

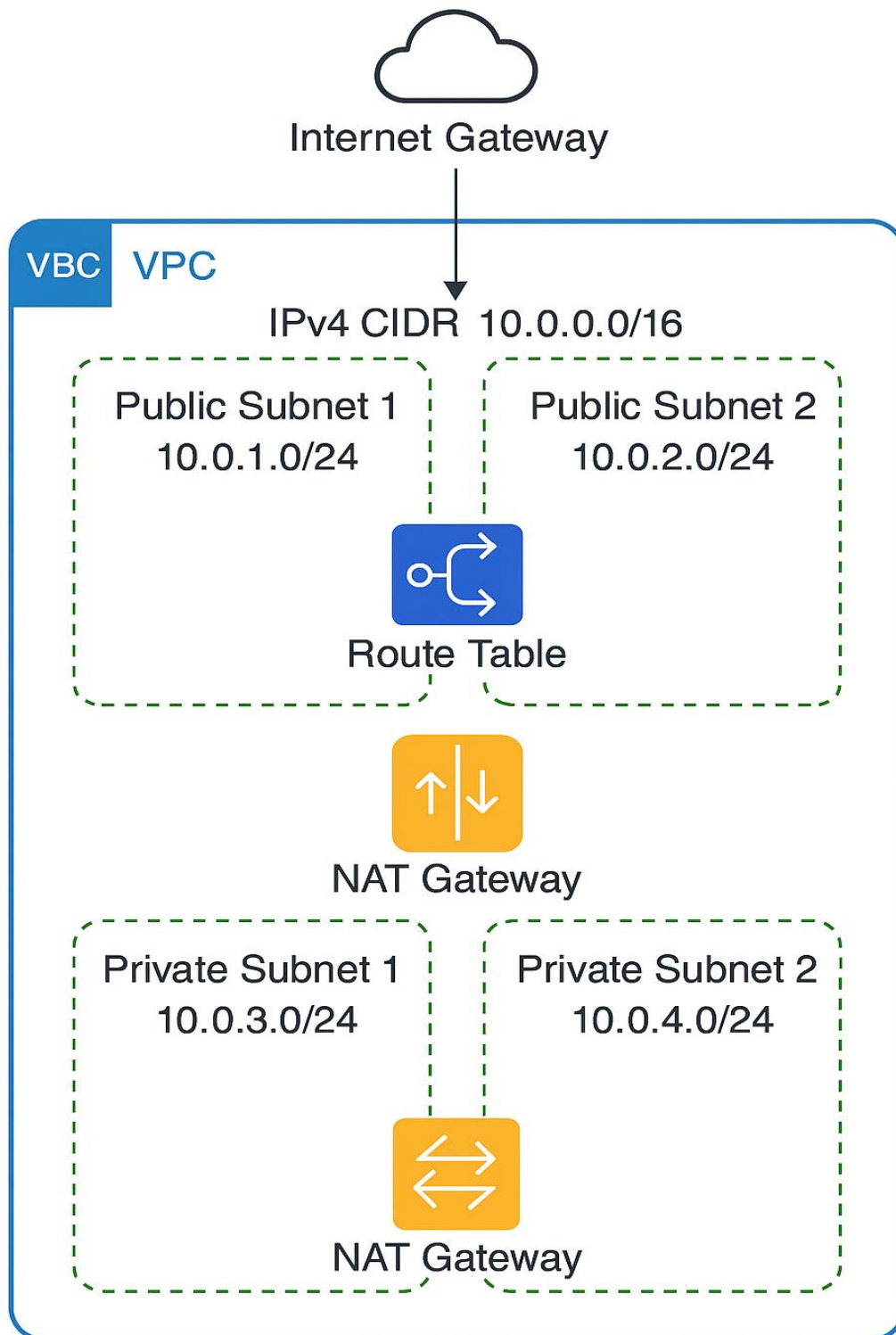


VIRTUAL PRIVATE CLOUD

For DevOps Engineers and Cloud Engineers



VPC architecture diagram



Creating a VPC in AWS

This guide helps you manually create a custom VPC in AWS with public and private subnets using the AWS Console.

Step 1: Navigate to VPC Dashboard

- Sign in to AWS Console.
- Go to VPC service → click "Create VPC".

Step 2: Create the VPC

- Name: my-custom-vpc
- IPv4 CIDR block: 10.0.0.0/16
- IPv6 block: None or auto-assigned
- Tenancy: Default
- Click Create VPC.

