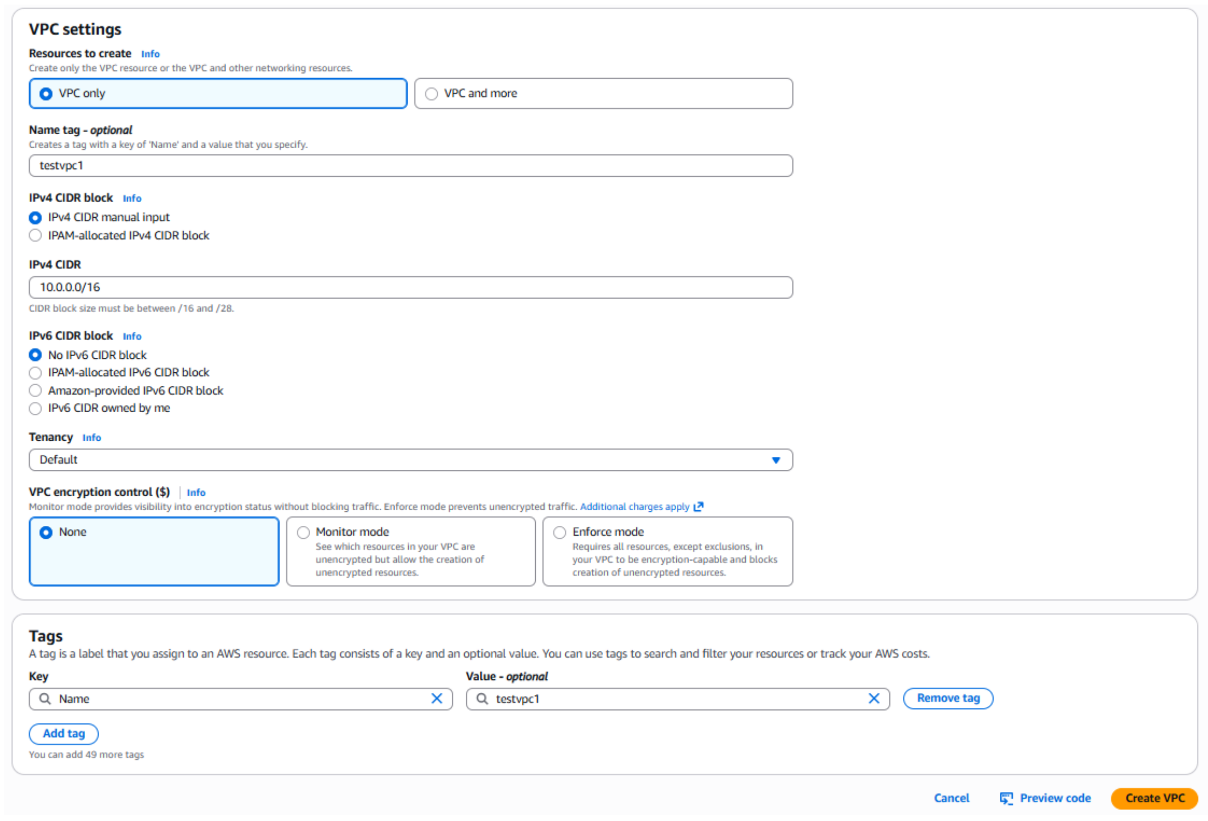
Design and Deploy a Multi-Subnet VPC with Bastion Host and NAT Gateway

(Step – by - Step Lab Guide)-ASWIN VTK

STEP 1: Create a VPC

1. Open **Amazon Web Services Management Console**
2. Go to **VPC → Your VPCs**
3. Click **Create VPC**
4. Select **VPC only**
5. Enter:
   - **Name**:
   - **IPv4 CIDR block**: `10.0.0.0/16`
6. Click **Create VPC**



VPC settings

Resources to create Info
Create only the VPC resource or the VPC and other networking resources.
[✓ VPC only] VPC and more

Name tag - *optional*
Creates a tag with a key of 'Name' and a value that you specify.
testvpc1

IPv4 CIDR block Info
[✓ IPv4 CIDR manual input]
IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.0.0.0/16
CIDR block size must be between /16 and /28.

IPv6 CIDR block Info
[✓ No IPv6 CIDR block]
IPAM-allocated IPv6 CIDR block
Amazon-provided IPv6 CIDR block
IPv6 CIDR owned by me

Tenancy Info
Default

VPC encryption control ($) Info
Monitor mode provides visibility into encryption status without blocking traffic. Enforce mode prevents unencrypted traffic. Additional charges apply
[✓ None]
Monitor mode — See which resources in your VPC are unencrypted but allow the creation of unencrypted resources.
Enforce mode — Requires all resources, except exclusions, in your VPC to be encryption-capable and blocks creation of unencrypted resources.

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

Key	Value - *optional*
Name	testvpc1

```
aws ec2 create-vpc --instance-tenancy 'default' --cidr-block '10.0.0.0/16' --tag-specifications
'{"resourceType":"vpc","tags":[{"key":"Name","value":"testvpc1"}]}'
```

STEP 2: Create Subnets (Public & Private)

Public Subnet

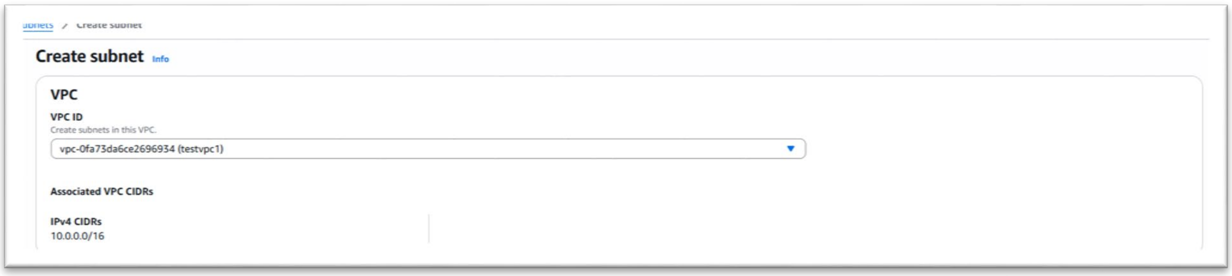
1. Go to **VPC → Subnets**
2. Click **Create subnet**
3. Select **My-VPC**
4. Subnet details:
   - **Name**: Public-Subnet
   - **AZ**: ap-south-1a
   - **CIDR**: 10.0.0.0/24
5. Click **Create subnet**

◆ **Private Subnet**

Repeat steps:

- **Name**: Private-Subnet
- **CIDR**: 10.0.1.0/24

✓ Subnets created.



Subnets / Create subnet

Create subnet Info

VPC

VPC ID

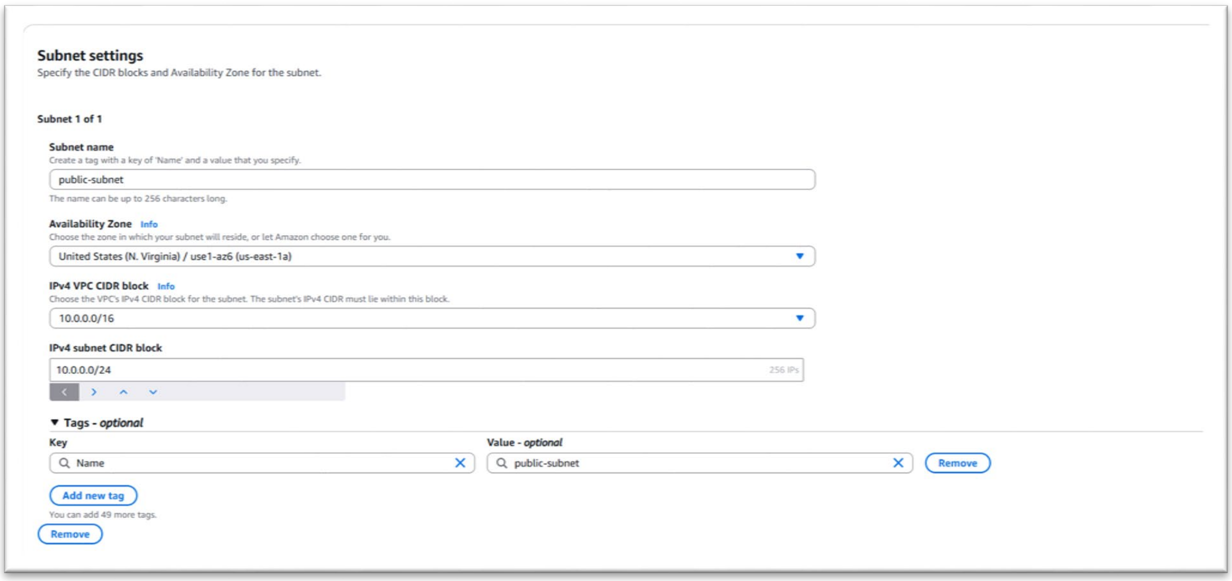
Create subnets in this VPC.

vpc-0fa73da6ce2696934 (testvpc1)

Associated VPC CIDRs

IPv4 CIDRs

10.0.0.0/16


Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name: public-subnet

Availability Zone: United States (N. Virginia) / use1-az6 (us-east-1a)

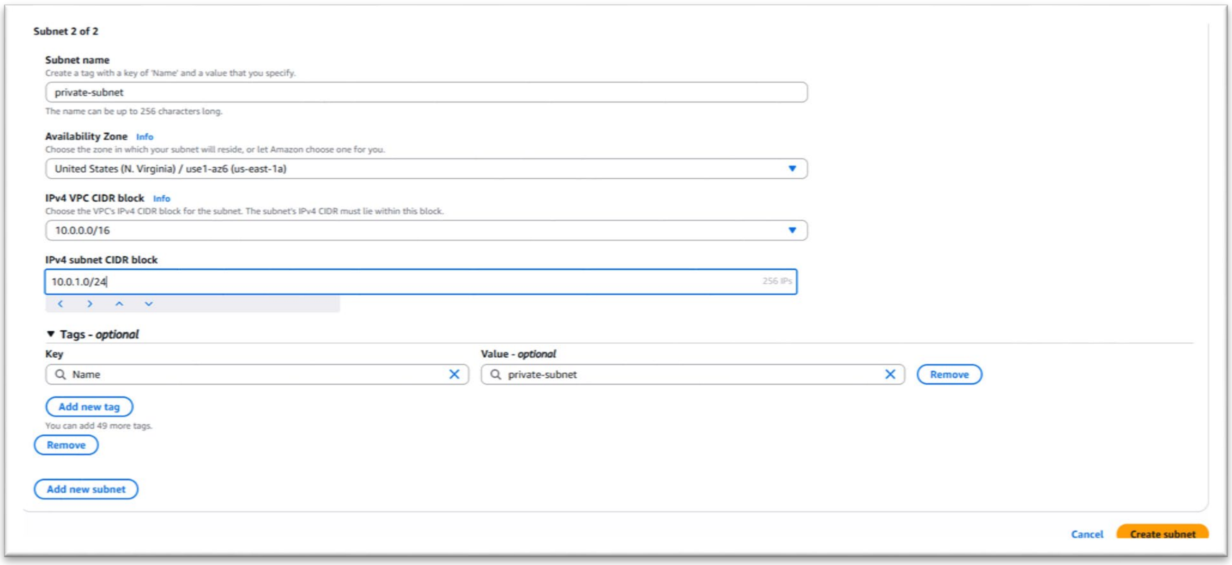
IPv4 VPC CIDR block: 10.0.0.0/16

IPv4 subnet CIDR block: 10.0.0.0/24 (256 IPs)

Tags - optional

Key	Value - optional
Name	public-subnet

You can add 49 more tags.


