NETWORK ADDRESS TRANSLATION (NAT) GATEWAY

An **AWS NAT Gateway** is a managed Network Address Translation (NAT) service that enables instances in a private subnet to connect to the internet or other AWS services, while preventing external systems from initiating connections to those instances. This setup is essential for scenarios where resources require outbound access without exposing them to unsolicited inbound traffic.

Types of NAT Gateways:

1. **Public NAT Gateway:**
    - **Purpose:** Allows instances in private subnets to access the internet.
    - **Configuration:** Deployed in a public subnet with an associated Elastic IP address.
    - **Routing:** Traffic from private subnets is routed to the NAT Gateway, which then communicates with the internet via an Internet Gateway.
2. **Private NAT Gateway:**
    - **Purpose:** Enables instances in private subnets to connect to other VPCs or on-premises networks without using the internet.
    - **Configuration:** Deployed in a private subnet without an Elastic IP address.
    - **Routing:** Traffic is routed through a Transit Gateway or Virtual Private Gateway to reach desired networks.

In AWS, both **NAT Gateways** and **NAT Instances** serve the purpose of enabling instances in private subnets to initiate outbound IPv4 traffic to the internet or other AWS services, while preventing unsolicited inbound traffic from external sources. Here's a comparison of the two:

NAT Gateway:

- **Managed Service:** AWS manages the deployment, maintenance, and scaling, reducing administrative overhead.
- **High Availability:** Designed with redundancy within each Availability Zone (AZ). For zone-independent architecture, deploy a NAT Gateway in each AZ.
- **Scalability:** Automatically scales up to 100 Gbps to accommodate varying traffic levels.
- **Cost:** Involves charges based on usage duration and data processed.
- **Security:** Cannot be associated with security groups; control traffic using Network ACLs.

NAT Instance:

- **Self-Managed:** Requires manual setup, configuration, and maintenance, including software updates and patches.
- **Availability:** Achieving high availability necessitates configuring failover between instances, often using scripts.
- **Scalability:** Limited by the instance type's bandwidth; scaling may require resizing or adding instances.
- **Cost:** Costs are based on the instance type, size, and duration of operation.
- **Security:** Can be associated with security groups to control inbound and outbound traffic.

Key Differences:

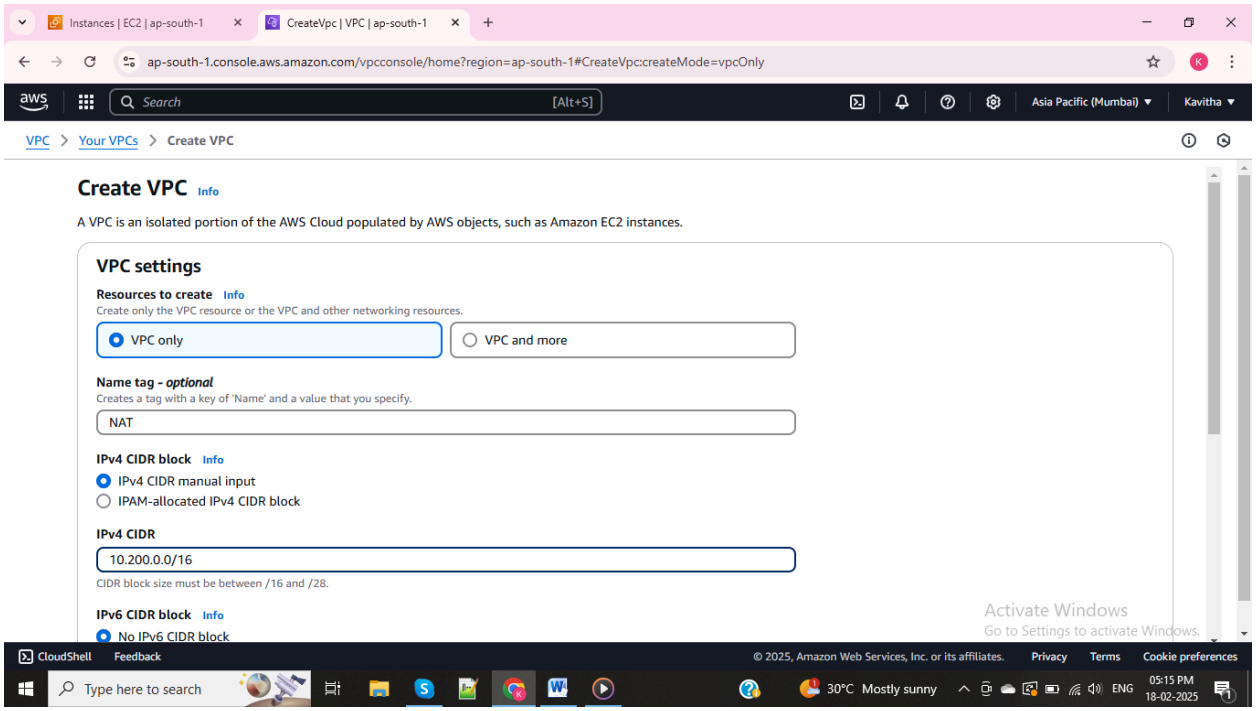
- **Management:** NAT Gateways are fully managed by AWS, whereas NAT Instances require user management.
- **Performance:** NAT Gateways offer higher bandwidth and automatic scaling; NAT Instances are constrained by the chosen instance's capacity.
- **Maintenance:** NAT Gateways require minimal maintenance; NAT Instances necessitate regular administrative tasks.
- **Security Configuration:** NAT Instances allow for security group associations; NAT Gateways do not.

Creating a **Virtual Private Cloud** (VPC) with a **Network Address Translation** (NAT) Gateway in AWS allows instances in private subnets to access the internet securely.

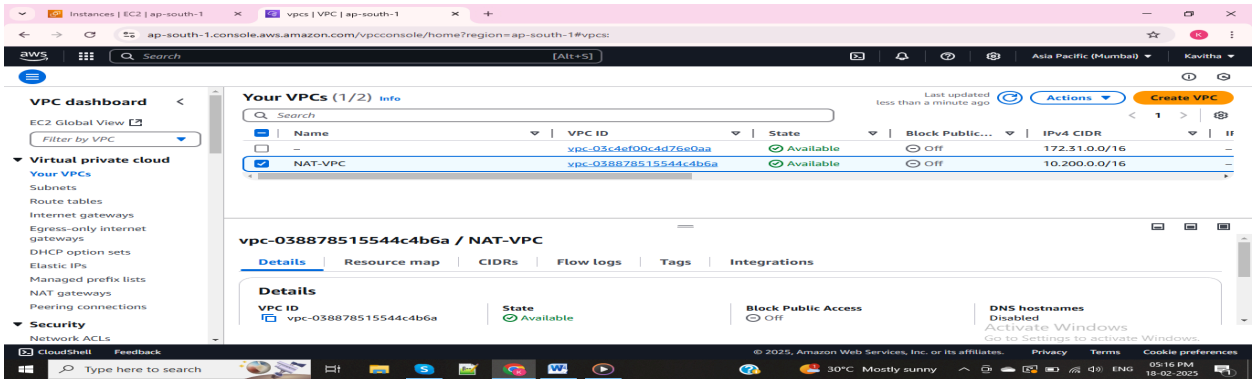
Go to **AWS** console Management:

1. Create a VPC:

- Navigate to the **VPC Dashboard** in the AWS Management Console.
- Click on **"Create VPC"**.
- Provide a **Name** and specify an **IPv4 CIDR block** (e.g., `10.200.0.0/16`).
- Choose **"Create"** to establish the VPC.

