

Virtual Private Cloud(VPC)

Author : Debajyoti Sahani

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

Major Concepts:

Subnets

Route Table

Internet Gateway

Nat Gateway

Security Groups

Network Access Control List(NACL)

Peering Connection

VPN

Subnets:

- Subnetting is the process of dividing a network into two or more subnets.
- An IP address has numbers that identify the network ID and the host ID.
- A subnet address borrows some of the bits from the host ID of the IP address.
- Subnetting is largely invisible to computer users who aren't also network administrators

Route Tables:

A *route table* contains a set of rules, called *routes*, that are used to determine where network traffic is directed.

Each subnet in your VPC must be associated with a route table; the table controls the routing for the subnet. A subnet can only be associated with one route table at a time, but you can associate multiple subnets with the same route table.

Internet Gateway:

An internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet. It therefore imposes no availability risks or bandwidth constraints on your network traffic.

An internet gateway serves two purposes: to provide a target in your VPC route tables for internet-routable traffic, and to perform network address translation (NAT) for instances that have been assigned public IPv4 addresses.

An internet gateway supports IPv4 and IPv6 traffic.

NAT Gateway:

You can use a network address translation (NAT) gateway to enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those instances.

NAT gateways are not supported for IPv6 traffic—use an egress-only internet gateway instead.

You are charged for creating and using a NAT gateway in your account. NAT gateway hourly usage and data processing rates apply.

Security Groups:

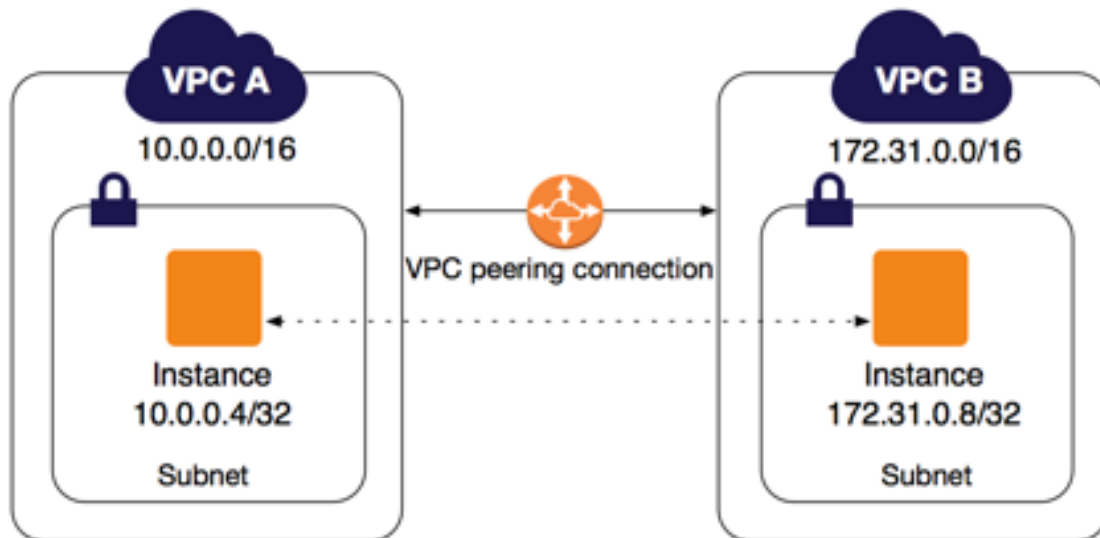
- A security group acts as a virtual firewall for your instance to control inbound and outbound traffic.
- When you launch an instance in a VPC, you can assign up to five security groups to the instance.
- Security groups act at the instance level, not the subnet level. Therefore, each instance in a subnet in your VPC could be assigned to a different set of security groups.
- If you don't specify a particular group at launch time, the instance is automatically assigned to the default security group for the VPC.

Network Access Control Lists(NACL):

A *network access control list (ACL)* is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets. You might set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your VPC.

VPC Peering:

- A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses.
- Instances in either VPC can communicate with each other as if they are within the same network. You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account. The VPCs can be in different regions (also known as an inter-region VPC peering connection).