Subnet 2 of 2

Subnet name: private-subnet

Availability Zone: United States (N. Virginia) / use1-az6 (us-east-1a)

IPv4 VPC CIDR block: 10.0.0.0/16

IPv4 subnet CIDR block: 10.0.1.0/24 (256 IPs)

Tags - optional

Key	Value - optional
Name	private-subnet

You can add 49 more tags.

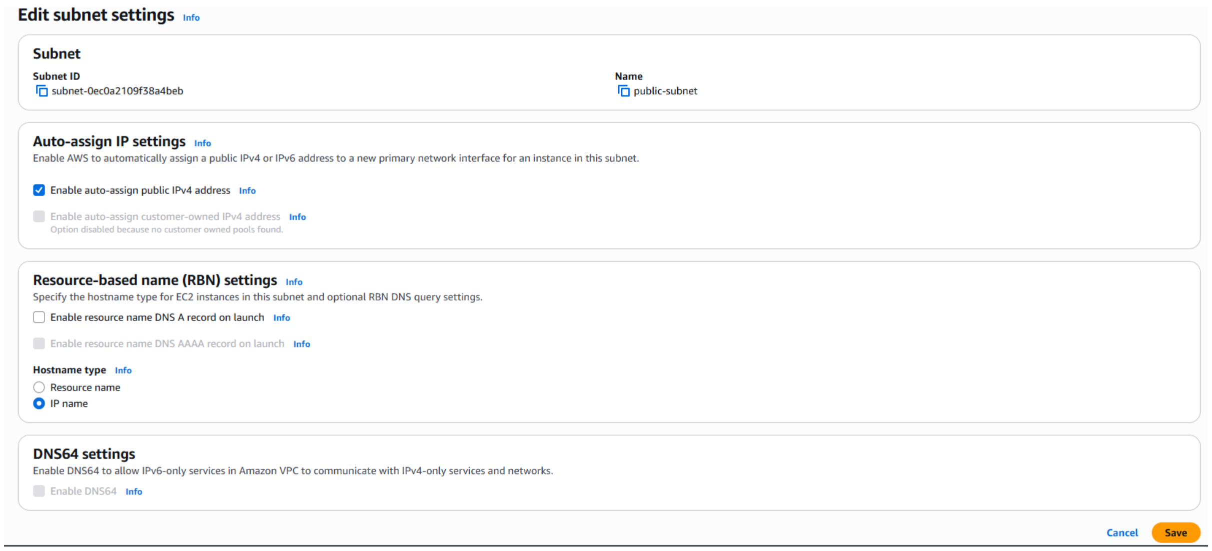
aws ec2 describe-subnets --max-results '1000'

```
aws ec2 create-subnet --vpc-id 'vpc-0fa73da6ce2696934' --cidr-block '10.0.1.0/24' --availability-zone-id 'use1-az6' --tag-specifications '{"resourceType":"subnet","tags":[{"key":"Name","value":"private-subnet"}]}'
```

```
aws ec2 create-subnet --vpc-id 'vpc-0fa73da6ce2696934' --cidr-block '10.0.0.0/24' --availability-zone-id 'use1-az6' --tag-specifications '{"resourceType":"subnet","tags":[{"key":"Name","value":"public-subnet"}]}'
```

STEP 3: Enable Auto-Assign Public IP (Public Subnet)

1. Select **Public-Subnet**
2. Click **Actions → Edit subnet settings**
3. Enable **Auto-assign public IPv4 address**
4. Save



Edit subnet settings Info

Subnet

Subnet ID: subnet-0ec0a2109f38a4beb

Name: public-subnet

Auto-assign IP settings Info

Enable AWS to automatically assign a public IPv4 or IPv6 address to a new primary network interface for an instance in this subnet.

- [✓] Enable auto-assign public IPv4 address Info
- [ ] Enable auto-assign customer-owned IPv4 address Info
  Option disabled because no customer owned pools found.

Resource-based name (RBN) settings Info

Specify the hostname type for EC2 instances in this subnet and optional RBN DNS query settings.

- [ ] Enable resource name DNS A record on launch Info
- [ ] Enable resource name DNS AAAA record on launch Info

Hostname type Info

- ○ Resource name
- ● IP name

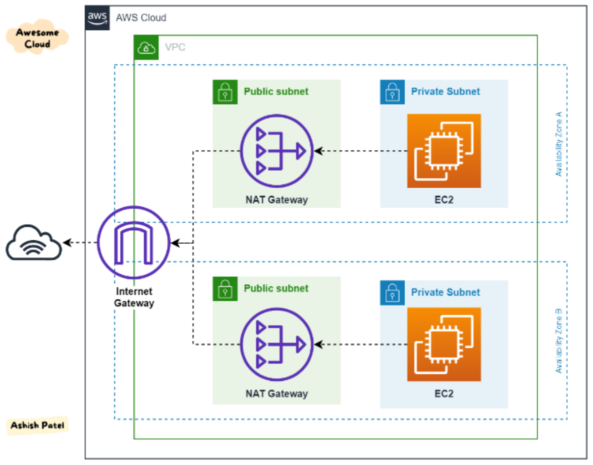
DNS64 settings

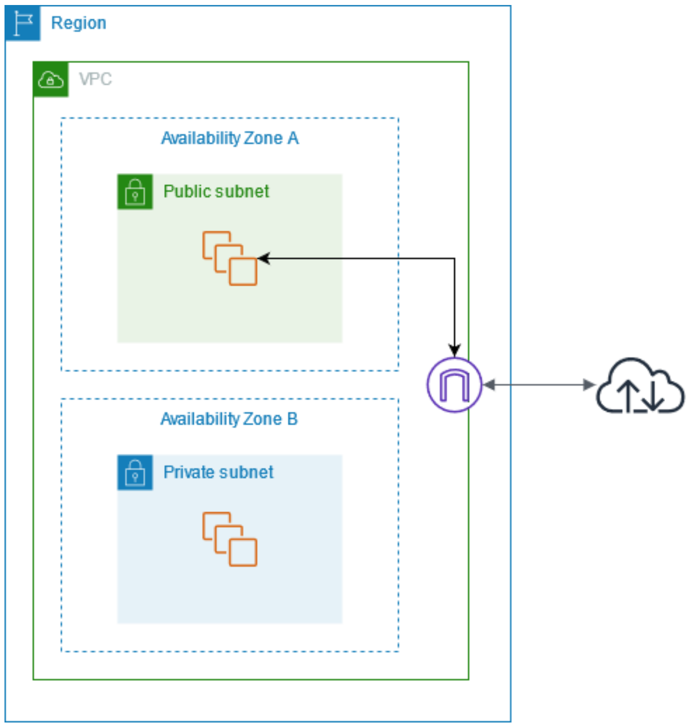
Enable DNS64 to allow IPv6-only services in Amazon VPC to communicate with IPv4-only services and networks.

- [ ] Enable DNS64 Info

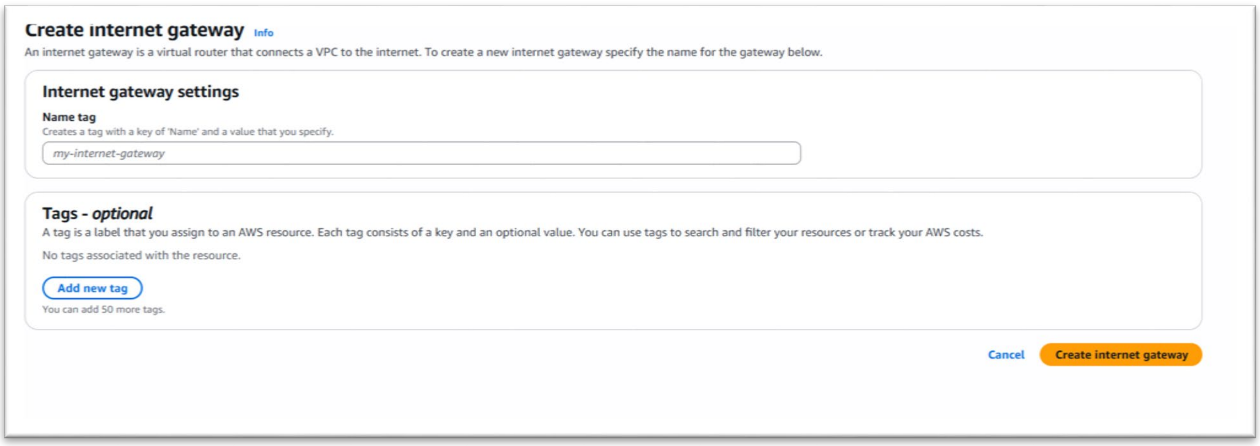
Cancel | Save

STEP 4: Create Internet Gateway (IGW)





1. Go to **VPC → Internet Gateways**
2. Click **Create internet gateway**
3. Name: `My-IGW`
4. Create **igw**



Create internet gateway Info

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

Internet gateway settings

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

my-internet-gateway

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

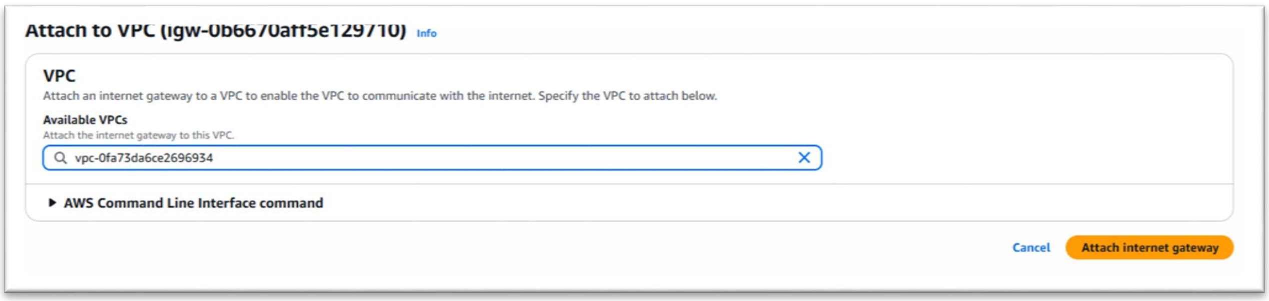
No tags associated with the resource.

Add new tag

You can add 50 more tags.