Step 3: Create Subnets

Create 2 public and 2 private subnets in different AZs.

Public Subnets:

1. Name: public-subnet-1 | AZ: us-east-1a | CIDR: 10.0.1.0/24
2. Name: public-subnet-2 | AZ: us-east-1b | CIDR: 10.0.2.0/24

Private Subnets:

1. Name: private-subnet-1 | AZ: us-east-1a | CIDR: 10.0.3.0/24
2. Name: private-subnet-2 | AZ: us-east-1b | CIDR: 10.0.4.0/24

Go to Subnets → Create subnet, choose your VPC and add them one by one.

Step 4: Create and Attach Internet Gateway

- Go to Internet Gateways → Create internet gateway
- Name: my-IGW
- Click Attach to VPC and choose my-custom-VPC

Step 5: Route Table for Public Subnets

- Go to Route Tables → Create route table
- Name: public-rt, select your VPC
- Edit routes: Add 0.0.0.0/0 → Target: Internet Gateway (my-IGW)
- Edit subnet associations: Attach public-subnet-1 and public-subnet-2

Step 6: NAT Gateway (for Private Subnets)

- Allocate a new Elastic IP
- Go to NAT Gateways → Create NAT Gateway

- Subnet: public-subnet-1
- Elastic IP: select the one you created
- Name: NAT-gateway

Step 7: Route Table for Private Subnets

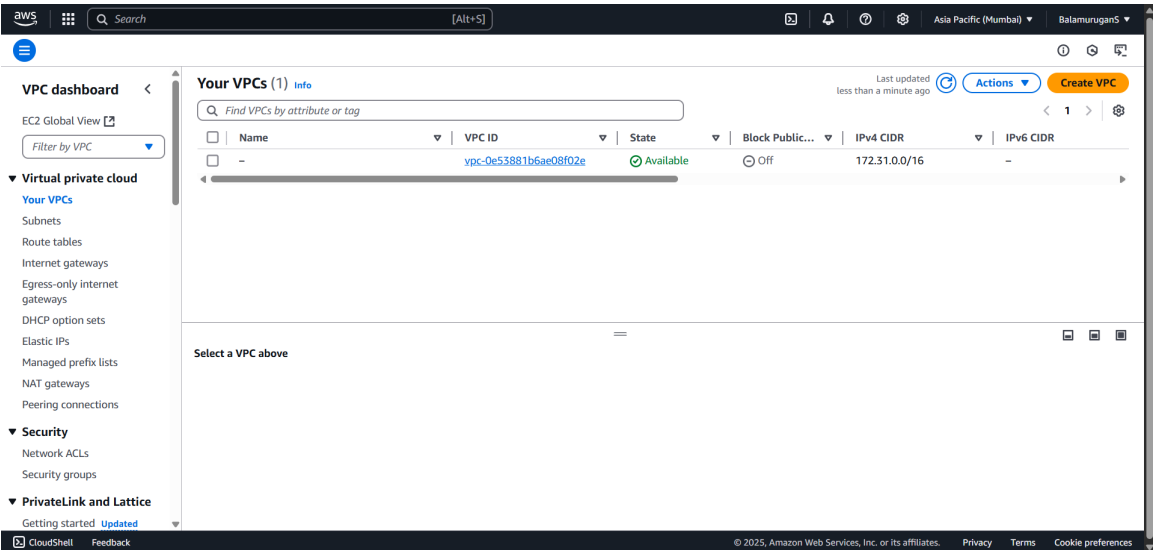
- Create new Route Table
- Name: private-rt, select your VPC
- Add route: 0.0.0.0/0 → Target: NAT Gateway
- Subnet association: Attach private-subnet-1 and private-subnet-2