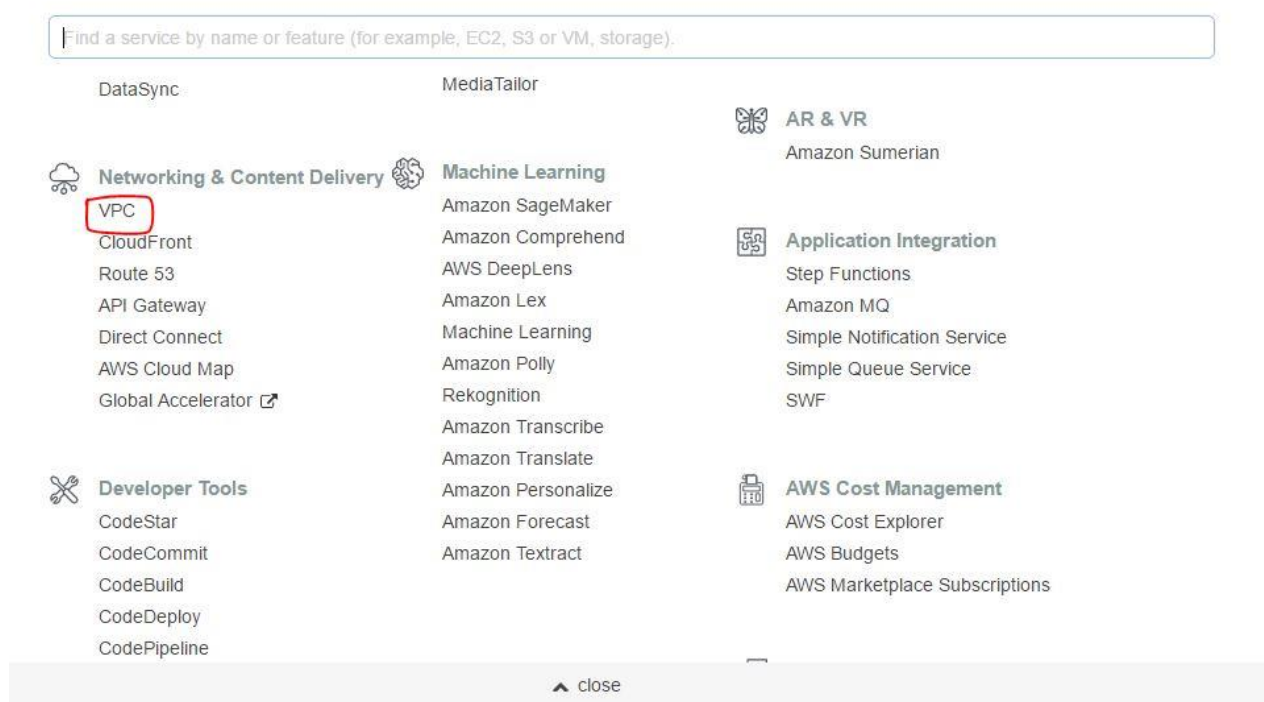
Virtual Private Network(VPN):

A VPN makes the private network (such as a company network) of an entity accessible through public infrastructure, primarily the internet. A VPN can allow users to exchange data efficiently across shared or public networks, as though they are directly linked to the private network.

You can create an IPsec VPN connection between your VPC and your remote network. On the AWS side of the Site-to-Site VPN connection, a *virtual private gateway* provides two VPN endpoints (tunnels) for automatic failover. You configure your *customer gateway* on the remote side of the Site-to-Site VPN connection.

VPC Lab

1st step will be going into the **AWS Management Console > Services > Networking and Content Delivery > VPC**



Now click on **Your VPCs > Create VPC**

VPC Dashboard

Filter by VPC:

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Endpoints

Endpoint Services

NAT Gateways

Peering Connections

Launch VPC Wizard **Launch EC2 Instances**

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

Resources by Region [Refresh Resources](#)

You are using the following Amazon VPC resources

VPCs See all regions	Mumbai 1	NAT Gateways See all regions	Mumbai 0
Subnets See all regions	Mumbai 2	VPC Peering Connections See all regions	Mumbai 0
Route Tables See all regions	Mumbai 1	Network ACLs See all regions	Mumbai 1
Internet Gateways See all regions	Mumbai 1	Security Groups See all regions	Mumbai 6
Egress-only Internet Gateways See all regions	Mumbai 0	Customer Gateways See all regions	Mumbai 0

Create VPC **Actions**

<input type="checkbox"/>	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
<input type="checkbox"/>		vpc-9da280f5	available	172.31.0.0...	-

Next provide the **Name** for your VPC > **IP Address** > **Create**

VPCs > Create 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 ⓘ

IPv4 CIDR block* ⓘ

IPv6 CIDR block ☒ No IPv6 CIDR Block ⓘ
☐ Amazon provided IPv6 CIDR block

Tenancy ⓘ

* Required

Cancel

Create

VPCs > Create VPC

Create VPC

✓ The following VPC was created:

VPC ID vpc-06e4eeab4ddf1cd9f

Close

Next click on **Subnets** from the Navigation pane > **Create subnet**

VPC Dashboard

Filter by VPC:
Select a VPC

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Endpoints

Endpoint Services

NAT Gateways

Peering Connections

Create VPC Actions

Filter by tags and attributes or search by keyword

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set
myvpc	vpc-06e4eeab4ddf1cd9f	available	10.0.0.0/16	-	dopt-20a5ac48
	vpc-9da280f5	available	172.31.0.0...	-	dopt-20a5ac48

VPC: vpc-06e4eeab4ddf1cd9f

Description CIDR Blocks Flow Logs Tags

VPC ID vpc-06e4eeab4ddf1cd9f Tenancy default

State available Default VPC No

IPv4 CIDR 10.0.0.0/16 IPv6 CIDR -

Create subnet Actions

Filter by tags and attributes or search by keyword

Name	Subnet ID	State	VPC	IPv4 CIDR
	subnet-b594dfdd	available	vpc-9da280f5	172.31.16.0/20
	subnet-c4c37b88	available	vpc-9da280f5	172.31.0.0/20

Next In the Create Subnet page **provide the name of your Subnet > IP Address > Availability Zone**

[Subnets](#) > Create subnet

Create subnet

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between a /16 netmask and /28 netmask, and can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag VPC to use for this Subnet

VPC*

VPC CIDRs

VPC CIDRs	Status	Status Reason
vpc-9da280f5		
vpc-06e4eeab4ddf1cd9f	myvpc	

Availability Zone

IPv4 CIDR block*

* Required

Cancel

Create

Again click on **Create Subnet** > Now provide the name(as private) > IP Address > Availability Zone(2nd One) > Create

[Subnets](#) > Create subnet

Create subnet

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between a /16 netmask and /28 netmask, and can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag VPC to use for this Subnet

VPC*

VPC CIDRs

CIDR	Status	Status Reason
10.0.0.0/16	associated	

Availability Zone

IPv4 CIDR block*

* Required

Cancel

Create

Now you can see your both subnets has been created (Public as well as Private) subnets.