Cancel | Create internet gateway

5. Select IGW → **Actions** → **Attach to VPC**
6. Attach to **My-VPC**



Attach to VPC (igw-0b6670aff5e129710) Info

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

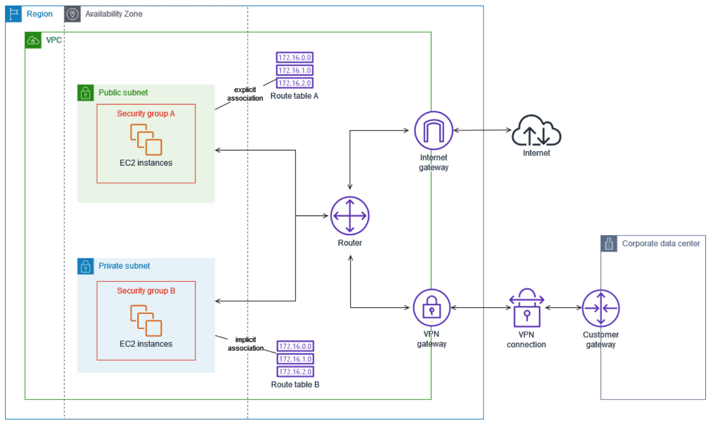
Available VPCs
Attach the internet gateway to this VPC.

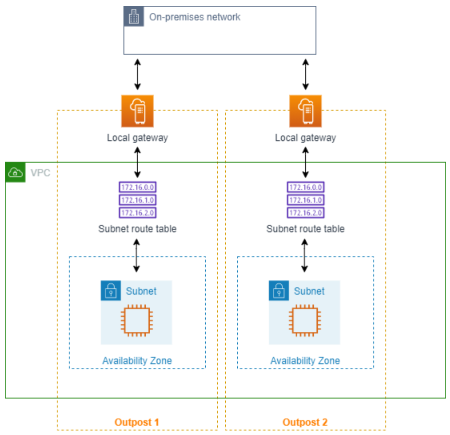
vpc-0fa73da6ce2696934

▸ **AWS Command Line Interface command**

Cancel | Attach internet gateway

STEP 5: Create Route Tables

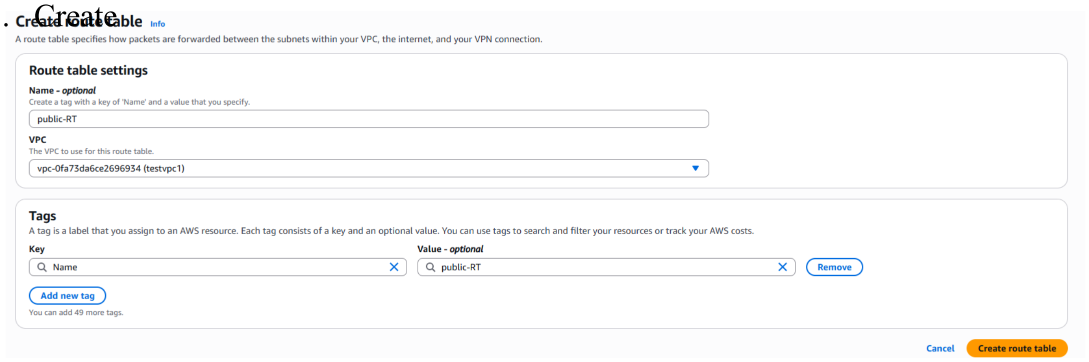




◆ Public Route Table

1. Go to **VPC** → **Route Tables**
2. Click **Create route table**
3. Name: `Public-RT`
4. VPC: select “`My-VPC`”

5. Create



Create route table Info

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

Route table settings

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

VPC
The VPC to use for this route table.
vpc-0fa73da6ce2696934 (testvpc1)

Tags

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

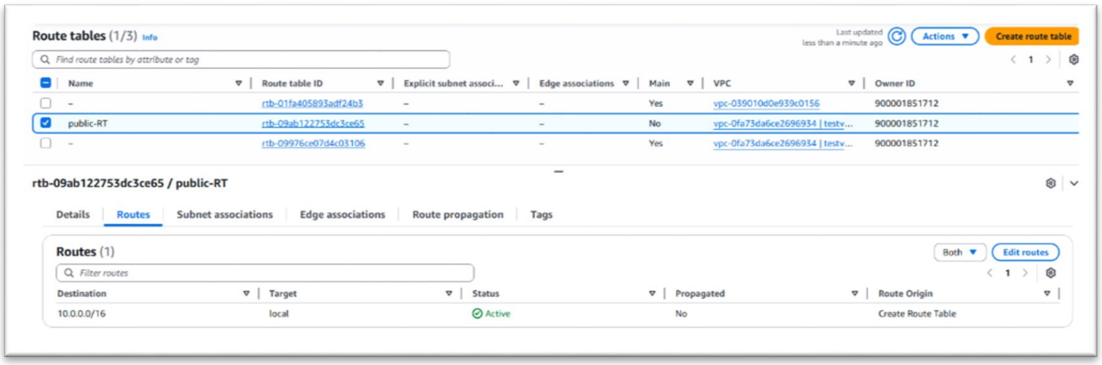
Key	Value - *optional*
Name	public-RT

Add new tag
You can add 49 more tags.

Cancel | Create route table

Add Route

1. Select `Public-RT`
2. Routes → **Edit routes**



Route tables (1/3) Info

Last updated less than a minute ago | Actions | Create route table

Find route tables by attribute or tag

	Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC	Owner ID
☐	-	rtb-01fa405893adf24b3	–	–	Yes	vpc-039010d0e939c0156	900001851712
☑	public-RT	rtb-09ab122753dc3ce65	–	–	No	vpc-0fa73da6ce2696934 | testv...	900001851712
☐	-	rtb-09976ce07d4c03106	–	–	Yes	vpc-0fa73da6ce2696934 | testv...	900001851712

rtb-09ab122753dc3ce65 / public-RT

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

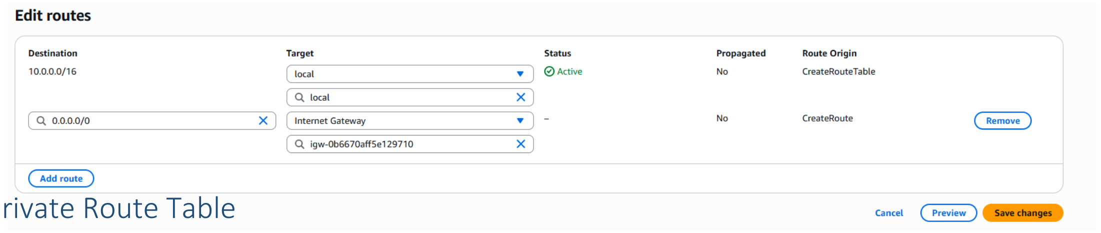
Routes (1)

Both | Edit routes

Filter routes

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/16	local	✓ Active	No	Create Route Table

3. Add:
   - Destination: `0.0.0.0/0`
   - Target: **Internet Gateway (My-IGW)**
4. Save



Edit routes

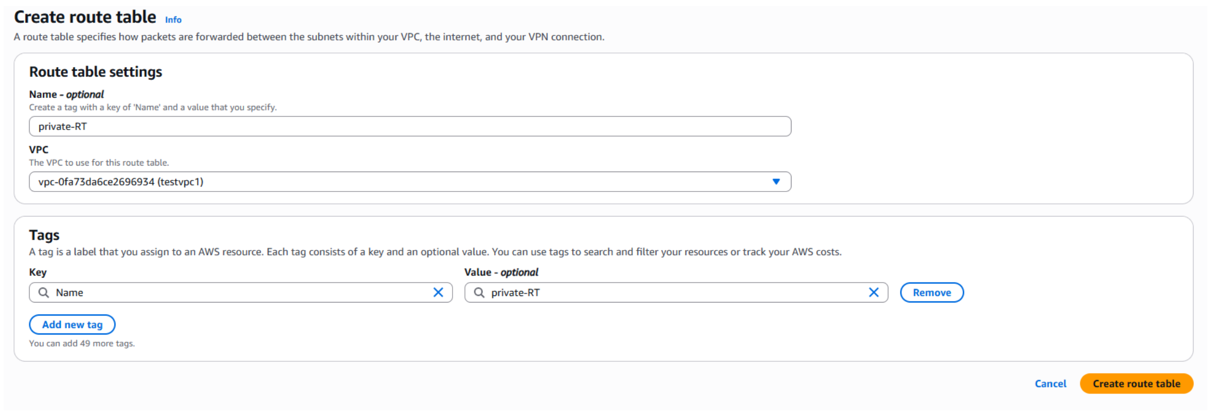
Destination	Target	Status	Propagated	Route Origin	
10.0.0.0/16	local / local	✓ Active	No	CreateRouteTable	
0.0.0.0/0	Internet Gateway / igw-0b6670aff5e129710	-	No	CreateRoute	Remove

Add route

Cancel | Preview | Save changes

Private Route Table

1. Create another route table
2. Name: `Private-RT`
3. VPC: `My-VPC`
4. Create



Create route table Info

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

Route table settings

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

VPC
The VPC to use for this route table.
vpc-0fa73da6ce2696934 (testvpc1)

Tags

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

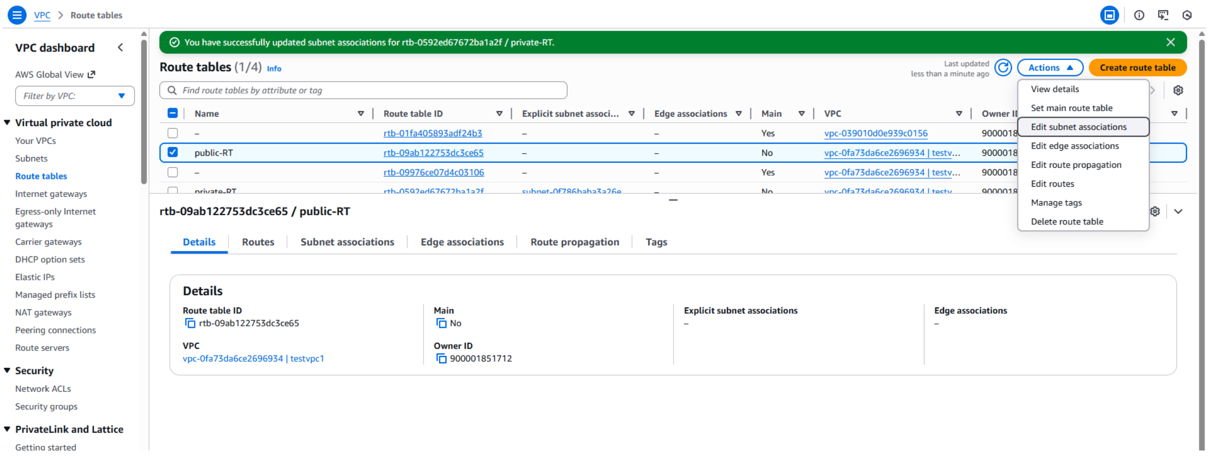
Key	Value - *optional*
Name	private-RT

Add new tag
You can add 49 more tags.

Cancel | Create route table

☞ (No internet route by default)

STEP 6: Associate Subnets with Route Tables



VPC > Route tables

You have successfully updated subnet associations for rtb-0592ed67672ba1a2f / private-RT.