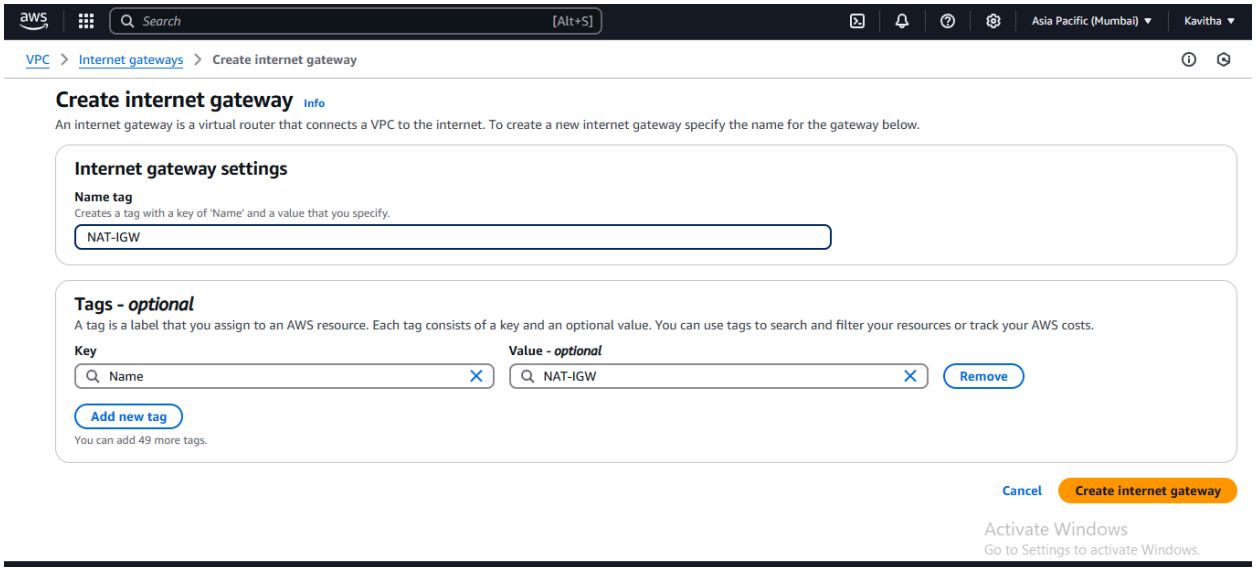


VPC created successfully

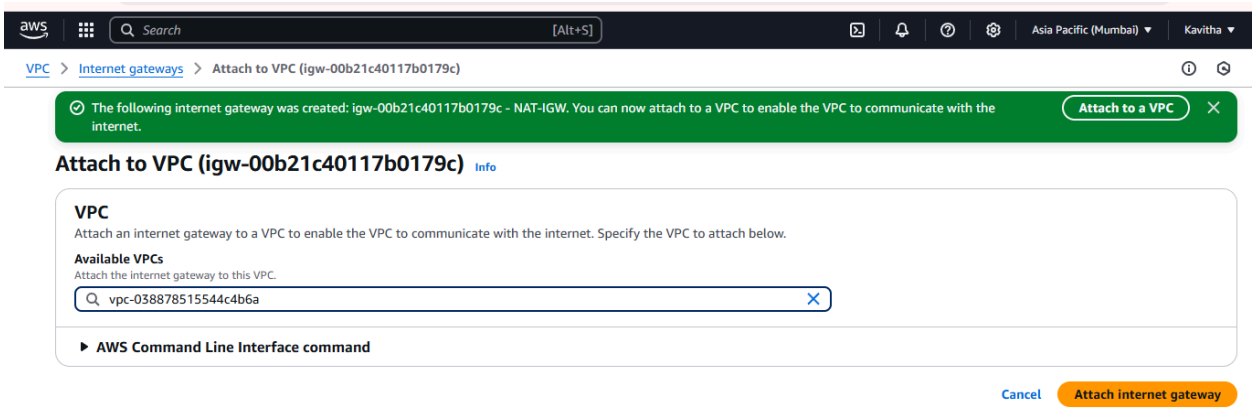


2. Create and Attach an Internet Gateway:

- In the VPC Dashboard, select **"Internet Gateways"** and click **"Create Internet Gateway"**.
- Provide a **Name** and choose **"Create"**.
- Select the newly created Internet Gateway, click **"Actions"**, and choose **"Attach to VPC"**
- Select your VPC and confirm the attachment.