Create subnet **Actions** ▾

Filter by tags and attributes or search by keyword

<input type="checkbox"/>	Name ▾	Subnet ID ▾	State ▾	VPC ▾	IPv4 CIDR ▾	Available IPv4 ▾
<input type="checkbox"/>		subnet-b594dfdd	available	vpc-9da280f5	172.31.16.0/20	4091
<input type="checkbox"/>		subnet-c4c37b88	available	vpc-9da280f5	172.31.0.0/20	4091
<input checked="" type="checkbox"/>	Private Sub...	subnet-0b4b5aae35ed0e3c7	available	vpc-06e4eeab4ddf1cd9f ...	10.0.2.0/24	251
<input checked="" type="checkbox"/>	public subnet	subnet-0db8c72961d802e6f	available	vpc-06e4eeab4ddf1cd9f ...	10.0.1.0/24	251

Next click on the Route Table from the navigation pane > click on Create Route Table

VPC Dashboard

Filter by VPC:
Select a VPC

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Create route table **Actions** ▾

Filter by tags and attributes or search by keyword

<input type="checkbox"/>	Name ▾	Route Table ID ▴	Explicitly Associated with	Main	VPC ID
<input type="checkbox"/>		rtb-03c505430542d0e0c	-	Yes	vpc-06e4eeab4ddf1cd9f .
<input type="checkbox"/>		rtb-828a69e9	-	Yes	vpc-9da280f5

In Create Route Table page provide the name of the Route table as publicRT and choose the VPC which you have created and click on Create.

[Route Tables](#) > Create 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 ⓘ

VPC* ⓘ

* Required

Filter by attributes

vpc-9da280f5	
vpc-06e4eeab4ddf1cd9f	myvpc

Cancel Create

Repeat this same thing again by click on Create Route Table and providing **the name as privateRT > VPC created > Create**

[Route Tables](#) > Create 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 ⓘ

VPC* ⓘ

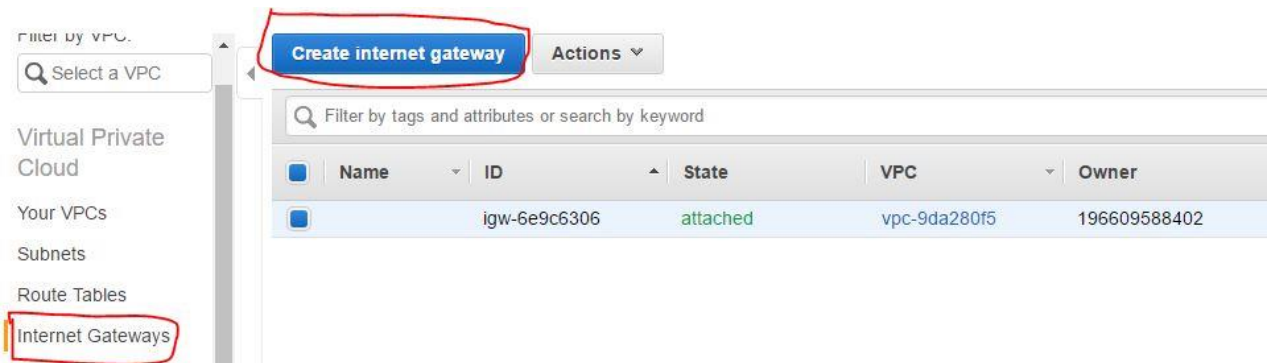
* Required

Filter by attributes

vpc-9da280f5	
vpc-06e4eeab4ddf1cd9f	myvpc

Cancel Create

Next choose Internet Gateway from the navigation pane > Create internet gateway



Provide the name of the Internet gateway > Create



Now as you have created your Internet gateway, it needs to be attached to some VPC, for that **click on the newly created Internet gateway > Actions > Attach to VPC > choose the newly created VPC > Attach**



Internet gateways > Attach to VPC

Attach to VPC

Attach an internet gateway to a VPC to enable communication with the internet. Specify the VPC you would like to attach below.

VPC*

Filter by attributes

VPC ID	Name
vpc-06e4eeab4ddf1cd9f	myvpc

* Required

Cancel Attach

Create internet gateway Actions

Filter by tags and attributes or search by keyword

	Name	ID	State	VPC	Owner
<input checked="" type="checkbox"/>	MyIGW	igw-0c37818948e...	attached	vpc-06e4eeab4dd...	196609588402
<input type="checkbox"/>		igw-6e9c6306	attached	vpc-9da280f5	196609588402

Next move to **Route Tables > choose publicRT > Routes > Edit Routes**

Filter by VPC. Select a VPC

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Endpoints

Endpoint Services

NAT Gateways

Peering Connections

Security

Create route table Actions

Filter by tags and attributes or search by keyword

	Name	Route Table ID	Explicitly Associated with	Main	VPC ID
<input checked="" type="checkbox"/>	publicRT	rtb-0034a4687f0327ba5	-	No	vpc-06e4eeab4ddf1cd9f ..
<input type="checkbox"/>		rtb-03c505430542d0e0c	-	Yes	vpc-06e4eeab4ddf1cd9f ..
<input type="checkbox"/>	privateRT	rtb-09189eda42b98ba9a	-	No	vpc-06e4eeab4ddf1cd9f ..
<input type="checkbox"/>		rtb-828a69e9	-	Yes	vnc-9da280f5