Route tables (1/4)

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC	Owner ID
–	rtb-01fa405893adf24b3	–	–	Yes	vpc-039010d0e939c0156	9000018
public-RT	rtb-09ab122753dc3ce65	–	–	No	vpc-0fa73da6ce2696934 | testv...	9000018
–	rtb-09976ce07d4c03106	–	–	Yes	vpc-0fa73da6ce2696934 | testv...	9000018

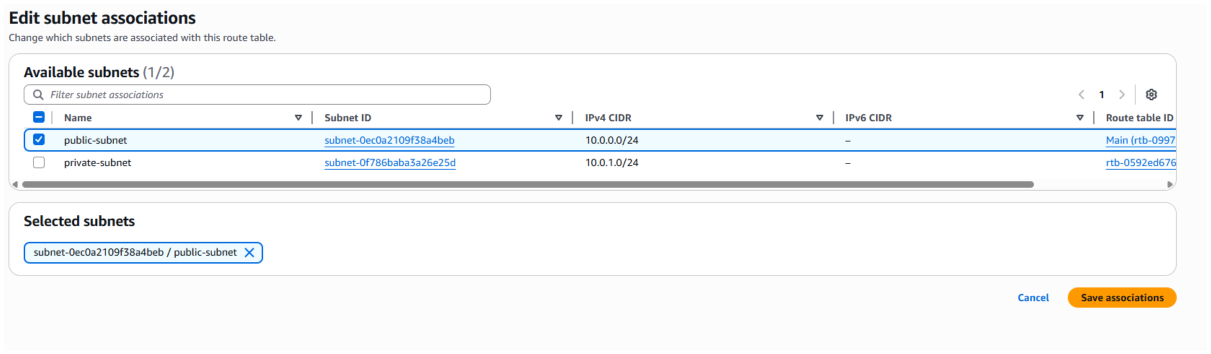
rtb-09ab122753dc3ce65 / public-RT

Details

Route table ID: rtb-09ab122753dc3ce65
Main: No
Explicit subnet associations: –
Edge associations: –
VPC: vpc-0fa73da6ce2696934 | testvpc1
Owner ID: 900001851712

Public Subnet

1. Select **Public-RT**
2. Subnet associations → **Edit**
3. Select **Public-Subnet**
4. Save



Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/2)

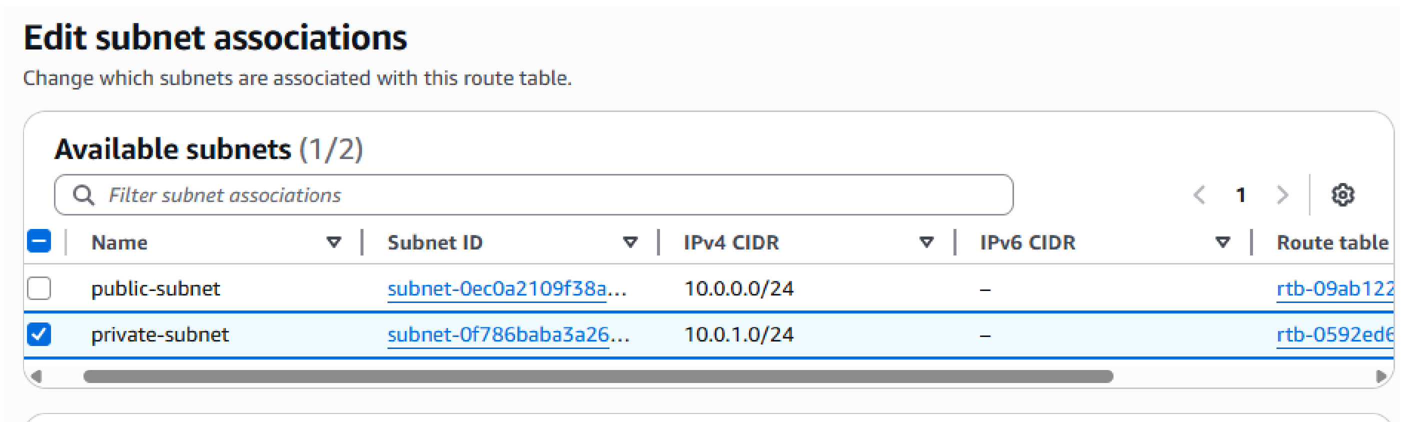
	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
✓	public-subnet	subnet-0ec0a2109f38a4beb	10.0.0.0/24	–	Main (rtb-0997
	private-subnet	subnet-0f786baba3a26e25d	10.0.1.0/24	–	rtb-0592ed676

Selected subnets

subnet-0ec0a2109f38a4beb / public-subnet

Private Subnet

1. Select **Private-RT**
2. Associate **Private-Subnet**
3. Save



Edit subnet associations

Change which subnets are associated with this route table.

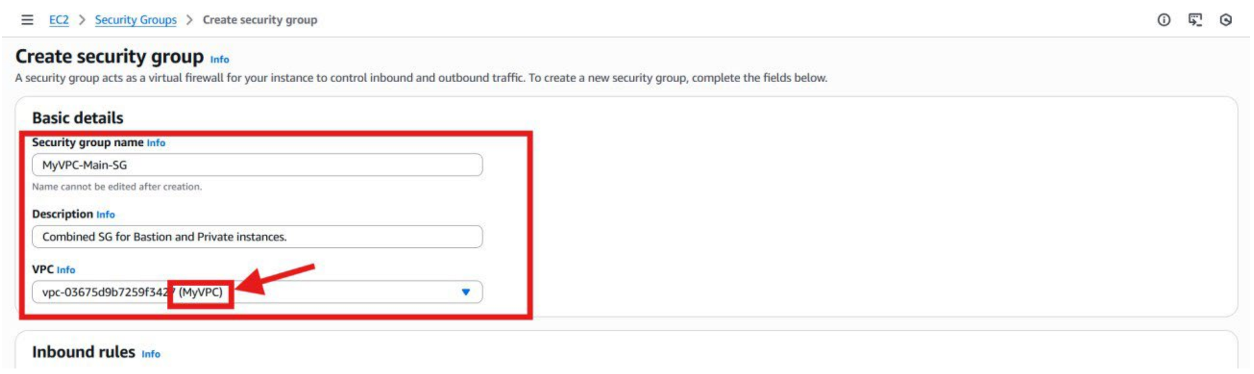
Available subnets (1/2)

	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table
	public-subnet	subnet-0ec0a2109f38a...	10.0.0.0/24	–	rtb-09ab122
✓	private-subnet	subnet-0f786baba3a26...	10.0.1.0/24	–	rtb-0592ed6

Step 7: Configure Security Groups

Create the Security Group

- Go to EC2> Security Groups > Create security group.
- Name: MyVPC-Main-SG.
- Description: Combined SG for Bastion and Private instances.
- VPC: Select MyVPC.


EC2 > Security Groups > Create security group
Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name: MyVPC-Main-SG

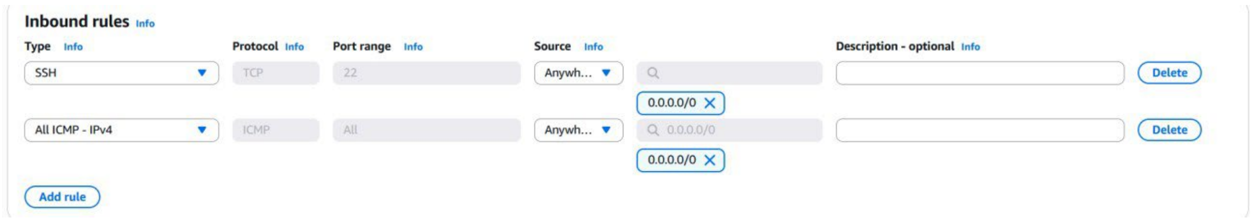
Name cannot be edited after creation.

Description: Combined SG for Bastion and Private instances.

VPC: vpc-03675d9b7259f3427 (MyVPC)

Inbound Rules:

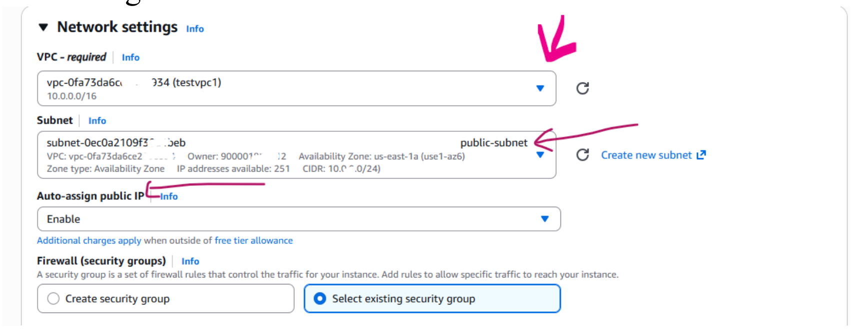
- Type: SSH | Port: 22 | Source: 0.0.0.0/0
- Type: ICMP | Port: all | Source: 0.0.0.0/0



Type	Protocol	Port range	Source		Description - optional
SSH	TCP	22	Anywh...	0.0.0.0/0	
All ICMP - IPv4	ICMP	All	Anywh...	0.0.0.0/0	

Step 8: Launch Public EC2 Instance

1. AMI: **Amazon Linux 2**
2. Instance Type: `t3.micro`
3. Network: Edit network
   - VPC: `myVPC`
   - Subnet: `Public-Subnet`
   - Auto-assign Public IP: **Enable**



Network settings Info

VPC - *required* | Info

vpc-0fa73da6c[redacted]934 (testvpc1)
10.0.0.0/16

Subnet | Info

subnet-0ec0a2109f3[redacted]eb public-subnet
VPC: vpc-0fa73da6ce2[redacted] Owner: 900001[redacted]2 Availability Zone: us-east-1a (use1-az6)
Zone type: Availability Zone IP addresses available: 251 CIDR: 10.0[redacted].0/24)

Create new subnet

Auto-assign public IP Info

Enable

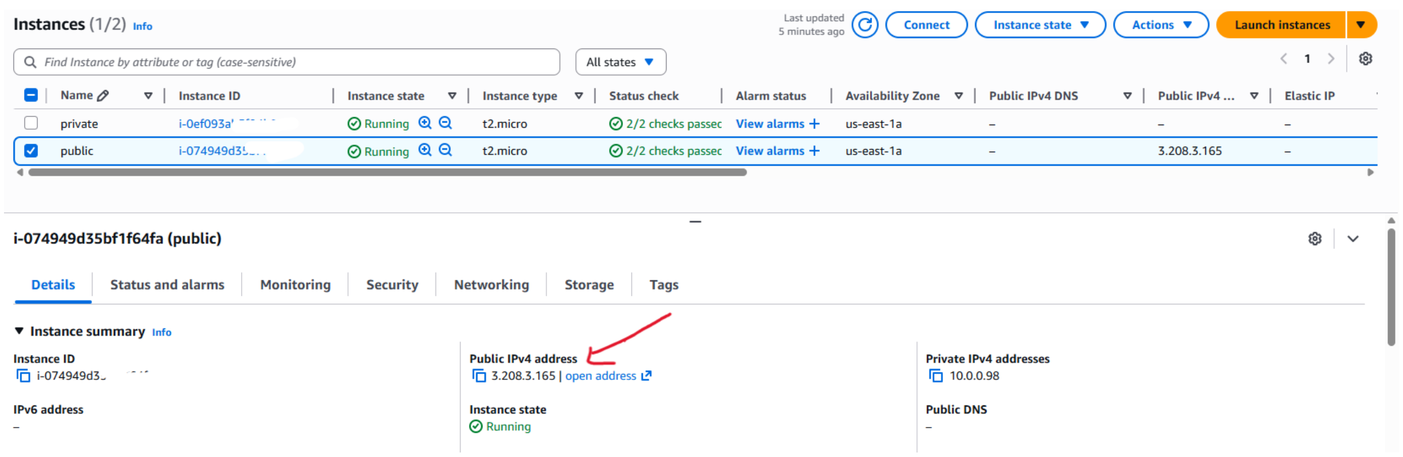
Additional charges apply when outside of free tier allowance

