

Step-by-Step Guide to Creating an AWS VPC

Bastion Host Setup or Jump Server Setup

A VPC, or Virtual Private Cloud, is a logically isolated virtual network within the AWS Cloud where you can launch your AWS resources. It provides complete control over your virtual networking environment, including your own IP address range, subnets, route tables, and network gateways.

Prerequisites

- AWS Account.
- Basic understanding of networking concepts like ip and CIDR.

Learning Goals

- Creating the VPC by using VPC only option.
- The typical VPC setup creation and testing
 - **Public Subnet** → EC2 that can be reached from the internet (e.g. web server, bastion host, load balancer). The EC2 has Elastic Ip.
 - **Private Subnet** → EC2 that cannot be reached directly from the internet (e.g. database, application backend, cache).
- To reach private subnet, first SSH into public subnet than try SSH into the private subnet.

Step 1: Create a VPC

- Create a VPC -> Vpc only -> Name
- **Ipv4 CIDR block**
 - Ipv4 CIDR block - Default and Manual selection of non-overlapping ip ✓
 - Ipv4 CIDR manual input/IPAM-allocated IPv4 CIDR block - **IP Address Manager (IPAM)** to automatically assign a CIDR block to your VPC. Instead of you picking a range, IPAM pulls an available CIDR block from a predefined pool. This prevents IP address conflicts, ensures compliance with your organization's networking rules, and automates the process, making it much easier to scale in large, multi-account environments.
- **Ipv4 CIDR**
 - Here I used the Ip range between 10.0.0.0/28 (16 Ip's).
 - Vpc ip are working with RFC 1918.

RFC 1918

RFC 1918 is a standard that defines private IPv4 address ranges that are not routable over the public internet. They're meant for internal/private networks.

The three ranges are:

- 10.0.0.0 – 10.255.255.255 (CIDR: 10.0.0.0/8)
- 172.16.0.0 – 172.31.255.255 (CIDR: 172.16.0.0/12)
- 192.168.0.0 – 192.168.255.255 (CIDR: 192.168.0.0/16)

AWS recommends you choose your VPC CIDR block from one of these ranges to keep your VPC private.

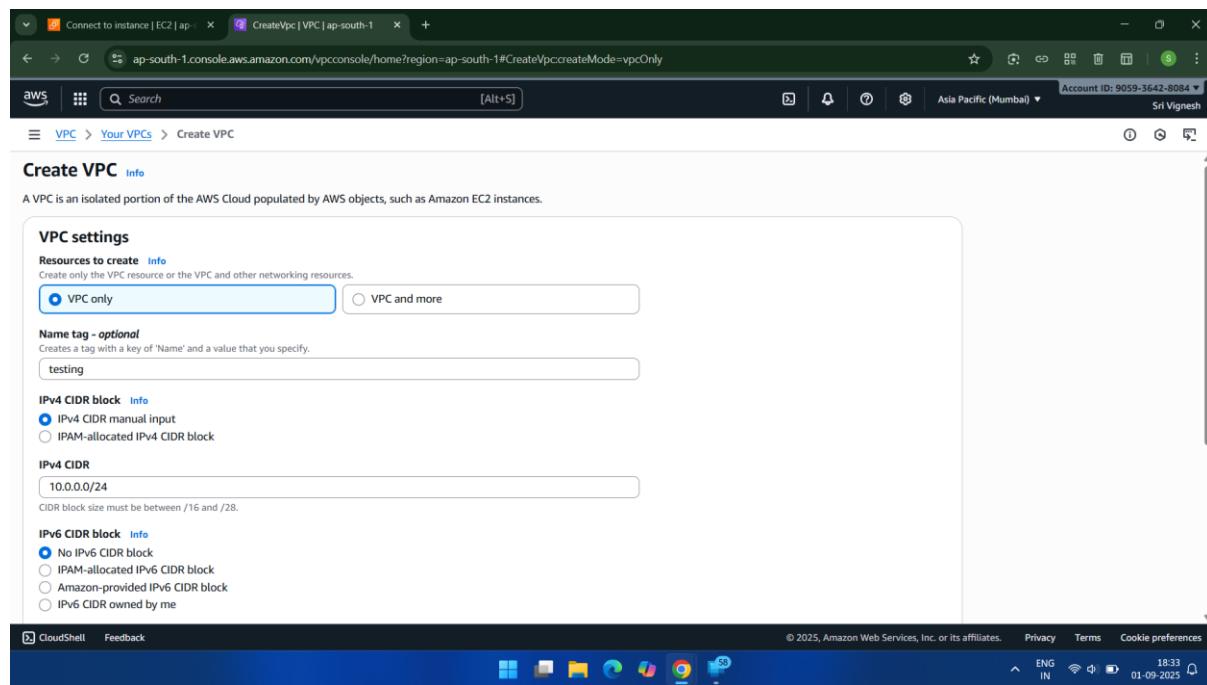
In AWS, every subnet loses 5 IPs that you can't assign to resources:

1. First IP (network address) – Identifies the subnet itself, not a device.
2. Second IP (VPC router) – Used by AWS as the default gateway for routing traffic.
3. Third IP (DNS server) – Reserved for the Amazon-provided DNS in that VPC.
4. Fourth IP (future use) – Held for AWS internal purposes.
5. Last IP (broadcast address) – Used for broadcasting to all devices in that subnet (even though AWS doesn't actually support broadcast traffic).

