Implementation of :- An Amazon S3 File Upload Notification System Using SNS

Objective of the Project

The main objective of this project is to **create a simple notification system** that sends an alert whenever a file is uploaded to an **Amazon S3 bucket** using **Amazon Simple Notification Service (SNS)**.

AWS Services Used in This Project and Their Usage

01 Amazon Simple Storage Service (Amazon S3)

- Amazon S3 is used to store the uploaded files/documents that are part of the project workflow.
- It acts as the core service where file upload events are detected and forwarded for notification processing.

02 Amazon Simple Notification Service (Amazon SNS)

- Amazon SNS is used to generate and deliver notifications when a file upload event occurs in the S3 bucket.
- It ensures timely alert delivery to subscribed endpoints, confirming successful file upload operations

03 Amazon Elastic Compute Cloud (Amazon EC2)

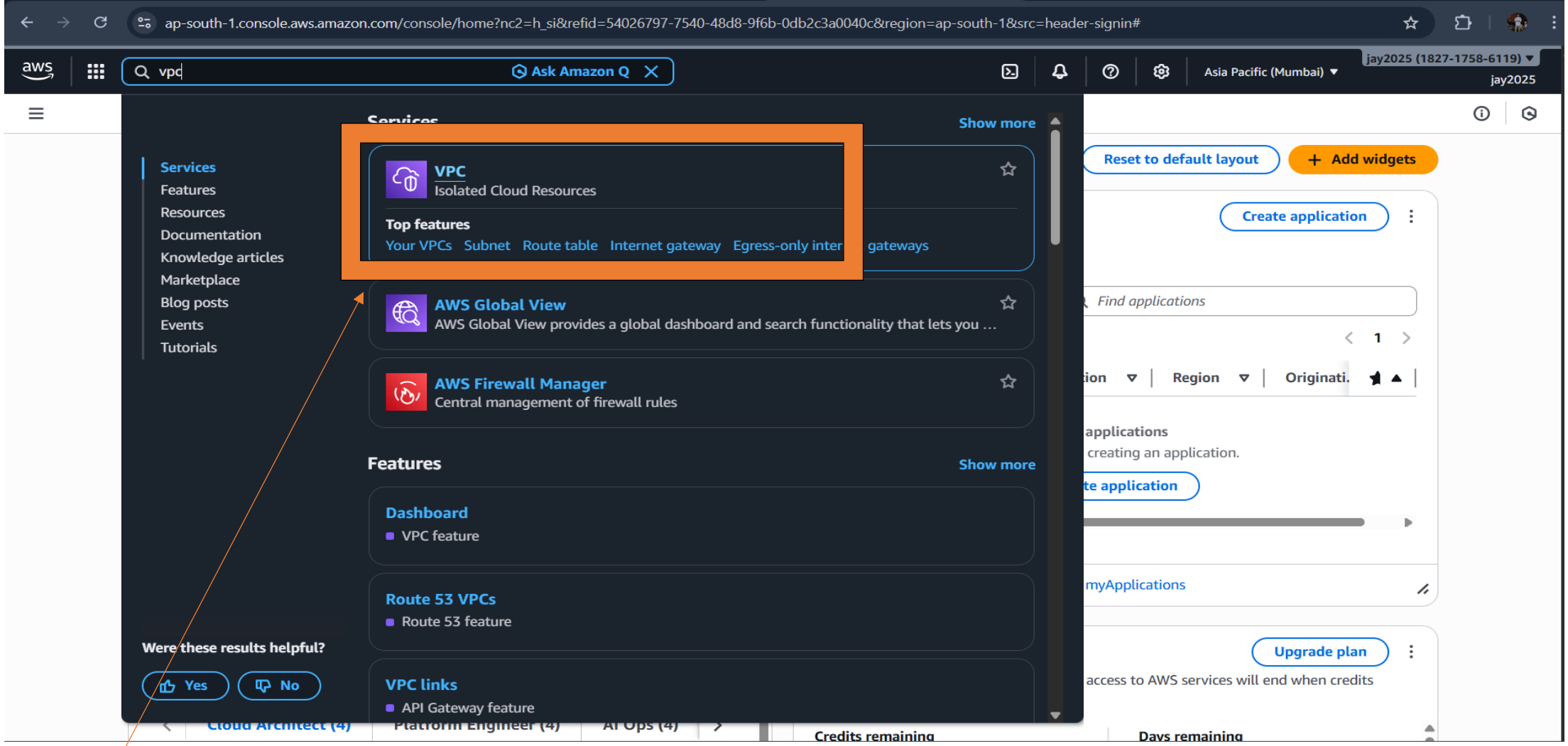
- Amazon EC2 is used as the execution environment to perform file upload operations to the S3 bucket.
- It also supports testing and validation of the notification system using AWS CLI commands.

04 AWS Identity and Access Management (IAM)

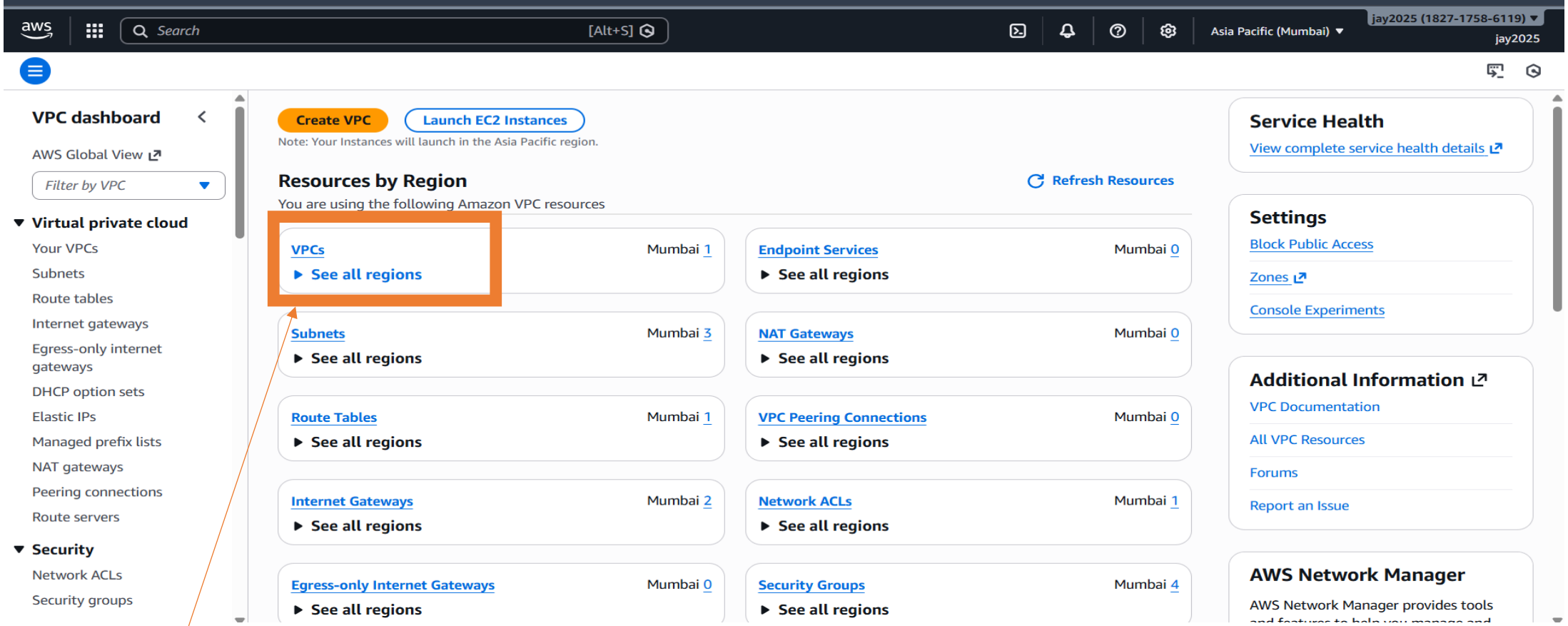
- IAM is used to define roles and permissions that allow secure interaction between S3, SNS, and EC2.
- It ensures controlled access by enforcing security policies for file uploads and notification publishing.

05 Amazon Virtual Private Cloud (Amazon VPC)

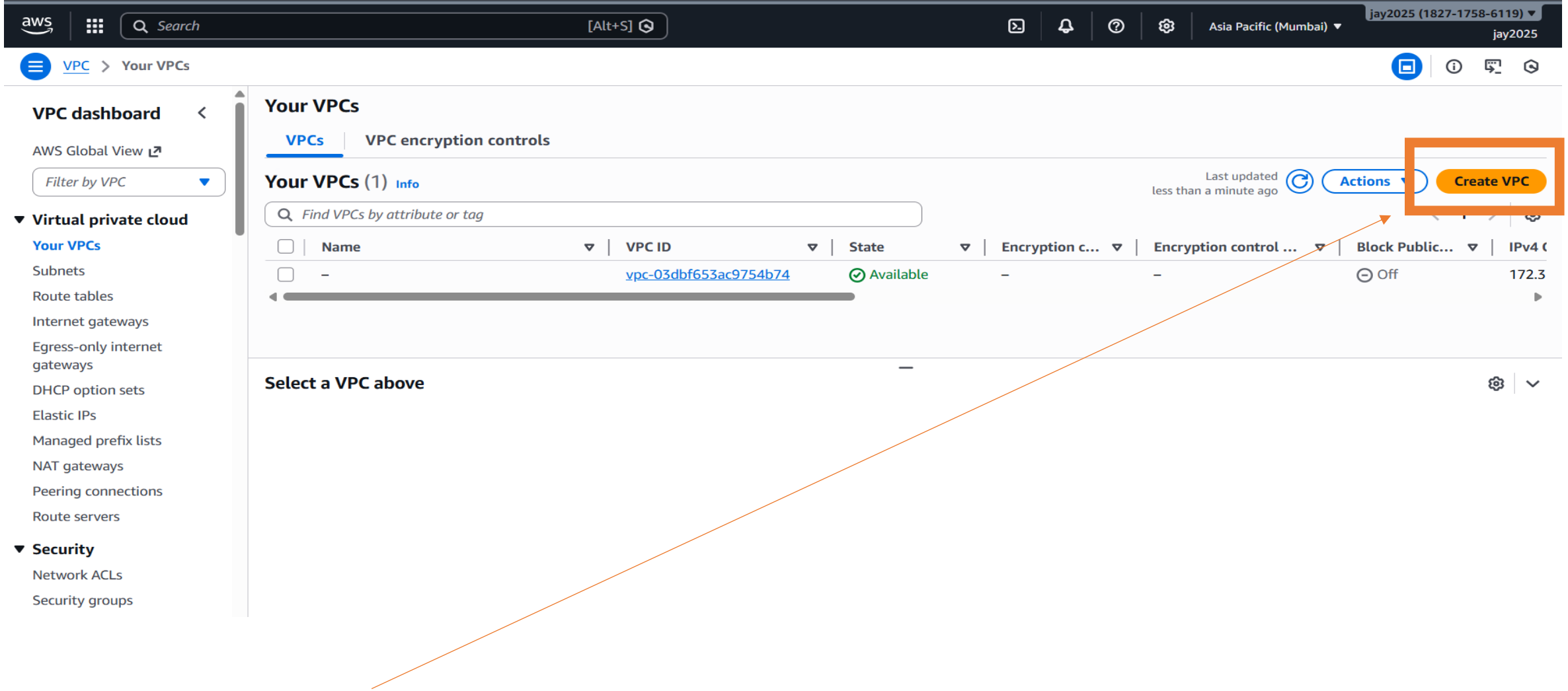
- Amazon VPC provides a secure and isolated network environment for the EC2 instance used in the project.
- It enhances security by managing network access and protecting cloud resources from unauthorized access.



This step is used to access the VPC service in AWS.
VPC provides a secure and isolated network for running the EC2 instance used in this project.
It ensures that the project resources are deployed in a controlled environment.

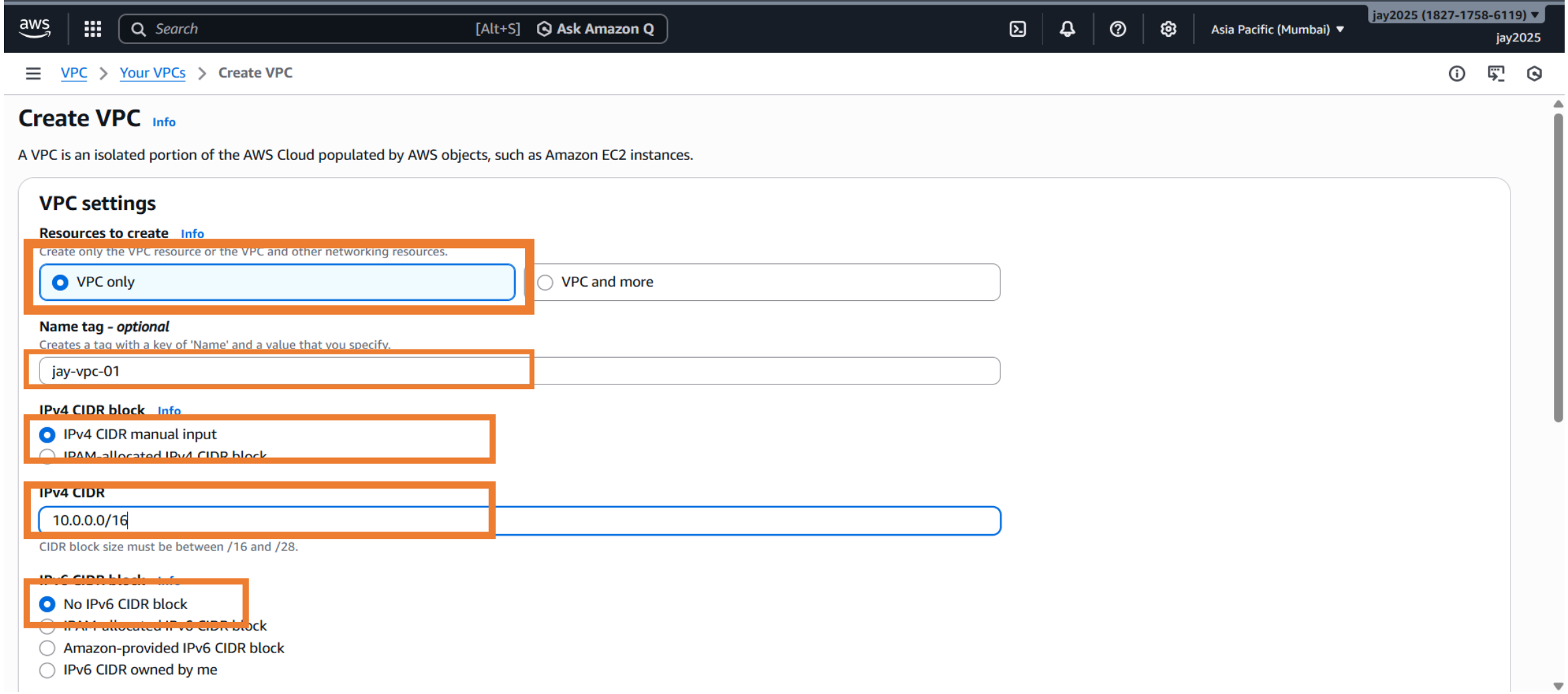


Click on VPCs



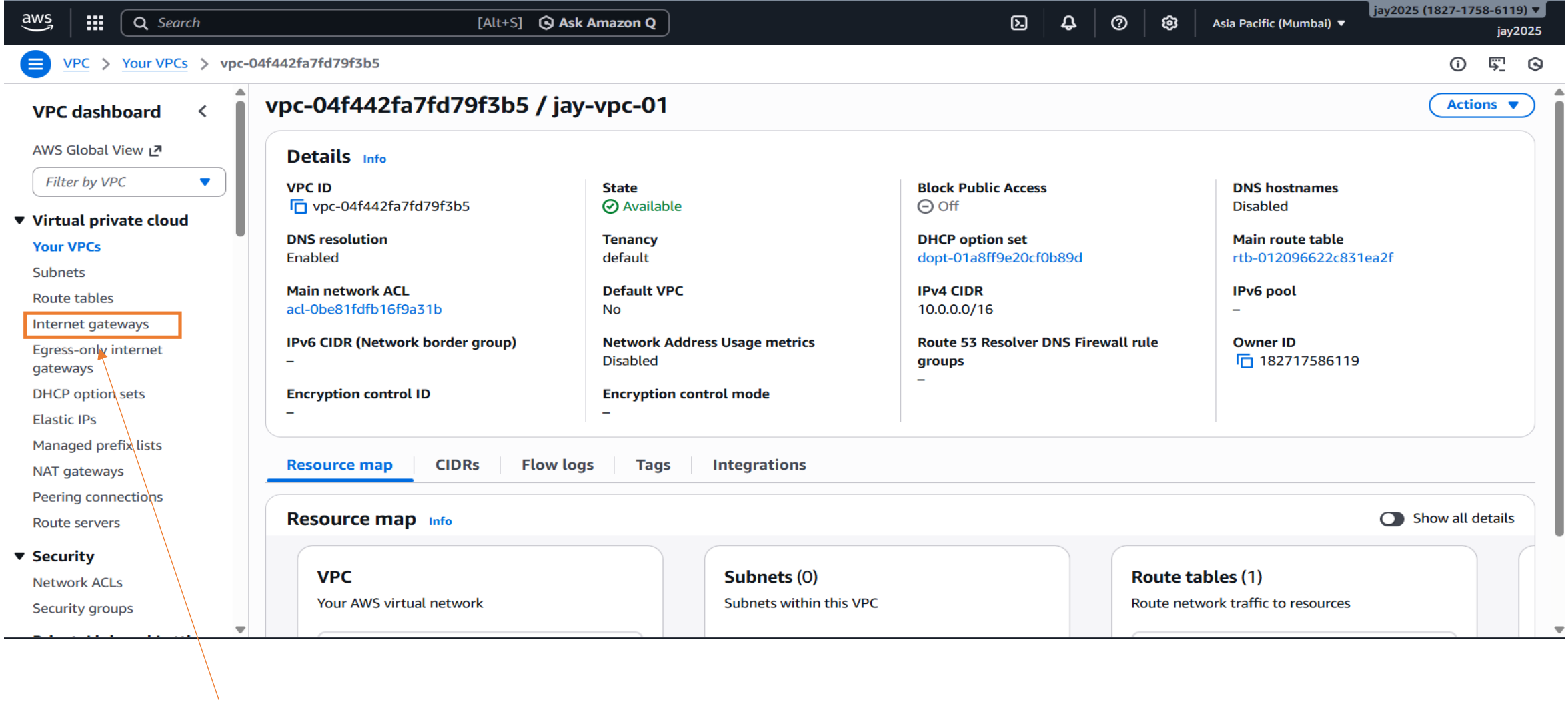
Click on Create VPC

- Create a Virtual Private Cloud for the project.