Summary

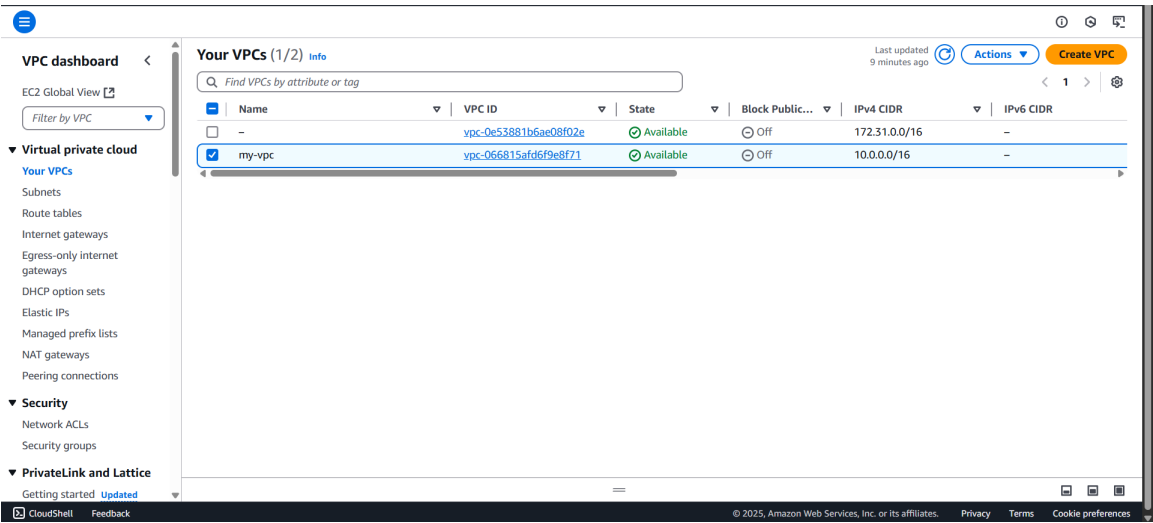
VPC setup includes:

- Custom VPC: 10.0.0.0/16
- 2 Public Subnets + IGW + Route Table
- 2 Private Subnets + NAT Gateway + Route Table

Default VPC



Creating Private New VPC

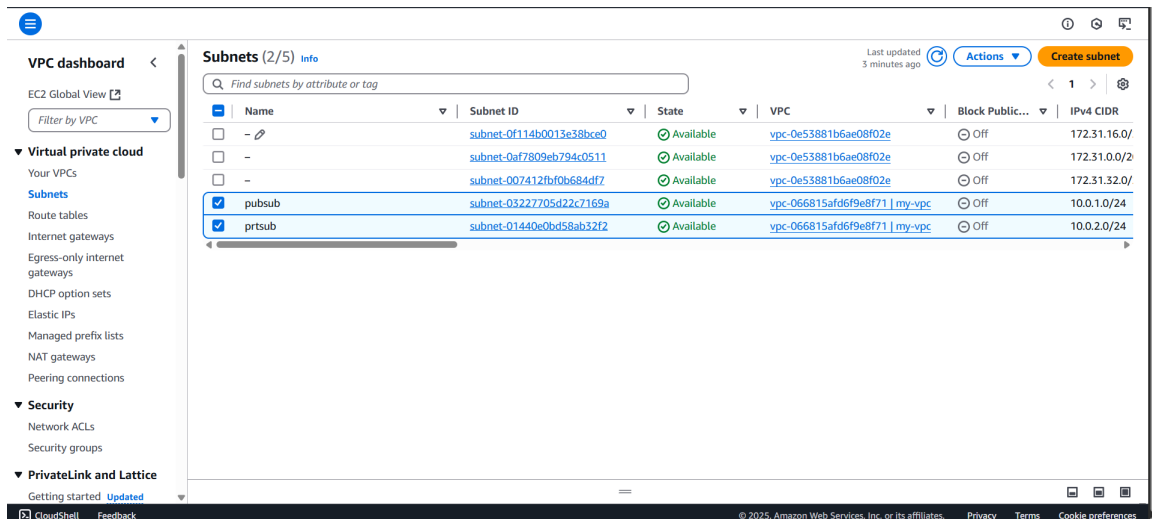


And we will create **SUBNETS**

- Private subnets
- Public subnets

-go to subnets

We can create public subnet and private subnet individually



The screenshot displays the AWS VPC console's 'Subnets' page. The left-hand navigation pane is open, showing the 'Virtual private cloud' section with 'Subnets' selected. The main content area, titled 'Subnets (2/5)', contains a table listing existing subnets. The table has columns for Name, Subnet ID, State, VPC, Block Public Access, and IPv4 CIDR. Two subnets are listed: 'pubsub' and 'prtsub', both in an 'Available' state. The 'pubsub' subnet is associated with 'my-vpc' and has a CIDR of 10.0.1.0/24. The 'prtsub' subnet is also associated with 'my-vpc' and has a CIDR of 10.0.2.0/24. At the top right of the console, there are buttons for 'Actions' and 'Create subnet'.