- **Ipv6 CIDR block**

- No IPv6 CIDR block: No IPv6 address range is associated with the VPC. ✓
- IPAM-allocated IPv6 CIDR block: An IPv6 CIDR block assigned to the VPC by AWS IP Address Manager (IPAM).
- Amazon-provided IPv6 CIDR block: A globally unique IPv6 CIDR block provided by AWS that you can associate with your VPC.
- IPv6 CIDR owned by me: A publicly routable IPv6 CIDR block that you own and have brought into AWS to use within your VPC.

- Create a VPC.



Sri Vignesh

Account ID: 9059-3642-8084

EC2 Global View

Filter by VPC:

VPC dashboard

You successfully created vpc-0d80104c3937f12d2 / testing

vpc-0d80104c3937f12d2 / testing

Details

VPC ID vpc-0d80104c3937f12d2	State Available	Block Public Access Off	DNS hostnames Disabled
DNS resolution Enabled	Tenancy default	DHCP option set dopt-007a45c7dd31ecad2	Main route table rtb-04b2b346bf72347e2
Main network ACL acl-00e1d8645717115e9	Default VPC No	IPv4 CIDR 10.0.0.0/24	IPv6 pool -
IPv6 CIDR (Network border group) -	Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID 905936428084

Resource map

Show all details

VPC Subnets (0) Route tables (1) Network Connections (0)

Your AWS virtual network Subnets within this VPC Subnets (0) Subnets within this VPC

Route tables (1) Route network traffic to resources rtb-04b2b346bf72347e2

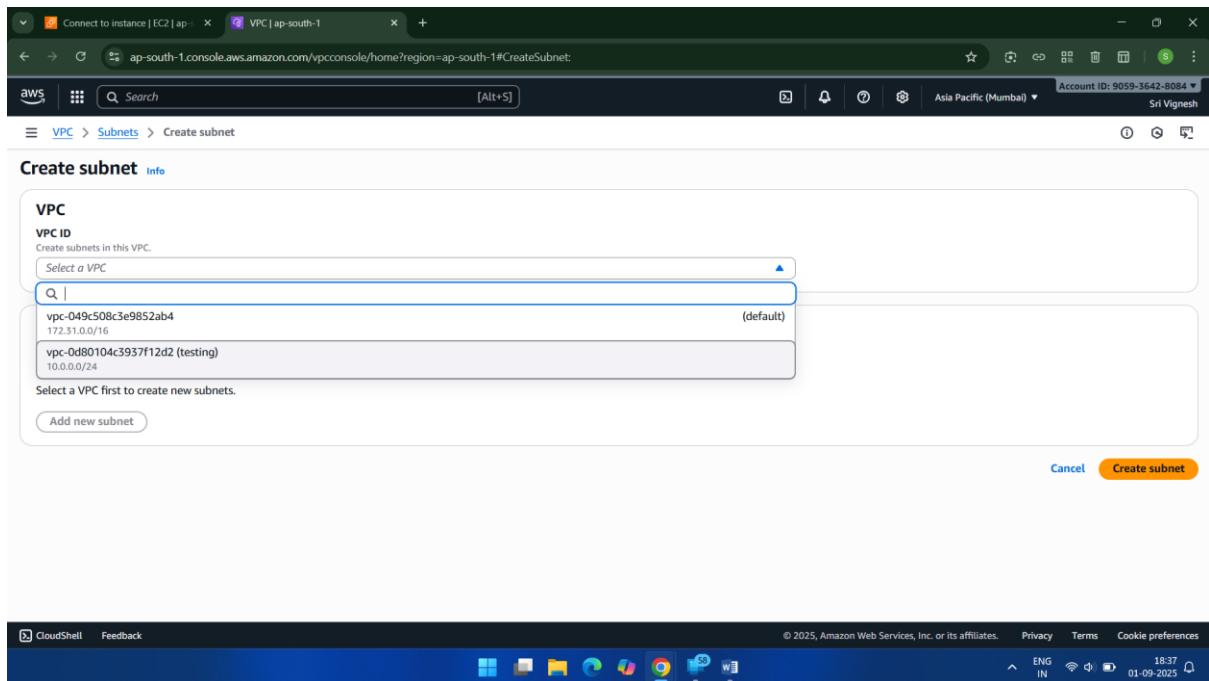
© 2025, Amazon Web Services, Inc. or its affiliates.

CloudShell Feedback ENG IN 18:33 01-09-2025

The screenshot shows the AWS VPC console interface. On the left, there's a navigation sidebar with sections like 'Virtual private cloud' (Subnets, Route tables, Internet gateways, Egress-only Internet gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections), 'Security' (Network ACLs, Security groups), and 'PrivateLink and Lattice'. The main content area displays the details of a VPC named 'vpc-0d80104c3937f12d2 / testing'. A green banner at the top says 'You successfully created vpc-0d80104c3937f12d2 / testing'. The 'Details' section contains various configuration parameters. Below it is a 'Resource map' section with tabs for 'VPC', 'Subnets', 'Route tables', and 'Network Connections'. The 'VPC' tab is selected, showing 'Your AWS virtual network' and a search bar. The 'Route tables' tab shows one route table named 'rtb-04b2b346bf72347e2'. The bottom of the page includes standard AWS footer links and a status bar with the date and time.