- Go to the VPC service in AWS.
- Click Create VPC → choose “Only VPC.”
- Provide a Name tag **jay-vpc-01**
- Define the CIDR block **10.0.0.0/16**
- Disable IPv6 not required we use ipv4

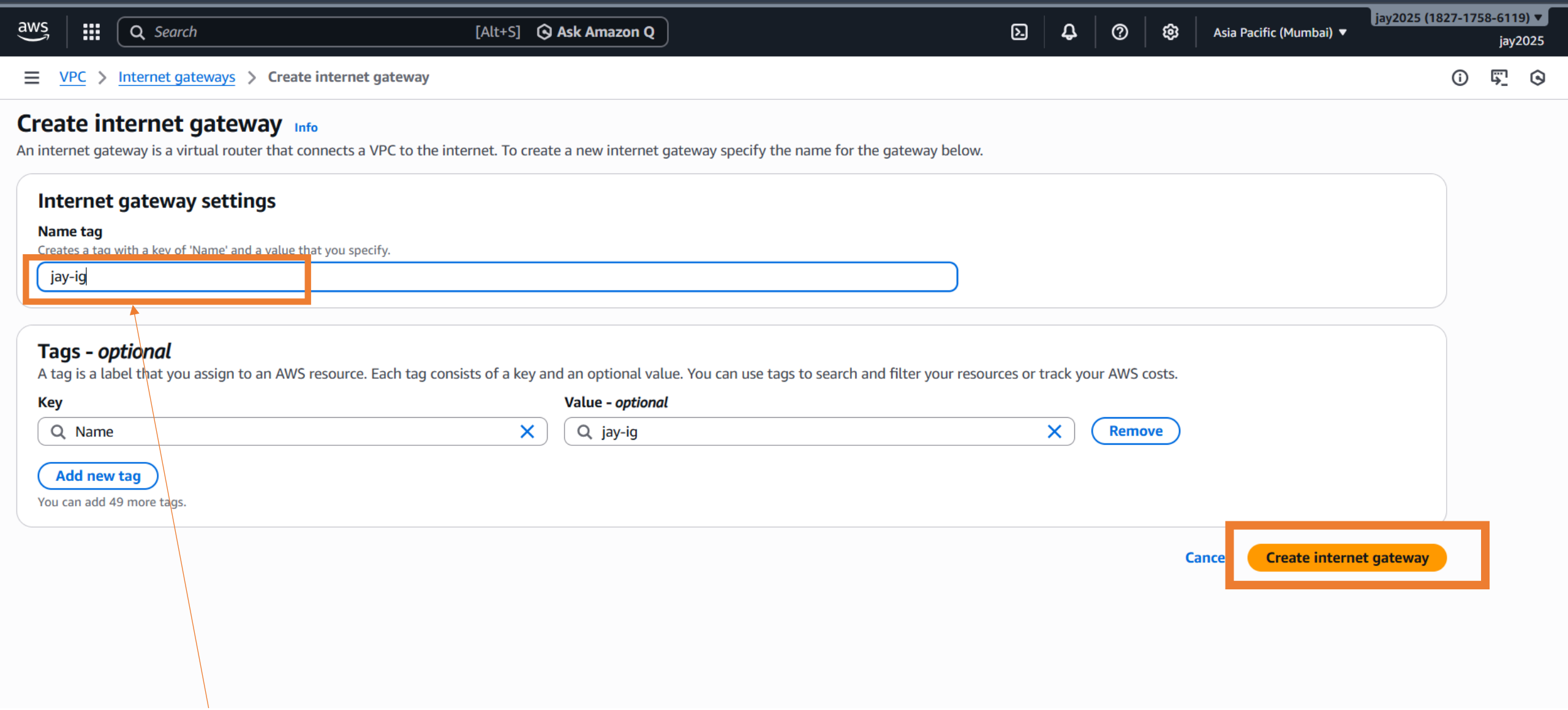
Why Use: Establishes a secure, isolated network environment for project.



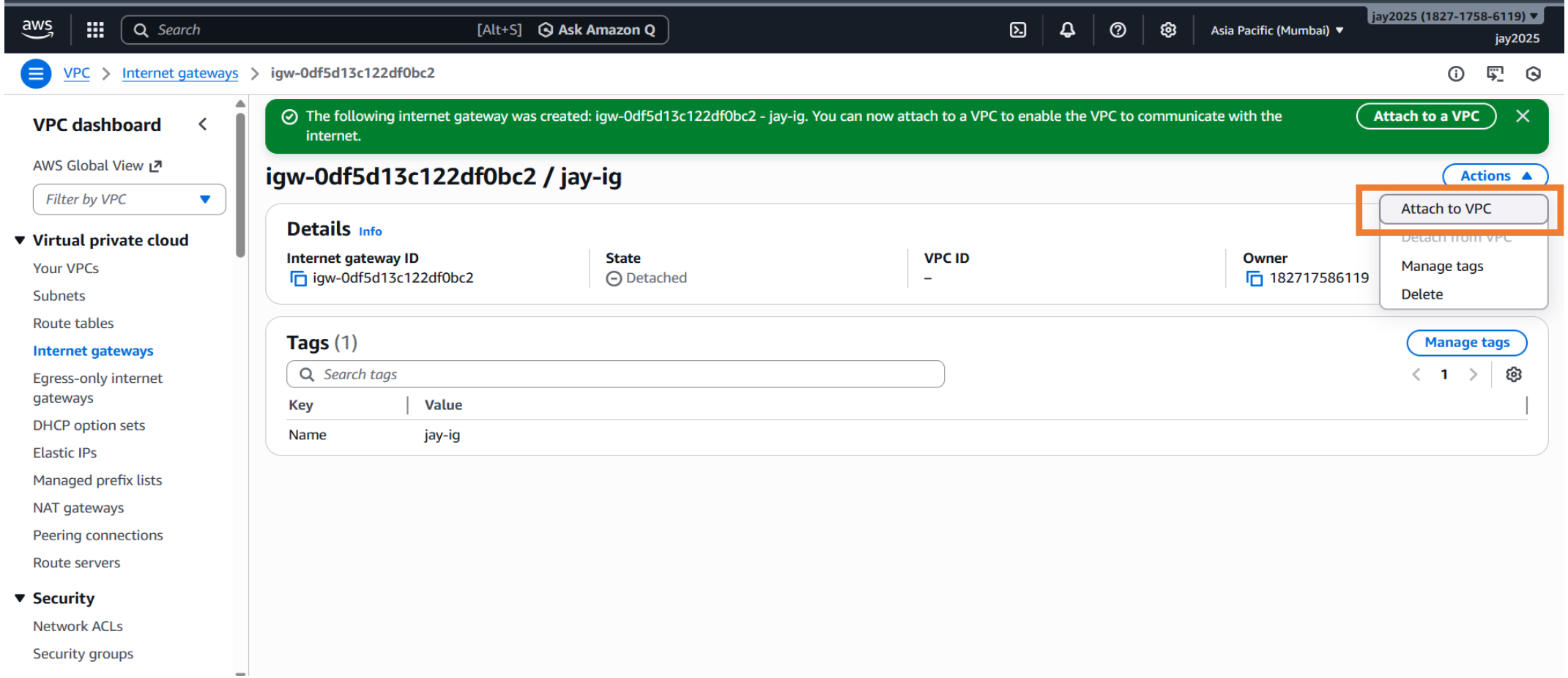
Click on internet gatways.

Why Use

- Required for EC2 instances in the public subnet to access the internet.
- Enables inbound and outbound traffic for public-facing applications.



Adding a Name tag **jay-ig** to the Internet Gateway.



Click on Attach vpc.

Why Use This Step

- Without attachment, the IGW exists but isn’t connected to network.

Attach to VPC (igw-0df5d13c122df0bc2) 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

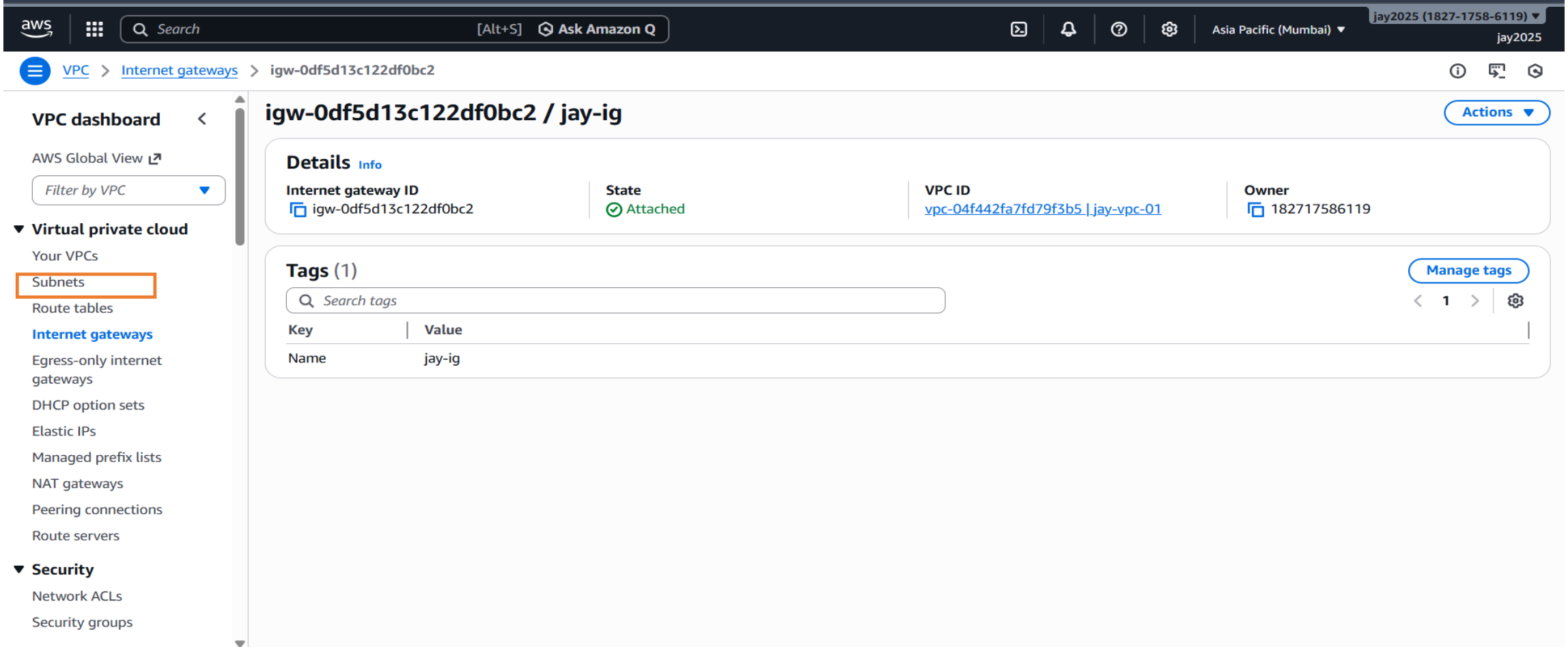
Attach the internet gateway to this VPC.

vpc-04f442fa7fd79f3b5

▶ **AWS Command Line Interface command**

Cancel | Attach internet gateway

Attach IGW to our vpc

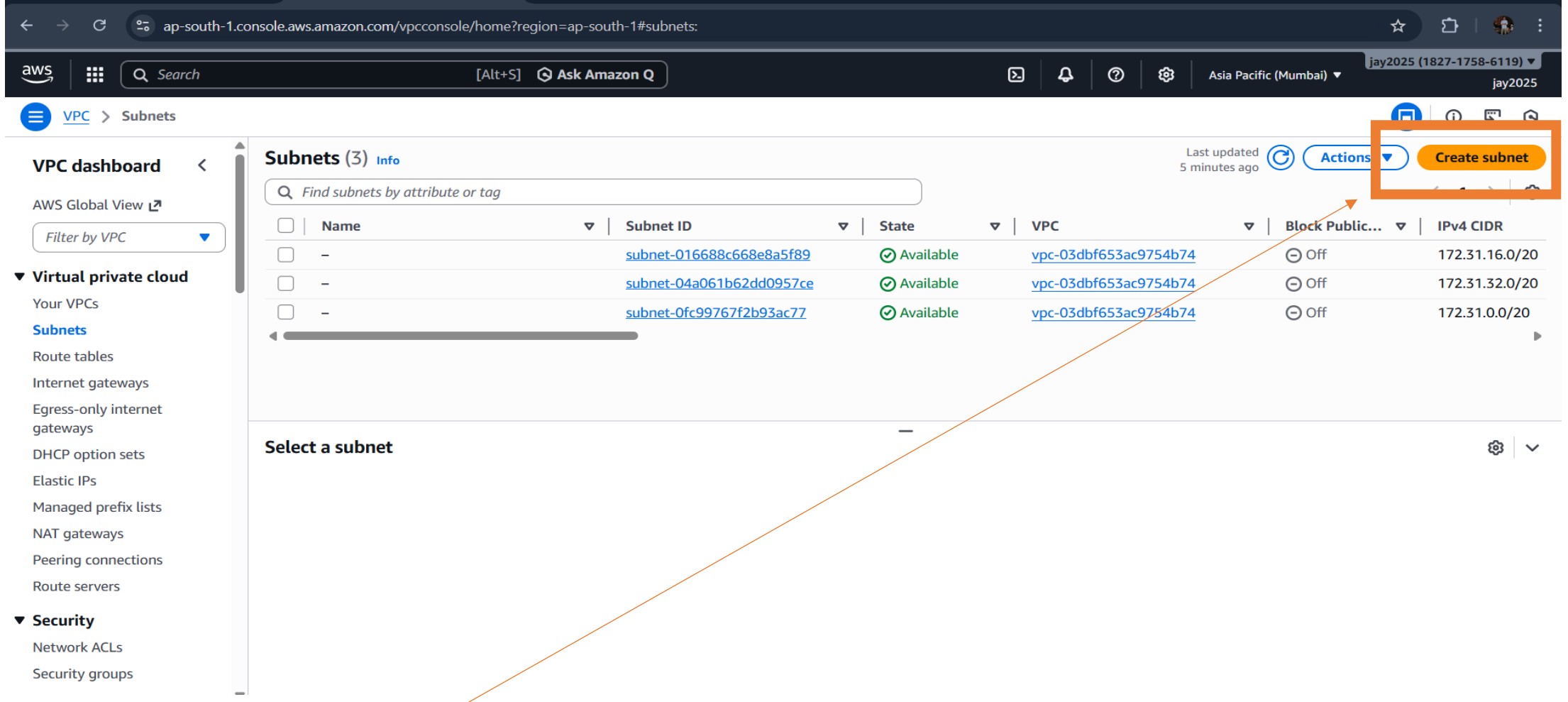


Click on subnet

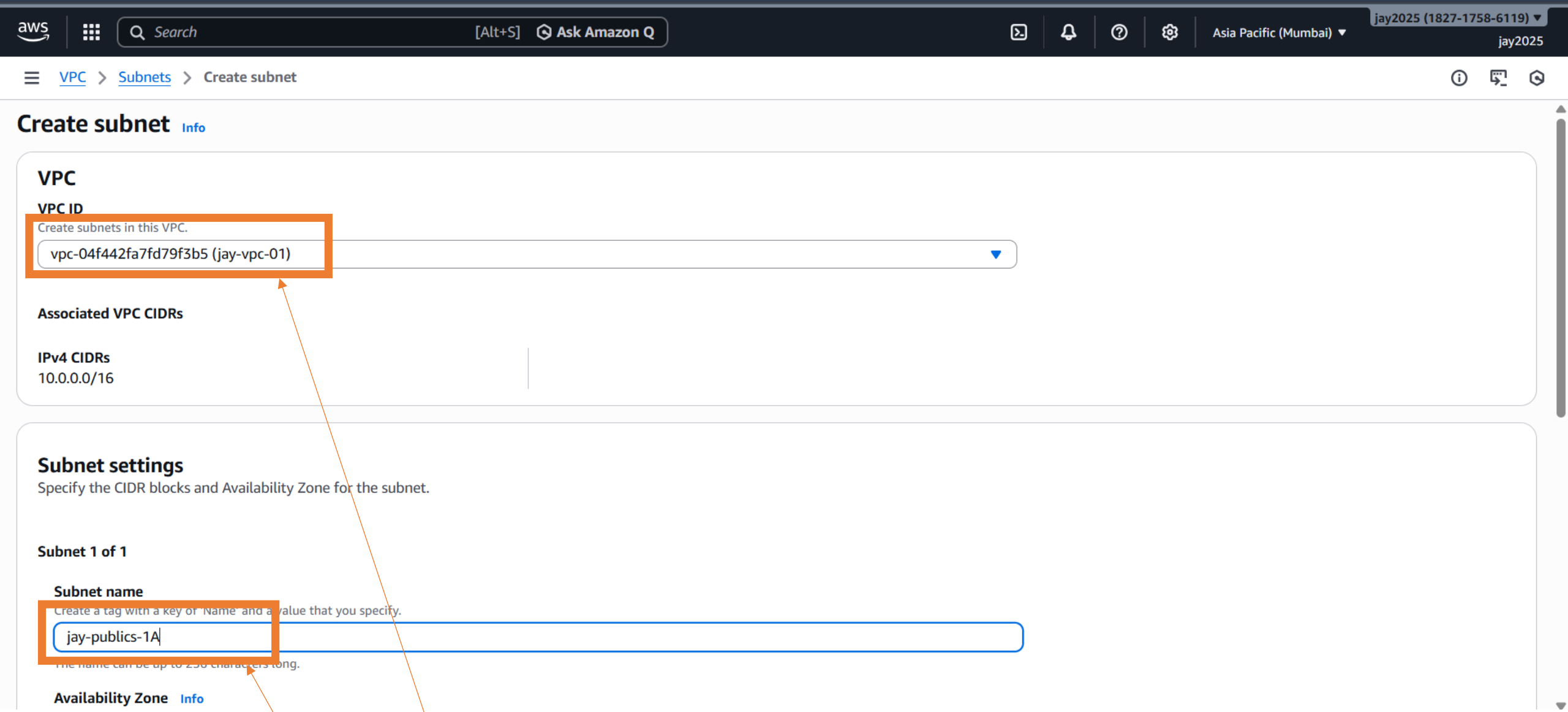
- Definition: Sub-divisions of VPC's IP range.
- Types:
- Public Subnet → connected to the internet via an Internet Gateway.
- Private Subnet → isolated, no direct internet access.