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
-	subnet-0f114b0013e38bce0	Available	vpc-0e53881b6ae08f02e	Off	172.31.16.0/
-	subnet-0af7809eb794c0511	Available	vpc-0e53881b6ae08f02e	Off	172.31.0.0/2
-	subnet-007412fbf0b684df7	Available	vpc-0e53881b6ae08f02e	Off	172.31.32.0/
pubsub	subnet-03227705d22c7169a	Available	vpc-066815afd6f9e8f71 my-vpc	Off	10.0.1.0/24
prtsub	subnet-01440e0bd58ab32f2	Available	vpc-066815afd6f9e8f71 my-vpc	Off	10.0.2.0/24

We will create **INTERNET GATEWAY**

The screenshot shows the AWS Management Console interface for an Internet Gateway. The breadcrumb navigation at the top reads: VPC > Internet gateways > igw-0f85c7f30c7c0d23d. The left-hand navigation pane is expanded to 'Virtual private cloud', with 'Internet gateways' selected. The main content area is titled 'igw-0f85c7f30c7c0d23d / my-igw'. It features a 'Details' section with the following information:

Internet gateway ID	State	VPC ID	Owner
igw-0f85c7f30c7c0d23d	Detached	-	024848447245

Below the details is a 'Tags' section with a search bar and a table:

Key	Value
Name	my-igw

The 'State' is 'Detached', indicating the gateway is not yet attached to a VPC.

And -go to action, select our VPC after to get attached our VPC

This screenshot shows the same AWS Management Console page, but the Internet Gateway is now in an 'Attached' state. The breadcrumb navigation remains: VPC > Internet gateways > igw-0f85c7f30c7c0d23d. The left navigation pane is still expanded to 'Virtual private cloud' > 'Internet gateways'. The main content area shows the 'igw-0f85c7f30c7c0d23d / my-igw' details. The 'State' is now 'Attached' (indicated by a green checkmark icon). The 'VPC ID' is now 'vpc-066815afd6f9e8f71 | my-vpc'. The 'Owner' remains '024848447245'. The 'Tags' section is identical to the previous screenshot.

We create **ROUNDTABLES**

We will create round table for public and private and select our VPC while we creating
PUBLIC

The screenshot shows the AWS Management Console interface for a VPC. The left sidebar contains navigation links for VPC dashboard, EC2 Global View, and various VPC resources. The main content area displays the details for the route table **rtb-0d74b344e5e08776a / pubrt**. A green notification bar at the top indicates that the route table was created successfully. The details section shows the route table ID, VPC ID, and owner ID. The routes section shows a single route for destination 10.0.0/16 with a local target and an active status.

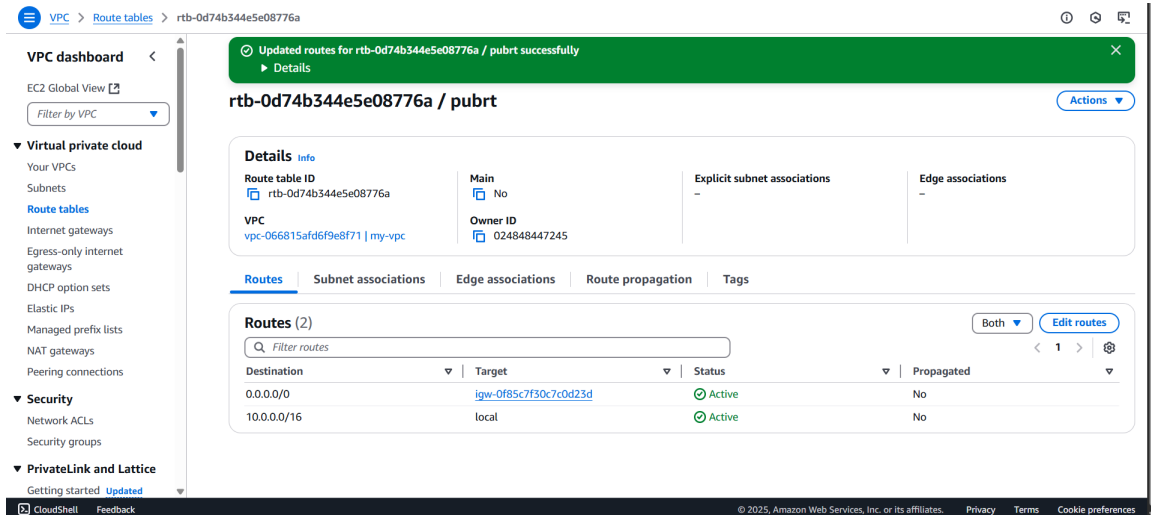
Destination	Target	Status	Propagated
10.0.0/16	local	Active	No