Step 2: Create a Subnet

- Subnet -> Create Subnet.
- Select the VPC need to associated with this new subnet.
- Subnet name.
- Select the availability zone – 1 subnet per availability zone.
 - For **public** subnet select – **ap-south-1a**
 - For **private** subnet select – **ap-south-1b**
- IPv4 subnet CIDR block – Manually enter the ip range
 - Here I select the range between **10.0.0.0/28** for public ranges 16 IP's.
 - For private range between **10.0.0.16/28** range gives 16 IP's.
- Click -> Create Subnet.



Creating Public Subnet

The screenshot shows the AWS VPC console interface for creating a new subnet. The top navigation bar includes tabs for 'Connect to instance | EC2 | ap-south-1', 'VPC | ap-south-1', and 'VPC | ap-south-1'. The main title is 'VPC | ap-south-1' with the sub-section 'Create subnet'. The account ID is 9059-3642-8084, and the region is Asia Pacific (Mumbai). The user is Sri Vignesh.

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

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

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

IPv4 VPC CIDR block [Info](#)
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

IPv4 subnet CIDR block
 16 IPs
A dropdown menu with arrows for navigating CIDR blocks.

Tags - optional
Key Value - optional [Remove](#)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 18:50 01-09-2025

Creating Private Subnet

The screenshot shows the AWS VPC console interface for creating a new subnet. The top navigation bar includes tabs for 'Connect to instance | EC2 | ap-south-1', 'VPC | ap-south-1', and 'VPC | ap-south-1'. The main title is 'VPC | ap-south-1' with the sub-section 'Create subnet'. The account ID is 9059-3642-8084, and the region is Asia Pacific (Mumbai). The user is Sri Vignesh.

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

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

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

IPv4 VPC CIDR block [Info](#)
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

IPv4 subnet CIDR block
 16 IPs
A dropdown menu with arrows for navigating CIDR blocks.

Tags - optional
[Remove](#)

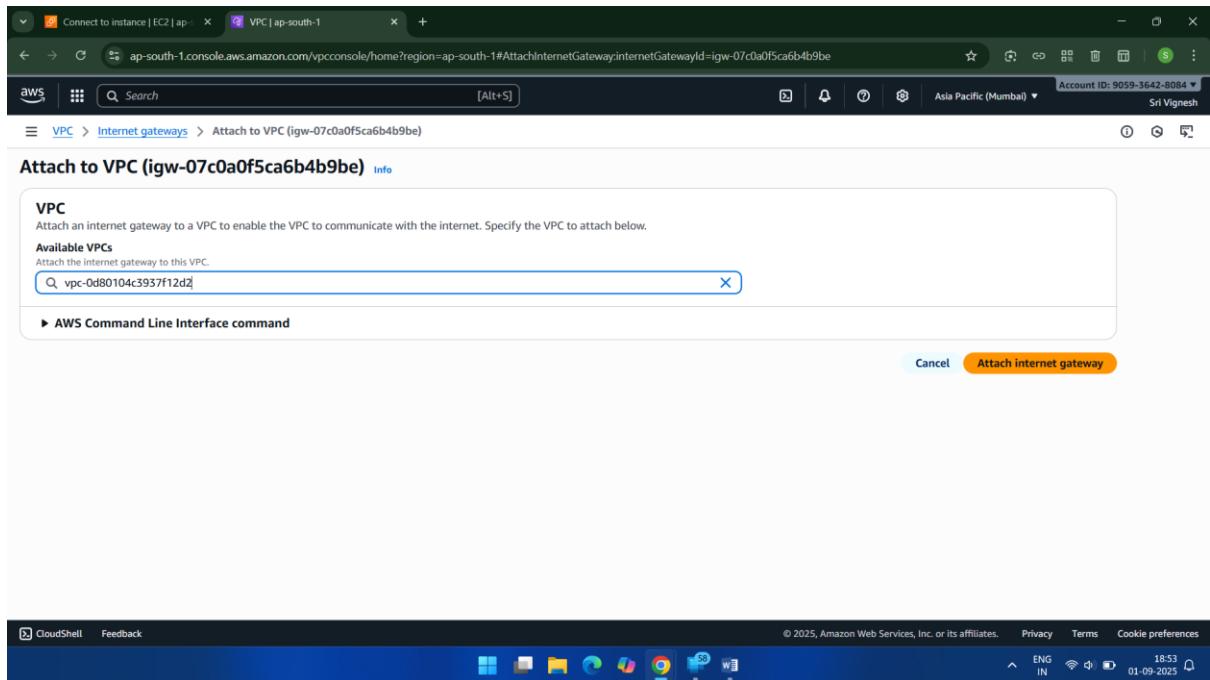
CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 18:51 01-09-2025

Step 3: Creating Internet gateway

- AWS → Create an Internet Gateway
- Name the Internet gateway
- Click → Create Internet gateway
- Attach the Internet Gateway to the VPC
 - Select the Ig → Action → Attach to VPC.
 - Select the VPC
 - Click → Attach to Internet Gateway.