Route Table: rtb-0034a4687f0327ba5

Summary **Routes** Subnet Associations Route Propagation Tags

Edit routes

View All routes

Destination	Target	Status
10.0.0.0/16	local	active

Now in that click on **Add route > In Destination provide value 0.0.0.0/0 > In target choose Internet Gateway and choose the one you created > Save Routes**

[Route Tables](#) > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No

Add route

* Required

Cancel

Save routes

[Route Tables](#) > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0			No

Add route

* Required

Cancel

Save routes

Egress Only Internet Gateway
Instance
Internet Gateway
NAT Gateway
Network Interface
Peering Connection
Transit Gateway
Virtual Private Gateway

[Route Tables](#) > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	igw-		No

Add route

* Required

Cancel

Save routes

igw-0c37818948e68ef15 MyIGW

Next in same **publicRT** click on **Subnet Associations** > **Edit subnet associations** > choose the public Subnet > **Save**

[Create route table](#) [Actions](#)

<input type="checkbox"/>	Name	Route Table ID	Explicitly Associated with	Main	VPC ID
<input checked="" type="checkbox"/>	publicRT	rtb-0034a4687f0327ba5	-	No	vpc-06e4eeab4ddf1cd9f ...
<input type="checkbox"/>		rtb-03c505430542d0e0c	-	Yes	vpc-06e4eeab4ddf1cd9f ...
<input type="checkbox"/>	privateRT	rtb-09189eda42b98ba9a	-	No	vpc-06e4eeab4ddf1cd9f ...
<input type="checkbox"/>		rtb-828a69e9	-	Yes	vpc-9da280f5

Route Table: rtb-0034a4687f0327ba5

[Summary](#) [Routes](#) [Subnet Associations](#) [Route Propagation](#) [Tags](#)

[Edit subnet associations](#)

Subnet ID	IPv4 CIDR	IPv6 CIDR
-----------	-----------	-----------

[Route Tables](#) > Edit subnet associations

Edit subnet associations

Route table rtb-0034a4687f0327ba5 (publicRT)

Associated subnets subnet-0db8c72961d802e6f

Filter by attributes or search by keyword			
<input type="checkbox"/>	Subnet ID	IPv4 CIDR	IPv6 CIDR
<input type="checkbox"/>	subnet-0b4b5aae35ed0e3c7 Private S...	10.0.2.0/24	-
<input checked="" type="checkbox"/>	subnet-0db8c72961d802e6f public sub...	10.0.1.0/24	-

* Required

[Cancel](#) [Save](#)

Create route table **Actions** ▾

Filter by tags and attributes or search by keyword

<input type="checkbox"/>	Name ▾	Route Table ID ▴	Explicitly Associated with	Main
<input checked="" type="checkbox"/>	publicRT	rtb-0034a4687f0327ba5	subnet-0db8c72961d802e6f	No
<input type="checkbox"/>		rtb-03c505430542d0e0c	-	Yes
<input type="checkbox"/>	privateRT	rtb-09189eda42b98ba9a	-	No
<input type="checkbox"/>		rtb-828a69e9	-	Yes

Route Table: rtb-0034a4687f0327ba5

Summary **Routes** **Subnet Associations** **Route Propagation** **Tags**

Edit subnet associations

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-0db8c72961d802e...	10.0.1.0/24	-

Next click on Nat Gateways from the navigation pane > Create NAT Gateway > choose the public subnet > Create Elastic IP > click on Create a Nat Gateway

Create NAT Gateway **Actions** ▾

Filter by tags and attributes or search by keyword

You do not have any NAT Gateways in this region

Click the Create NAT Gateway button to create your first NAT Gateway

Create NAT Gateway

Cloud

- Your VPCs
- Subnets
- Route Tables
- Internet Gateways
- Egress Only Internet Gateways
- DHCP Options Sets
- Elastic IPs
- Endpoints
- Endpoint Services
- NAT Gateways**

NAT Gateways > Create NAT Gateway

Create NAT Gateway

Create a NAT gateway and assign it an Elastic IP address. [Learn more.](#)

Subnet*

Search subnets by ID or name or VPC e.g. "subnet-1a2b3c4d"

Filter by attributes

Subnet ID	Subnet Name	VPC ID	VPC Name
subnet-0b4b5aae35ed0e3c7	Private Subnet	vpc-06e4eeab4ddf1cd9f	myvpc
subnet-b594dfdd	-	vpc-9da280f5	-
subnet-c4c37b88	-	vpc-9da280f5	-
subnet-0db8c72961d802e6f	public subnet	vpc-06e4eeab4ddf1cd9f	myvpc

Elastic IP Allocation ID*

Create New EIP

* Required

Cancel Create a NAT Gateway

NAT Gateways > Create NAT Gateway

Create NAT Gateway

Create a NAT gateway and assign it an Elastic IP address. [Learn more.](#)

Subnet*

subnet-0db8c72961d802e6f

Create New EIP

Elastic IP Allocation ID*

Enter an allocation ID or select an EIP

Create New EIP

* Required

Cancel Create a NAT Gateway

Once its created instead of Close click on Edit route tables

NAT Gateways > Create 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-03cad81ccd28b624f