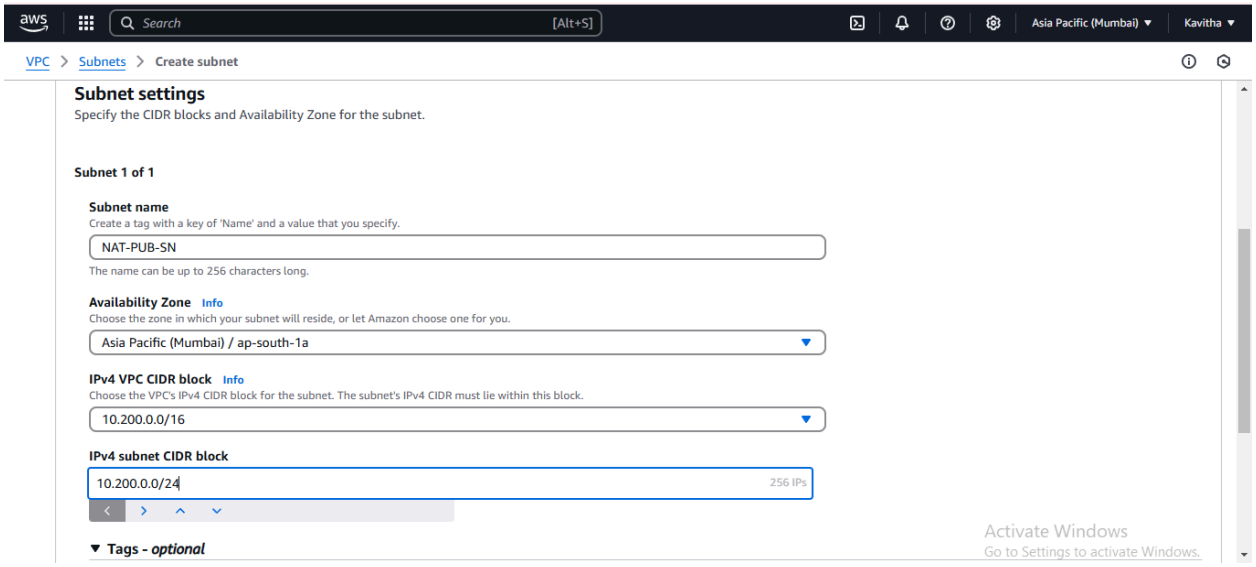


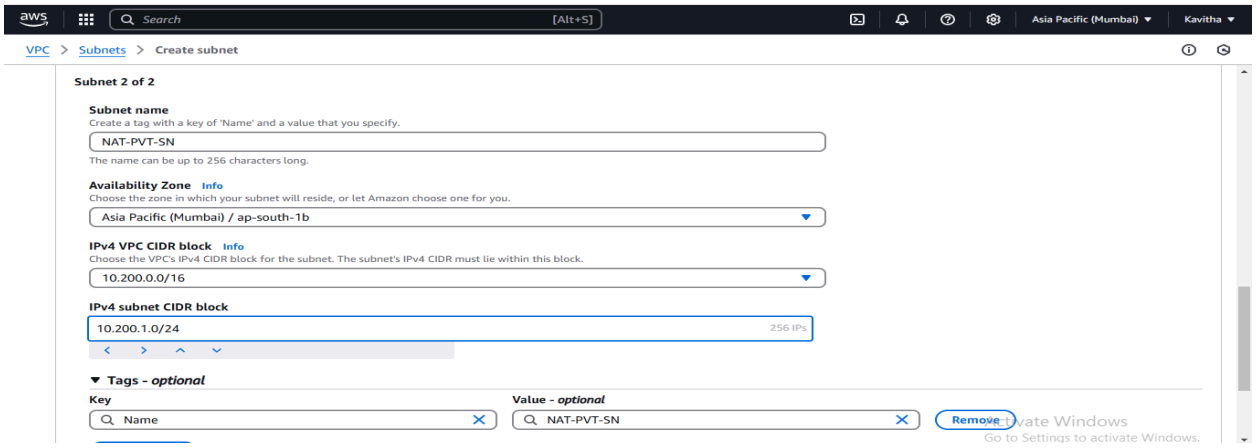
Attach the Internet gateway to VPC



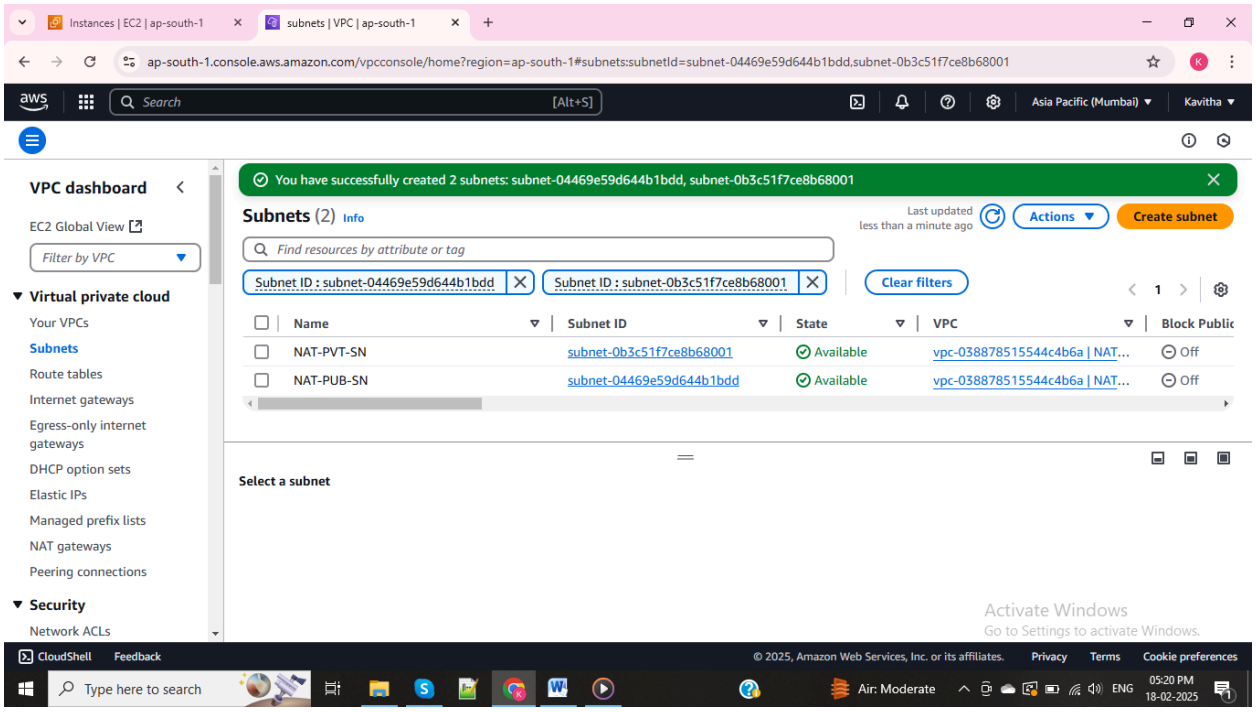
3. Create Subnets:

- **Public Subnet:**
  - In the VPC Dashboard, select **"Subnets"** and click **"Create Subnet"**.
  - Assign a **Name**, select the **VPC** created earlier, and specify an **IPv4 CIDR block** (e.g., `10.200.0.0/24`).
  - Designate an **Availability Zone** as needed.
  - Ensure **Auto-assign public IPv4 address** is **enabled**.
- **Private Subnet:**
  - Repeat the steps above to create another subnet.
  - Assign a **Name**, select the **same VPC**, and specify a different **IPv4 CIDR block** (e.g., `10.200.1.0/24`).
  - Ensure **Auto-assign public IPv4 address** is **disabled**.





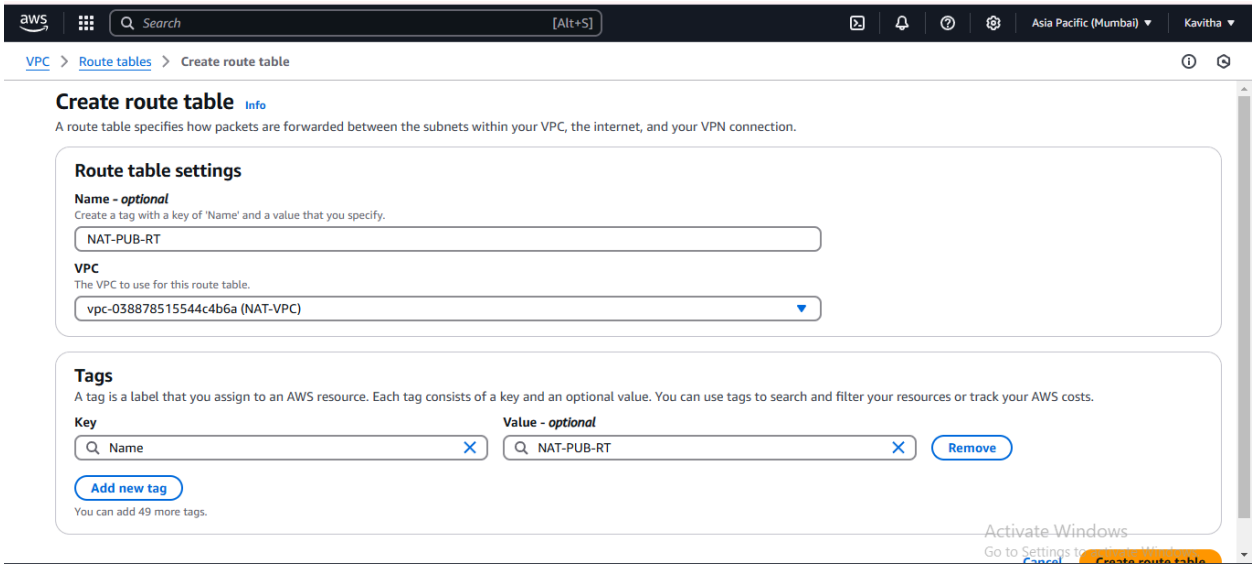
Public and private subnets created successfully



