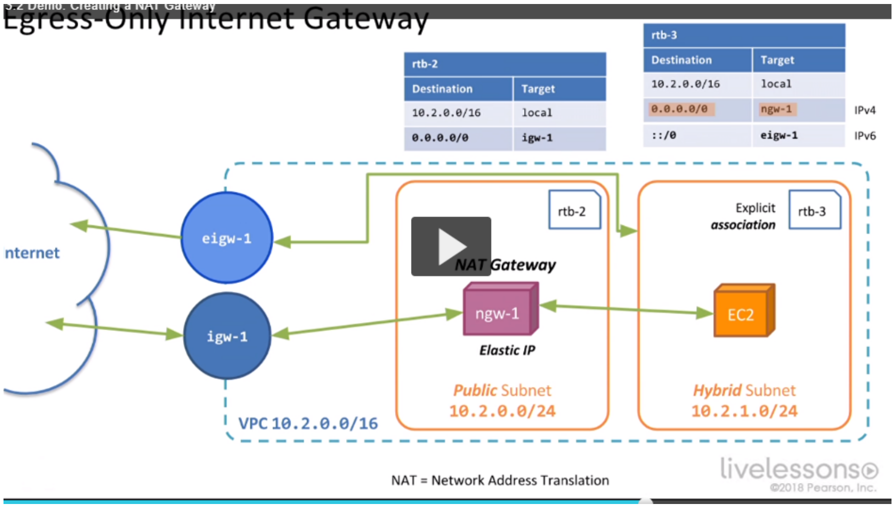
Now it successfully connects to Internet,but when we delete the NAT Instance there is no more Internet

Nat Gateway – So the widely preferred method is NAT Gateway which is in HA, scalable upto 10Gb, maintained by Amazon and do not sit behind a security group. Also we cannot SSH to Nat Gateway, so it's secure. The NAT gateway is Ipv4 and for Ipv6 we use Egress Gateways. Use the public subnet for Nat Gateways and assign an Elastic IP to it.

NAT Gateway will take some time, after its created we update the route table associated with private subnet to add a route to Internet using Nat Gateway.



The Diagram looks like above with NAT Gateway



3.2 Demo: Creating a NAT Gateway

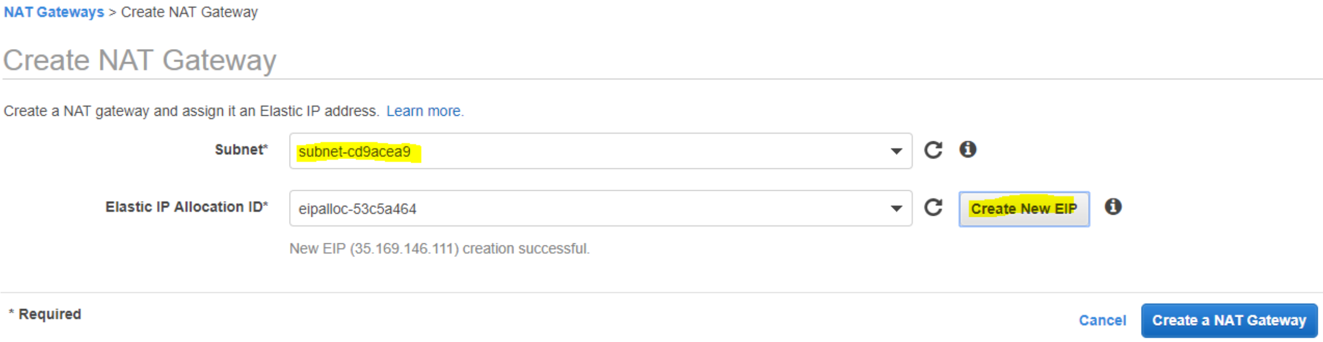
Egress-Only Internet Gateway

rtb-2	
Destination	Target
10.2.0.0/16	local
0.0.0.0/0	igw-1

rtb-3		
Destination	Target	
10.2.0.0/16	local	
0.0.0.0/0	ngw-1	IPv4
::/0	eigw-1	IPv6

NAT = Network Address Translation

NAT Gateway created



NAT Gateways > Create NAT Gateway

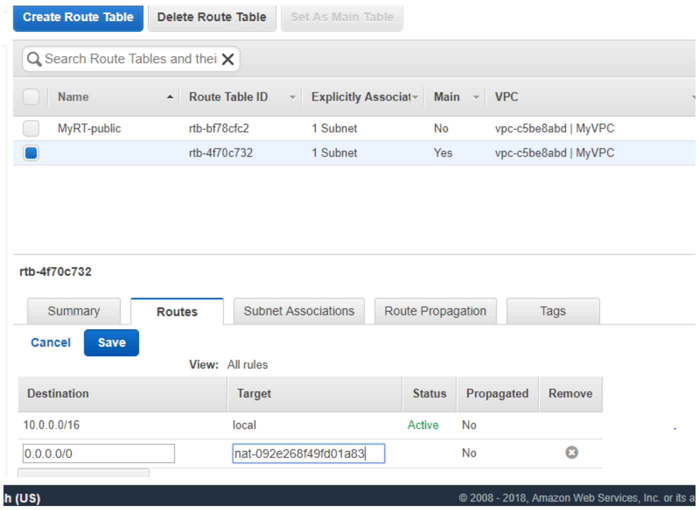
Create NAT Gateway

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

Subnet*	subnet-cd9acea9
Elastic IP Allocation ID*	eipalloc-53c5a464

New EIP (35.169.146.111) creation successful.

The Route Table Modified



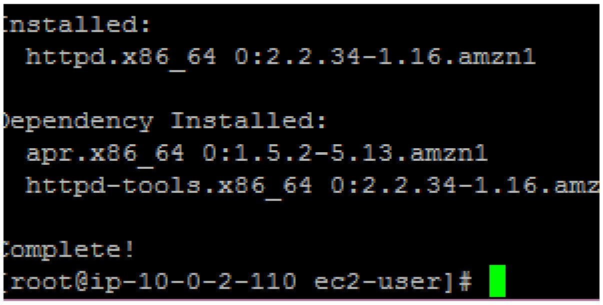
	Name	Route Table ID	Explicitly Associat	Main	VPC
□	MyRT-public	rtb-bf78cfc2	1 Subnet	No	vpc-c5be8abd | MyVPC
■		rtb-4f70c732	1 Subnet	Yes	vpc-c5be8abd | MyVPC

rtb-4f70c732

View: All rules

Destination	Target	Status	Propagated	Remove
10.0.0.0/16	local	Active	No	
0.0.0.0/0	nat-092e268f49fd01a83		No	⊗

Internet Accessible from EC2 private Instance:



```
Installed:
  httpd.x86_64 0:2.2.34-1.16.amzn1

Dependency Installed:
  apr.x86_64 0:1.5.2-5.13.amzn1
  httpd-tools.x86_64 0:2.2.34-1.16.amz

Complete!
[root@ip-10-0-2-110 ec2-user]# █
```