PRIVATE

The screenshot shows the AWS Management Console interface for a VPC. The left sidebar contains navigation links for VPC dashboard, EC2 Global View, and various VPC resources. The main content area displays the details for the route table **rtb-08d29b8491b9afb4 / pvtrt**. A green notification bar at the top indicates that the route table was created successfully. The details section shows the route table ID, VPC ID, and owner ID. The routes section shows a single route for destination 10.0.0/16 with a local target and an active status.

Destination	Target	Status	Propagated
10.0.0/16	local	Active	No

Click the public route tables – go to routes – edit routes – add routes – and select the Internet gateway – select the Internet gateway ID – and set the public IP 0.0.0.0/0



VPC dashboard < VPC > Route tables > rtb-0d74b344e5e08776a

Updated routes for rtb-0d74b344e5e08776a / pubrt successfully
Details

rtb-0d74b344e5e08776a / pubrt

Actions

Details Info

Route table ID
rtb-0d74b344e5e08776a

VPC
vpc-066815afd6f9e8771 | my-vpc

Main
No

Owner ID
024848447245

Explicit subnet associations
-

Edge associations
-

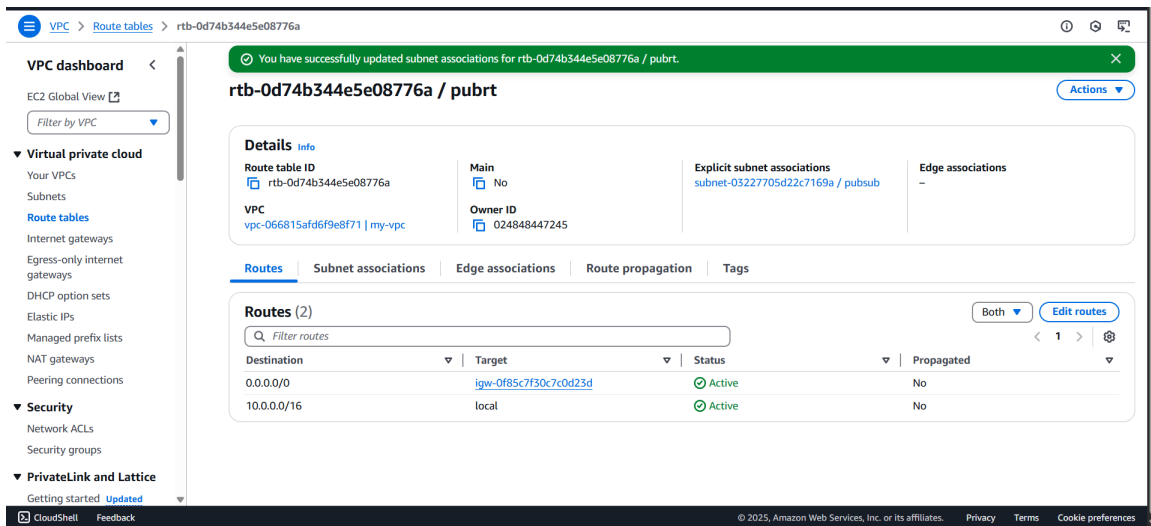
Routes Subnet associations Edge associations Route propagation Tags

Routes (2)

Filter routes

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0f85c7f30c7c0d23d	Active	No
10.0.0.0/16	local	Active	No

And – go to subnet association – edit subnet association – select public subnets



VPC dashboard < VPC > Route tables > rtb-0d74b344e5e08776a

You have successfully updated subnet associations for rtb-0d74b344e5e08776a / pubrt.

rtb-0d74b344e5e08776a / pubrt

Actions

Details Info

Route table ID
rtb-0d74b344e5e08776a

VPC
vpc-066815afd6f9e8771 | my-vpc

Main
No

Owner ID
024848447245

Explicit subnet associations
subnet-03227705d22c7169a / pubsub

Edge associations
-

Routes Subnet associations Edge associations Route propagation Tags

Routes (2)