The screenshot shows the 'Create internet gateway' wizard in the AWS VPC console. The 'Internet gateway settings' section has a 'Name tag' input field containing 'test_ig'. The 'Tags - optional' section shows a single tag 'test_ig' with key 'Name' and value 'test_ig'. At the bottom right are 'Cancel' and 'Create internet gateway' buttons.

The screenshot shows the 'Internet gateways (1/2)' page in the AWS VPC console. It lists one gateway named 'test_ig' with ID 'igw-07c0a0f5ca6b4b9be'. The 'Actions' menu for this gateway includes options like 'View details', 'Attach to VPC', 'Detach from VPC', 'Manage tags', and 'Delete internet gateway'. The left sidebar shows the 'Virtual private cloud' section with 'Internet gateways' selected.



Step 4: Creating Route table

- By default there is a Route table and Need to create two route tables each for Public and Private subnet.
- This says **Main** that can be used for **Public subnet**.
- Select the RouteTable states as Main -> Actions -> Edit route.
- Edit Route
 - By default there is a Destination- this will route the traffic within the VPC.
 - Add another route
 - **Destination:** Enter 0.0.0.0 /0
 - **Target:** Select Internet Gateway
- Save Changes -> (After this need to associate to vpc & create another route table to pvt subnet)

Route table ID	Explicit subnet associations	Edge associations	Main	VPC
rtb-04b2b346bf72347e2	-	-	Yes	vpc-0d80104c3937f12d2 testing 905936...
rtb-0d8870ecc5772e338	-	-	Yes	vpc-049c508c3e9852ab4 905936...

Edit routes

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/24	local	Active	No	CreateRouteTable
0.0.0.0/0	Internet Gateway	-	No	CreateRoute

Add route Remove Cancel Preview Save changes

- Select the **Public** subnet.
- Action-> Edit route table association
- Select the correct route table id
- Click Save

subnet-0dd5f1a87faff8ec2 / test_pub

Details

Subnet ID	Subnet ARN	State
subnet-0dd5f1a87faff8ec2	arn:aws:ec2:ap-south-1:905936428084:subnet/subnet-0dd5f1a87faff8ec2	Available
IPv4 CIDR	IPv6 CIDR	Block Pu
10.0.0.0/28	-	Off
Availability Zone	VPC	Create flow log
aps1-az1 (ap-south-1a)	vpc-0d80104c3937f12d2 testing	Edit subnet settings
Network ACL	Auto-assign public IPv4 address	Edit IPv6 CIDRs
acl-00e1d8645717115e9	No	Edit network ACL association
Auto-assign customer-owned IPv4 address	Outpost ID	Block Pu
No	-	Off
IPv6 CIDR reservations	Hostname type	Create flow log
-	IP name	Edit subnet settings
Resource name DNS AAAA record	Owner	Edit IPv6 CIDRs
Disabled	905936428084	Edit network ACL association

Actions ▾

Actions ▾ Create flow log
Edit subnet settings
Edit IPv6 CIDRs
Edit network ACL association
Edit route table association
Edit CIDR reservations
Share subnet
Manage tags
Delete

Flow logs Route table Network ACL CIDR reservations Sharing Tags

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 19:11 01-09-2025

Edit route table association

Route table settings

Route table ID: rtb-04b2b346bf72347e2

Routes (2)

Destination	Target
10.0.0.0/24	local
0.0.0.0/0	igw-07c0a0f5ca6b4b9be

Cancel **Save**

- Create Another Route table For **Private Subnet**.
- Name and Select the **VPC** .
- Click -> Create route table.
- Action -> Edit route
 - **Destination:** Enter `0.0.0.0/0`
 - **Target:** Select **Nat Gateway ✓** - Create NAT gateway
- Click -> Save changes.

Create route table

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.
test-pvt

VPC
The VPC to use for this route table.
vpc-0d80104c3937f12d2 (testing)

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 - <i>optional</i>
Q Name	Q test-pvt

Add new tag
You can add 49 more tags.

Create route table

The screenshot shows the AWS VPC Route Tables console. The URL is ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#EditRoutes:RouteTableId=rtb-0210d970839c2179c. The account ID is 9059-3642-8084. The page title is "Edit routes". There is one route entry:

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/24	local	Active	No	CreateRouteTable

Buttons at the bottom include "Add route", "Cancel", "Preview", and "Save changes". A message says "You have not made any changes."