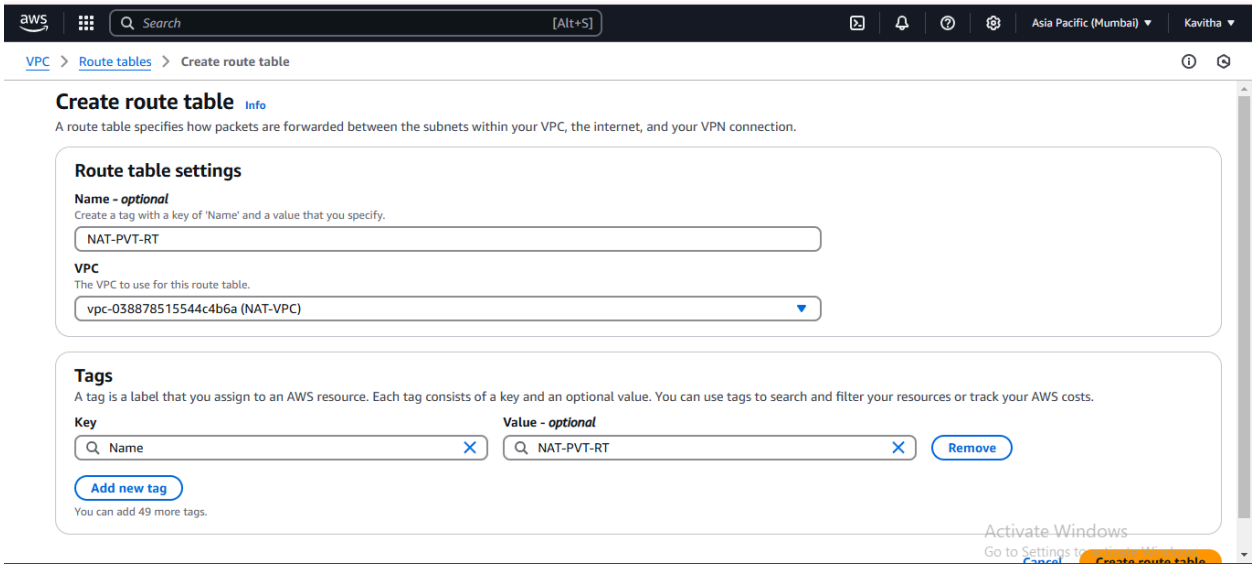
4. Configure Route Tables:

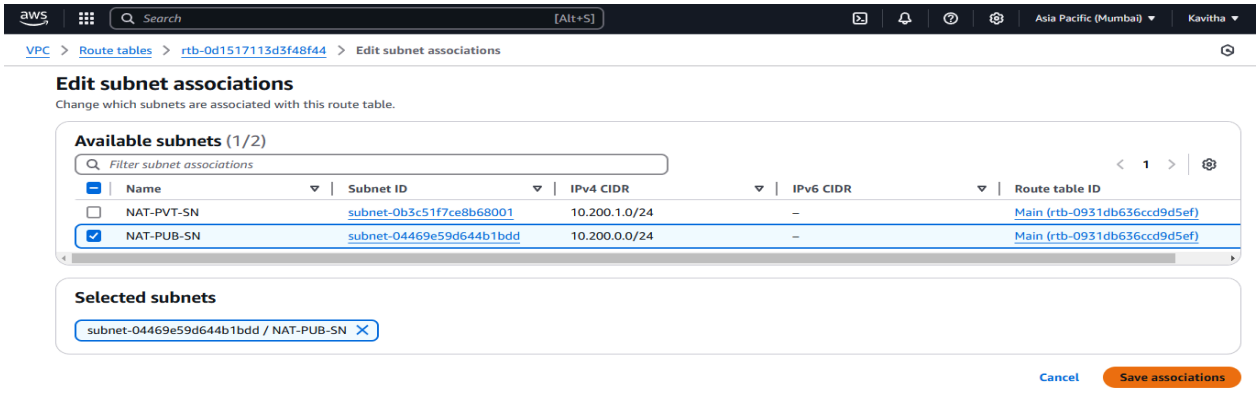
- **Public Route Table:**
  - In the VPC Dashboard, select **"Route Tables"** and click **"Create Route Table"**.
  - Assign a **Name**, select your **VPC**, and choose **"Create"**.
  - With the new route table selected, navigate to the **"Routes"** tab and click **"Edit routes"**.
  - Add a route with **Destination** `0.0.0.0/0` and **Target** as the Internet Gateway.
  - Save the changes.
  - Navigate to the **"Subnet Associations"** tab, click **"Edit subnet associations"**, and associate the **public subnet**.
- **Private Route Table:**
  - Repeat the steps to create another route table for the private subnet.
  - No routes need to be added at this stage.
  - Associate this route table with the **private subnet**.