Why Use:

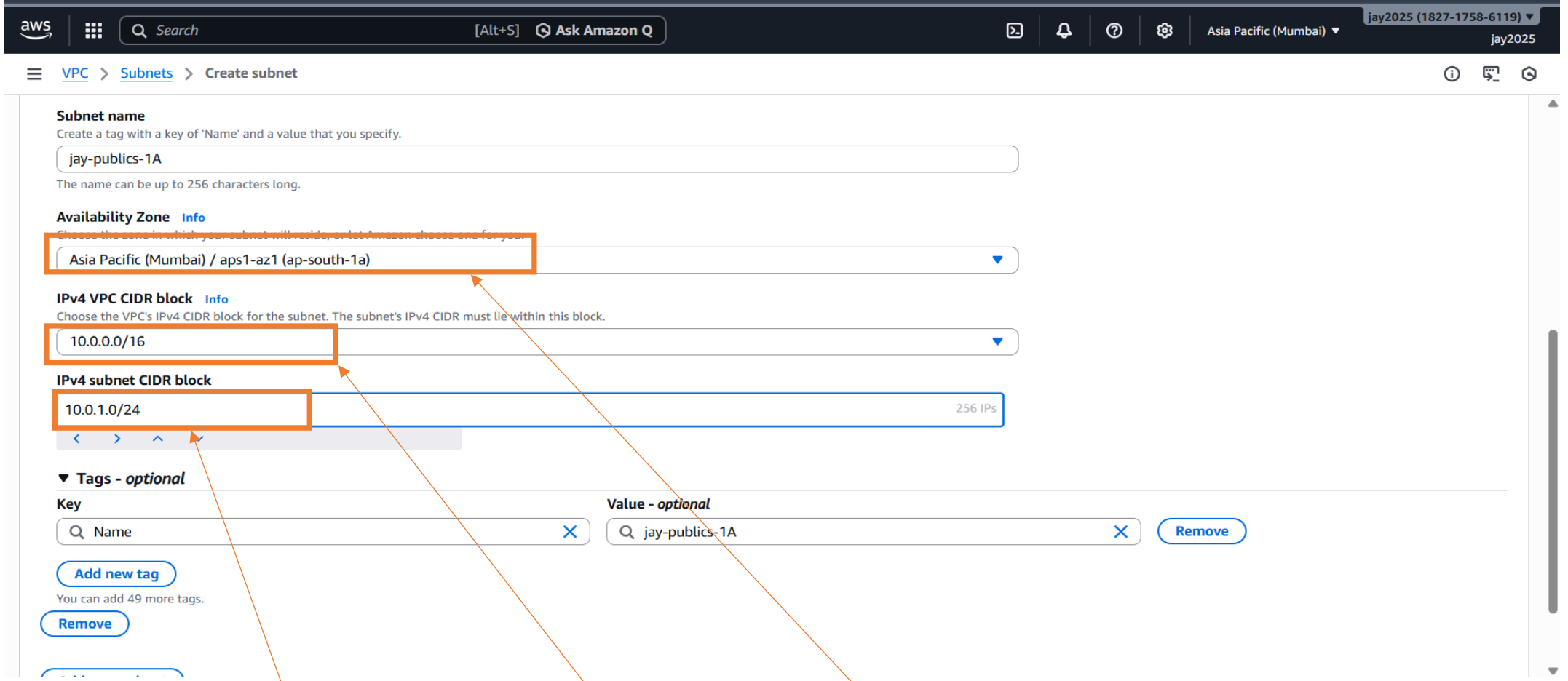
- Organizes resources by accessibility.
- Public subnet for web servers, private subnet for databases or backend services.



Creat new subnet



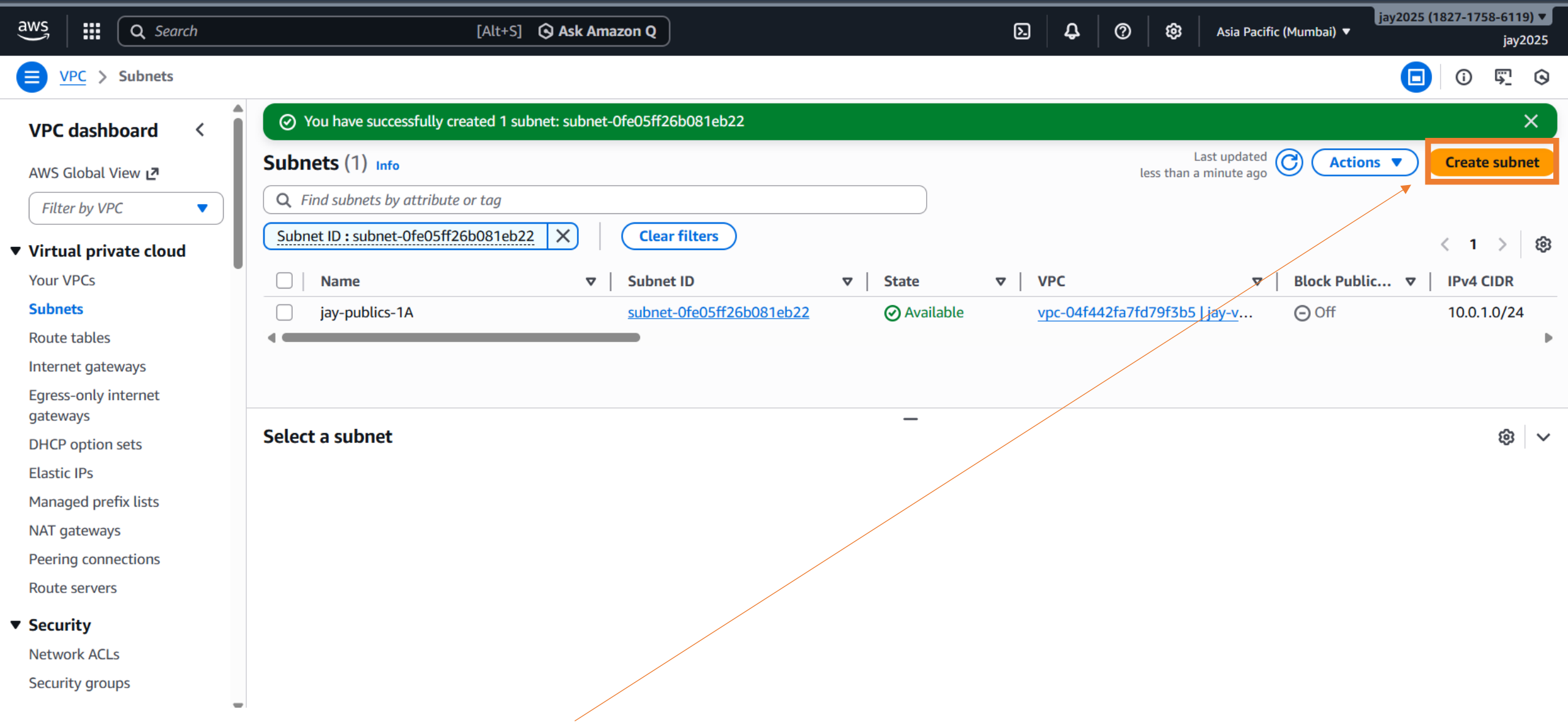
Connect to my vpc and Defining a Name tag **jay-publics-1A** to mark this subnet as the public zone for internet-accessible resources.



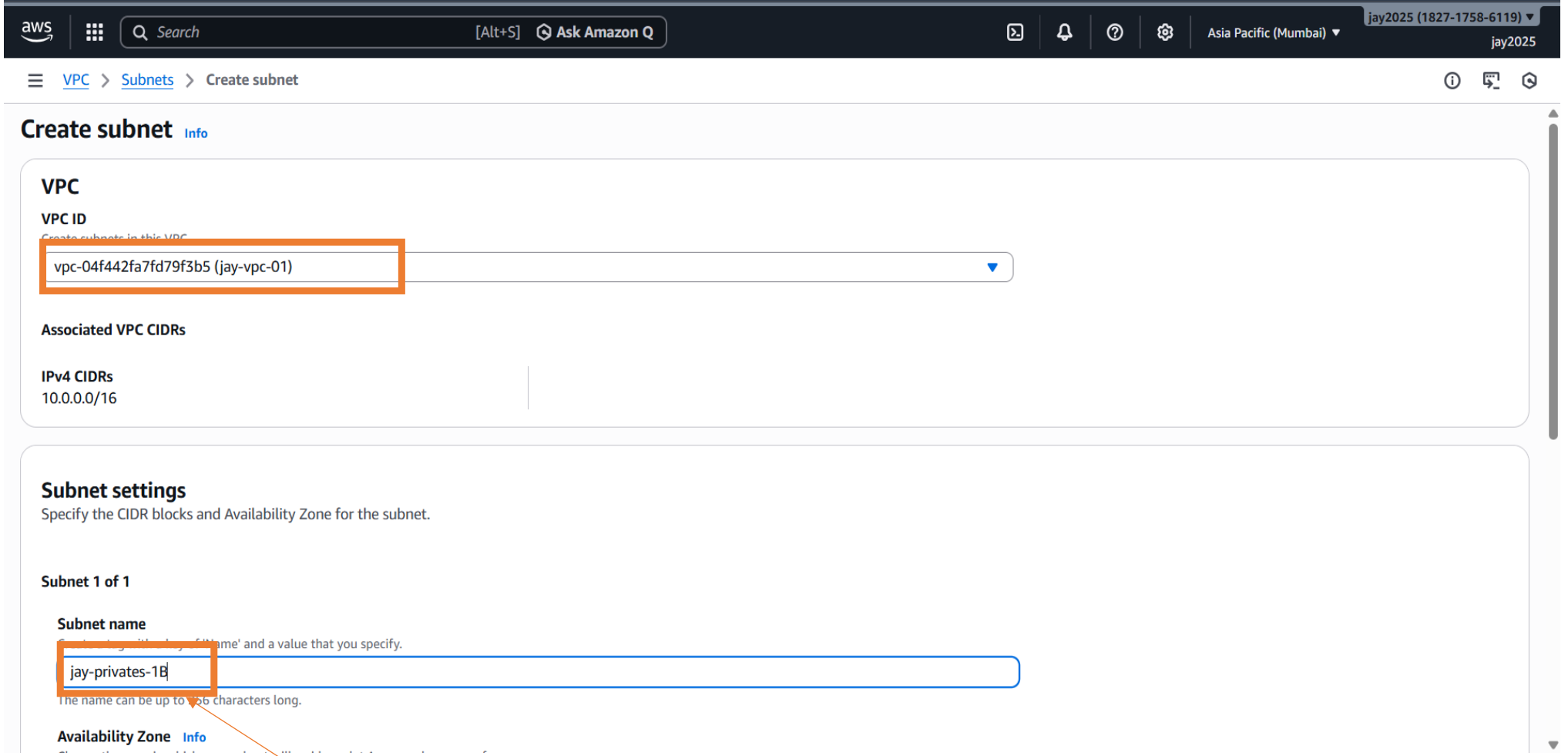
- Availability Zone: Select the AZ where the subnet will be created **ap-south-1a.**
- IPv4 CIDR Block: Define the IP range for the subnet **10.0.0.0/16**
- VPC Association: Attach the subnet to my project VPC **10.0.1.0/24**

Use

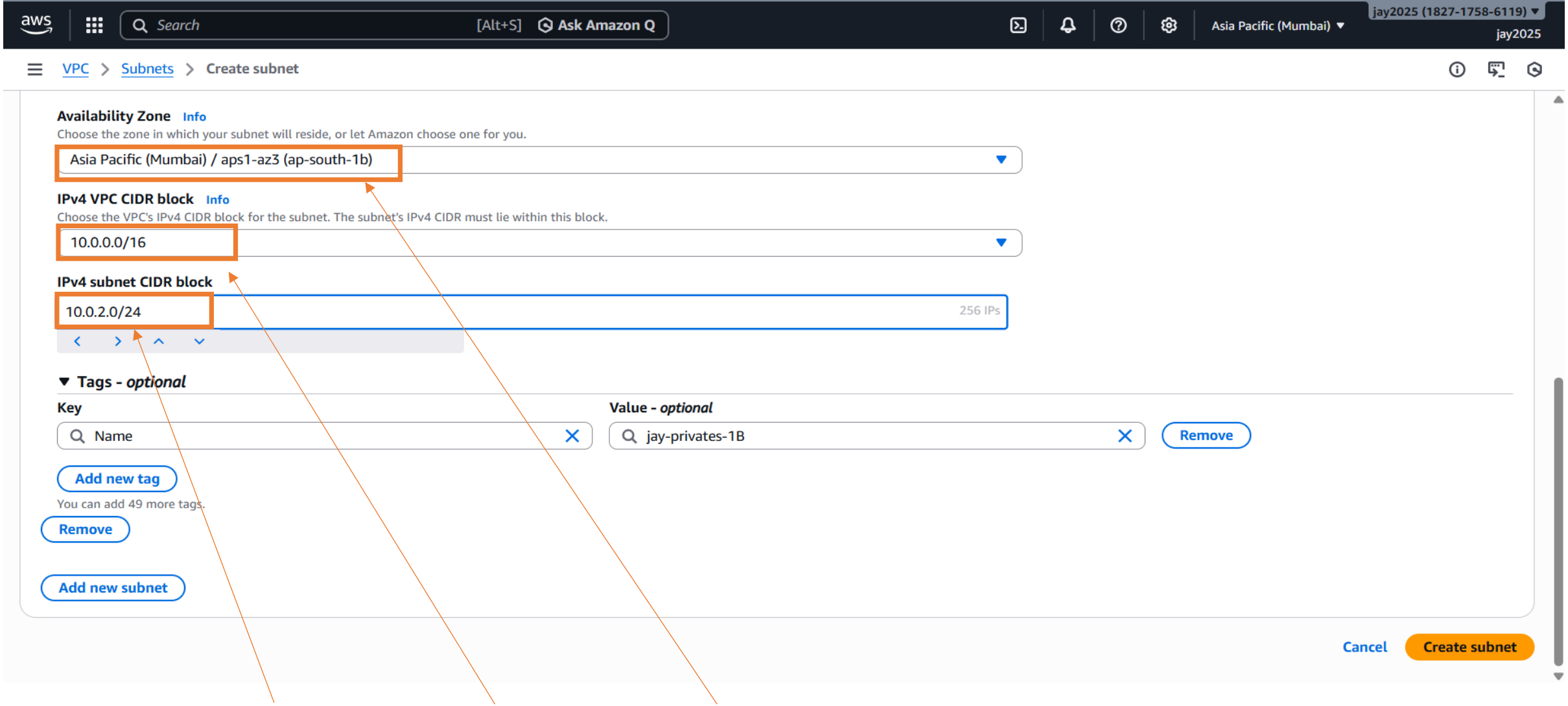
The public subnet is used to host resources (like EC2 instances) that need direct internet access. By enabling auto-assign public IPs, these resources can communicate with external services and users.



Repeat the same process for the private subnet with some changes.



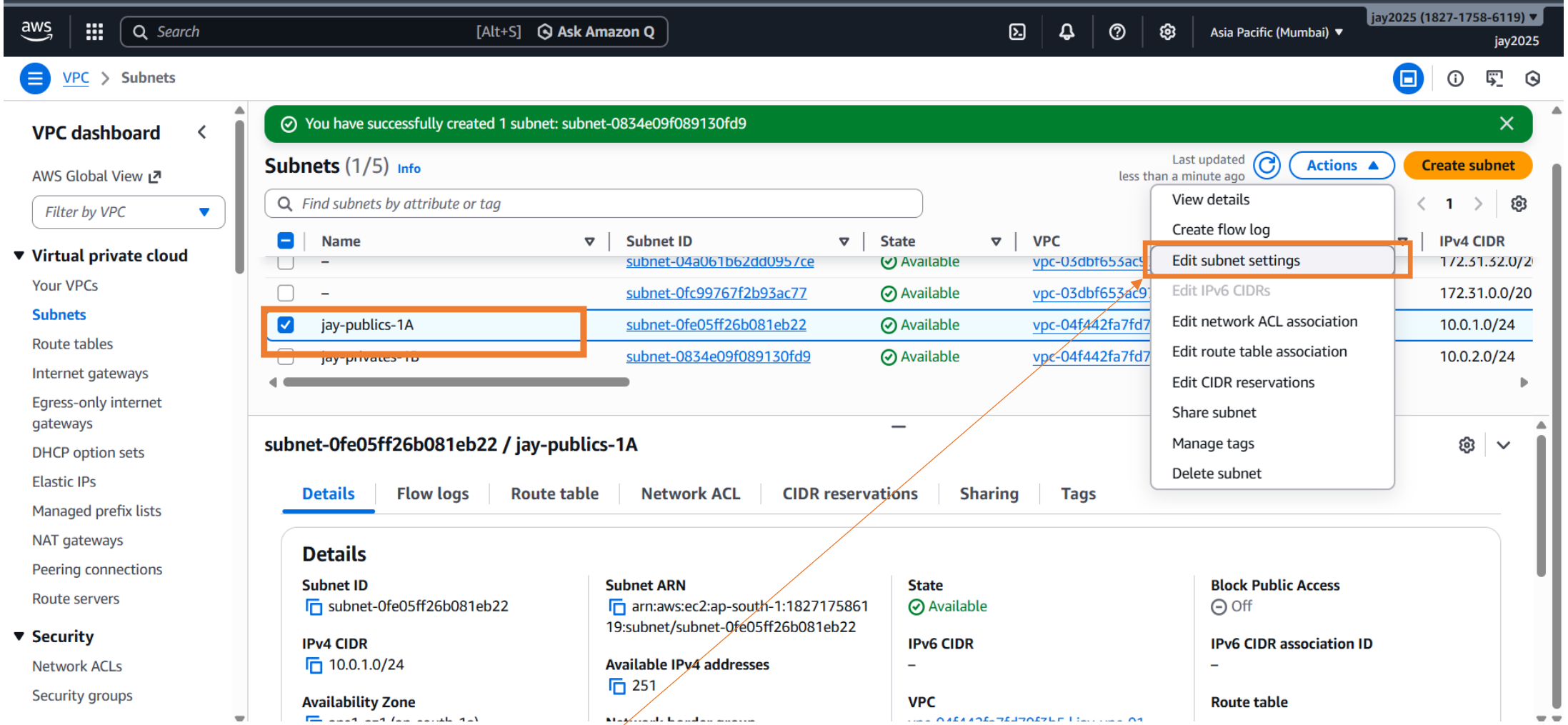
Assigning a Name tag **jay-private-1B** to indicate this subnet is isolated for backend resources.