Edit route tables

Close

In Route tables click on privateRT > Routes > Edit Routes

Create route table

Actions

Q

Filter by tags and attributes or search by keyword

K < 1 to 4

<input type="checkbox"/>	Name	Route Table ID	Explicitly Associated with	Main	VPC ID	Owner
<input type="checkbox"/>	publicRT	rtb-0034a4687f0327ba5	subnet-0db8c72961d802e6f	No	vpc-06e4eeab4ddf1cd9f ...	196609588402
<input type="checkbox"/>		rtb-03c505430542d0e0c	-	Yes	vpc-06e4eeab4ddf1cd9f ...	196609588402
<input checked="" type="checkbox"/>	privateRT	rtb-045094db343d7e615	-	No	vpc-06e4eeab4ddf1cd9f ...	196609588402
<input type="checkbox"/>		rtb-828a69e9	-	Yes	vpc-9da280f5	196609588402

Route Table: rtb-045094db343d7e615

Summary

Routes

Subnet Associations

Route Propagation

Tags

Edit routes

View All routes

Click on Add Route > In Destination provide the value 0.0.0.0/0 > In target provide Nat Gateway > Select the newly created one > Save routes

Route Tables > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No

Add route

* Required

Cancel

Save routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
<input type="text" value="0.0.0.0/0"/>	<input type="text" value=""/>		No <input type="button" value="X"/>

* Required

Egress Only Internet Gateway

Instance

Internet Gateway

NAT Gateway

Network Interface

Peering Connection

Transit Gateway

Virtual Private Gateway

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
<input type="text" value="0.0.0.0/0"/>	<input type="text" value="nat-03cad81ccd28b624f"/>		No <input type="button" value="X"/>

* Required

Next click on the Subnet Associations > Edit subnet associations > choose the private subnet > Save

Create route tableActions

Filter by tags and attributes or search by keyword

<input type="checkbox"/>	Name	Route Table ID	Explicitly Associated with	Main	VPC ID
<input type="checkbox"/>	publicRT	rtb-0034a4687f0327ba5	subnet-0db8c72961d802e6f	No	vpc-06e4eeab4ddf1cd9f ...
<input type="checkbox"/>		rtb-03c505430542d0e0c	-	Yes	vpc-06e4eeab4ddf1cd9f ...
<input checked="" type="checkbox"/>	privateRT	rtb-045094db343d7e615	-	No	vpc-06e4eeab4ddf1cd9f ...
<input type="checkbox"/>		rtb-828a69e9	-	Yes	vpc-9da280f5

Route Table: rtb-045094db343d7e615

Summary	Routes	Subnet Associations	Route Propagation	Tags
<input type="button" value="Edit subnet associations"/>				
Subnet ID		IPv4 CIDR	IPv6 CIDR	

[Route Tables](#) > Edit subnet associations

Edit subnet associations

Route table rtb-045094db343d7e615 (privateRT)

Associated subnets

<input type="text"/> Filter by attributes or search by keyword				
<input type="checkbox"/>	Subnet ID ▾	IPv4 CIDR ▾	IPv6 CIDR ▾	Current Route Table
<input checked="" type="checkbox"/>	subnet-0b4b5aae35ed0e3c7 Private S...	10.0.2.0/24	-	Main
<input type="checkbox"/>	subnet-0db8c72961d802e6f public sub...	10.0.1.0/24	-	rtb-0034a4687f0327ba5

* Required

Now we have covered our VPC configuration, our main objective here is to create 2 instances where in 1 Instance we will be attaching it to Public Subnet and the other one to the Private Subnet, checking the Internet Connection for the Public Instance which has been connected to the public Subnet.

Further for private Instance which have been attached to the Private Subnet, will not be providing Public IP to it, so we have to enter through our public Instance and also to check if Internet connection is available to it or not.

For this lets create the 1st instance attaching the VPC you created > choose the Public Subnet > Enable Auto assign public Ip

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, assign an access management role to the instance, and more.

Number of instances	1	Launch into Auto Scaling Group
Purchasing option	<input type="checkbox"/> Request Spot instances	
Network	vpc-06e4eeab4ddf1cd9f myvpc	Create new VPC
Subnet	subnet-0db8c72961d802e6f public subnet ap-southeast-1 250 IP Addresses available	Create new subnet
Auto-assign Public IP	Enable	
Placement group	<input type="checkbox"/> Add instance to placement group	
Capacity Reservation	Open	Create new Capacity Reservation
IAM role	None	Create new IAM role

Cancel Previous **Review and Launch** Next: Add Storage

Create a new security key for the instance and launch the usual way.

Next create 2nd Instance, in Configure Instance Details provide Network as the VPC created > Subnet > Private Subnet > Auto-assign Public IP > Subnet setting-Disable

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

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, assign an access management role to the instance, and more.

Number of instances Launch into Auto Scaling Group

Purchasing option ☐ Request Spot instances

Network Create new VPC

Subnet Create new subnet
251 IP Addresses available

Auto-assign Public IP

Placement group ☐ Add instance to placement group

Capacity Reservation Create new Capacity Reservation

IAM role Create new IAM role

Cancel Previous **Review and Launch** Next: Add Storage

Next go to your Public Instance > copy the Public IP > enter into the Console > Use command "yum update -y"

Here you will find out internet is available to this instance.