Public Route Table



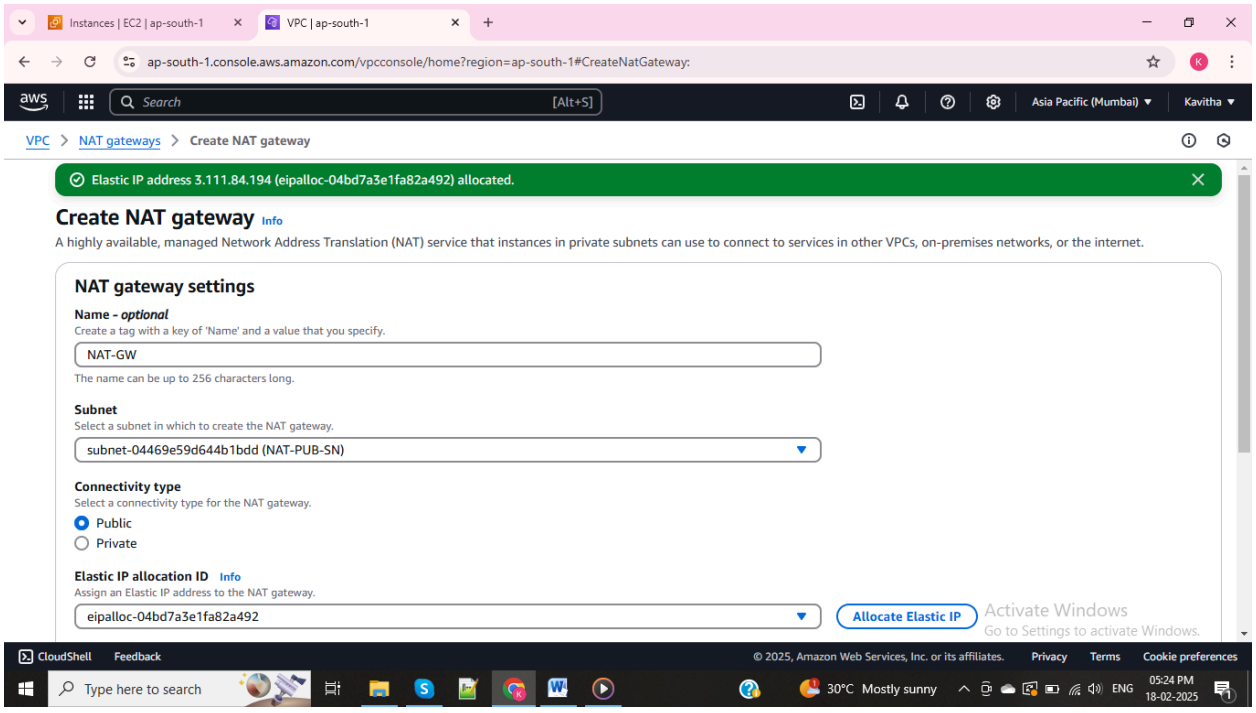
Private Route Table



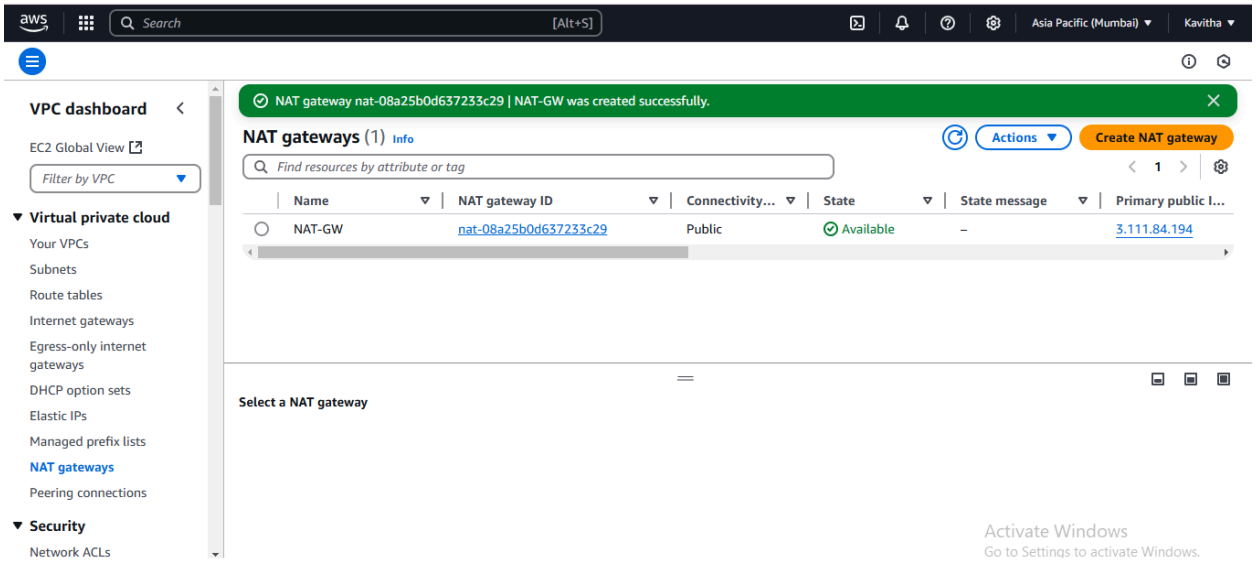


5. Create a NAT Gateway:

- In the VPC Dashboard, select **"NAT Gateways"** and click **"Create NAT Gateway"**.
- Assign a **Name**, select the **public subnet**, and allocate a new **Elastic IP**.
- Choose **"Create NAT Gateway"**.