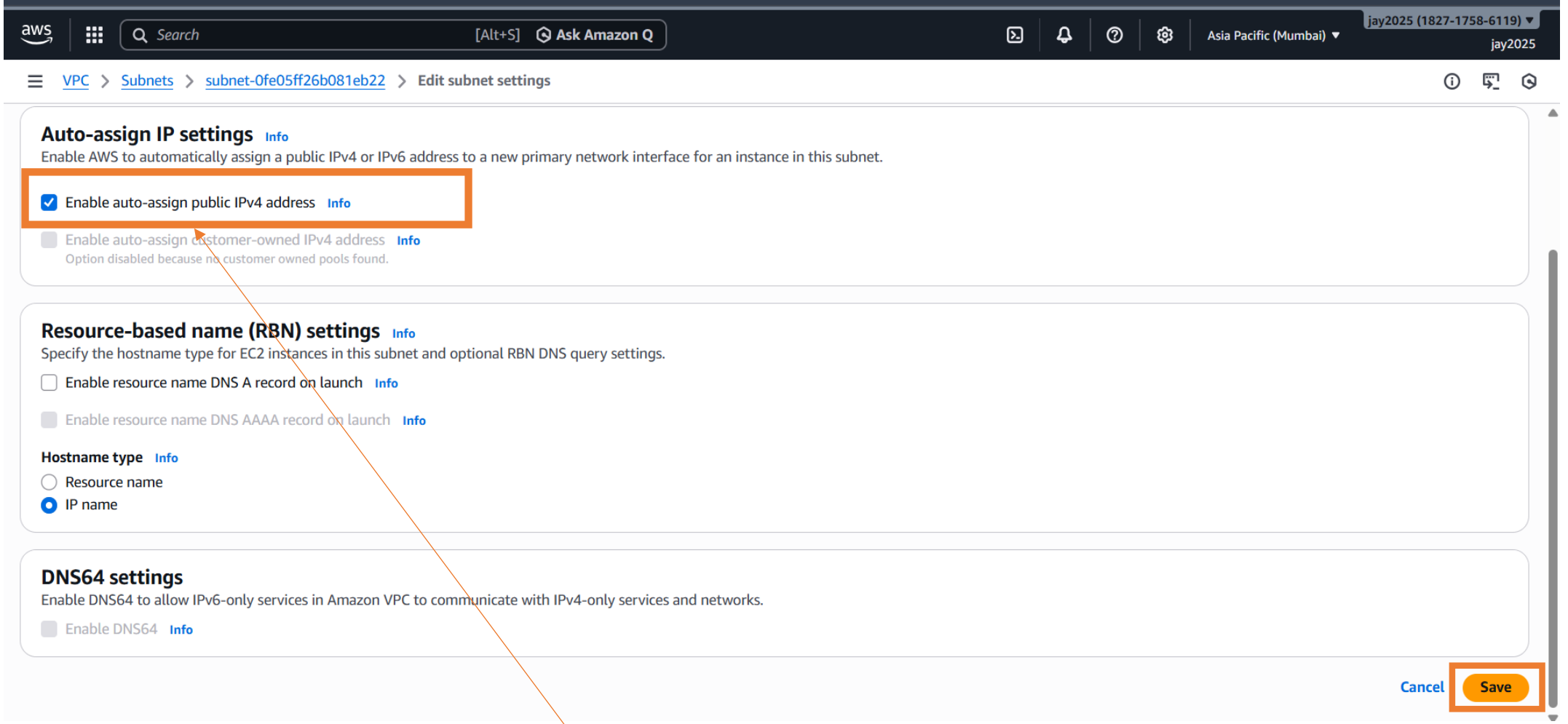
"A private subnet is created in Availability **Zone ap-south-1B** with IPv4 CIDR block **10.0.2.0/24,** associated to my vpc."

Use

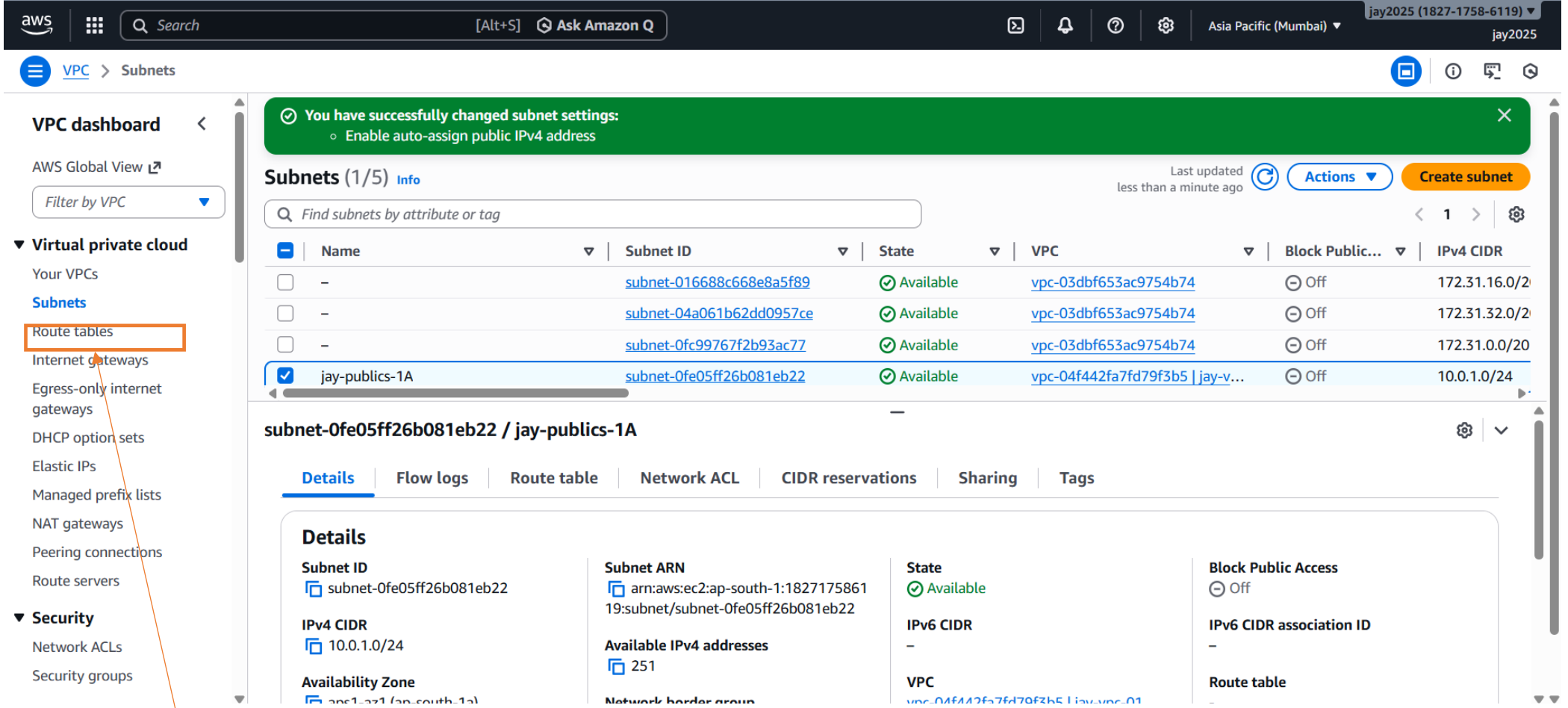
These resources remain isolated from direct internet access, ensuring higher security while still being able to reach the internet through a NAT Gateway if required



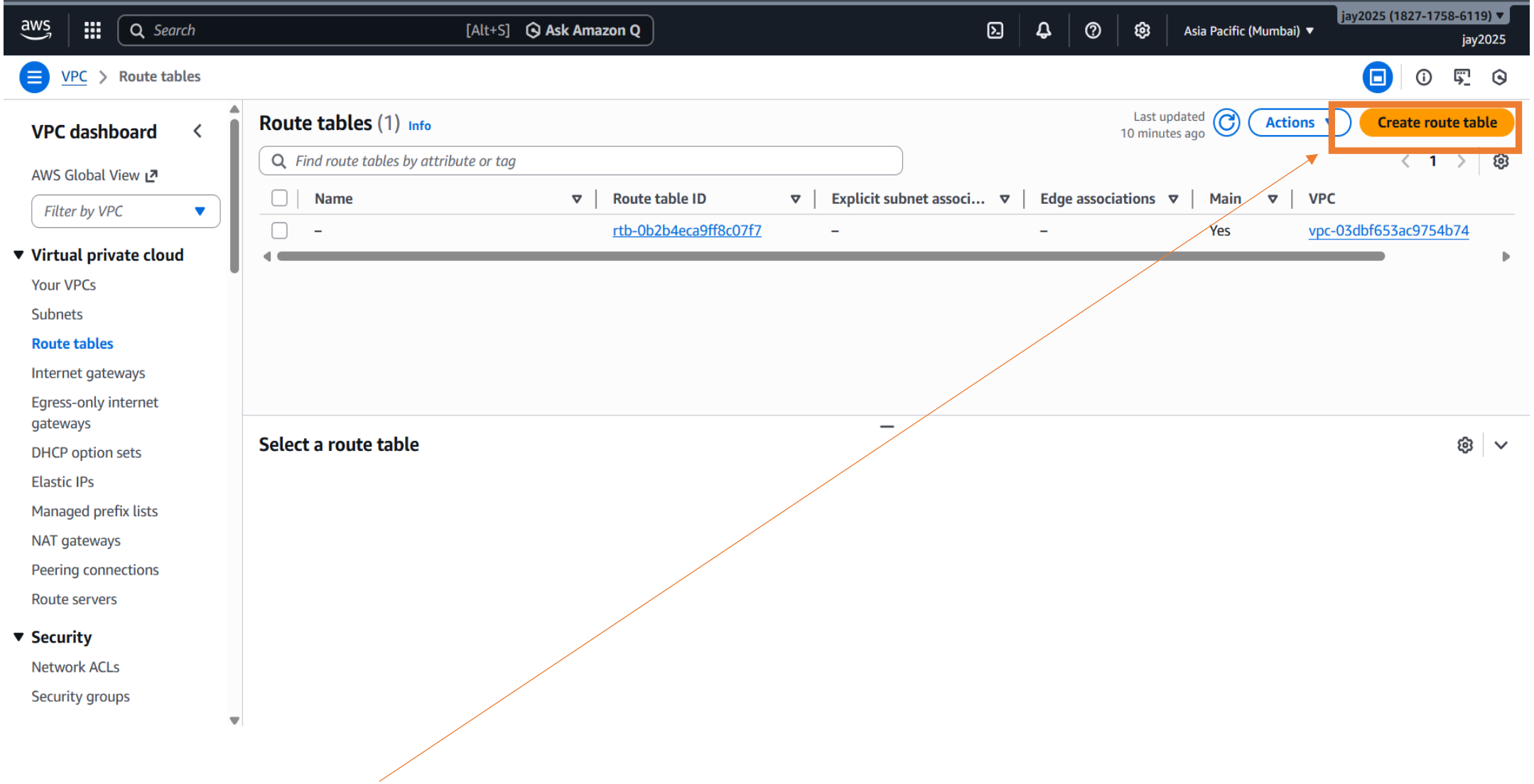
Going to subnet setting in public subnet



Enable this option so EC2 instances launched here automatically get a public IP.



Click on Route tables

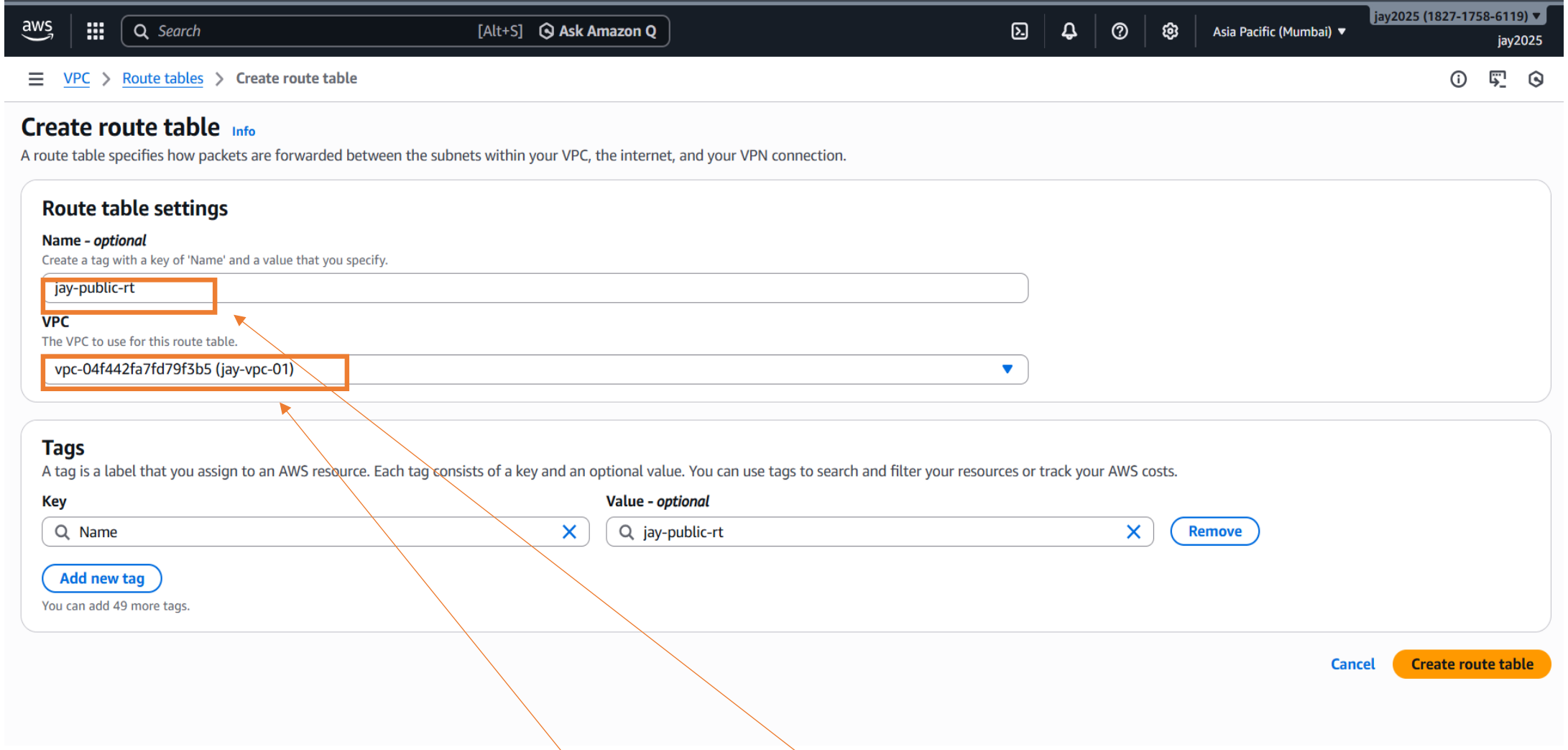


Creat Route table

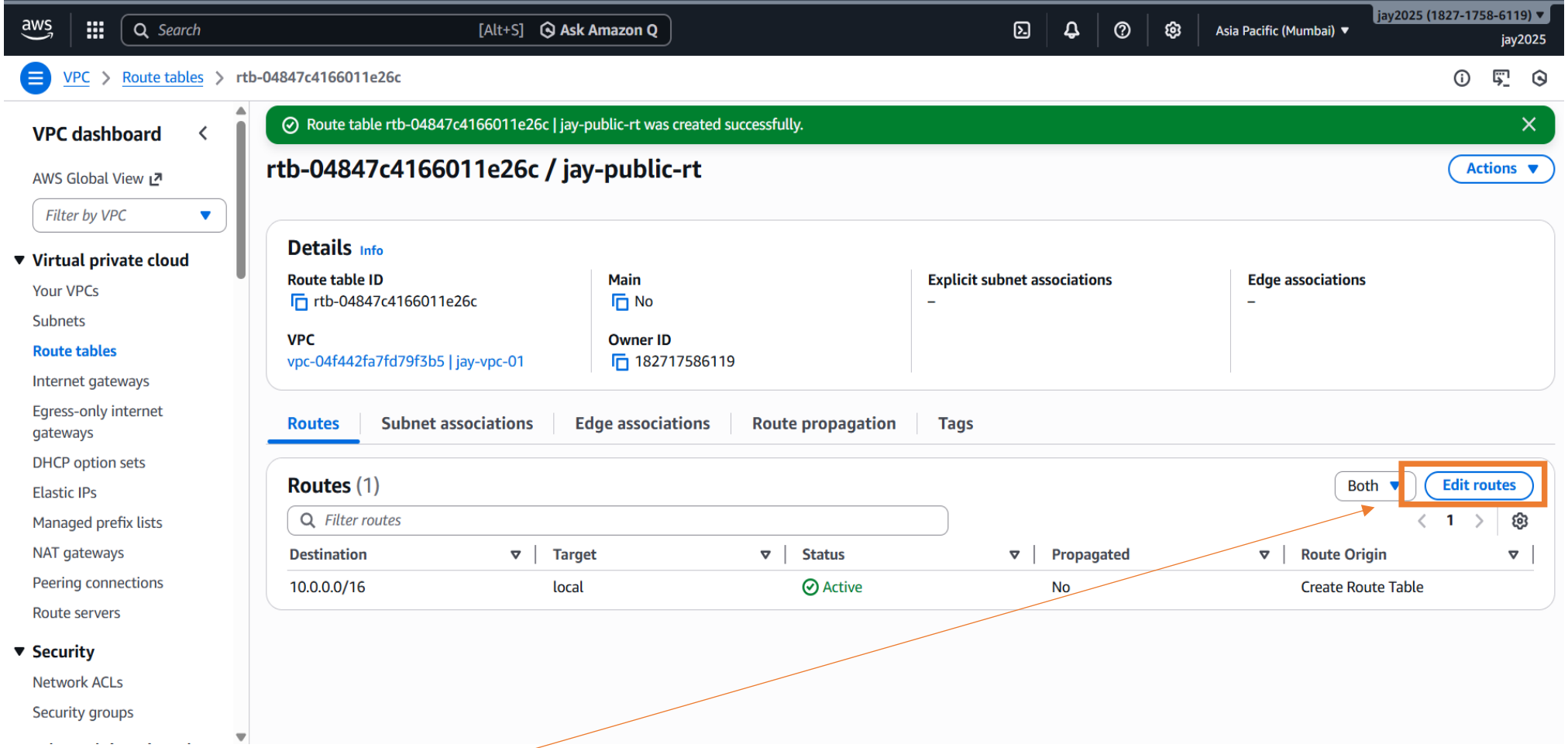
- Definition: Sets of rules that determine how traffic flows within my VPC.