Public Instance	i-0ffe5e1e91d61dc88	t2.micro	ap-south-1a	running	2/2 checks ...	None	
-----------------	---------------------	----------	-------------	---------	----------------	------	--

Instance: i-0ffe5e1e91d61dc88 (Public Instance)		Public IP: 13.126.135.20			
Description	Status Checks	Monitoring	Tags		
Instance ID	i-0ffe5e1e91d61dc88		Public DNS (IPv4)	-	
Instance state	running		IPv4 Public IP	13.126.135.20	
Instance type	t2.micro		IPv6 IPs	-	


```
root@ip-10-0-1-90:~  
Running transaction check  
Running transaction test  
Transaction test succeeded  
Running transaction  
  Updating    : 1:openssl-libs-1.0.2k-16.amzn2.0.3.x86_64      1/5  
  Updating    : 1:openssl-1.0.2k-16.amzn2.0.3.x86_64         2/5  
  Installing  : kernel-4.14.101-91.76.amzn2.x86_64           3/5  
  Cleanup     : 1:openssl-1.0.2k-16.amzn2.0.2.x86_64         4/5  
  Cleanup     : 1:openssl-libs-1.0.2k-16.amzn2.0.2.x86_64    5/5  
  Verifying   : 1:openssl-libs-1.0.2k-16.amzn2.0.3.x86_64    1/5  
  Verifying   : kernel-4.14.101-91.76.amzn2.x86_64           2/5  
  Verifying   : 1:openssl-1.0.2k-16.amzn2.0.3.x86_64         3/5  
  Verifying   : 1:openssl-libs-1.0.2k-16.amzn2.0.2.x86_64    4/5  
  Verifying   : 1:openssl-1.0.2k-16.amzn2.0.2.x86_64         5/5  
  
Installed:  
  kernel.x86_64 0:4.14.101-91.76.amzn2  
  
Updated:  
  openssl.x86_64 1:1.0.2k-16.amzn2.0.3  
  openssl-libs.x86_64 1:1.0.2k-16.amzn2.0.3  
  
Complete!  
[root@ip-10-0-1-90 ~]#
```

Now we need to enter into the private instance for which we don't have any public IP assigned, for this you need to go the pem file of your key attached to the private Instance > Open the file > copy the details > In console of public Instance > create on file with the same name > paste the content of the pem file > save

testlinux1.pem - Notepad

File Edit Format View Help

```
-----BEGIN RSA PRIVATE KEY-----
MIIIEowIBAAKCAQEAg1EVBv7shKCH7Q9nH60FhYK7tgxUNEayJ7thPzytI+SzIz7k4P9bWhg+Bi8/
6L+e95HDqJR3NwIf6EnjzQ2hoDSPRfiD7a+BdS+0o0J31ZXm0F861wKOCvQHVVfnRTFN6tj1DqKH
W36rGBnSVVIqoDLF5v34sv3U0WhCOCm149XRfcLbLoy7qnyDkocs/stkY2MoPB/hdYm5/A/BPW1G
IaRrGr0VnhQZNUJxxK8j0Cp4nhHeEz8G/Z11M9X8eysK/GcSSguBKF6UxnbOGrhj1oySiDYy9dAM
7c+uukXLbsnyP3EhrQKIW2hzGV/RuesdCqeCfBHwRN+B1I/20XvotwIDAQABAoIBAA6GXG40qnit
DPvitURhoPY41Ky5RmMV4sTUTBhjJl8on8vpPAXLGP6pKcsjVM1YzKuTmvG9okYaXQ8jhSf8NwFy
btqNd4JF5Gri4Pch9gFQ+FvZ/LclmDKL9XeyJmmX34d3gfuhbJlpNGGREZvQX9iiZRN8yhJ6Uxzq
jFCMBItzxYuvvc1f0v4jRDsd4mdZ8M3q2q2cv593G9Ltf/79+Lnp6tbr8rF131CsAxTtiGUbrAB/
nHo8gMEpMHcyF1AEY1EO3ALfsefNbL+Gg4EW2vGp8x3/83ICrum0GdnW0UVNIXfnH2UIcHRFbQbn
/UHhKV956AotyTUTruv7Bop1xkkCgYEAuvZf1NoFNPKER+IsfWjLxWcUF5oGdQahYohGdk7pVIJa
c9yj/8LvyEW9FxuKLTJLHk18VePkV3nbDDy8p3UFTUnHoVymmYQ6LsH+esjgJ5S2pNOqb2Y6SYm+
mDBkqgia06yYGNVDU0AHn4Hcw2IxMFqIWOq73aQZFUP1qy20bbsCgYEAs86AY8UvDHRLZ7ai4xo9
6V1L5qB+vA++GCqH91WaBpp/p1R1xB/YbxNjviVThigVNGUCjQd0ts0qdej9D9987CgubVxBdMdn
8fAnjvsrHnfWHHz/HBj057dyC0ZwZXnAReng/7m2d6aqabSBud64twWKFCDdCKyJ9qgLiHAIAZUC
gYAJHkoau2dG741SQRpASbIbPKSVWiPkTn54lsdE6MUBTgnWwJU8L0wA/LfqYw2shWn+Iuc66jW0
B/qPbcIwLXIkhafw1q2chLLPTuhq6ANTuyOQAI+MaC4ttNIky4Urt2E62EVdvhR5jCIUEc5mnNK
dXYt3MezdGIRkFXO9CJyiQKBgQCqS2xCEiFLVGG0auW+3b3BZgFa/qrJhsdtr+pxngVxN6arXkzj
YZ31JaIUftoSd6gNpZVS6Jbef1Y1smLrQ6nXKP8/yF6eMZ1bBS8SNkRSaQE6nhAoFmLti0rR7vbI
xRUbojTdPpDYpdtEnhbU6XJctTxfb/ioHHHWULYLL+h+QKBgGonXaBEGH3IUyUE87eSiwPnudqb
8XsLTiDsLeLxbwaHxrmJfbYaRr7GzNHU34BEPK4n0cAbulcyMOSCqclafBGGUJCWMeVys6giqVBY
o5WHmJnygs7v4Szd5b33IZjV3S6a7G3PJjbS5Tx6M37j0I1iKgCzSdsk6C0Y0DXCZ4Sy
-----END RSA PRIVATE KEY-----
```