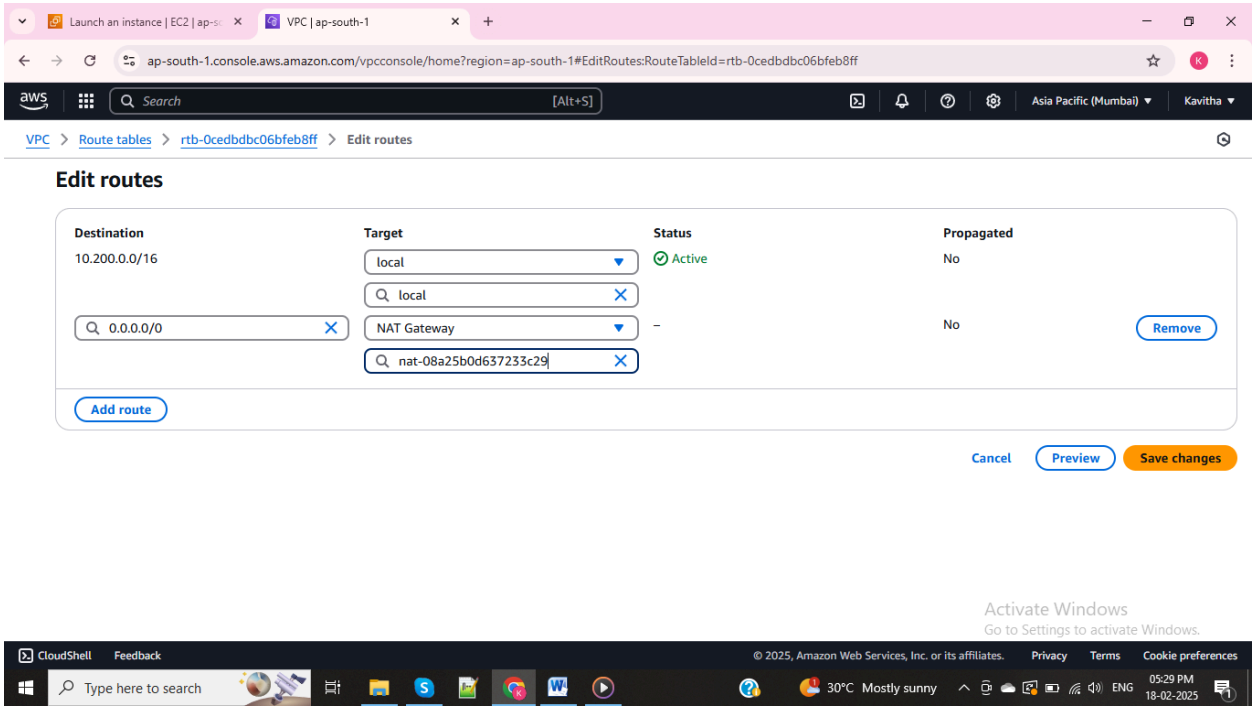


NAT GATEWAY created successfully



6. Update Private Route Table:

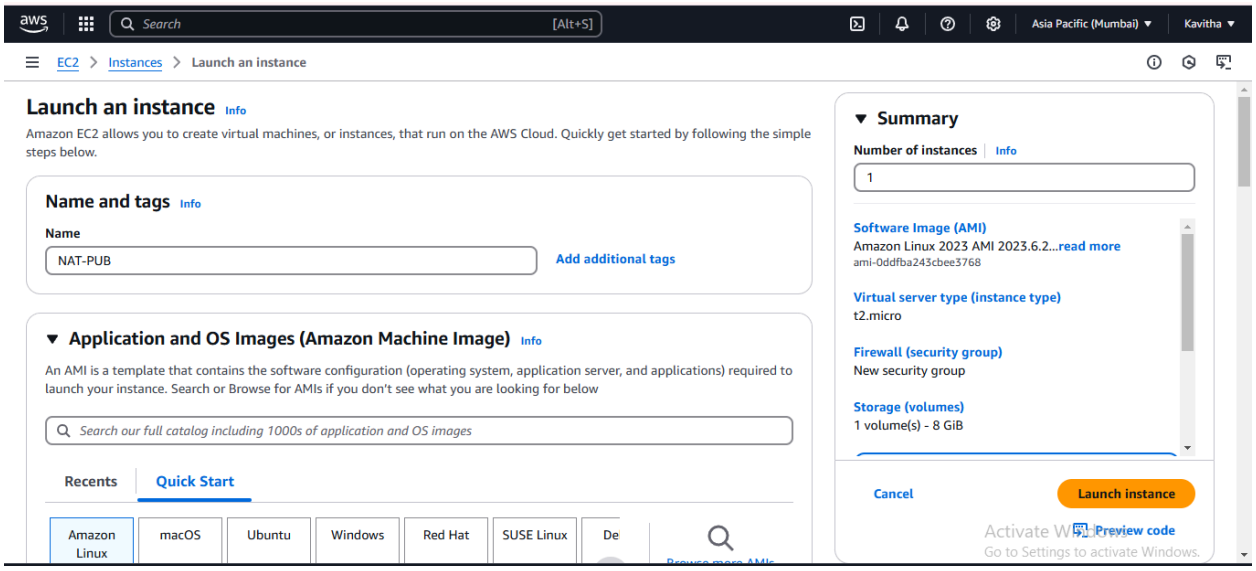
- Select the **private route table** created earlier.
- Navigate to the **"Routes"** tab and click **"Edit routes"**.
- Add a route with **Destination** `0.0.0.0/0` and **Target** as the NAT Gateway.
- Save the changes.

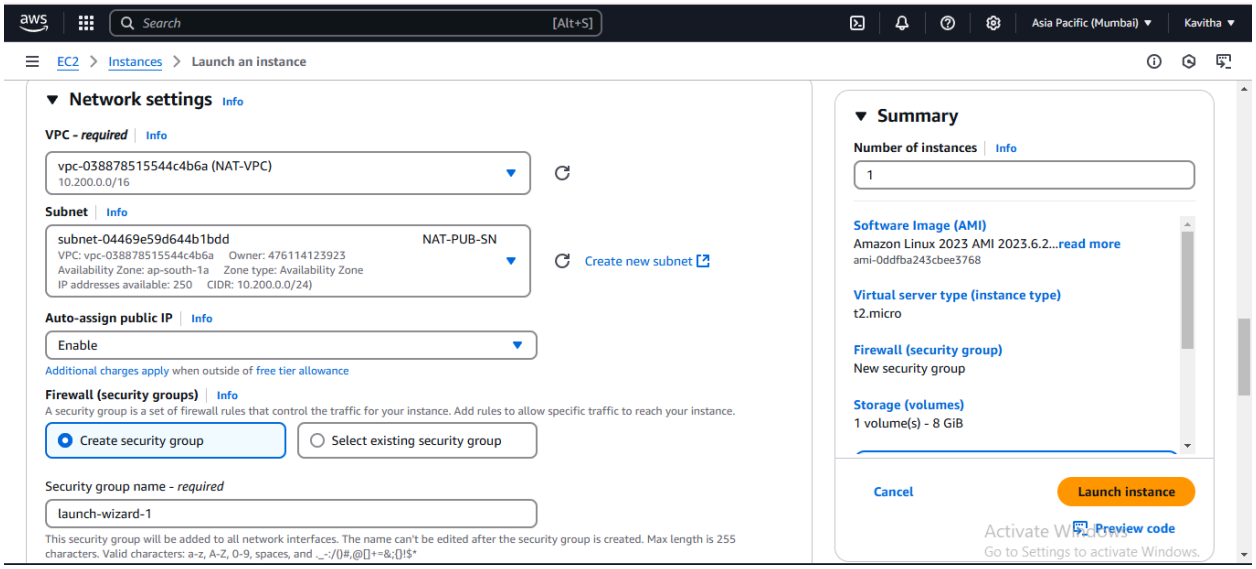


To enable secure communication between instances in a private subnet and those in a public subnet within your Virtual Private Cloud (VPC).

1. **Launch a Public Instance**
Purpose: Serves as an intermediary, allowing secure SSH access to instances in the private subnet.
Steps:

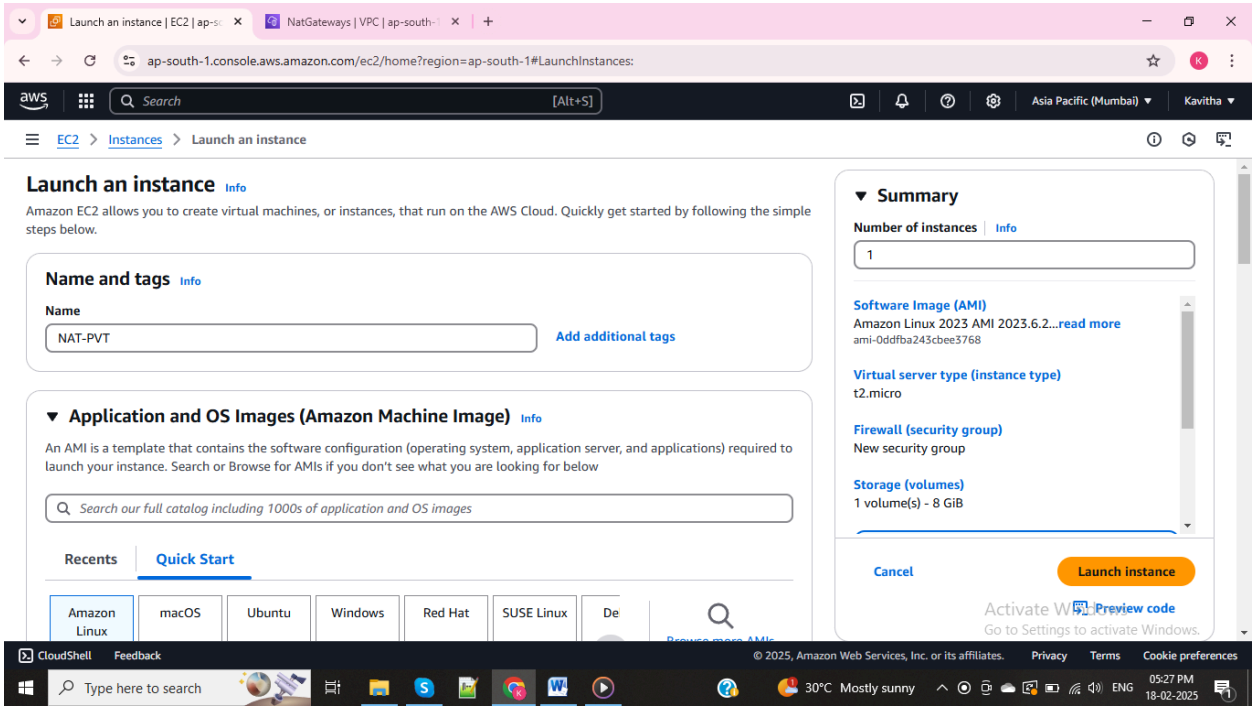
- Launch an EC2 instance in the **public subnet** of created VPC.
- Assign it a **public IP address**
- Attach a **security group** that permits inbound SSH (port 22) access.