Why Use

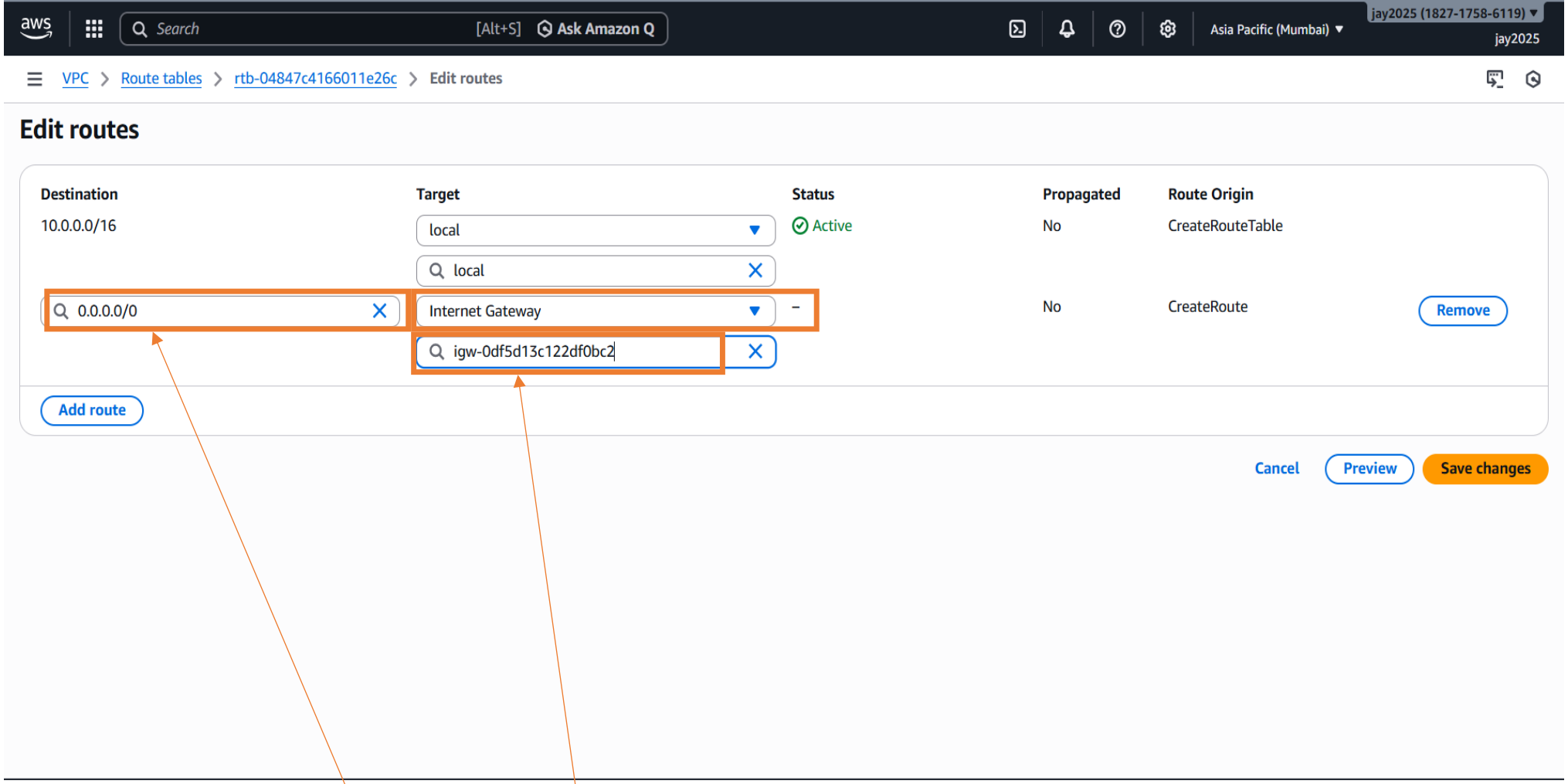
- Public route table → sends traffic to IGW.
- Private route table → sends traffic to NAT Gateway.



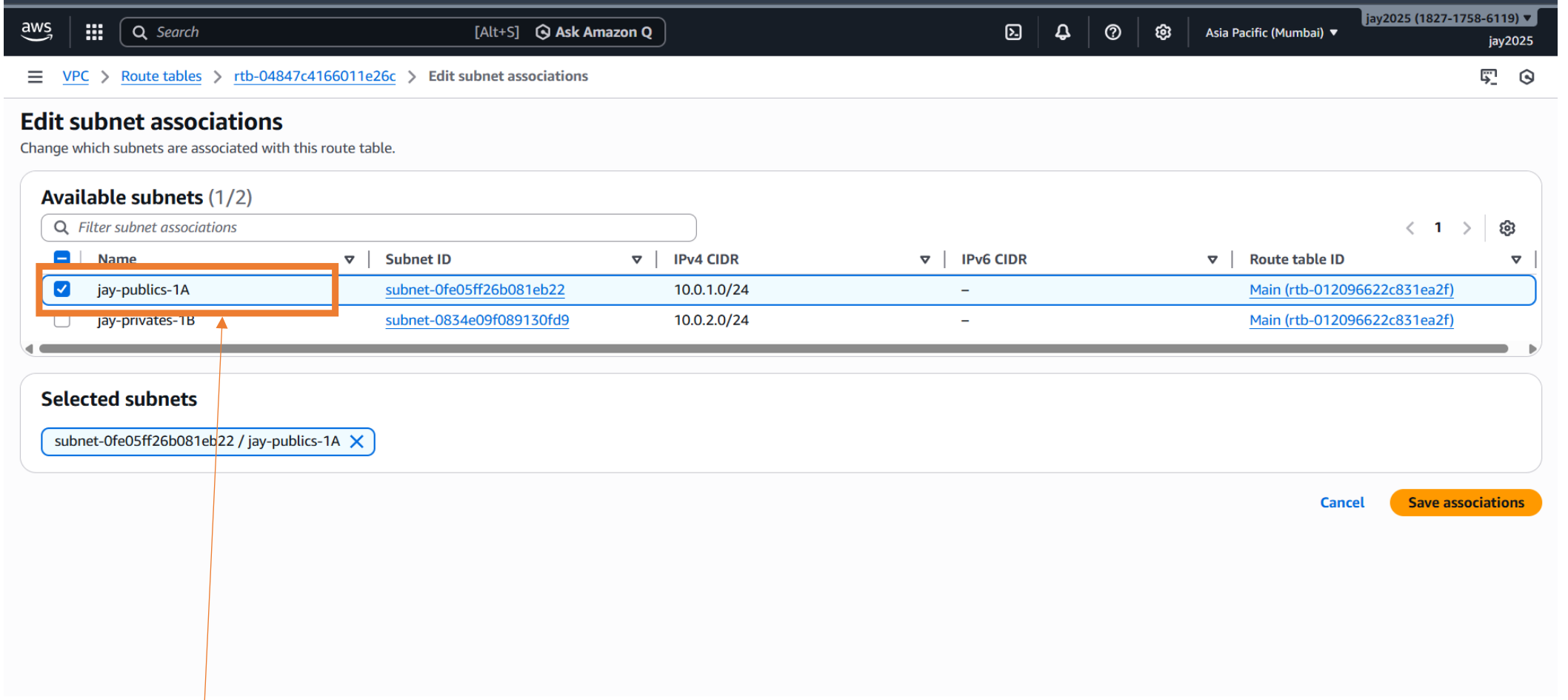
Naming the route table **jay-public-rt** and connect to my vpc



Now going to edit the route.

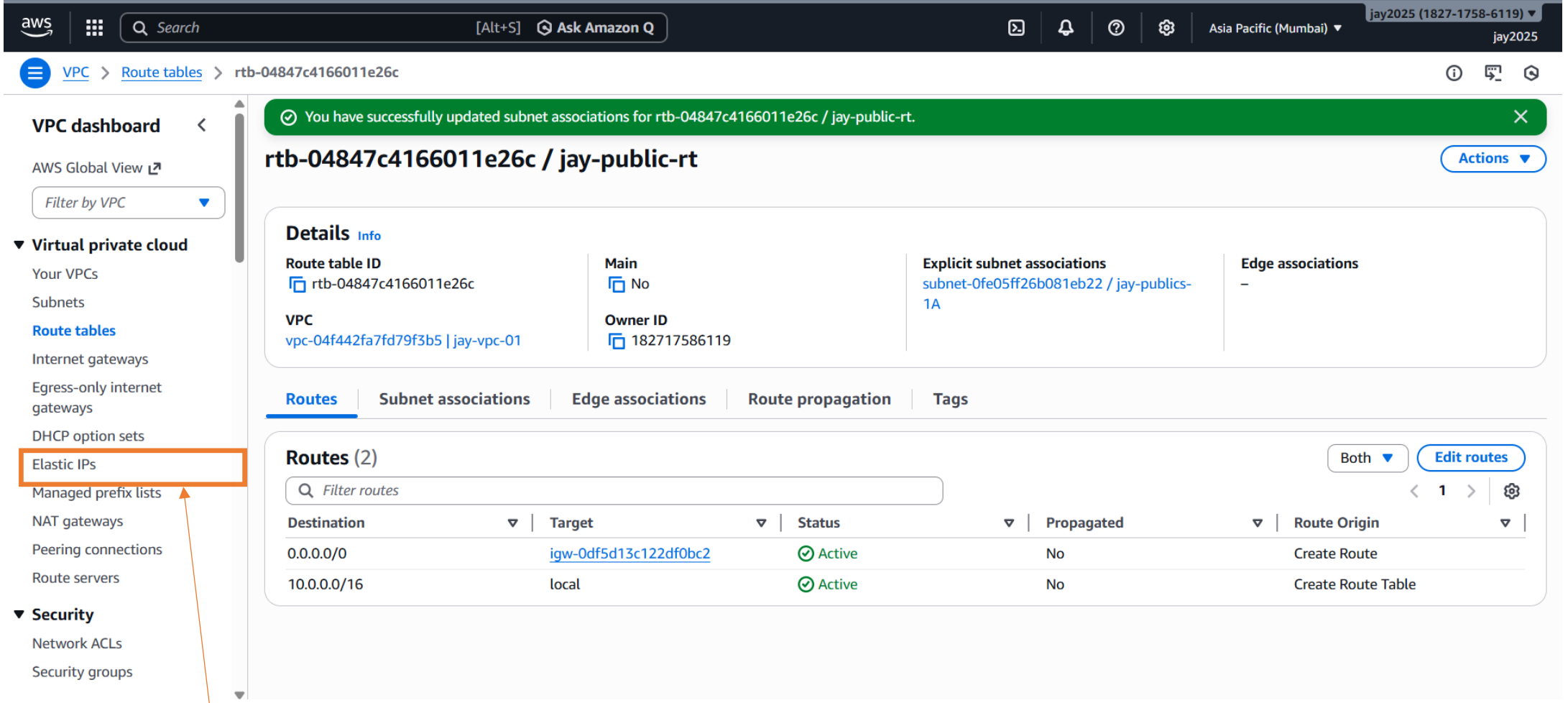


The public route table is edited to send all outbound traffic (0.0.0.0/0 means all IPv4) from the public subnet to the Internet Gateway (jay-ig). This enables EC2 instances in the public subnet to access the internet.”

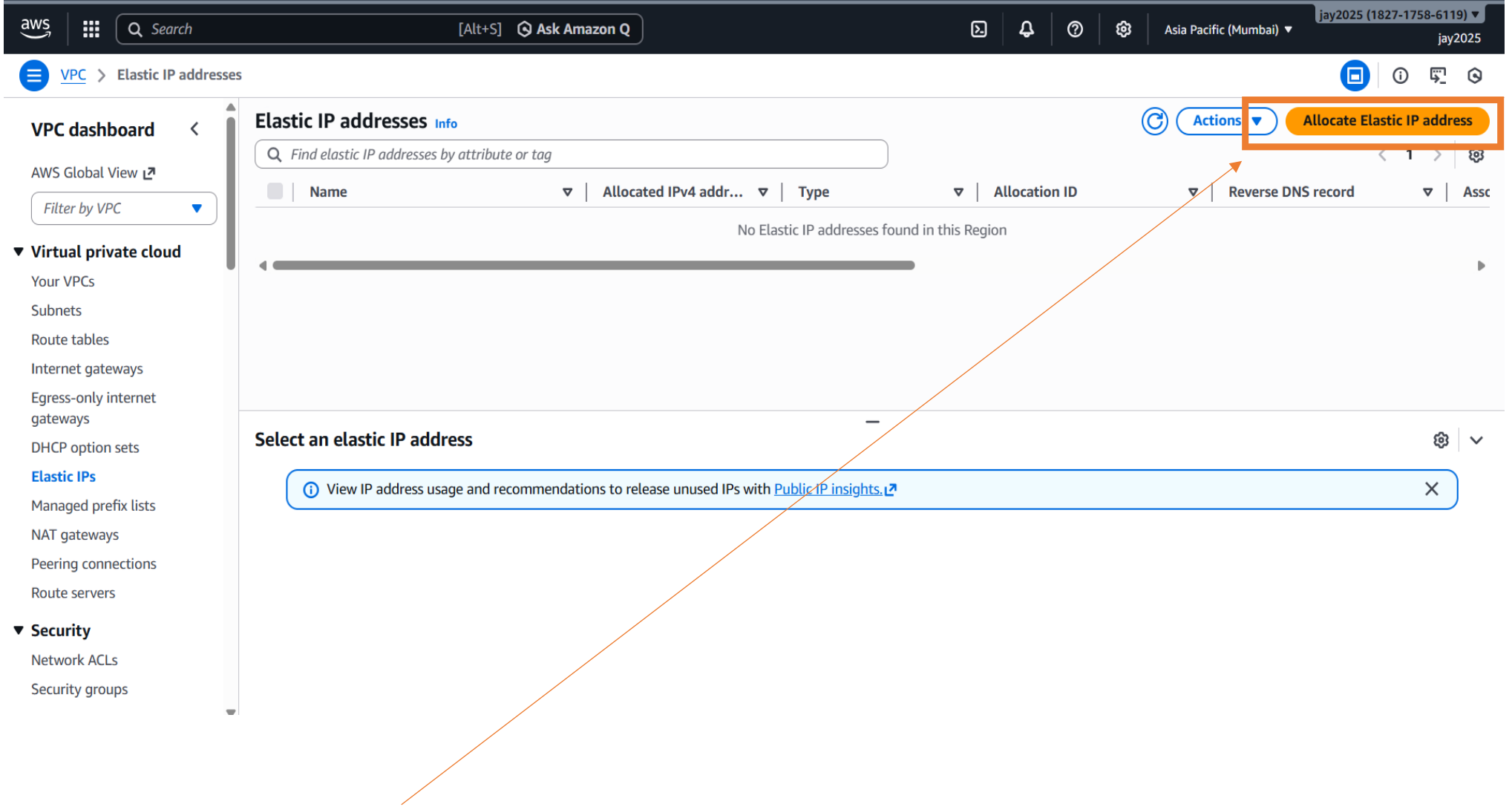


Now going into Edit associations

The public subnet is associated with the public route table, enabling traffic to flow through the Internet Gateway for internet access.