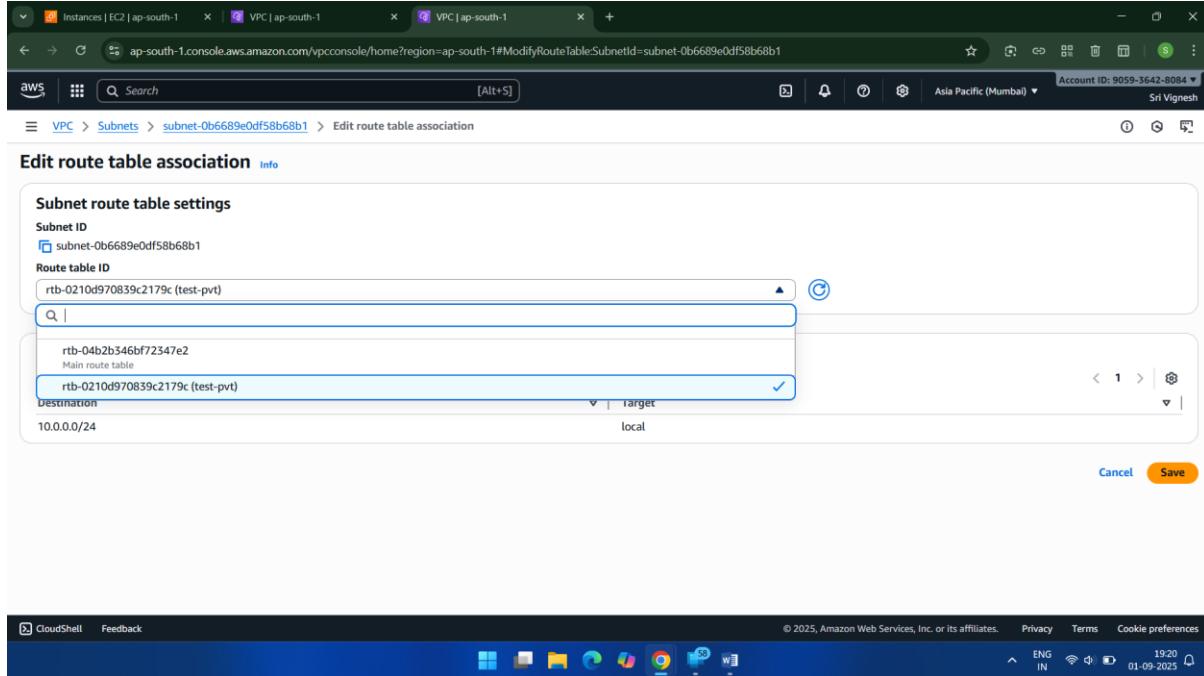
The screenshot shows the AWS VPC Route Tables console. The URL is ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#EditRoutes:RouteTableId=rtb-0210d970839c2179c. The account ID is 9059-3642-8084. The page title is "Edit routes". There are two route entries:

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/24	local	Active	No	CreateRouteTable
0.0.0.0/0	NAT Gateway	Active	No	CreateRoute

A "Remove" button is visible next to the second route entry. Buttons at the bottom include "Add route", "Cancel", "Preview", and "Save changes".



- Select the **Private** subnet.
- Action-> Edit route table association
- Select the correct route table id
- Click Save.



Step 5: Creating NAT(Network Address Translation) gateway

- NAT gateway -> Name.
- Select the Private Subnet.
- [Connectivity type](#)
 - Public NAT
 - Private NAT

Public NAT Gateway

A **public** NAT gateway is used for instances in a private subnet that need to send outbound traffic to the internet while preventing any inbound connections from the internet. It is created in a public subnet and requires an Elastic IP address. When an instance in a private subnet sends a request, the public NAT gateway translates the instance's private IP address to its own Elastic IP address before sending the traffic to the internet gateway.

Private NAT Gateway

A **private** NAT gateway is used to enable communication between instances in a private subnet and other VPCs or your on-premises network. It is created in a private subnet and does **not** require an Elastic IP address. This type of NAT gateway can be used to route traffic through a transit gateway or a virtual private gateway, but traffic routed to an internet gateway will be dropped.

- Elastic IP allocation ID
 - Select -> Allocate Elastic IP.
 - This will give you an elastic IP.
- Save and add this in that **Private Route table** Destinations.

Create NAT gateway Info

A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.

NAT gateway settings

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

The name can be up to 256 characters long.

Subnet
Select a subnet in which to create the NAT gateway.

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.

► Additional settings Info

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 2018 01-09-2025

Edit routes

Destination	Target	Status	Propagated	Route Origin
10.0.0.0/24	local	Active	No	CreateRouteTable
Q_ 0.0.0.0	Q_local	X	-	-
Q_ 0.0.0.0	NAT Gateway	-	No	CreateRoute
Q_ nat-0a8700d1ebc3d931a	Q_nat-0a8700d1ebc3d931a	X	-	-

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG IN 2019 01-09-2025



Step 6: Creating Security Group

- Create Security group for **Public Subnet**.
- Name -> add Description
- Select the VPC
- Inbound rules
 - Add type: SSH -> source: Anywhere 0.0.0.0/0
 - Add type: HTTP -> source: Anywhere 0.0.0.0/0
 - Add type: HTTPS -> source: Anywhere 0.0.0.0/0
- Outbound rules
 - Add type: All Traffic-> source: Anywhere 0.0.0.0/0
- Click -> Create security group.

The screenshot shows the AWS VPC Security Groups creation interface. It includes sections for Inbound rules (SSH, HTTP, HTTPS) and Outbound rules (All traffic). A note about allowing all IP addresses is present. There are also sections for Tags and optional tags, and a final 'Create security group' button.