Filter routes

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0f85c7f30c7c0d23d	Active	No
10.0.0.0/16	local	Active	No

We go to Private round tables - go to subnet association – add private subnets

VPC dashboard <

EC2 Global View [↗](#)

Filter by VPC ▾

▼ **Virtual private cloud**

- Your VPCs
- 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

▼ **PrivateLink and Lattice**

Getting started [Updated](#)

[CloudShell](#) [Feedback](#)

Route tables (1/4) [Info](#)

Last updated 2 minutes ago [Actions](#) [Create route table](#)

Find route tables by attribute or tag

<input type="checkbox"/>	Name	Route table ID	Explicit subnet associ...	Edge associations	Main
<input type="checkbox"/>	-	rtb-0bcb8f45213bbffe0	-	-	Yes
<input type="checkbox"/>	-	rtb-0a820f412f1d424b9	-	-	Yes
<input type="checkbox"/>	pubrt	rtb-0d74b344e5e08776a	subnet-03227705d22c71...	-	No
<input checked="" type="checkbox"/>	pvtrt	rtb-08d29b8491b9afb4	subnet-01440e0bd58ab3...	-	No

rtb-08d29b8491b9afb4 / pvtrt

[Details](#) [Routes](#) [Subnet associations](#) [Edge associations](#) [Route propagation](#) [Tags](#)

Explicit subnet associations (1) [Edit subnet associations](#)

Find subnet association

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
prtsub	subnet-01440e0bd58ab32f2	10.0.2.0/24	-

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

We will create **SECURITY GROUPS**

We create security groups for Public and private

Customize your inbound rules for public SG ex; HTTP, HTTPS, RDP, SSH

The screenshot shows the AWS Management Console interface for a VPC. The left sidebar contains navigation links for VPC dashboard, EC2 Global View, Virtual private cloud, Security, and PrivateLink and Lattice. The main content area displays the details of a security group named 'pubsg' with ID 'sg-07004c61265282e71'. A green notification banner at the top states 'Security group (sg-07004c61265282e71 | pubsg) was created successfully'. Below the details, the 'Inbound rules' tab is selected, showing a table with four rules:

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-0876a57632faa09bb	IPv4	RDP	TCP	3389
-	sgr-05d458a6cfb7bc304	IPv4	HTTP	TCP	80
-	sgr-0c31a64749cc49759	IPv4	HTTPS	TCP	443
-	sgr-03026498d36903ac9	IPv4	SSH	TCP	22

And we create private SG

Go to – inbound rules – click the source – select the PUBLIC SG – select the ALL TCP and click create SG

The screenshot shows the AWS Management Console interface for a VPC. The left sidebar contains navigation links for VPC dashboard, EC2 Global View, Virtual private cloud, Security, and PrivateLink and Lattice. The main content area displays the details of a security group named 'pvtsg' with ID 'sg-041d0363238f84a7c'. A green notification banner at the top states 'Security group (sg-041d0363238f84a7c | pvtsg) was created successfully'. Below the details, the 'Inbound rules' tab is selected, showing a table with one rule:

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-04b62661d2fa789d2	-	All TCP	TCP	0 - 65535

We will create **EC2 INSTANCES**

Create ec2 for **public and private**

Inside of public instance

Select your key pair – edit networking – select your VPC – select your public subnet – enable auto assign IP – edit your fire wall – select your public SG – Launch instance

PUBLIC INSTANCE

The screenshot displays the AWS Management Console interface for EC2 instances. On the left, a navigation menu includes 'EC2', 'Dashboard', 'EC2 Global View', 'Events', 'Instances' (expanded), 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Capacity Reservations', 'Images' (expanded), 'AMIs', 'AMI Catalog', 'Elastic Block Store' (expanded), 'Volumes', 'Snapshots', and 'Lifecycle Manager'. The main content area shows a table of instances with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 address. Two instances are listed: 'Application' (ID: i-0c71426c6c55baf4b, state: Running, type: t2.micro) and 'public' (ID: i-078a98461e8e10f31, state: Running, type: t3.micro). Below the table, the details for instance 'i-0c71426c6c55baf4b (docker)' are shown, including tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. The 'Instance summary' section displays the Instance ID, IPv6 address, Public IPv4 address (3.108.254.0), Private IPv4 addresses (172.31.15.76), and Public IPv4 DNS (ec2-3-108-254-0.ap-south-1.compute.amazonaws.com).