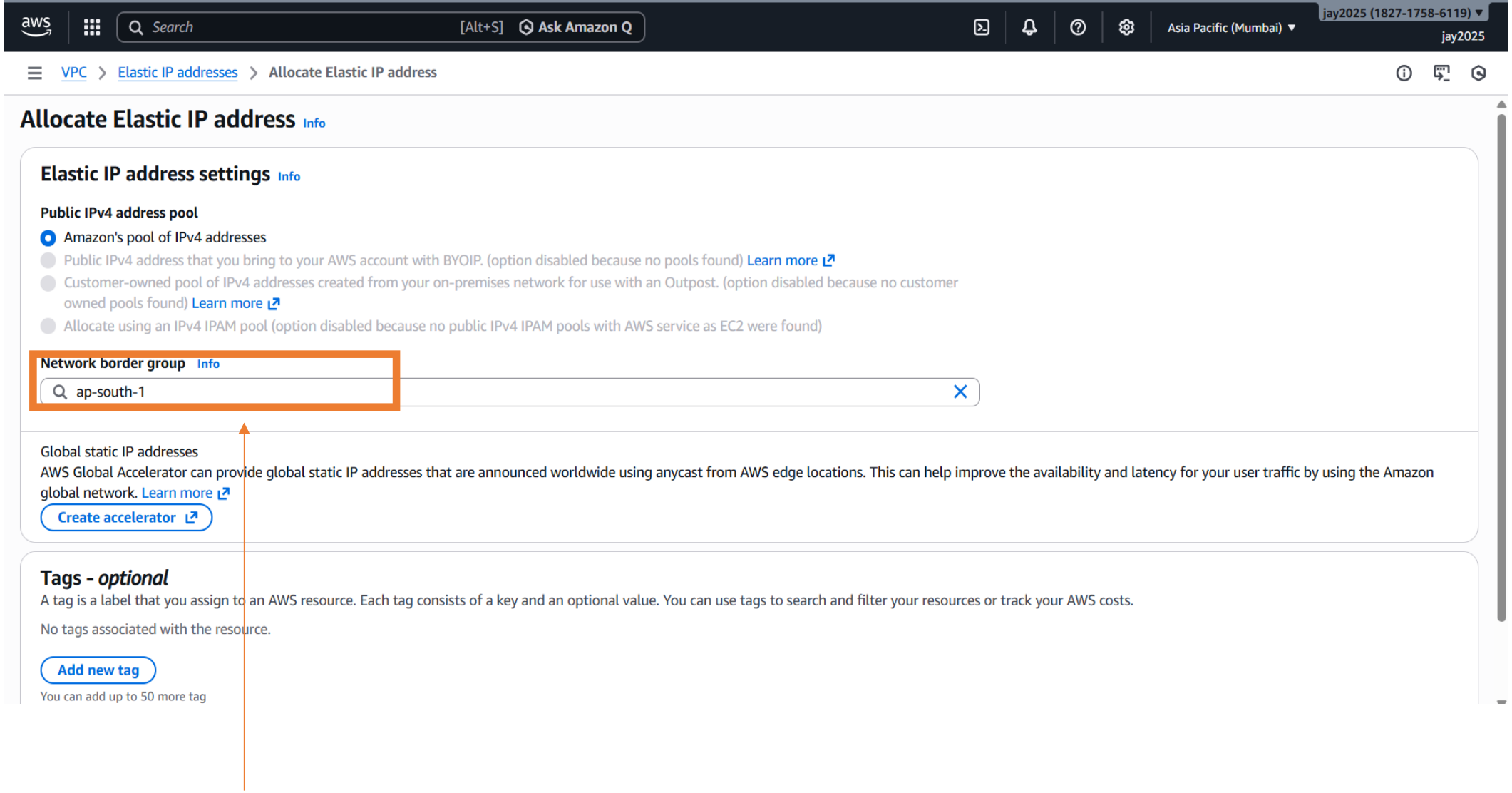


Click on Elastic IPs

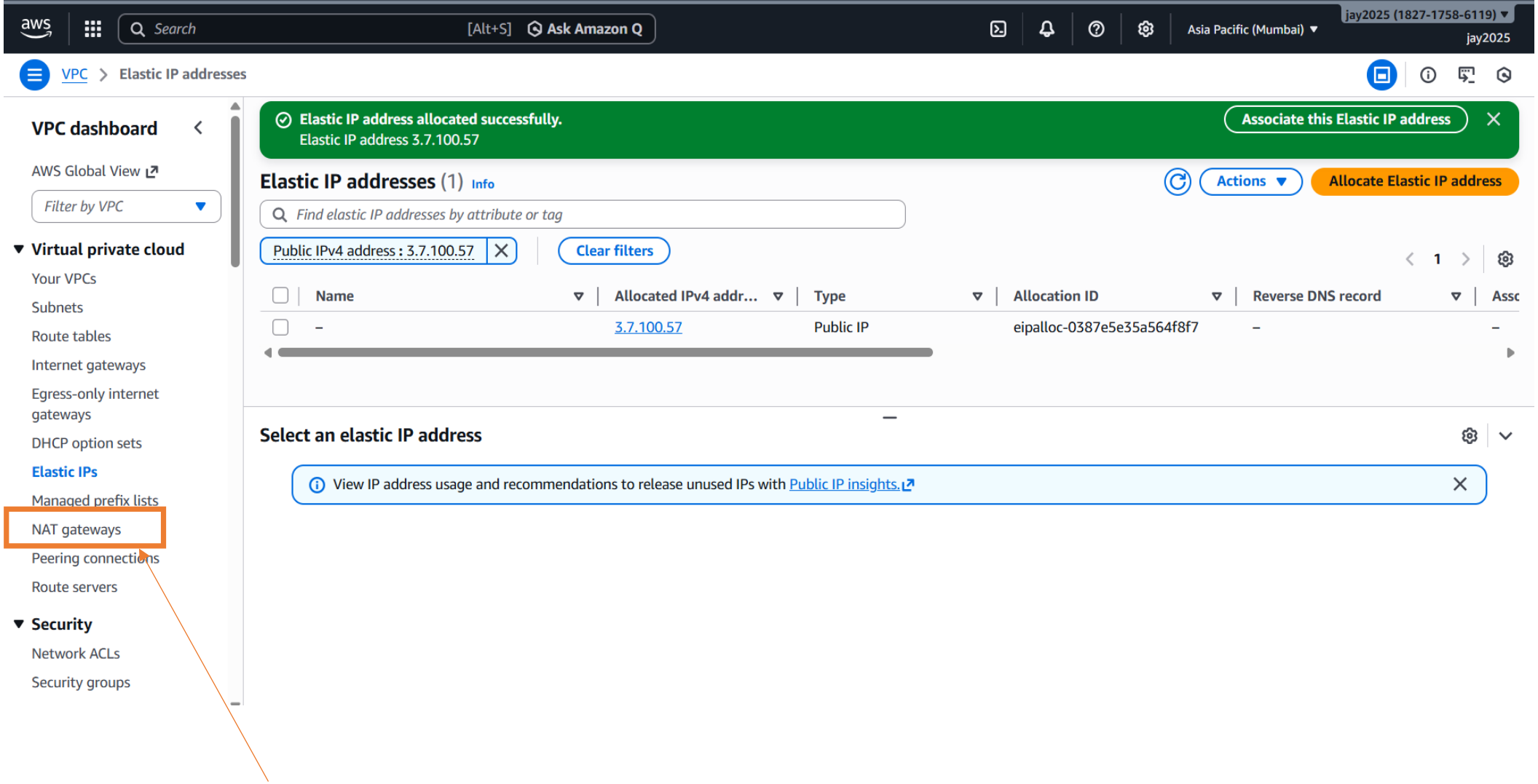


Click on Allocate Elastic IP addrees

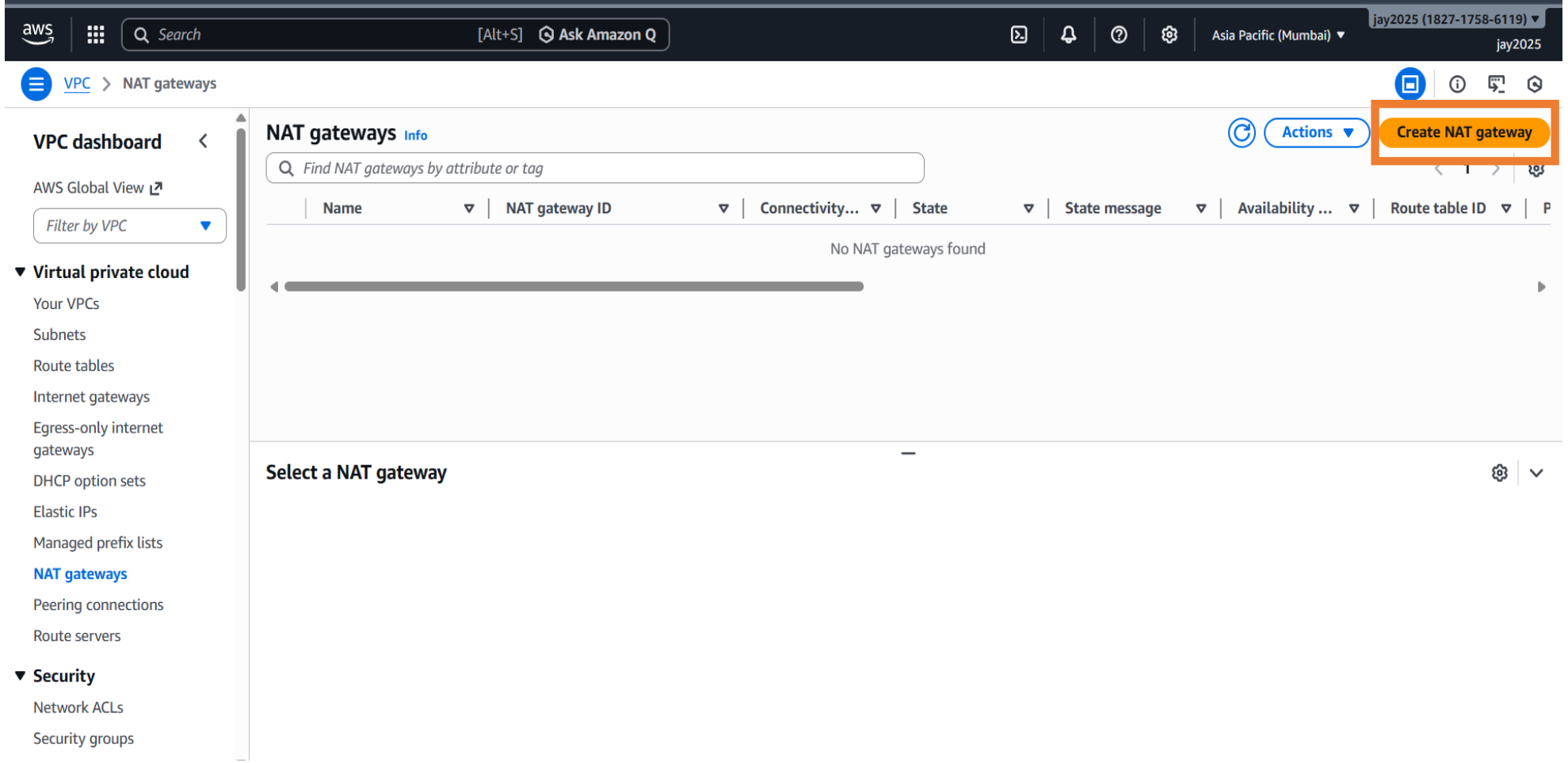
An Elastic IP is allocated to provide a static public IPv4 address. This address is attached to the NAT Gateway, ensuring consistent outbound internet access for private subnet resources.



Click on Available zone

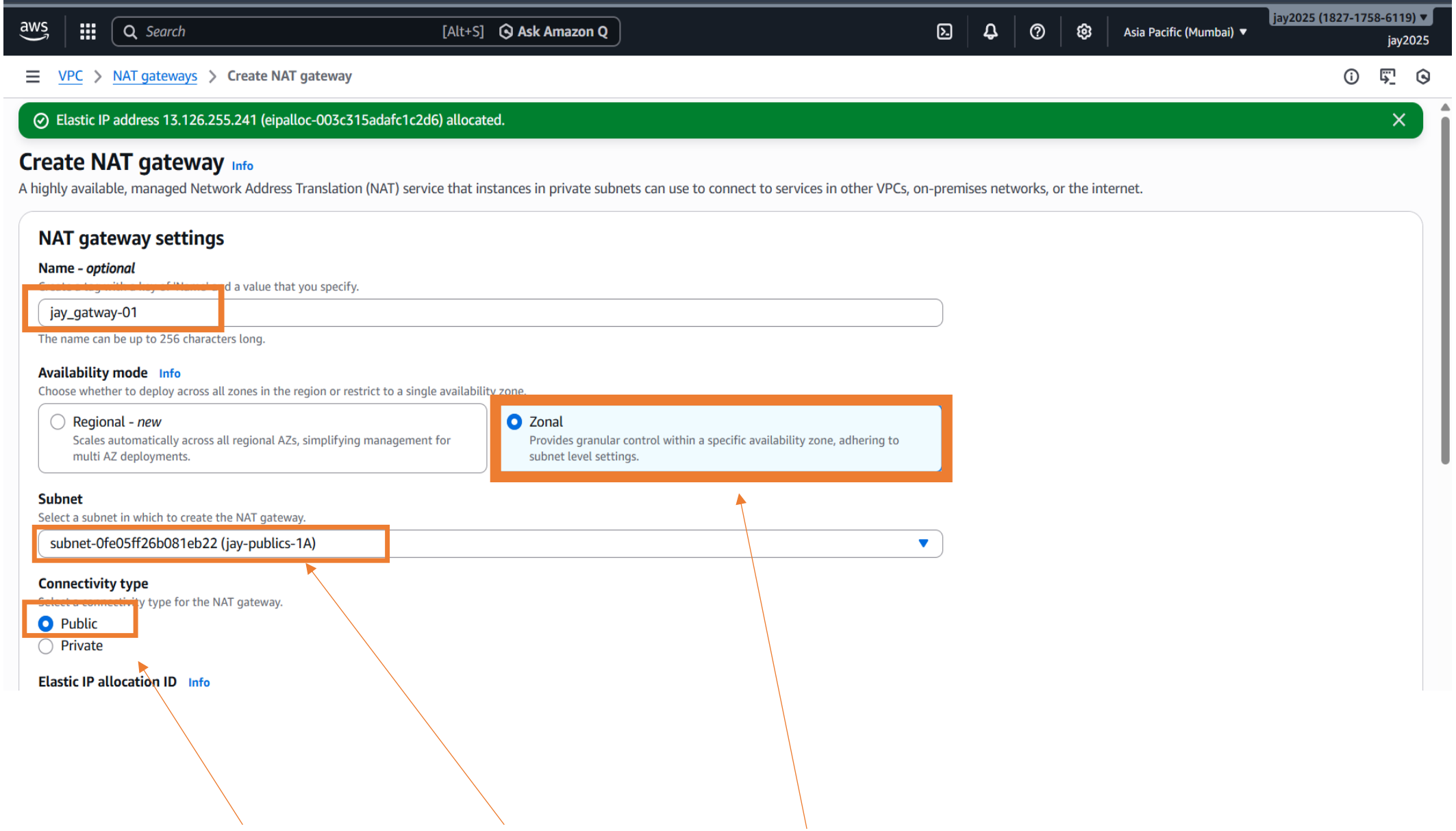


Click on NAT gateway

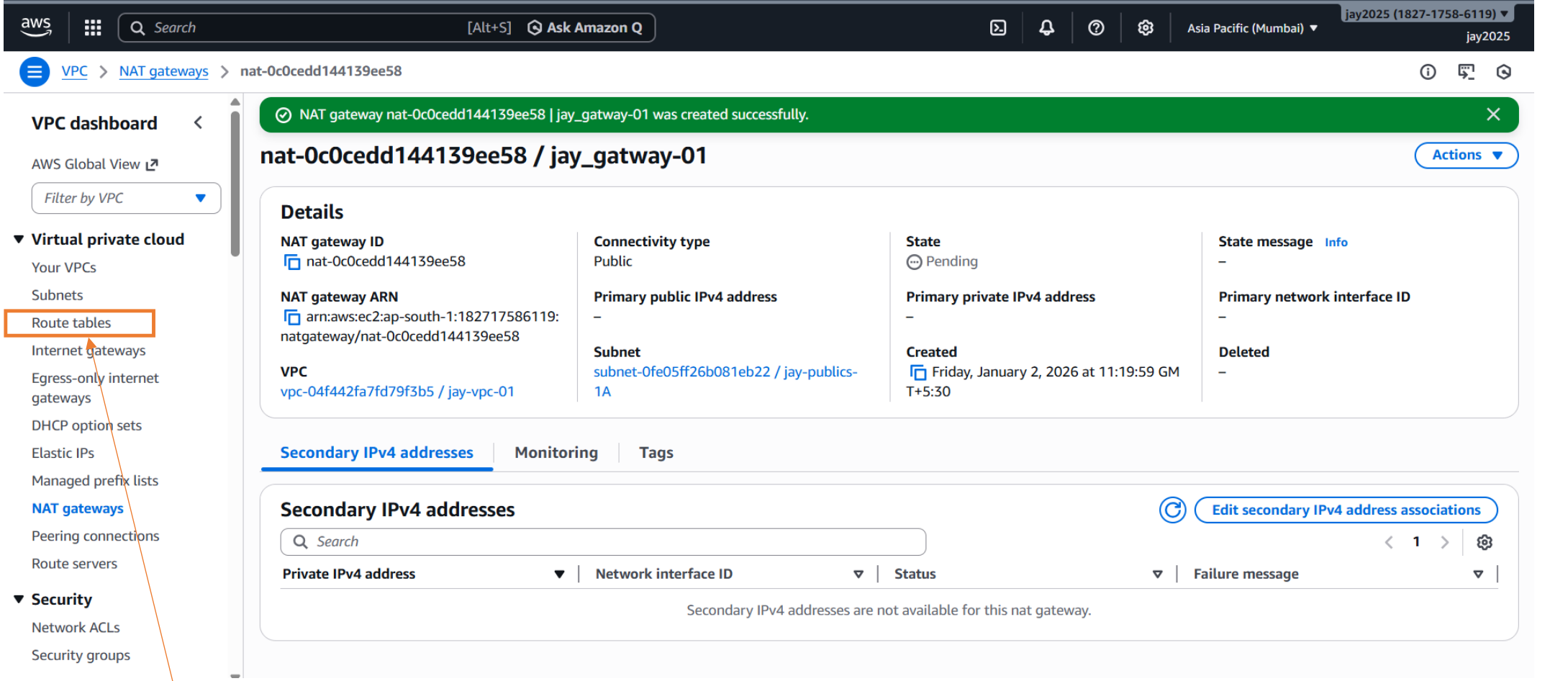


Creat NAT gateway

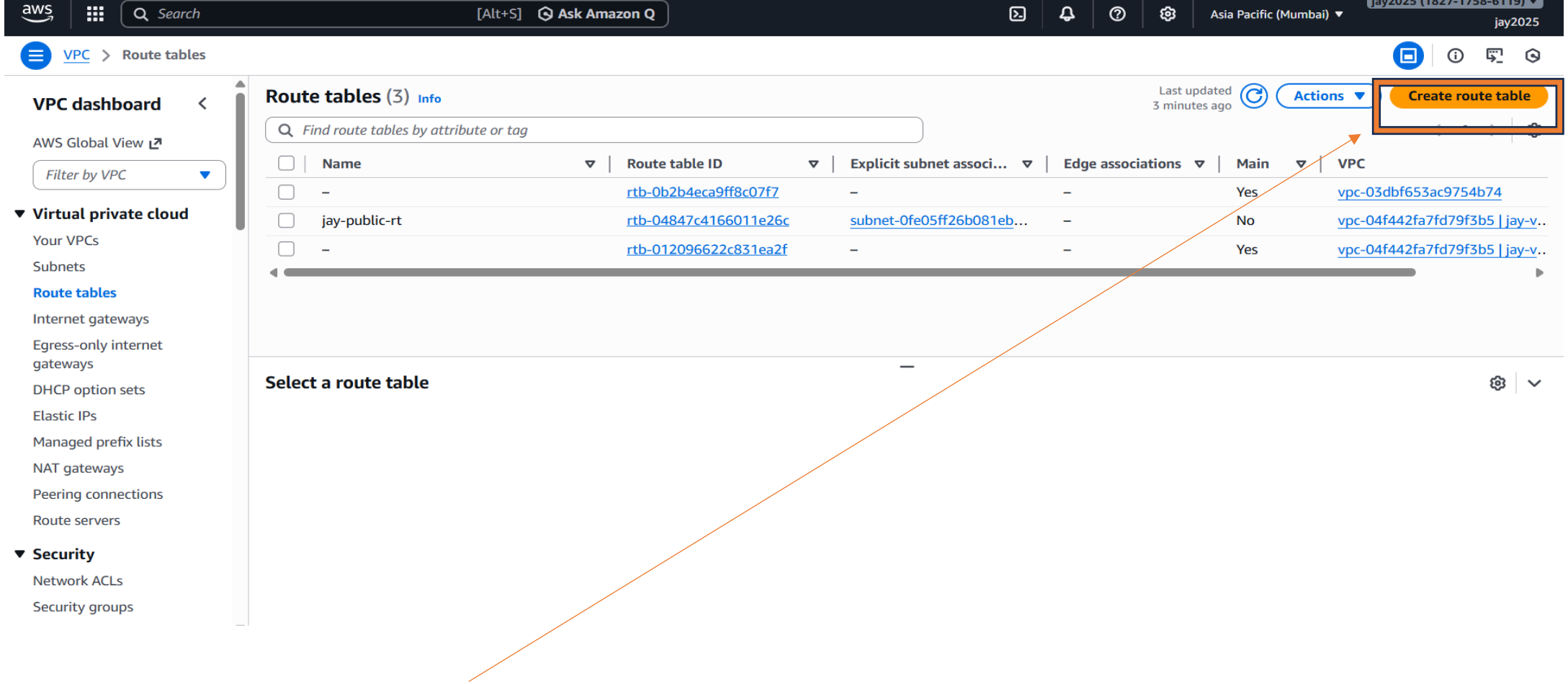
A NAT Gateway (Network Address Translation Gateway) is a managed AWS service that allows instances in private subnets to connect to the internet or other AWS services, while preventing unsolicited inbound connections from the internet.