root@ip-10-0-1-90:~

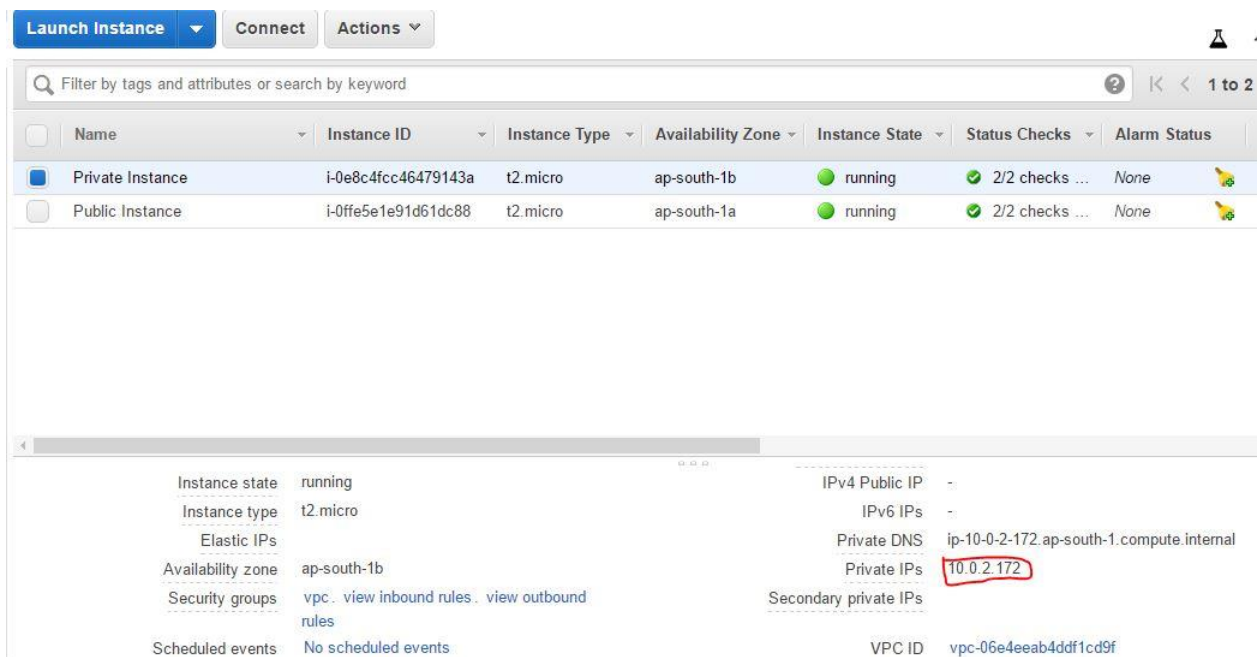
```
-----BEGIN RSA PRIVATE KEY-----
MIIIEowIBAAKCAQEAg1EVBv7shKCH7Q9nH60FhYK7tgxUNEayJ7thPzytI+SzIz7k4P9bWhg+Bi8/
6L+e95HDqJR3NwIf6EnjzQ2hoDSPRfiD7a+BdS+0o0J31ZXm0F861wKOCvQHVVfnRTFN6tj1DqKH
W36rGBnSVVIqoDLF5v34sv3U0WhCOCm149XRfcLbLoy7qnyDkocs/stkY2MoPB/hdYm5/A/BPW1G
IaRrGr0VnhQZNUJxxK8j0Cp4nhHeEz8G/Z11M9X8eysK/GcSSguBKF6UxnbOGrhj1oySiDYy9dAM
7c+uukXLbsnyP3EhrQKIW2hzGV/RuesdCqeCfBHwRN+B1I/20XvotwIDAQABAoIBAA6GXG40qnit
DPvitURhoPY41Ky5RmMV4sTUTBhjJl8on8vpPAXLGP6pKcsjVM1YzKuTmvG9okYaXQ8jhSf8NwFy
btqNd4JF5Gri4Pch9gFQ+FvZ/LclmDKL9XeyJmmX34d3gfuhbJlpNGGREZvQX9iiZRN8yhJ6Uxzq
jFCMBItzxYuvvc1f0v4jRDsd4mdZ8M3q2q2cv593G9Ltf/79+Lnp6tbr8rF131CsAxTtiGUbrAB/
nHo8gMEpMHcyF1AEY1EO3ALfsefNbL+Gg4EW2vGp8x3/83ICrum0GdnW0UVNIXfnH2UIcHRFbQbn
/UHhKV956AotyTUTruv7Bop1xkkCgYEAuvZf1NoFNPKER+IsfWjLxWcUF5oGdQahYohGdk7pVIJa
c9yj/8LvyEW9FxuKLTJLHk18VePkV3nbDDy8p3UFTUnHoVymmYQ6LsH+esjgJ5S2pNOqb2Y6SYm+
mDBkqgia06yYGNVDU0AHn4Hcw2IxMFqIWOq73aQZFUP1qy20bbsCgYEAs86AY8UvDHRLZ7ai4xo9
6V1L5qB+vA++GCqH91WaBpp/p1R1xB/YbxNjviVThigVNGUCjQd0ts0qdej9D9987CgubVxBdMdn
8fAnjvsrHnfWHHz/HBj057dyC0ZwZXnAReng/7m2d6aqabSBud64twWKFCDdCKyJ9qgLiHAIAZUC
gYAJHkoau2dG741SQRpASbIbPKSVWiPkTn54lsdE6MUBTgnWwJU8L0wA/LfqYw2shWn+Iuc66jW0
B/qPbcIwLXIkhafw1q2chLLPTuhq6ANTuyOQAI+MaC4ttNIky4Urt2E62EVdvhR5jCIUEc5mnNK
dXYt3MezdGIRkFXO9CJyiQKBgQCqS2xCEiFLVGG0auW+3b3BZgFa/qrJhsdtr+pxngVxN6arXkzj
YZ31JaIUftoSd6gNpZVS6Jbef1Y1smLrQ6nXKP8/yF6eMZ1bBS8SNkRSaQE6nhAoFmLti0rR7vbI
xRUbojTdPpDYpdtEnhbU6XJctTxfb/ioHHHWULYLL+h+QKBgGonXaBEGH3IUyUE87eSiwPnudqb
8XsLTiDsLeLxbwaHxrmJfbYaRr7GzNHU34BEPK4n0cAbulcyMOSCqclafBGGUJCWMeVys6giqVBY
o5WHmJnygs7v4Szd5b33IZjV3S6a7G3PJjbS5Tx6M37j0I1iKgCzSdsk6C0Y0DXCZ4Sy
-----END RSA PRIVATE KEY-----
:wq!
```