Firewall (security groups) | Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group [✓ Select existing security group]

4. Launch with key pair



Instances (1/2) Info

Find Instance by attribute or tag (case-sensitive)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
☐	private	i-0ef093a'[redacted]	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1a	–	–	–
☑	public	i-074949d3[redacted]	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1a	–	3.208.3.165	–

i-074949d35bf1f64fa (public)

Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags

Instance summary Info

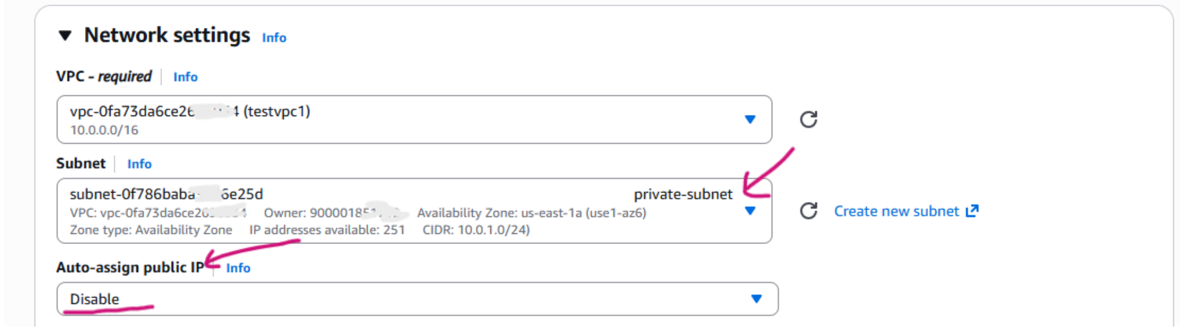
Instance ID	**Public IPv4 address**	**Private IPv4 addresses**
i-074949d3[redacted]	3.208.3.165 | open address	10.0.0.98
IPv6 address	**Instance state**	**Public DNS**
–	Running	–

Public IP seen here

Step 9: Launch Private EC2 Instance

1. EC2 → Launch Instance
2. AMI: Amazon Linux 2
3. Network: Edit network
   - VPC: `myVPC`
   - Subnet: `private-Subnet`

- Auto-assign Public IP: Disable



Network settings Info

VPC - *required* Info

vpc-0fa73da6ce26[redacted]4 (testvpc1)
10.0.0.0/16

Subnet Info

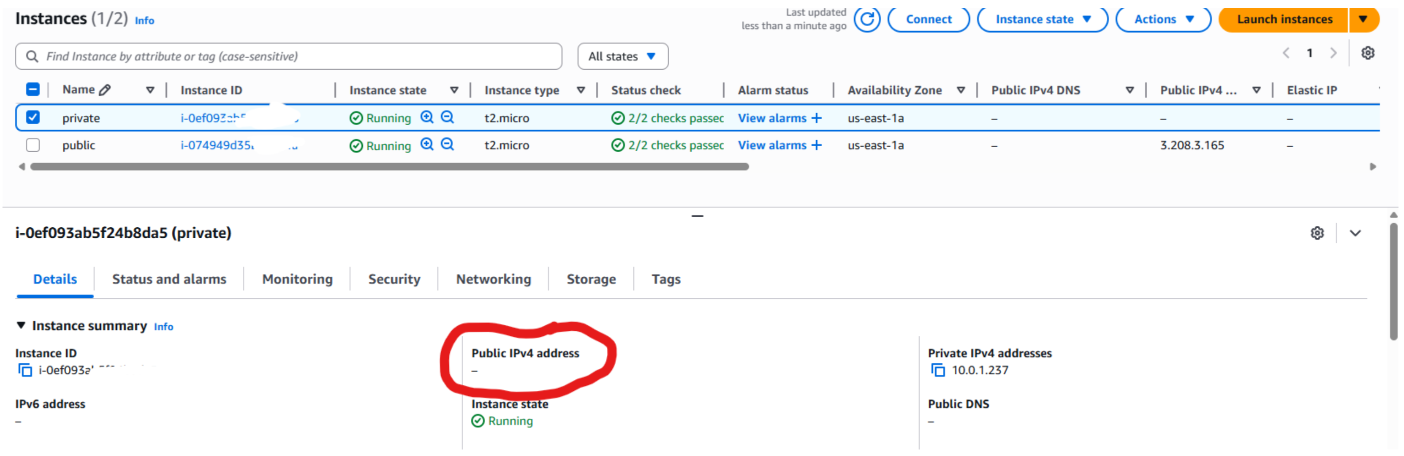
subnet-0f786baba[redacted]6e25d private-subnet
VPC: vpc-0fa73da6ce26[redacted] Owner: 90000185[redacted] Availability Zone: us-east-1a (use1-az6)
Zone type: Availability Zone IP addresses available: 251 CIDR: 10.0.1.0/24)

Create new subnet

Auto-assign public IP Info

Disable

4. Launch with key pair
5. Subnet: **Private Subnet**
6. Security Group:
7. Launch



Instances (1/2) Info

Last updated less than a minute ago

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
✓	private	i-0ef093[redacted]	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1a	–	–	–
	public	i-074949d35[redacted]	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1a	–	3.208.3.165	–

i-0ef093ab5f24b8da5 (private)

Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags

Instance summary Info

Instance ID
i-0ef093[redacted]

Public IPv4 address
–

Private IPv4 addresses
10.0.1.237

IPv6 address
–

Instance state
Running

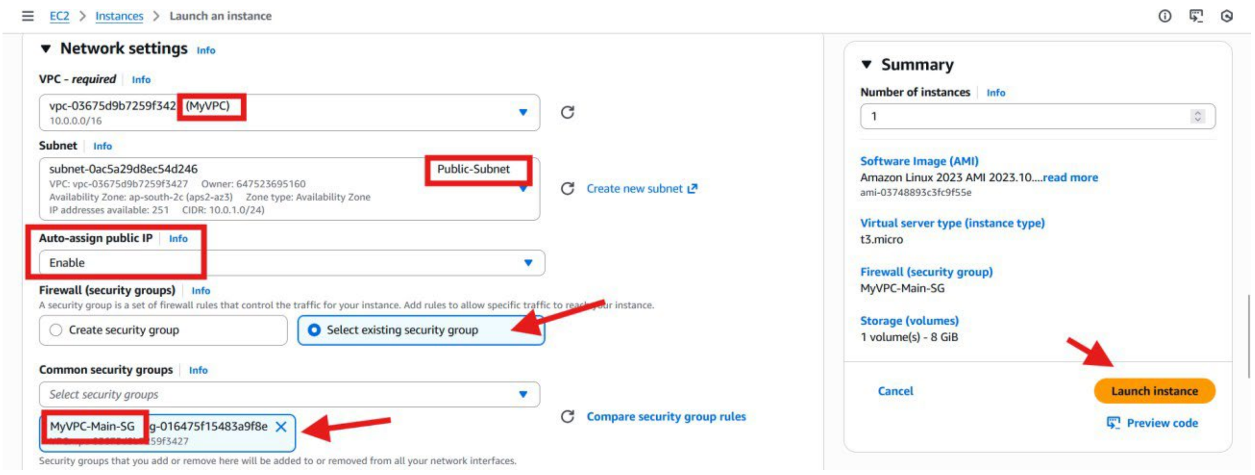
Public DNS
–

See here there is no public IP

Step 8: Launch Instances & Demonstrate

A. Launch the Bastion Host (Public)

1. Go to **EC2** > **Launch Instance**.
2. **Name:** Public-EC2.
3. **Network Settings:** Select **MyVPC** and **Public-Subnet**.
4. **Security Group:** Select the existing **MyVPC-Main-SG**.
5. **Launch.**



EC2 > Instances > Launch an instance

Network settings

VPC - required: vpc-03675d9b7259f342 (MyVPC) 10.0.0.0/16

Subnet: subnet-0ac5a29d8ec54d246 Public-Subnet
VPC: vpc-03675d9b7259f3427 Owner: 647523695160
Availability Zone: ap-south-2c (aps2-az3) Zone type: Availability Zone
IP addresses available: 251 CIDR: 10.0.1.0/24)

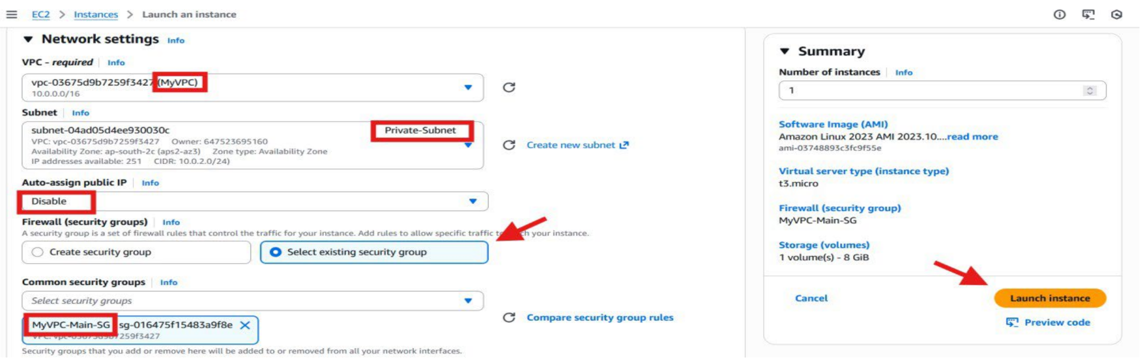
Auto-assign public IP: Enable

Firewall (security groups)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
Create security group [✓ Select existing security group]

Common security groups: MyVPC-Main-SG g-016475f15483a9f8e
Security groups that you add or remove here will be added to or removed from all your network interfaces.

Summary
Number of instances: 1
Software Image (AMI): Amazon Linux 2023 AMI 2023.10....read more ami-03748893c3fc9f55e
Virtual server type (instance type): t3.micro
Firewall (security group): MyVPC-Main-SG
Storage (volumes): 1 volume(s) - 8 GiB

B. Launch the Private Instance (Private)

1. **Launch Instance** again.
2. **Name:** Private-EC2.
3. **Network Settings:** Select MyVPC and Private-Subnet.
   - Note: Ensure "Auto-assign public IP" is Disabled.
4. **Security Group:** Select the existing **MyVPC-Main-SG**.
5. **Launch**
6. 