NAT Gateway is created in a public subnet within Availability Zone ap-soth-1a , using connectivity type Public and attached to Elastic IP . This setup enables secure outbound internet access for private subnet resources.

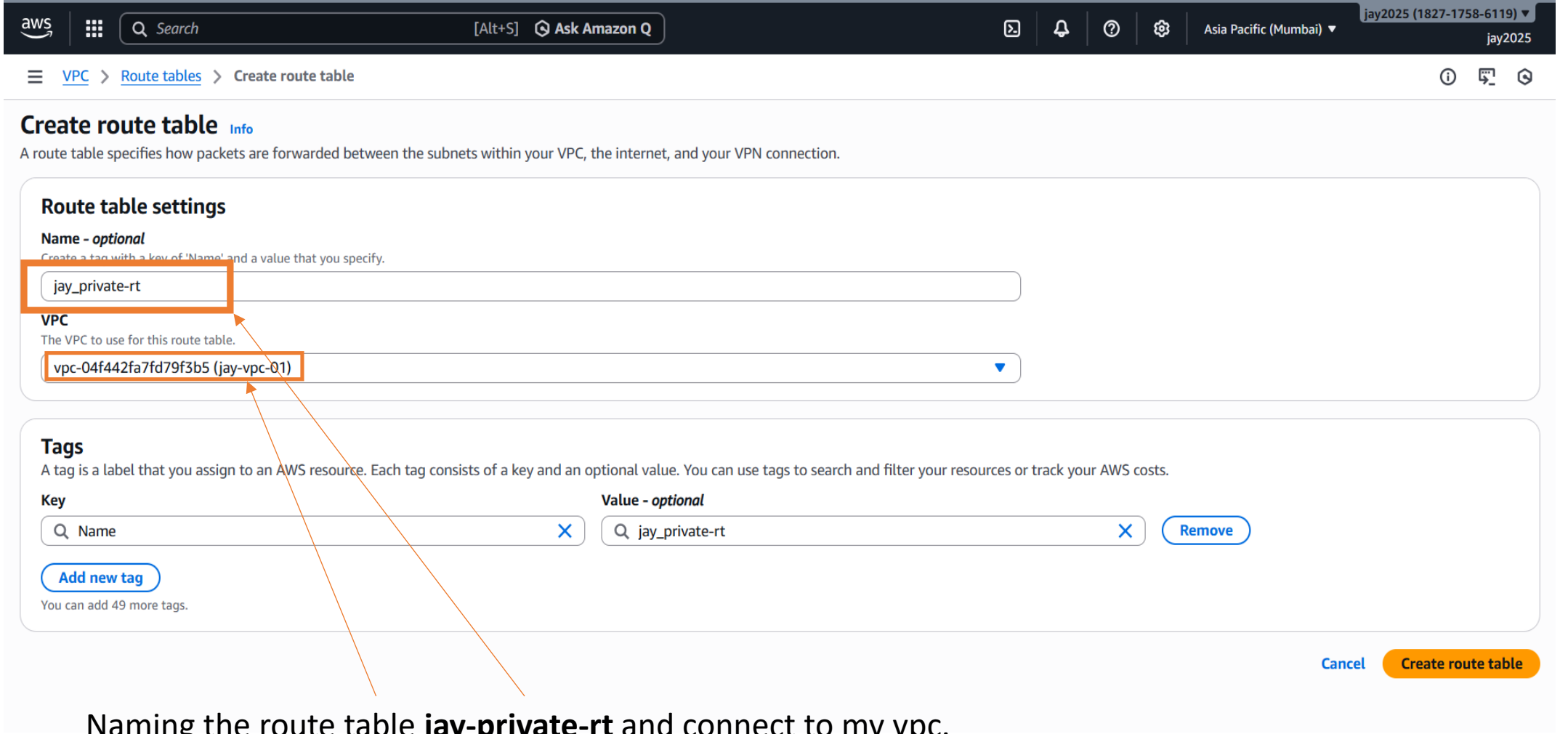


Click on Route tables

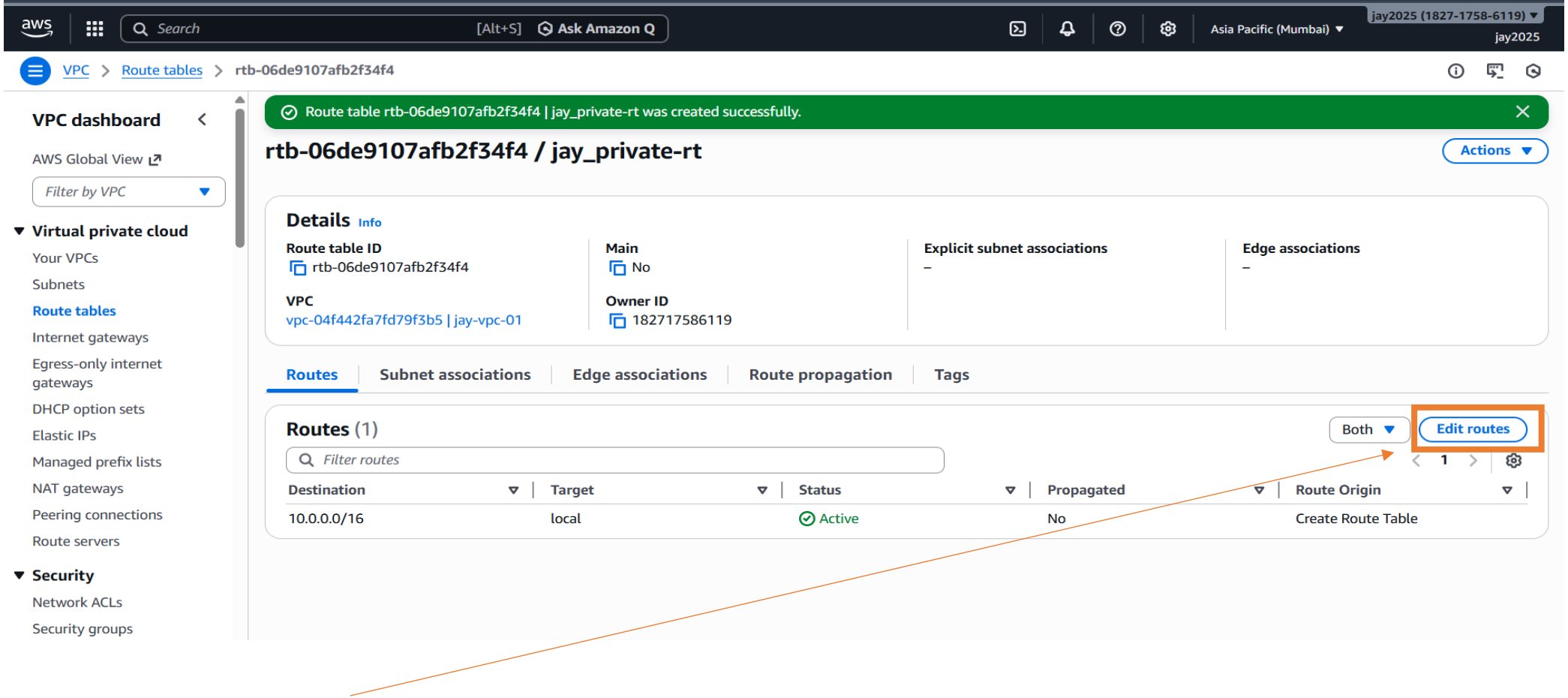


Creat private route table.

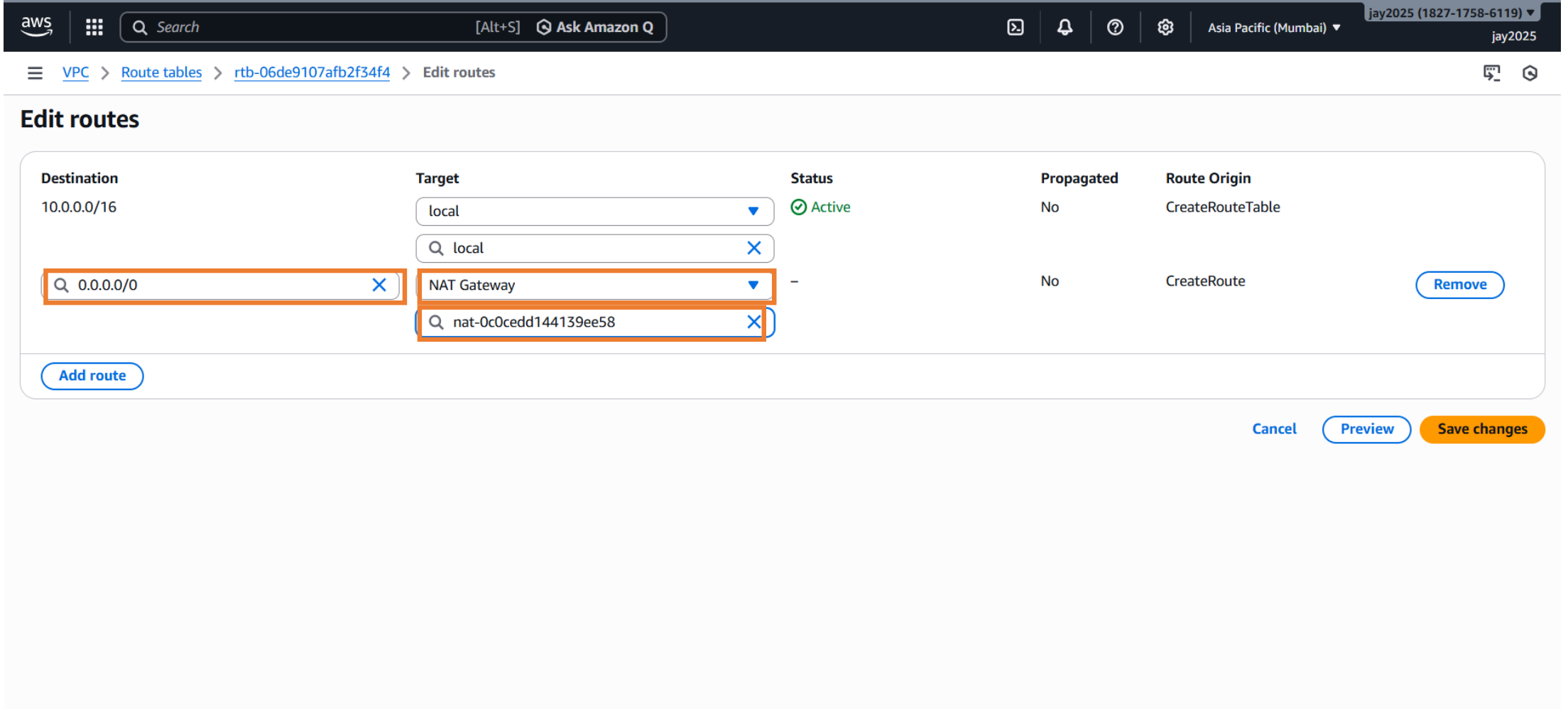
The private route table is configured to send all outbound traffic from the private subnet to the NAT Gateway. This allows private subnet instances to securely access the internet while remaining isolated from inbound connections.



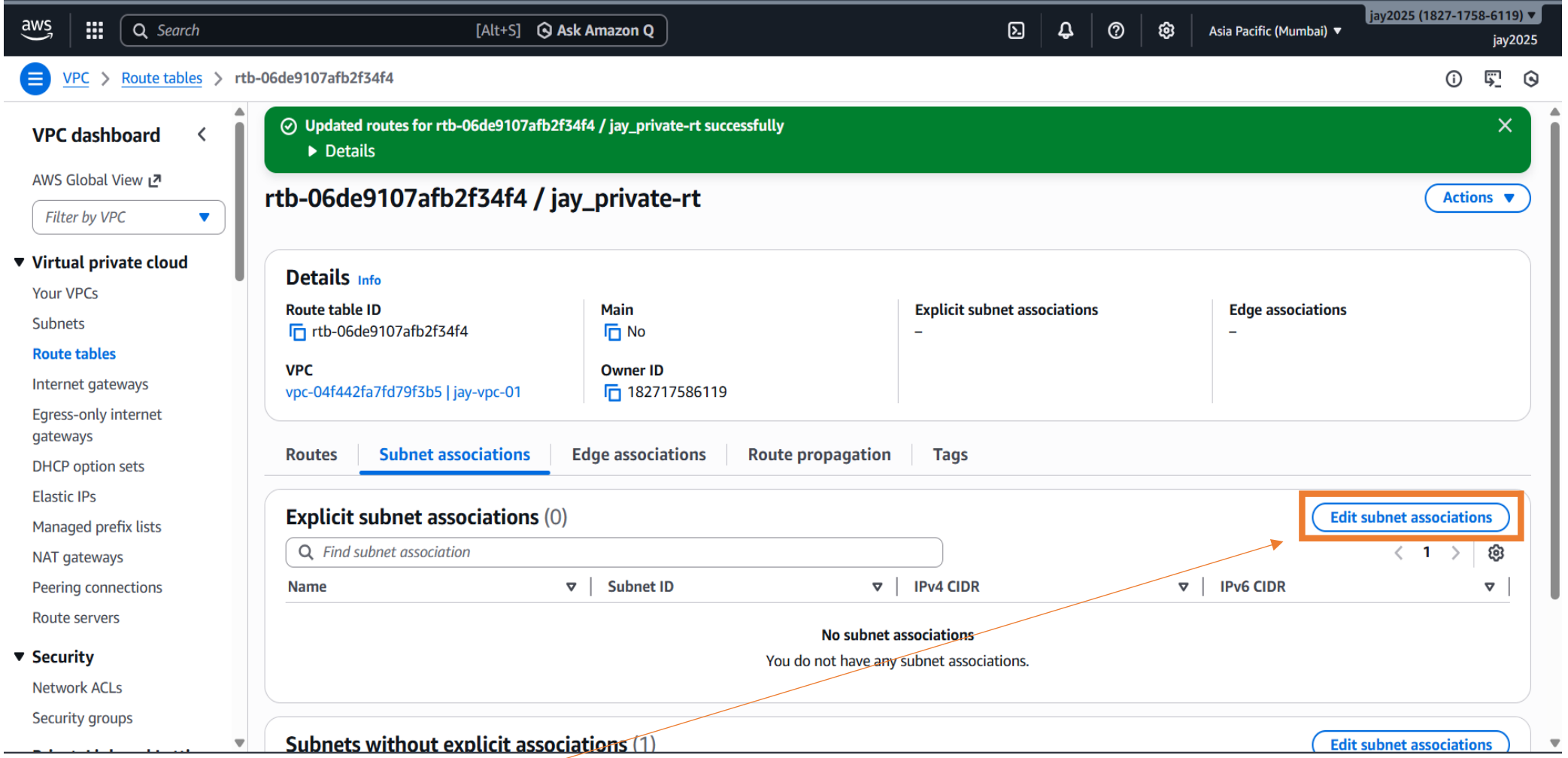
Naming the route table **jay-private-rt** and connect to my vpc.



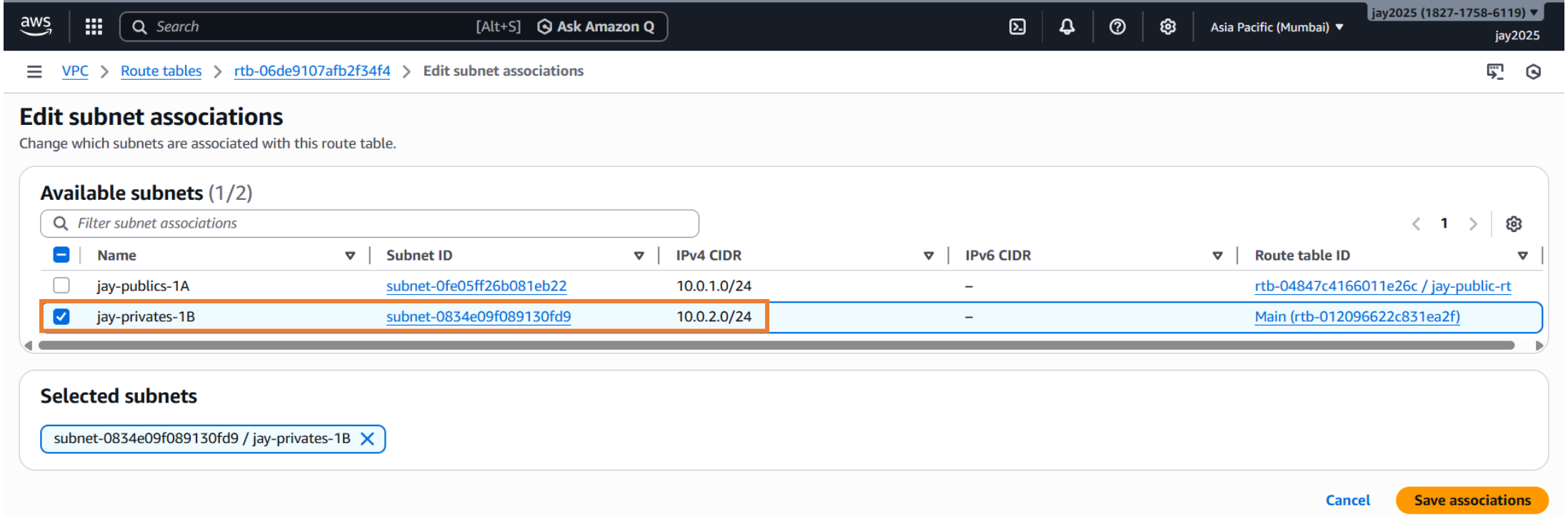
Now going to edit routes.