Give permissions to the file created using the following command

chmod 400 testlinux1.pem

```
[root@ip-10-0-1-90 ~]# vi testlinux1.pem
[root@ip-10-0-1-90 ~]# chmod 400 testlinux1.pem
[root@ip-10-0-1-90 ~]#
```

Now copy the Private Instance Private IP Address



The screenshot shows the AWS Management Console interface. At the top, there are buttons for 'Launch Instance', 'Connect', and 'Actions'. Below these is a search bar and a table of instances. The table has columns for Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, and Alarm Status. Two instances are listed: 'Private Instance' and 'Public Instance'. The 'Private Instance' is selected, and its details are shown below the table. The details include Instance state (running), Instance type (t2.micro), Elastic IPs, Availability zone (ap-south-1b), Security groups (vpc, view inbound rules, view outbound rules), Scheduled events (No scheduled events), IPv4 Public IP (-), IPv6 IPs (-), Private DNS (ip-10-0-2-172.ap-south-1.compute.internal), Private IPs (10.0.2.172, highlighted with a red box), Secondary private IPs, and VPC ID (vpc-06e4eeab4ddf1cd9f).

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
Private Instance	i-0e8c4fcc46479143a	t2.micro	ap-south-1b	running	2/2 checks ...	None
Public Instance	i-0ffe5e1e91d61dc88	t2.micro	ap-south-1a	running	2/2 checks ...	None

Instance state	running	IPv4 Public IP	-
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-10-0-2-172.ap-south-1.compute.internal
Availability zone	ap-south-1b	Private IPs	10.0.2.172
Security groups	vpc, view inbound rules, view outbound rules	Secondary private IPs	
Scheduled events	No scheduled events	VPC ID	vpc-06e4eeab4ddf1cd9f

Next go to your console of Public Instance, type the command

ssh -i testlinux1.pem ec2-user@10.0.2.172

Note : testlinux1.pem is the IP Address which I used for my demonstration, you can provide your created file name and also the IP Address for the private Instance may be difference.

```
ec2-user@ip-10-0-2-172:~  
kernel.x86_64 0:4.14.101-91.76.amzn2  
  
Updated:  
openssl.x86_64 1:1.0.2k-16.amzn2.0.3  
openssl-libs.x86_64 1:1.0.2k-16.amzn2.0.3  
  
Complete!  
[root@ip-10-0-1-90 ~]# vi testlinux1.pem  
[root@ip-10-0-1-90 ~]# chmod 400 testlinux1.pem  
[root@ip-10-0-1-90 ~]# ssh -i testlinux1.pem ec2-user@10.0.2.172  
The authenticity of host '10.0.2.172 (10.0.2.172)' can't be established.  
ECDSA key fingerprint is SHA256:BKY+5fjZKBdpmqjm0eqTTlLoy7PKOor2YoCPHUOEObO.  
ECDSA key fingerprint is MD5:45:ab:cb:bb:03:09:27:d0:0f:92:85:3e:de:e8:fb:9e.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '10.0.2.172' (ECDSA) to the list of known hosts.  
  
      _ | _ | _ )  
      _ | ( _ | /   Amazon Linux 2 AMI  
      _ | \ _ | _ |  
  
https://aws.amazon.com/amazon-linux-2/  
1 package(s) needed for security, out of 3 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-10-0-2-172 ~]$
```

Next check if internet is available to this instance or not by using update command or you can also use ping command, follow the image below

```
fec2-user@ip-10-0-2-172 ~]$ ping www.google.com
PING www.google.com (172.217.160.164) 56(84) bytes of data.
64 bytes from bom05s12-in-f4.1e100.net (172.217.160.164): icmp_seq=1 ttl=52 time=2.12 ms
64 bytes from bom05s12-in-f4.1e100.net (172.217.160.164): icmp_seq=2 ttl=52 time=1.81 ms
64 bytes from bom05s12-in-f4.1e100.net (172.217.160.164): icmp_seq=3 ttl=52 time=1.81 ms
64 bytes from bom05s12-in-f4.1e100.net (172.217.160.164): icmp_seq=4 ttl=52 time=1.84 ms
^C
--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.813/1.898/2.122/0.136 ms
```

Here we can see internet is available for the private instance.