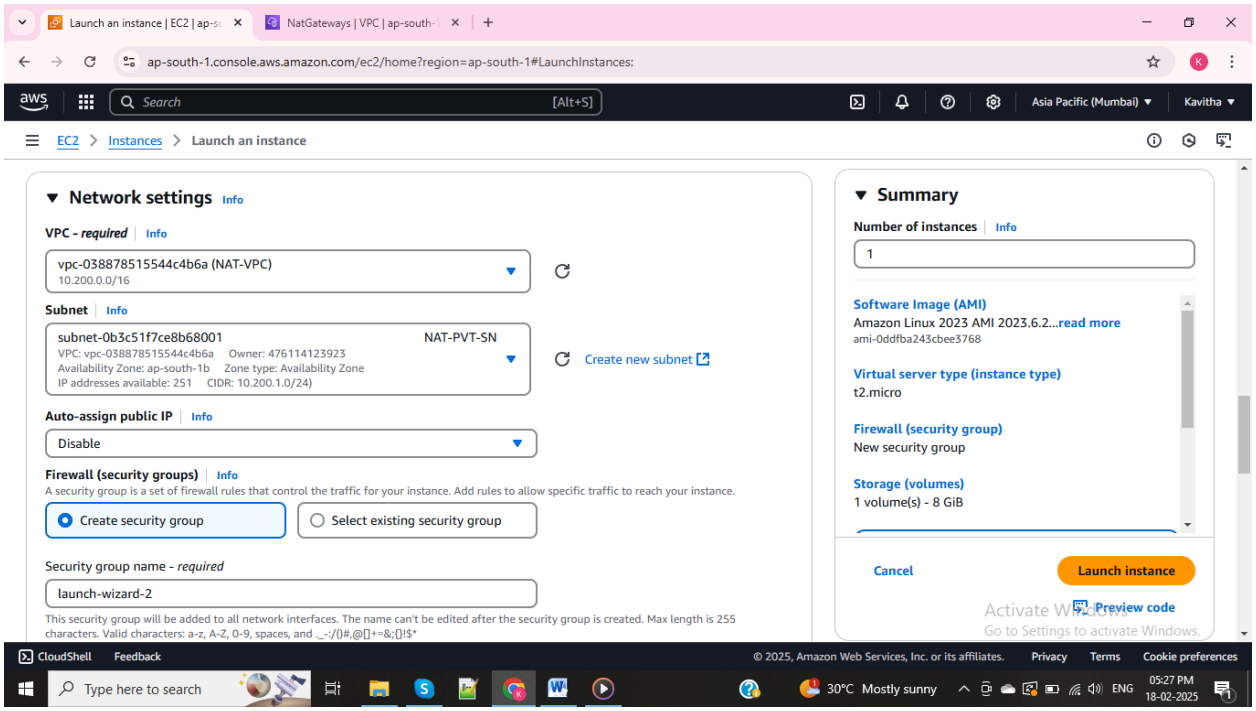




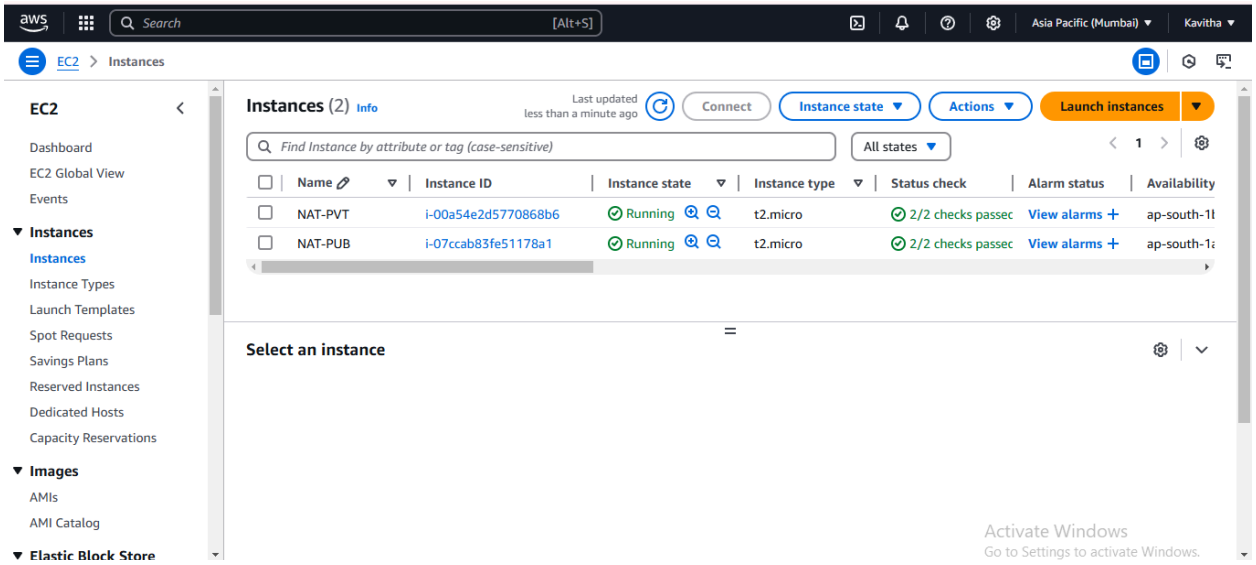
2. Launch a Private Instance:

- **Purpose:** The target instance residing in the private subnet without direct internet exposure.
- **Steps:**
  - Launch an EC2 instance in the **private subnet** of created VPC.
  - Ensure it does **not** have a public IP address.
  - Attach a **security group** that allows inbound SSH (port 22) traffic from the **public instance's security group**.