EC2 > Instances > Launch an instance

Network settings

VPC - required: vpc-03675d9b7259f3427 (MyVPC) 10.0.0.0/16

Subnet: subnet-04ad05d4ee930030c Private-Subnet
VPC: vpc-03675d9b7259f3427 Owner: 647523695160
Availability Zone: ap-south-2c (aps2-az3) Zone type: Availability Zone
IP addresses available: 251 CIDR: 10.0.2.0/24)

Auto-assign public IP: Disable

Firewall (security groups)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
Create security group [✓ Select existing security group]

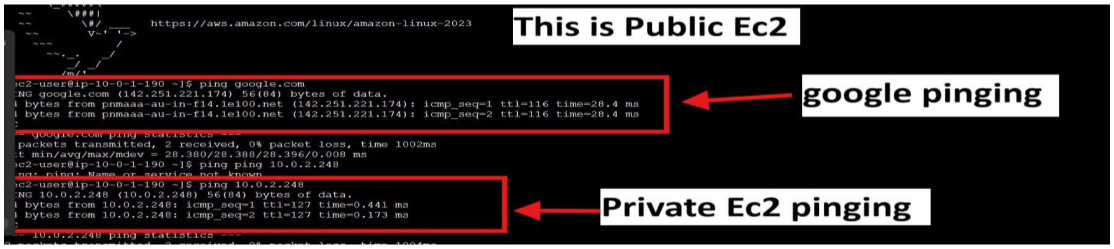
Common security groups: MyVPC-Main-SG sg-016475f15483a9f8e
Security groups that you add or remove here will be added to or removed from all your network interfaces.

Summary
Number of instances: 1
Software Image (AMI): Amazon Linux 2023 AMI 2023.10....read more ami-03748893c3fc9f55e
Virtual server type (instance type): t3.micro
Firewall (security group): MyVPC-Main-SG
Storage (volumes): 1 volume(s) - 8 GiB

Testing the Bastion Hosting

This means SSH into our private instance from our public instance.

1. **Connect to Public-EC2:** Open your terminal and SSH into the **Public-EC2** instance.

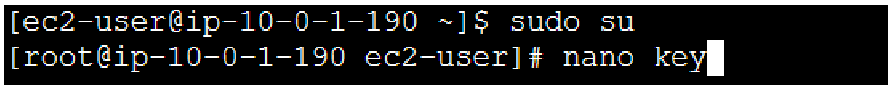


```
~~       \###|
~~        \#/  ___   https://aws.amazon.com/linux/amazon-linux-2023
 ~~        V~' '->
  ~~~         /
    ~~._.   _/
       _/ _/
      _/m/'
ec2-user@ip-10-0-1-190 ~]$ ping google.com
NG google.com (142.251.221.174) 56(84) bytes of data.
 bytes from pnmaaa-au-in-f14.1e100.net (142.251.221.174): icmp_seq=1 ttl=116 time=28.4 ms
 bytes from pnmaaa-au-in-f14.1e100.net (142.251.221.174): icmp_seq=2 ttl=116 time=28.4 ms

- google.com ping statistics ---
 packets transmitted, 2 received, 0% packet loss, time 1002ms
t min/avg/max/mdev = 28.380/28.388/28.396/0.008 ms
ec2-user@ip-10-0-1-190 ~]$ ping ping 10.0.2.248
ng: ping: Name or service not known
ec2-user@ip-10-0-1-190 ~]$ ping 10.0.2.248
NG 10.0.2.248 (10.0.2.248) 56(84) bytes of data.
 bytes from 10.0.2.248: icmp_seq=1 ttl=127 time=0.441 ms
 bytes from 10.0.2.248: icmp_seq=2 ttl=127 time=0.173 ms

-- 10.0.2.248 ping statistics ---
```

2. Copy the SSH Key to the Public-EC2 so that we can use it to connect to the Private-EC2.

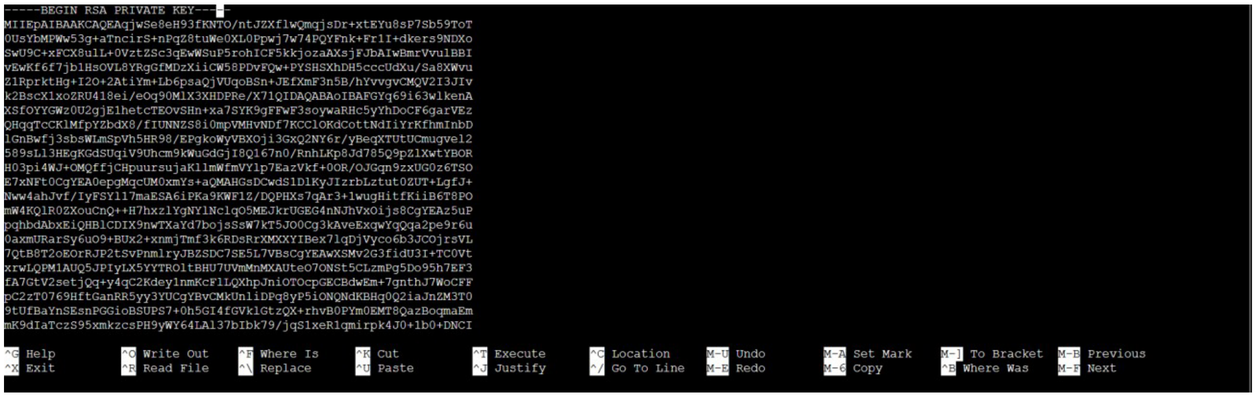
   Follow the comments below:

   $ sudo su
   # nano key



```
[ec2-user@ip-10-0-1-190 ~]$ sudo su
[root@ip-10-0-1-190 ec2-user]# nano key
```

   Nano text editor will open past the public SSH key there.
   Should be something like this:



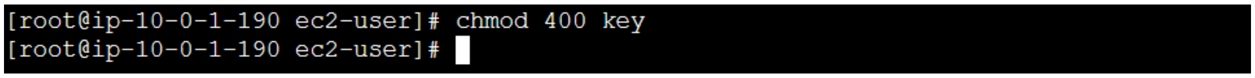
```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAqjwSe8eH93fKNTO/ntJZXflwQmqjsDr+xtEYu8sP7Sb59ToT
0UsYbMPWw53g+aTncirS+nPqZ8tuWe0XL0Ppwj7w74PQYFnk+Fr1I+dkers9NDXo
SwU9C+xFCX8ulL+0VztZSc3qEwWSuP5rohICF5kkjozaAXsjFJbAIwBmrVvulBBI
vEwKf6f7jblHsOVL8YRgGfMDzXiiCW58PDvFQw+PYSHSXhDH5cccUdXu/Sa8XWvu
Z1RprktHg+I2O+2AtiYm+Lb6psaQjVUqoBSn+JEfXmF3n5B/hYvvgvCMQV2I3JIv
k2BscX1xoZRU418ei/eOq90MlX3XHDPRe/X71QIDAQABAoIBAFGYq69i63wlkenA
XSfOYYGWz0U2gjE1hetcTEOvSHn+xa7SYK9gFFwF3soywaRHc5yYhDoCF6garVEz
QHqqTcCKlMfpYZbdX8/fIUNNZS8i0mpVMHvNDf7KCClOKdCottNdIiYrKfhmInbD
lGnBwfj3sbsWLmSpVh5HR98/EPgkoWyVBXOji3GxQ2NY6r/yBeqXTUtUCmugvel2
589sLl3HEgKGdSUqiV9Uhcm9kWuGdGjI8Q167n0/RnhLKp8Jd785Q9pZlXwtYBOR
H03pi4WJ+OMQffjCHpuursujaKllmWfmVYlp7EazVkf+0OR/OJGqn9zxUG0z6TSO
E7xNFt0CgYEA0epgMqcUM0xmYs+aQMAHGsDCwdS1DlKyJIzrbLztut0ZUT+LgfJ+
Nww4ahJvf/IyFSYl17maESA6iPKa9KWF1Z/DQPHXs7qAr3+1wugHitfKiiB6T8PO
mW4KQlR0ZXouCnQ++H7hxzlYgNYlNclqO5MEJkrUGEG4nNJhVxOijs8CgYEAz5uP
pqhbdAbxEiQHBlCDIX9nwTXaYd7bojsSsW7kT5JO0Cg3kAveExqwYqQqa2pe9r6u
0axmURarSy6uO9+BUx2+xnmjTmf3k6RDsRrXMXXYIBex7lqDjVyco6b3JCOjrsVL
7QtB8T2oEOrRJP2tSvPnmlryJBZSDC7SE5L7VBsCgYEAwXSMv2G3fidU3I+TC0Vt
xrwLQPM1AUQ5JPIyLX5YYTROltBHU7UVmMnMXAUteO7ONSt5CLzmPg5Do95h7EF3
fA7GtV2setjQq+y4qC2Kdey1nmKcFlLQXhpJniOTOcpGECBdwEm+7gnthJ7WoCFF
pC2zT0769HftGanRR5yy3YUCgYBvCMkUnliDPq8yP5iONQNdKBHq0Q2iaJnZM3T0
9tUfBaYnSEsnPGGioBSUPS7+0h5GI4fGVklGtzQX+rhvB0PYm0EMT8QazBoqmaEm
mK9dIaTczS95xmkzcsPH9yWY64LAl37bIbk79/jqS1xeR1qmirpk4J0+1b0+DNCI

^G Help       ^O Write Out  ^F Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-] To Bracket M-B Previous
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy      ^B Where Was   M-F Next
```

   To save in nano use: **Ctrl+S**

   To exit the nano use: **Ctrl+X**

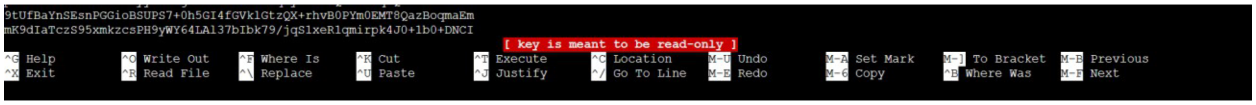
   Next step is to change the key to be **read-only**, use the following comment:

   # chmod 400 key



```
[root@ip-10-0-1-190 ec2-user]# chmod 400 key
[root@ip-10-0-1-190 ec2-user]#
```

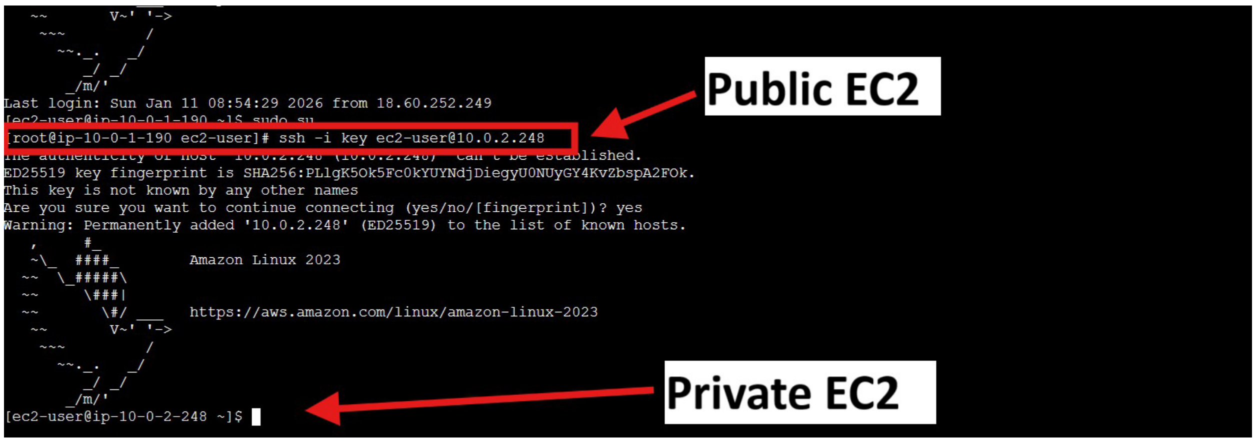
# nano key (again to check if the file only has read only permission)