Inbound rules:

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	Anywhere	0.0.0.0/0
HTTP	TCP	80	Anywhere	0.0.0.0/0
HTTPS	TCP	443	Anywhere	0.0.0.0/0

Outbound rules:

Type	Protocol	Port range	Destination	Description - optional
All traffic	All	All	Custom	0.0.0.0/0

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.

Create security group

- Create Security group for **Private Subnet**.
- Name -> add Description
- Select the VPC
- Inbound rules
 - Add type: All Traffic -> source: Custom -> **Public Security group**.
 - This will only allow traffic from Public subnet.
- Outbound rules
 - Add type: All Traffic-> source: Anywhere 0.0.0.0/0
- Click -> Create security group.

The screenshot shows the 'Create security group' page in the AWS VPC console. In the 'Basic details' section, the security group name is 'test-pvt'. The description is 'Security group for private subnet instances'. The VPC is set to 'vpc-0d80104c3937f12d2 (testing)'. The 'Inbound rules' section indicates 'This security group has no inbound rules.' An 'Add rule' button is available. The 'Outbound rules' section is also visible.

The screenshot shows the 'Create security group' page with rules added. In the 'Inbound rules' section, a rule is defined with 'Type' as 'All traffic', 'Protocol' as 'All', 'Port range' as 'All', 'Source' as 'Custom' (with 'sg-0bda351d21b0adad' selected), and 'Description' as 'sg-0bda351d21b0adadf'. The 'Outbound rules' section shows a rule with 'Type' as 'All traffic', 'Protocol' as 'All', 'Port range' as 'All', 'Destination' as 'Anywh...', and 'Description' as '0.0.0.0/0'. A warning message at the bottom states: '⚠️ Rules with destination of 0.0.0.0/0 or ::/0 allow your instances to send traffic to any IPv4 or IPv6 address. We recommend setting security group rules to be more restrictive and to only allow traffic to specific known IP addresses.'

Step 7: Creating Security Group

- Create 2 Instances with same OS and Key pair. One for **Public and Private subnet**.
- **Network Settings**
 - Click Edit
 - Select the VPC
 - Select the **Public subnet 1a**.
 - Auto assign IP: **Enable** (For public subnet)
 - Firewall: Select existing Security Group.
 - Select -> Public Security group.
 - Click -> Launch Instance.

Name and tags

Name: t1-pub

Application and OS Images (Amazon Machine Image)

Search: Search our full catalog including 1000s of application and OS images

Recent **Quick Start**

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type

ami-02d26659fd82cf299 (64-bit (x86)) / ami-0b9093ea00afed92 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd64...read more

Virtual server type (instance type): t2.micro

Firewall (security group): test-pub

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where applicable).

Launch instance **Preview code**

Description

Ubuntu Server 24.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture: 64-bit (x86)

AMI ID: ami-02d26659fd82cf299

Publish Date: 2025-08-21

Username: ubuntu

Verified provider

Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Windows base pricing: 0.017 USD per Hour On-Demand RHEL base pricing: 0.0268 USD per Hour

On-Demand Linux base pricing: 0.0124 USD per Hour

On-Demand Ubuntu Pro base pricing: 0.0142 USD per Hour On-Demand SUSE base pricing: 0.0124 USD per Hour

Additional costs apply for AMIs with pre-installed software

Key pair (login)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 24.04, amd64...read more

Virtual server type (instance type): t2.micro

Firewall (security group): test-pub

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where applicable).

Launch instance **Preview code**

The screenshot shows the 'Launch an instance' wizard in the AWS Management Console. The current step is 'Network settings'. The 'Subnet' dropdown is set to 'test_pub'. Under 'Auto-assign public IP', 'Enable' is selected. In the 'Firewall (security groups)' section, 'Select existing security group' is chosen. The 'Common security groups' dropdown shows 'test-pub sg-0bda351d21b0adad'. The right panel displays the 'Summary' of the instance configuration, including the AMI (Canonical, Ubuntu, 24.04), instance type (t2.micro), and storage (1 volume(s) - 8 GiB). A 'Free tier' notification is present. At the bottom right are 'Cancel', 'Launch instance', and 'Preview code' buttons.

This screenshot continues the 'Launch an instance' wizard. The 'Configure storage' section shows a single volume configuration: 1x 8 GiB gp3. An info box states that free-tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. The 'Advanced' tab is visible above the storage section. The right panel's 'Summary' shows the same instance details as the previous step. A 'Free tier' notification is also present. The bottom right features 'Cancel', 'Launch instance', and 'Preview code' buttons.

- Creation of instance for **Private subnet**.
- **Network Settings**
 - Click Edit
 - Select the VPC
 - Select the **Private subnet 1b**.
 - Auto assign IP: **Disable** (For Private subnet)
 - Firewall: Select existing Security Group.
 - Select -> **Private Security group**.
 - Click -> Launch Instance.

The screenshot shows the AWS EC2 'Launch an instance' page. In the 'Name and tags' section, the name 't1-pvt' is entered. Under 'Application and OS Images (Amazon Machine Image)', the Canonical, Ubuntu, 24.04 AMI is selected. The 'Virtual server type (instance type)' is set to t2.micro. A 'Free tier' notification is visible, stating that in the first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where applicable). The 'Launch instance' button is highlighted.