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 address
Application	i-0c71426c6c55baf4b	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-3-108-254-0.ap-south-1.compute.amazonaws.com
public	i-078a98461e8e10f31	Running	t3.micro	Initializing	View alarms +	ap-south-1b	-

i-0c71426c6c55baf4b (docker)

Instance summary

Instance ID: i-0c71426c6c55baf4b

IPv6 address: -

Public IPv4 address: 3.108.254.0 | [open address](#)

Private IPv4 addresses: 172.31.15.76

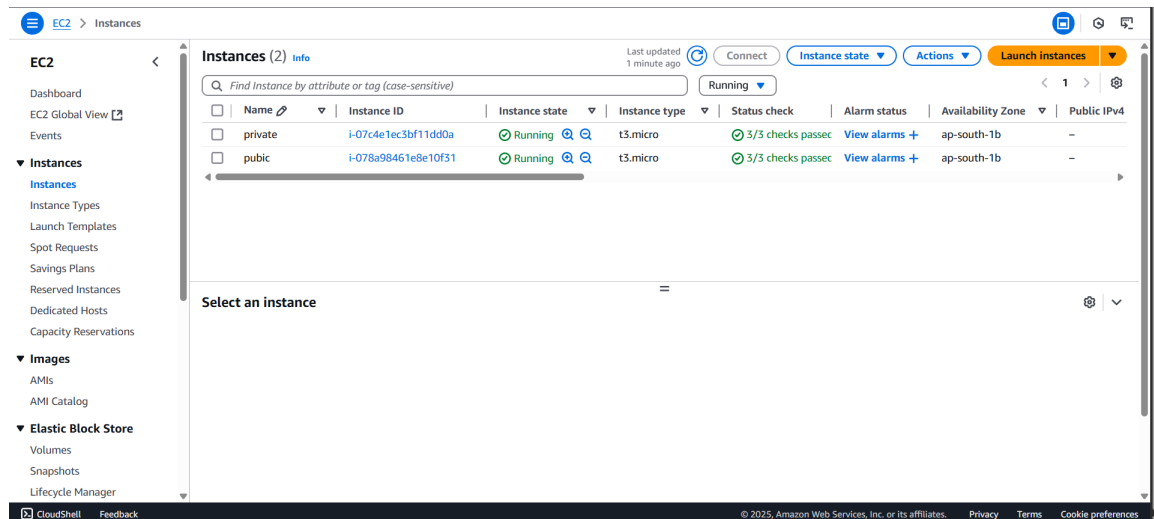
Instance state: Running

Public IPv4 DNS: ec2-3-108-254-0.ap-south-1.compute.amazonaws.com

Create private instance – as same as creating public instance

But select your private subnets – disable auto assign IP – select your private SG – Launch instance