```
9tUfBaYnSEsnPGGioBSUPS7+0h5GI4fGVklGtzQX+rhvB0PYm0EMT8QazBoqmaEm
mK9dIaTczS95xmkzcsPH9yWY64LAl37bIbk79/jqSlxeRlqmirpk4J0+1b0+DNCI
                                        [ key is meant to be read-only ]
^G Help         ^O Write Out    ^W Where Is     ^K Cut          ^T Execute      ^C Location     M-U Undo        M-A Set Mark    M-] To Bracket  M-B Previous
^X Exit         ^R Read File    ^\ Replace      ^U Paste        ^J Justify      ^/ Go To Line   M-E Redo        M-6 Copy        ^B Where Was    M-F Next
```

3. **Connect to Private Instance:** From inside the Public-EC2, SSH into the private instance using its **Private IP**.

   # ssh -i key ec2-user@<Private-Instance-IP>

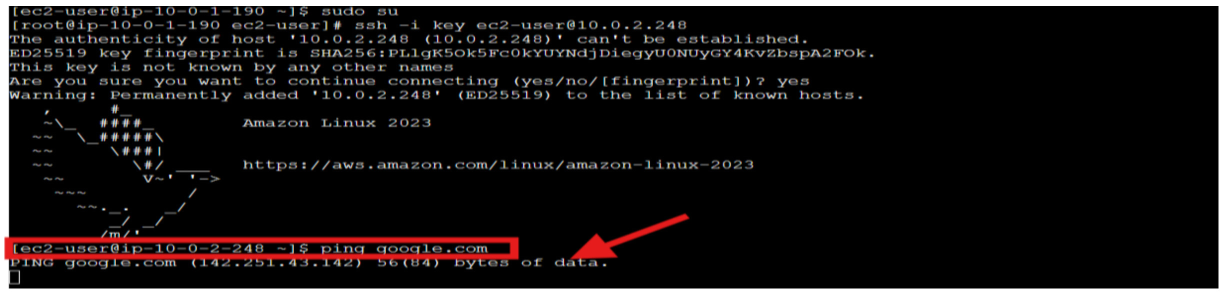


```
  ~~       V~' '->
   ~~~         /
     ~~._.   _/
        _/ _/
      _/m/'
Last login: Sun Jan 11 08:54:29 2026 from 18.60.252.249
[ec2-user@ip-10-0-1-190 ~]$ sudo su
[root@ip-10-0-1-190 ec2-user]# ssh -i key ec2-user@10.0.2.248
The authenticity of host '10.0.2.248 (10.0.2.248)' can't be established.
ED25519 key fingerprint is SHA256:PLlgK5Ok5Fc0kYUYNdjDiegyU0NUyGY4KvZbspA2FOk.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.248' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-2-248 ~]$
```

Verify we don’t have internet access without NAT Gateway in Private-EC2 by

   $ ping google.com

As you can see, we can’t be able to ping from Private-EC2 as below:

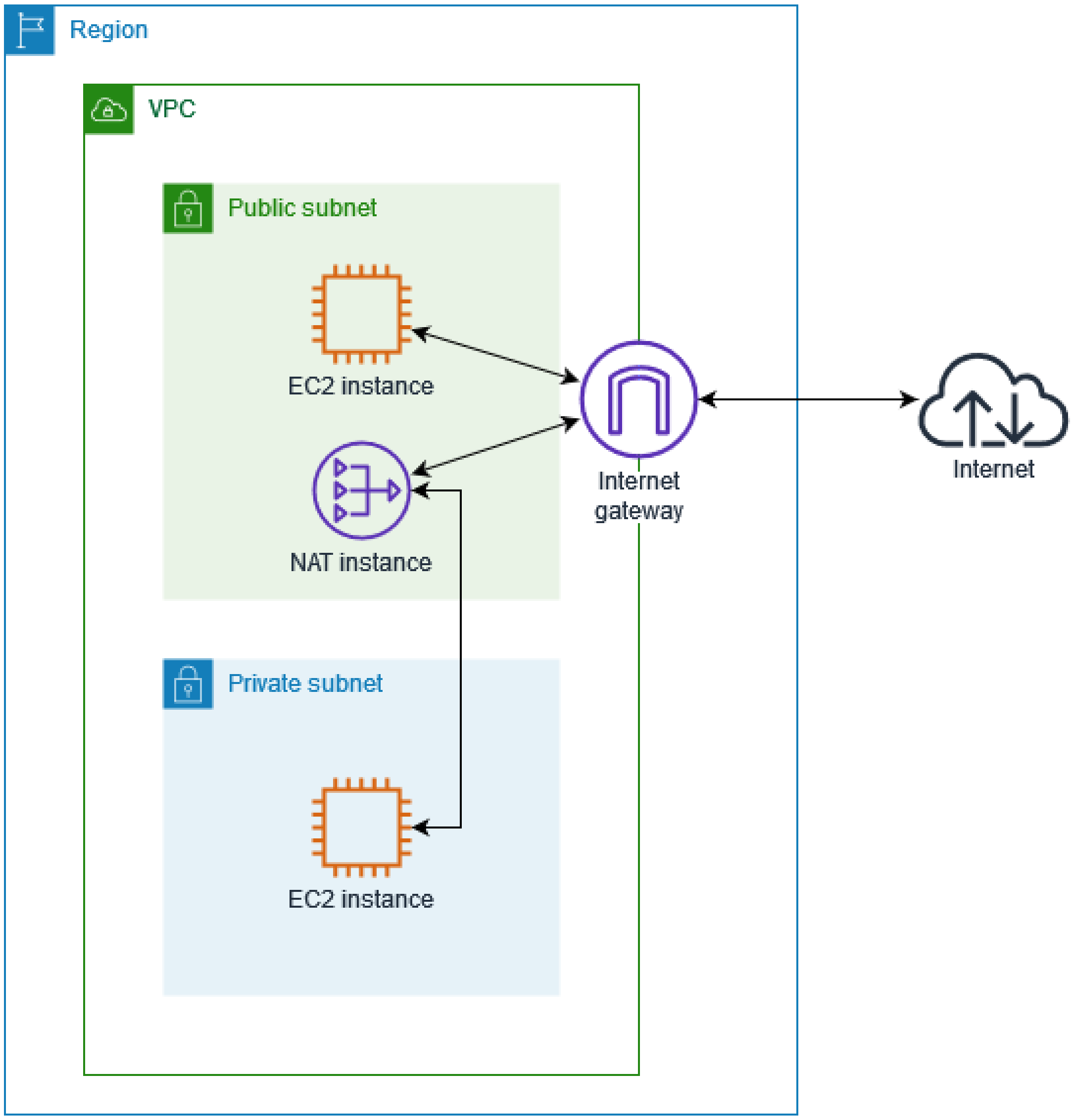


```
[ec2-user@ip-10-0-1-190 ~]$ sudo su
[root@ip-10-0-1-190 ec2-user]# ssh -i key ec2-user@10.0.2.248
The authenticity of host '10.0.2.248 (10.0.2.248)' can't be established.
ED25519 key fingerprint is SHA256:PLlgK5Ok5Fc0kYUYNdjDiegyU0NUyGY4KvZbspA2FOk.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.248' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-2-248 ~]$ ping google.com
PING google.com (142.251.43.142) 56(84) bytes of data.
```

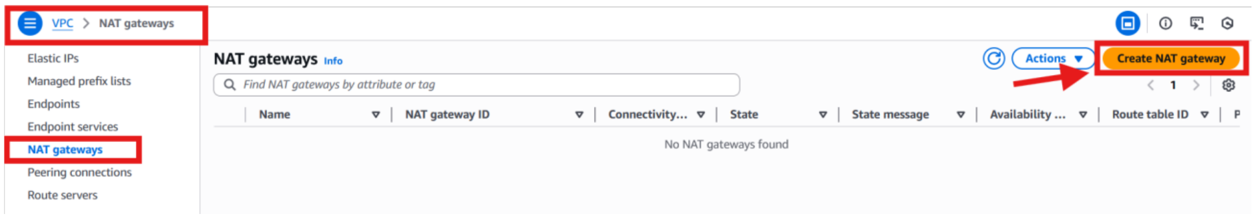
Create the NAT Gateway (For Private Subnet)

This allows your private instances to "go out" to the internet without let anyone "coming in."

The following is the architecture of the NAT Gateway:



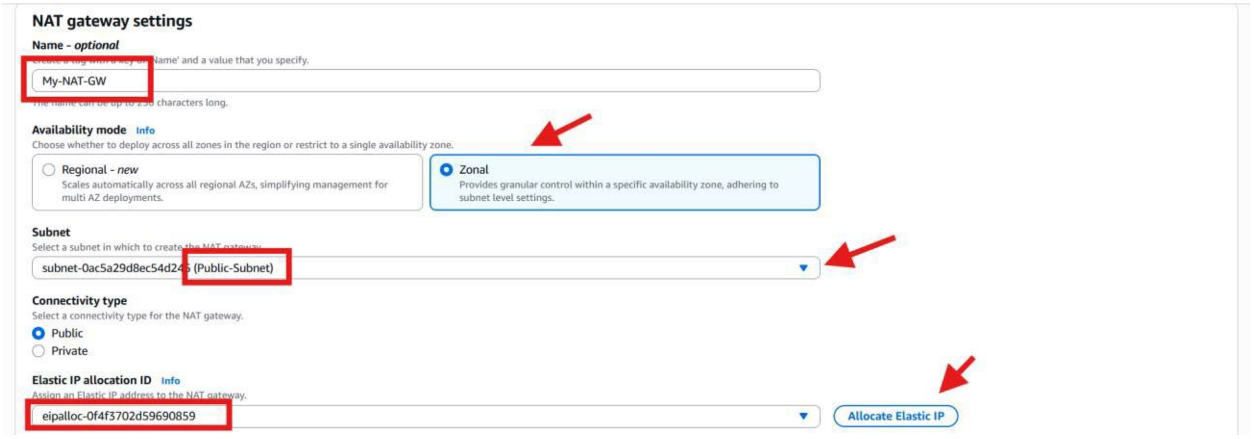
1. Go to VPC service page.
2. In the left sidebar, click **NAT Gateways**, then **Create NAT gateway**.