The screenshot shows the 'Network settings' section of the AWS EC2 'Launch an instance' page. It includes fields for 'VPC - required' (set to 'vpc-0d80104c3937f12d2 (testing)'), 'Subnet' (set to 'subnet-0b6689e0df5fb6b1'), 'Auto-assign public IP' (disabled), and 'Firewall (security groups)' (set to 'Select existing security group'). A security group named 'test-pvt' is selected. The 'Common security groups' section lists 'test-pvt sg-0b57bad03b0c8810c'. A note at the bottom states: 'Security groups that you add or remove here will be added to or removed from all your network interfaces.' A 'Free tier' notification is also present.

Step 8: Testing

- Step 1: Copying pem file into Public Instance
 - From Local Copy ".Pem" File to the EC2 Instance

For Amazon Linux use the below code in Local:

```
scp -i C:/Users/vicky/Downloads/ss.pem C:/Users/vicky/Downloads/ss.pem ec2-user@<public-ec2-public-ip>:/home/ec2-user/
```

For Ubuntu use the below code in Local:

```
scp -i C:/Users/vicky/Downloads/ss.pem C:/Users/vicky/Downloads/ss.pem  
ubuntu@3.108.62.142:/home/ubuntu/
```

The above command will Store the ".pem" file into the public EC2 instance in location : "/home/ubuntu/" from "C:\Users\vicky\Downloads\ss.pem".

The screenshot shows a Windows terminal window with the following command history:

```
ubuntu@ip-10-0-0-10: ~ + | ~
Warning: Identity file C:/Users/vicky/Downloads/ss.pem not accessible: No such file or directory.
ec2-user@3.108.62.142: Permission denied (publickey).
scp: Connection closed

C:\Users\vicky>scp -i C:/Users/vicky/Downloads/ss.pem C:/Users/vicky/Downloads/ss.pem ec2-user@3.108.62.142:/home/ec2-user/
Warning: Identity file C:/Users/vicky/Downloads/ss.pem not accessible: No such file or directory.
ec2-user@3.108.62.142: Permission denied (publickey).
scp: Connection closed

C:\Users\vicky>scp -i C:/Users/vicky/Downloads/ss.pem C:/Users/vicky/Downloads/ss.pem ubuntu@3.108.62.142:/home/ubuntu/
100% 1678 26.0KB/s 00:00

C:\Users\vicky>ssh -i "C:/Users/vicky/Downloads/ss.pem" ubuntu@3.108.62.142
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1011-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support:      https://ubuntu.com/pro

System information as of Tue Sep 2 14:00:21 UTC 2025

System load: 0.0 Processes: 109
Usage of /: 28.2% of 6.71GB Users logged in: 0
Memory usage: 22% IPv4 address for enx0: 10.0.0.10
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Mon Sep 1 16:00:01 2025 from 27.61.50.77
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

- Step 2: SSH into the public instance
 - Command : `cd /home/ubuntu`
 - You can find the ".pem" file.
 - Command : `sudo chmod 400 <name>.pem`
 - For security purpose its mandatory to change the permissions
 - Command : `ssh -i <name>.pem ubuntu@3.108.62.142`
 - You can able to login to the private subnet. By which we can confirm by this process and login to the private instance.
 - Command : `curl ifconfig.me`
 - Using this command in public instance this'll return public ip of the instance.
 - While using this on private instance this'll return public ip of NAT gateway.

```
ubuntu@ip-10-0-0-10:~ + - x
C:\Users\vicky>scp -i C:/Users/vicky/Downloads/ss.pem C:/Users/vicky/Downloads/ss.pem ubuntu@3.108.62.142:/home/ubuntu/
ss.pem                                                 100% 1678     26.0KB/s   00:00

C:\Users\vicky>ssh -i "C:/Users/vicky/Downloads/ss.pem" ubuntu@3.108.62.142
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1011-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Tue Sep 2 14:00:21 UTC 2025

System load: 0.0          Processes:           109
Usage of /: 28.2% of 6.71GB Users logged in: 0
Memory usage: 22%          IPv4 address for enX0: 10.0.0.10
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