PRIVATE INSTANCE

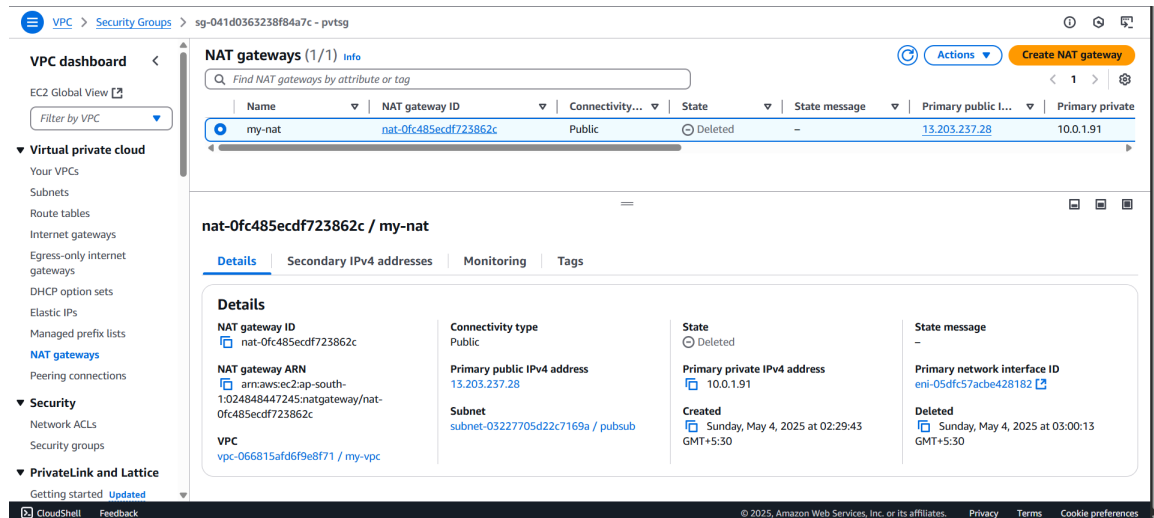


The screenshot displays the AWS Management Console for the EC2 service. The left-hand navigation pane shows the 'EC2' section expanded, with 'Instances' selected. The main content area is titled 'Instances (2)' and shows a table of two running instances. The first instance is named 'private' with ID 'i-07c4e1ec3bf11dd0a', and the second is named 'public' with ID 'i-078a98461e8e10f31'. Both are t3.micro instances in the 'ap-south-1b' availability zone. Below the table, there is a section titled 'Select an instance'.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
private	i-07c4e1ec3bf11dd0a	Running	t3.micro	3/3 checks passed	View alarms +	ap-south-1b	-
public	i-078a98461e8e10f31	Running	t3.micro	3/3 checks passed	View alarms +	ap-south-1b	-

Creating NAT GATEWAY

Go to create – name your NAT gateway – select your public subnets – click you Elastic allocate IP – create Nat gateway



Connect you PUBLIC INSTACE

Through SSH

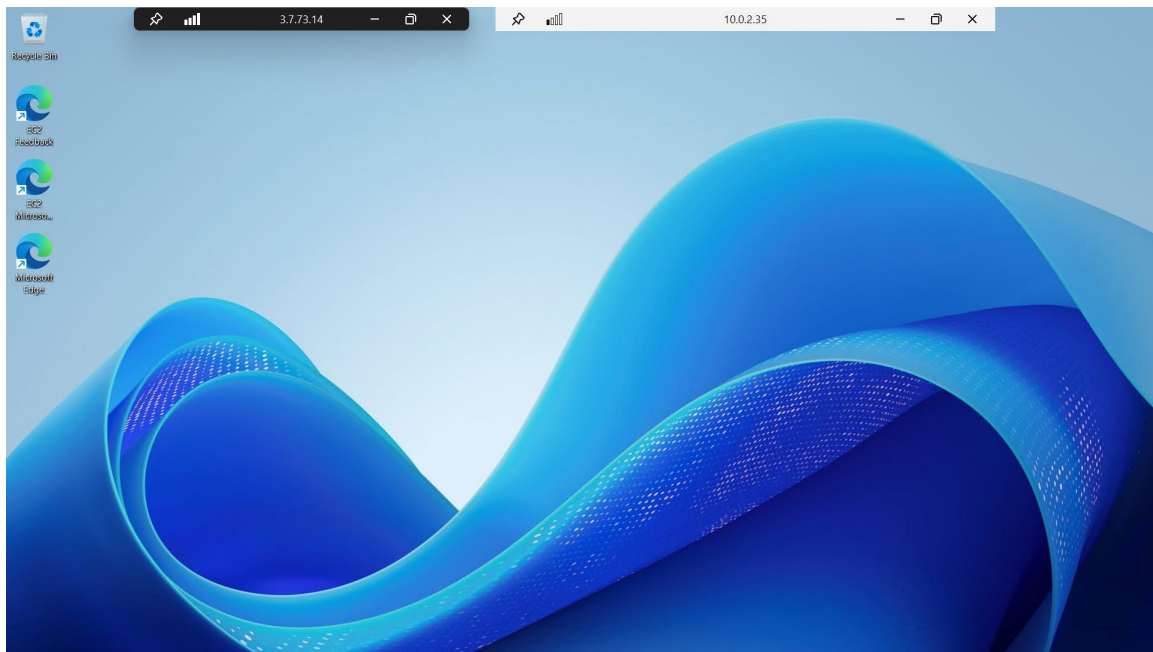
Successfully launched PUBLIC EC2



After that connect your **PRIVATE INSTANCE**

Generated password – copy your Private IP address – Go to your PUBLIC EC2 machine –

Select Remote desktop – paste your IP address inside of your public machine and user name and password



Successfully launched private ec2 inside of public ec2 with internet connecting for using NAT gateway

THANK YOU