Name	NAT gateway ID	Connectivity...	State	State message	Availability ...	Route table ID	P

No NAT gateways found

3. **Name:** My-NAT-GW.
4. Select Availability Zone: Zonal
5. **Subnet: CRITICAL:** You must select the **Public-Subnet**.
6. Elastic IP allocation ID: Click Allocate Elastic IP.



NAT gateway settings

Name - *optional*

...Name' and a value that you specify.

My-NAT-GW

... characters long.

Availability mode Info

Choose whether to deploy across all zones in the region or restrict to a single availability zone.

- Regional - *new* — Scales automatically across all regional AZs, simplifying management for multi AZ deployments.
- [✓ Zonal] — Provides granular control within a specific availability zone, adhering to subnet level settings.

Subnet

Select a subnet in which to create the NAT gateway.

subnet-0ac5a29d8ec54d245 (Public-Subnet)

Connectivity type

Select a connectivity type for the NAT gateway.

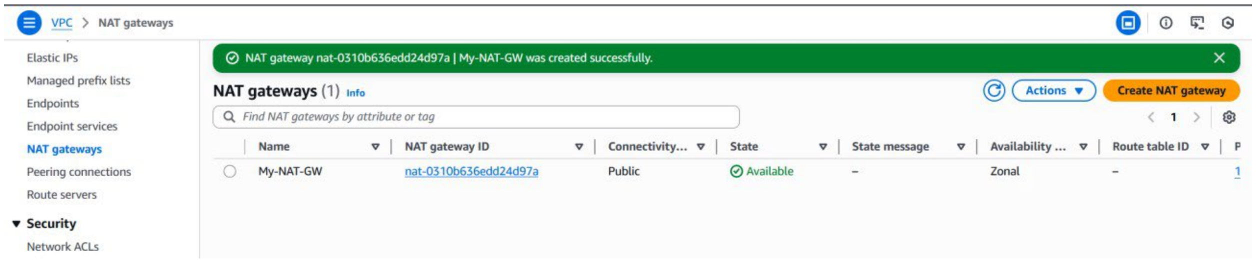
[✓ Public] Private

Elastic IP allocation ID Info

Assign an Elastic IP address to the NAT gateway.

eipalloc-0f4f3702d59690859

7. Click **Create NAT gateway**.



NAT gateway nat-0310b636edd24d97a | My-NAT-GW was created successfully.

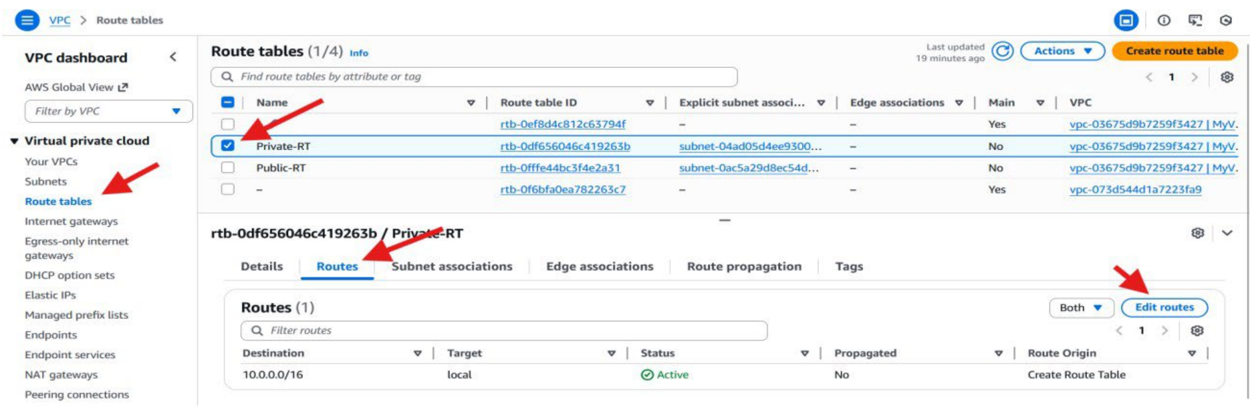
NAT gateways (1) Info

Find NAT gateways by attribute or tag

Name	NAT gateway ID	Connectivity...	State	State message	Availability ...	Route table ID	P
My-NAT-GW	nat-0310b636edd24d97a	Public	Available	–	Zonal	–	1

Update Private Route Table:

8. Go back to **Route Tables**.
9. find **Private-RT** add a route for 0.0.0.0/0 pointing to the **NAT Gateway** you just created.



Route tables (1/4) Info

Find route tables by attribute or tag

	Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
☐		rtb-0ef8d4c812c63794f	–	–	Yes	vpc-03675d9b7259f3427 | MyV.
☑	Private-RT	rtb-0df656046c419263b	subnet-04ad05d4ee9300...	–	No	vpc-03675d9b7259f3427 | MyV.
☐	Public-RT	rtb-0fffe44bc3f4e2a31	subnet-0ac5a29d8ec54d...	–	No	vpc-03675d9b7259f3427 | MyV.
☐	–	rtb-0f6bfa0ea782263c7	–	–	Yes	vpc-073d544d1a7223fa9

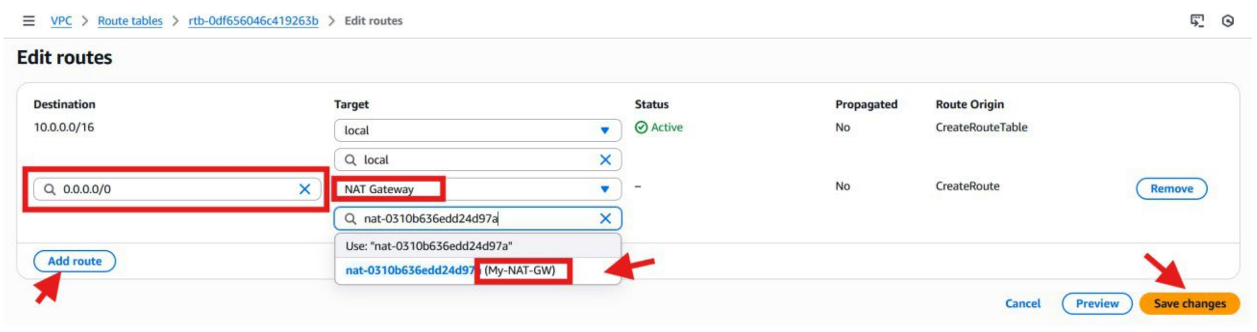
rtb-0df656046c419263b / Private-RT

Routes (1)

Filter routes

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/16	local	Active	No	Create Route Table

10. Add the NAT Route:
    1. With **Private-RT** selected, click the **Routes** tab and select **Edit routes**.
    2. Click **Add route**.
    3. **Destination:** 0.0.0.0/0.
    4. **Target:** Select **NAT Gateway** and choose the My-NAT-GW you just created.
    5. Click **Save changes**.

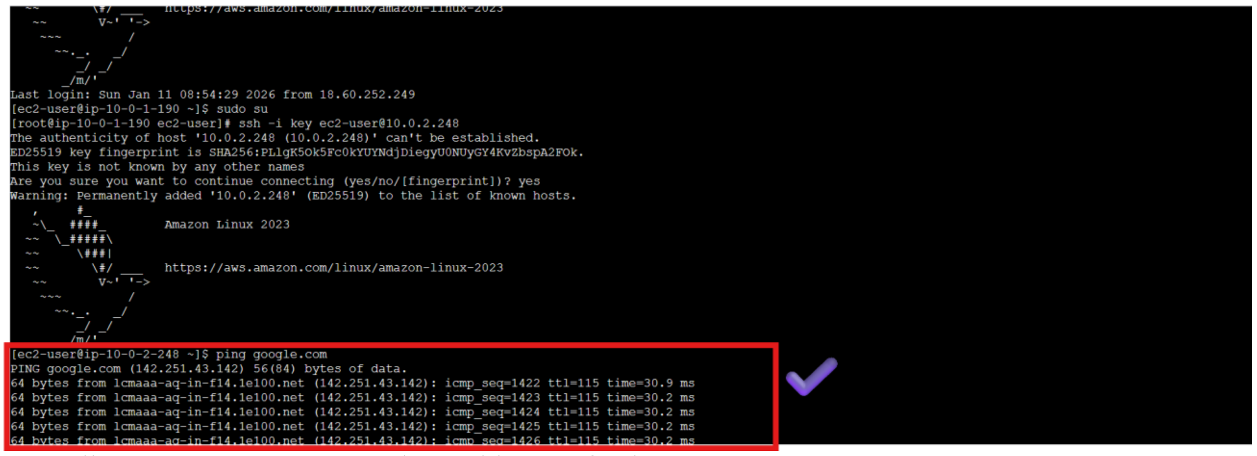

VPC > Route tables > rtb-0df656046c419263b > Edit routes
Edit routes

Destination	Target	Status	Propagated	Route Origin	
10.0.0.0/16	local	Active	No	CreateRouteTable	
0.0.0.0/0	NAT Gateway nat-0310b636edd24d97a	-	No	CreateRoute	Remove

local
Use: "nat-0310b636edd24d97a"
nat-0310b636edd24d97a (My-NAT-GW)
Add route
Cancel
Preview
Save changes

4. **Verify NAT Gateway:** Once inside the private instance, run a ping test to see if it can reach the internet (via the NAT Gateway) even though it has no public IP.

   $ ping google.com



```
 ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
  ~~       V~' '->
   ~~~         /
     ~~._.   _/
        _/ _/
      _/m/'
Last login: Sun Jan 11 08:54:29 2026 from 18.60.252.249
[ec2-user@ip-10-0-1-190 ~]$ sudo su
[root@ip-10-0-1-190 ec2-user]# ssh -i key ec2-user@10.0.2.248
The authenticity of host '10.0.2.248 (10.0.2.248)' can't be established.
ED25519 key fingerprint is SHA256:PLlgK5Ok5FcOkYUYNdjDiegyU0NUyGY4KvZbspA2FOk.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.248' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-2-248 ~]$ ping google.com
PING google.com (142.251.43.142) 56(84) bytes of data.
64 bytes from lcmaaa-aq-in-f14.1e100.net (142.251.43.142): icmp_seq=1422 ttl=115 time=30.9 ms
64 bytes from lcmaaa-aq-in-f14.1e100.net (142.251.43.142): icmp_seq=1423 ttl=115 time=30.2 ms
64 bytes from lcmaaa-aq-in-f14.1e100.net (142.251.43.142): icmp_seq=1424 ttl=115 time=30.2 ms
64 bytes from lcmaaa-aq-in-f14.1e100.net (142.251.43.142): icmp_seq=1425 ttl=115 time=30.2 ms
64 bytes from lcmaaa-aq-in-f14.1e100.net (142.251.43.142): icmp_seq=1426 ttl=115 time=30.2 ms
```

If you see replies, your NAT Gateway is working perfectly!

Security Best Practices

- Bastion allows **controlled SSH access**
- Private EC2 has **no public exposure**
- NAT Gateway provides **outbound-only internet**
- Use **IAM roles** instead of keys where possible

Conclusion

- Multi-subnet VPC design
- Secure bastion-based access
- NAT Gateway for private workloads
- Real-world AWS production architecture