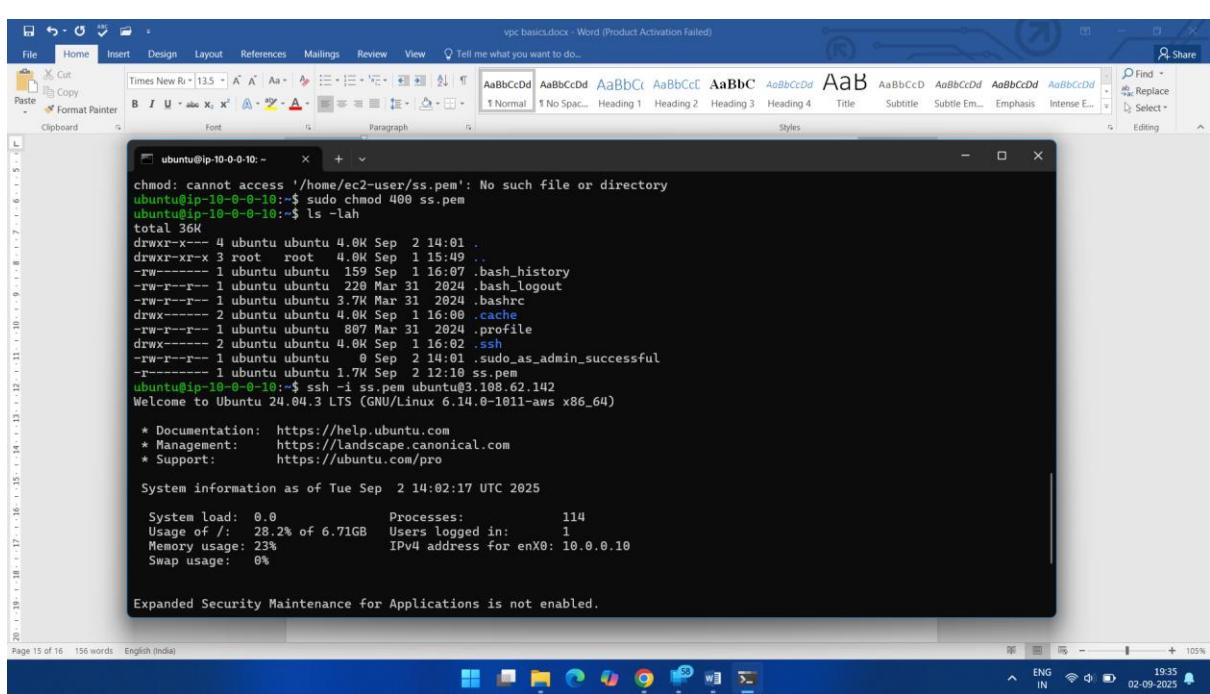
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Mon Sep 1 16:00:01 2025 from 27.61.50.77
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-0-10:~$ ls
ss.pem
ubuntu@ip-10-0-0-10:~$ ssh -i ss.pem ubuntu@3.108.62.142
The authenticity of host '3.108.62.142 (3.108.62.142)' can't be established.
ED25519 key fingerprint is SHA256:67Cu9Egsrb3hDPbS8HfKevdDHEOuLhObByYN03vldik.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.108.62.142' (ED25519) to the list of known hosts.

ubuntu@ip-10-0-0-10:~$
```



vpc_basics.docx - Word (Product Activation Failed)

File Home Insert Design Layout References Mailings Review View Tell me what you want to do...

Font Paragraph Styles

Clipboard

ubuntu@ip-10-0-0-10:~ + - x
chmod: cannot access '/home/ec2-user/ss.pem': No such file or directory
ubuntu@ip-10-0-0-10:~\$ sudo chmod 400 ss.pem
ubuntu@ip-10-0-0-10:~\$ ls -lah
total 36K
drwxr-x--- 4 ubuntu ubuntu 4.0K Sep 2 14:01 .
drwxr-xr-x 3 root root 4.0K Sep 1 15:49 ..
-rw----- 1 ubuntu ubuntu 159 Sep 1 16:07 .bash_history
-rw-r--r-- 1 ubuntu ubuntu 220 Mar 31 2024 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3.7K Mar 31 2024 .bashrc
drwxr----- 2 ubuntu ubuntu 4.0K Sep 1 16:00 .cache
-rw-r--r-- 1 ubuntu ubuntu 807 Mar 31 2024 .profile
drwxr----- 2 ubuntu ubuntu 4.0K Sep 1 16:02 .ssh
-rw-r--r-- 1 ubuntu ubuntu 0 Sep 2 14:01 .sudo_as_admin_successful
-rw----- 1 ubuntu ubuntu 1.7K Sep 2 12:10 ss.pem
ubuntu@ip-10-0-0-10:~\$ ssh -i ss.pem ubuntu@3.108.62.142
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1011-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Tue Sep 2 14:02:17 UTC 2025

System load: 0.0 Processes: 114
Usage of /: 28.2% of 6.71GB Users logged in: 1
Memory usage: 23% IPv4 address for enX0: 10.0.0.10
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

Page 15 of 16 156 words English (India)

ENG IN 19:35 02-09-2025

```

ubuntu@ip-10-0-0-10: ~ + 
-rw-r--r-- 1 ubuntu ubuntu 0 Sep 2 14:01 .sudo_as_admin_successful
-r----- 1 ubuntu ubuntu 1.7K Sep 2 12:10 ss.pem
ubuntu@ip-10-0-0-10:~$ ssh -i ss.pem ubuntu@3.108.62.142
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1011-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Tue Sep 2 14:02:17 UTC 2025

System load: 0.0 Processes: 114
Usage of /: 28.2% of 6.71GB Users logged in: 1
Memory usage: 23% IPv4 address for enX0: 10.0.0.10
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Tue Sep 2 14:00:21 2025 from 27.61.57.212
ubuntu@ip-10-0-0-10:~$ |

```

The terminal window shows system information, including memory usage, swap usage, and network details. It also indicates that no updates are available immediately.

Step 9: Testing

- Final Flow will be like