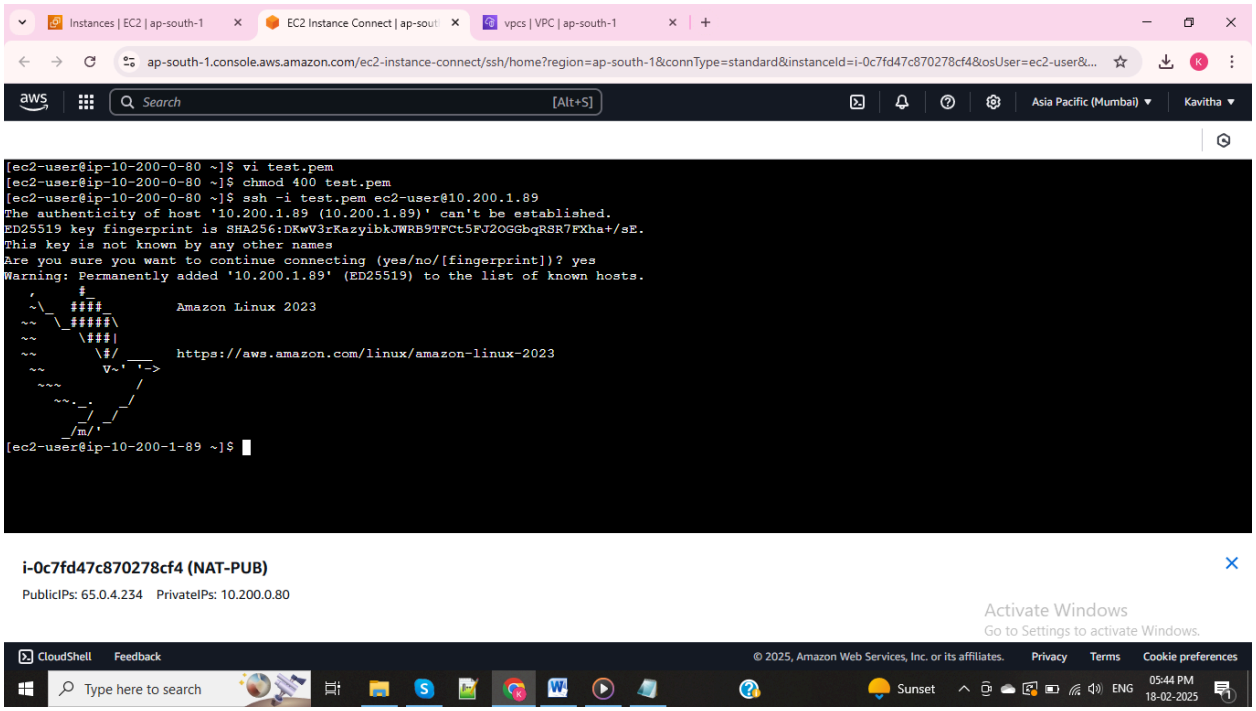

Launch an instance | EC2 | ap-s
NatGateways | VPC | ap-south-
ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:
Search
[Alt+S]
Asia Pacific (Mumbai)
Kavitha
EC2 > Instances > Launch an instance
Launch an instance Info
Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.
Name and tags Info
Name
NAT-PVT
Add additional tags
Application and OS Images (Amazon Machine Image) Info
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below
Search our full catalog including 1000s of application and OS images
Recents
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Linux
De
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.6.2...read more
ami-0ddfba243cbee3768
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Cancel
Launch instance
Preview code
Activate Windows
Go to Settings to activate Windows.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
30°C Mostly sunny
ENG
05:27 PM
18-02-2025


Launch an instance | EC2 | ap-s
NatGateways | VPC | ap-south-
ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:
Search
[Alt+S]
Asia Pacific (Mumbai)
Kavitha
EC2 > Instances > Launch an instance
Network settings Info
VPC - required Info
vpc-038878515544c4b6a (NAT-VPC)
10.200.0.0/16
Subnet Info
subnet-0b3c51f7ce8b68001 NAT-PVT-SN
VPC: vpc-038878515544c4b6a Owner: 476114123923
Availability Zone: ap-south-1b Zone type: Availability Zone
IP addresses available: 251 CIDR: 10.200.1.0/24)
Create new subnet
Auto-assign public IP Info
Disable
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
Security group name - required
launch-wizard-2
This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.6.2...read more
ami-0ddfba243cbee3768
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Cancel
Launch instance
Preview code
Activate Windows
Go to Settings to activate Windows.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
30°C Mostly sunny
ENG
05:27 PM
18-02-2025

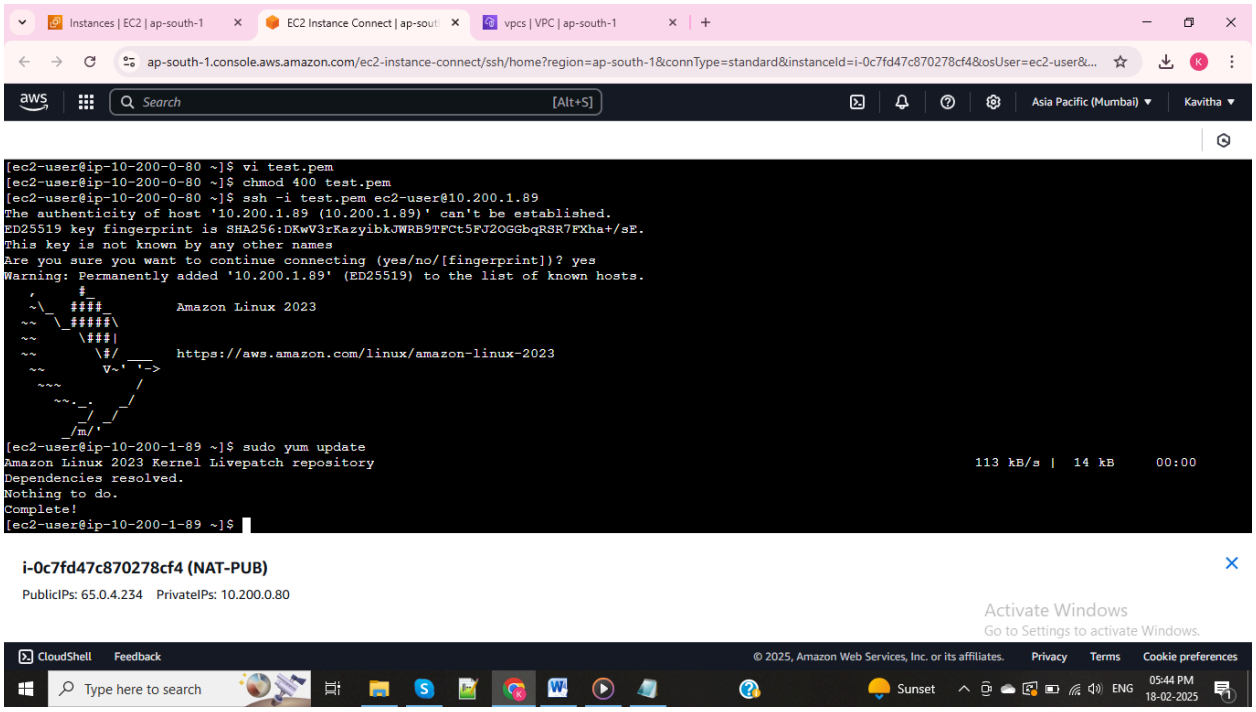
Public and Private Instance created



Connect the **Public Instance** --> Update the server --> Create a **.pem** file and Key was pasted in that file and save.
Change permission for the key --> **chmod 400 test.pem**
To connect private Instance give --> **ssh -i test.pem ec2-user@(private Ip of private Instance)**



Inside private Instance, server was updated by **sudo yum update**



Checking the Internet by ping google.com