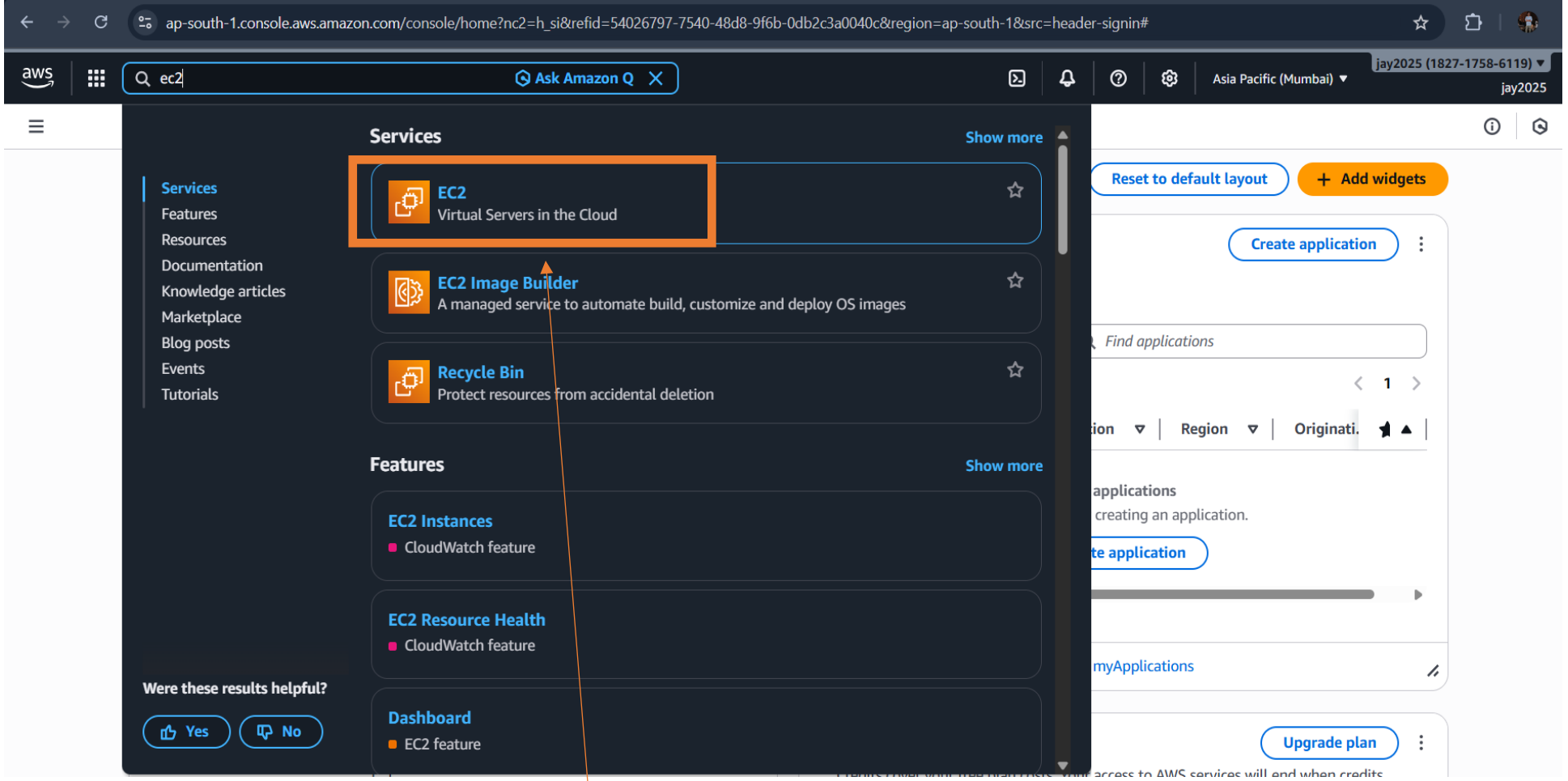
The private route table is edited to send all outbound traffic (0.0.0.0/0) from the private subnet to the NAT Gateway. This allows private subnet instances to securely access the internet while remaining isolated from inbound connections.



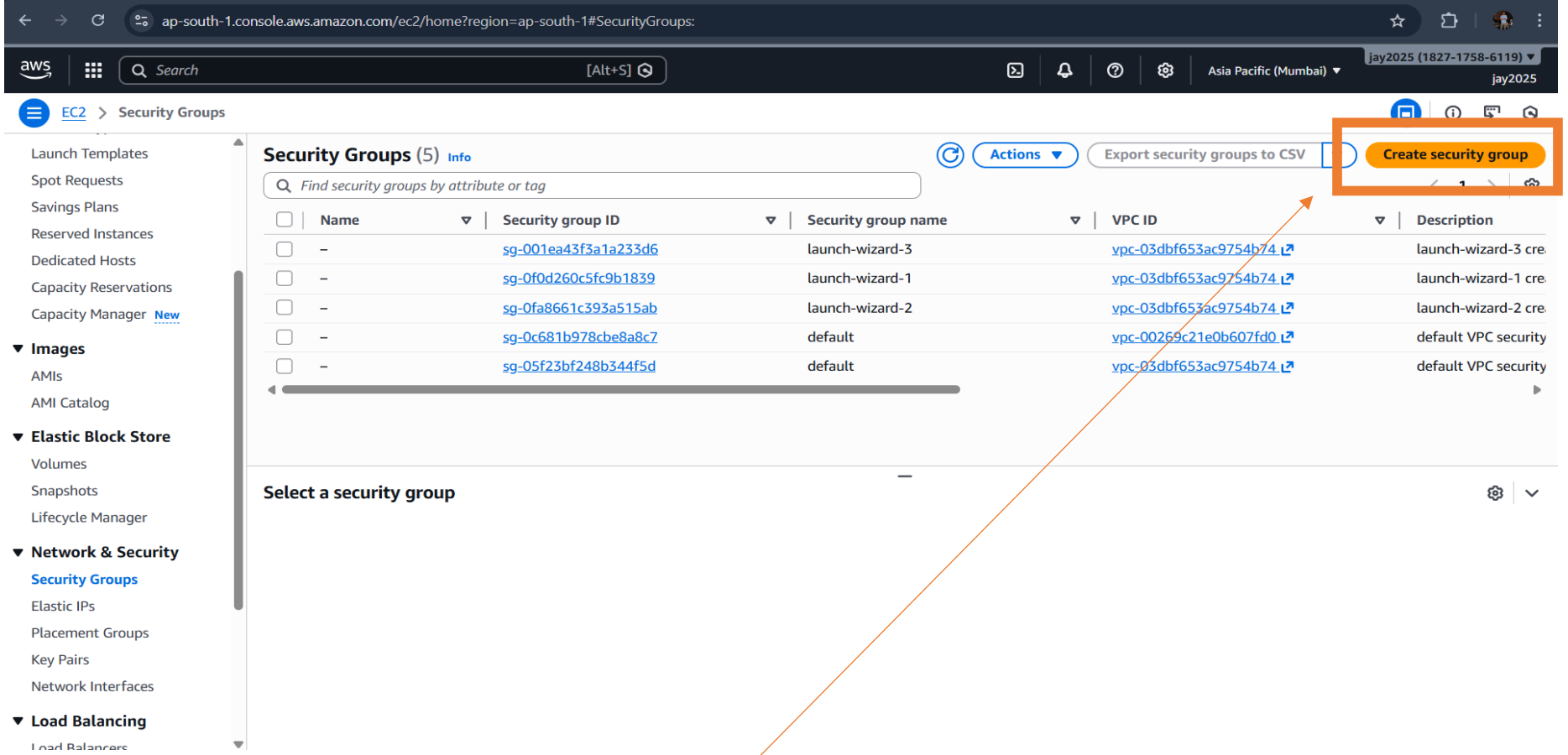
Now edit association



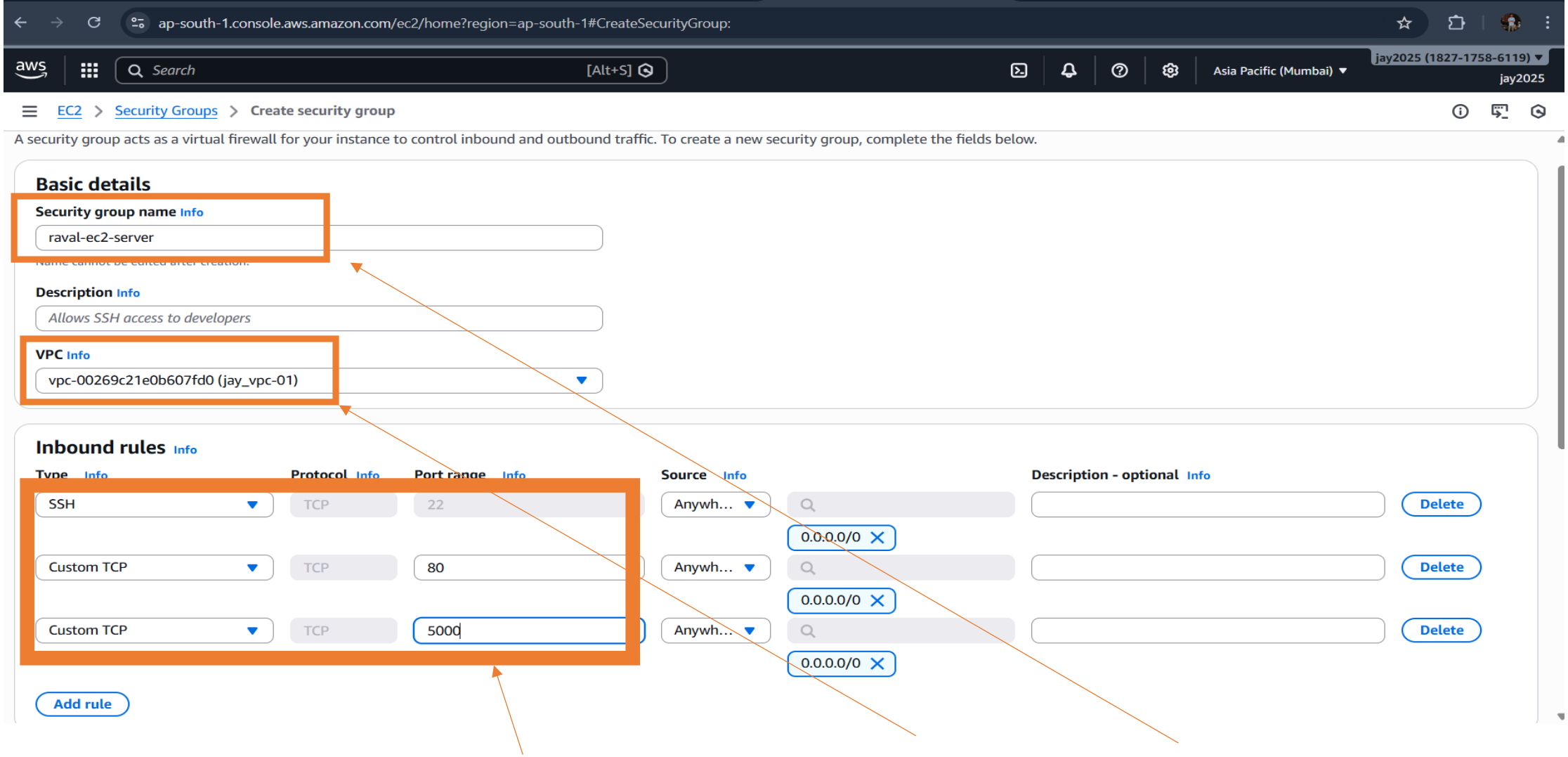
The private subnet is associated with the private route table Saving this association ensures that all traffic from the private subnet is routed securely through the NAT Gateway.



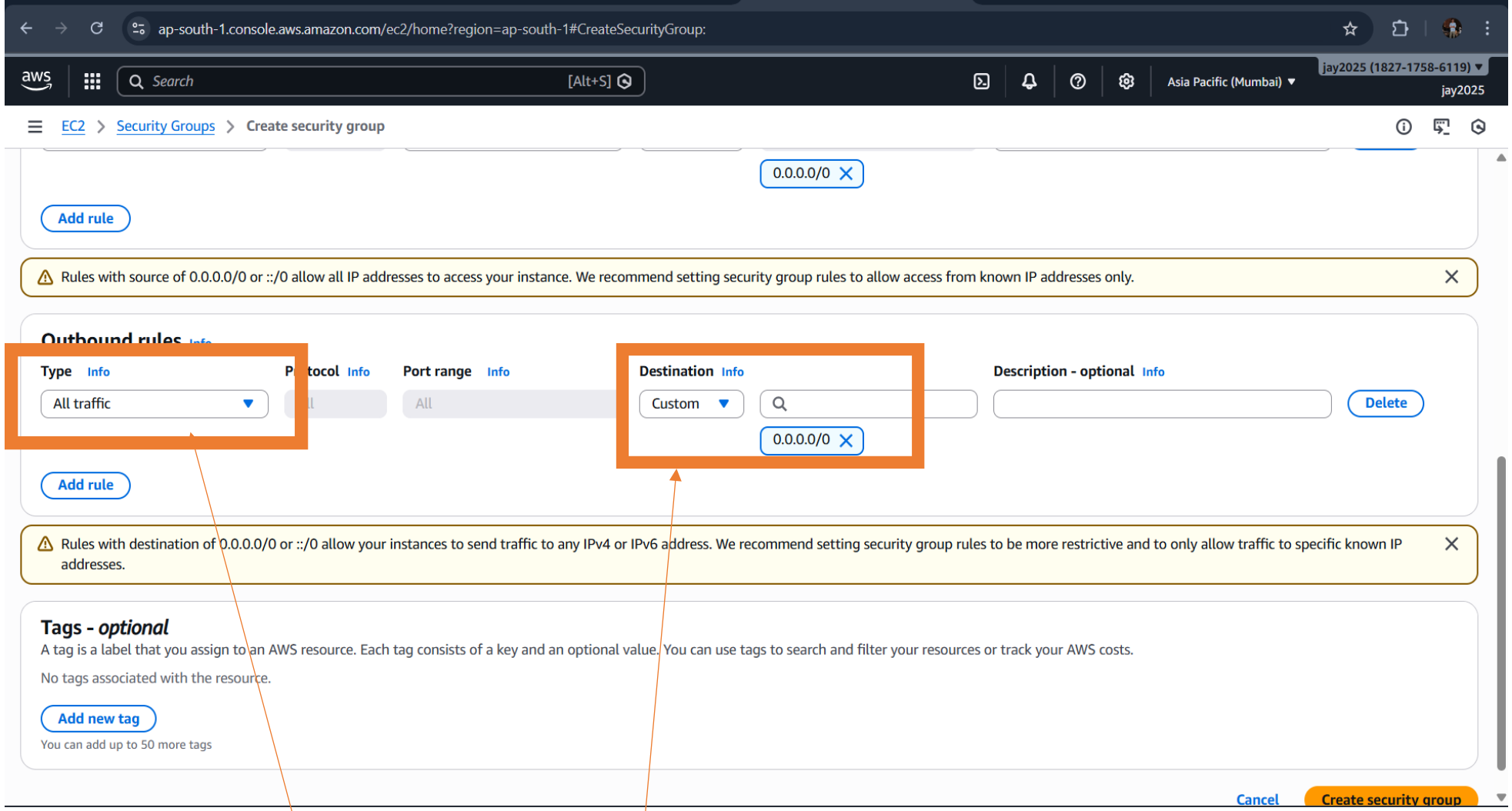
In this slide, we search and select the **EC2 service** from the AWS console.



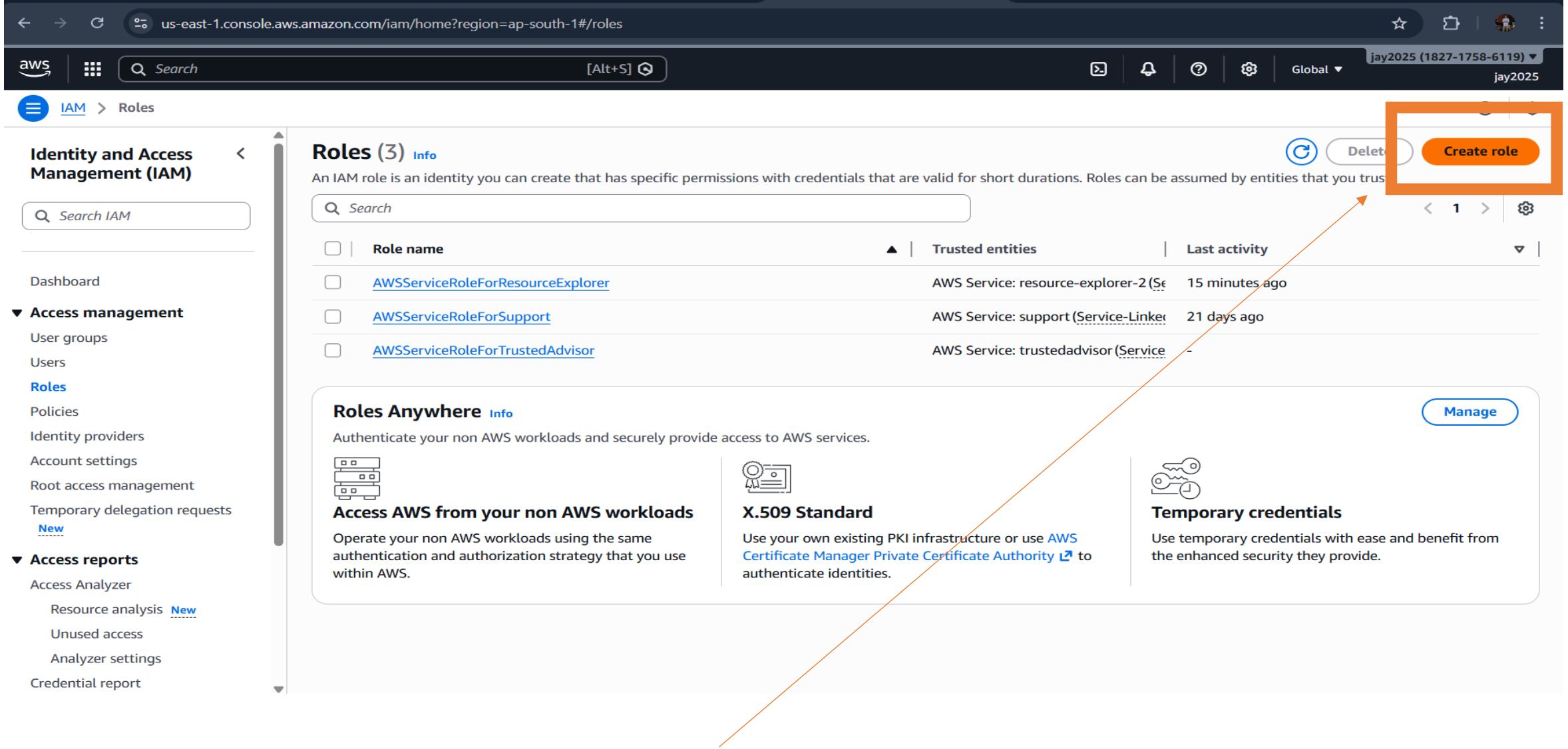
Open the Security Groups section under EC2.



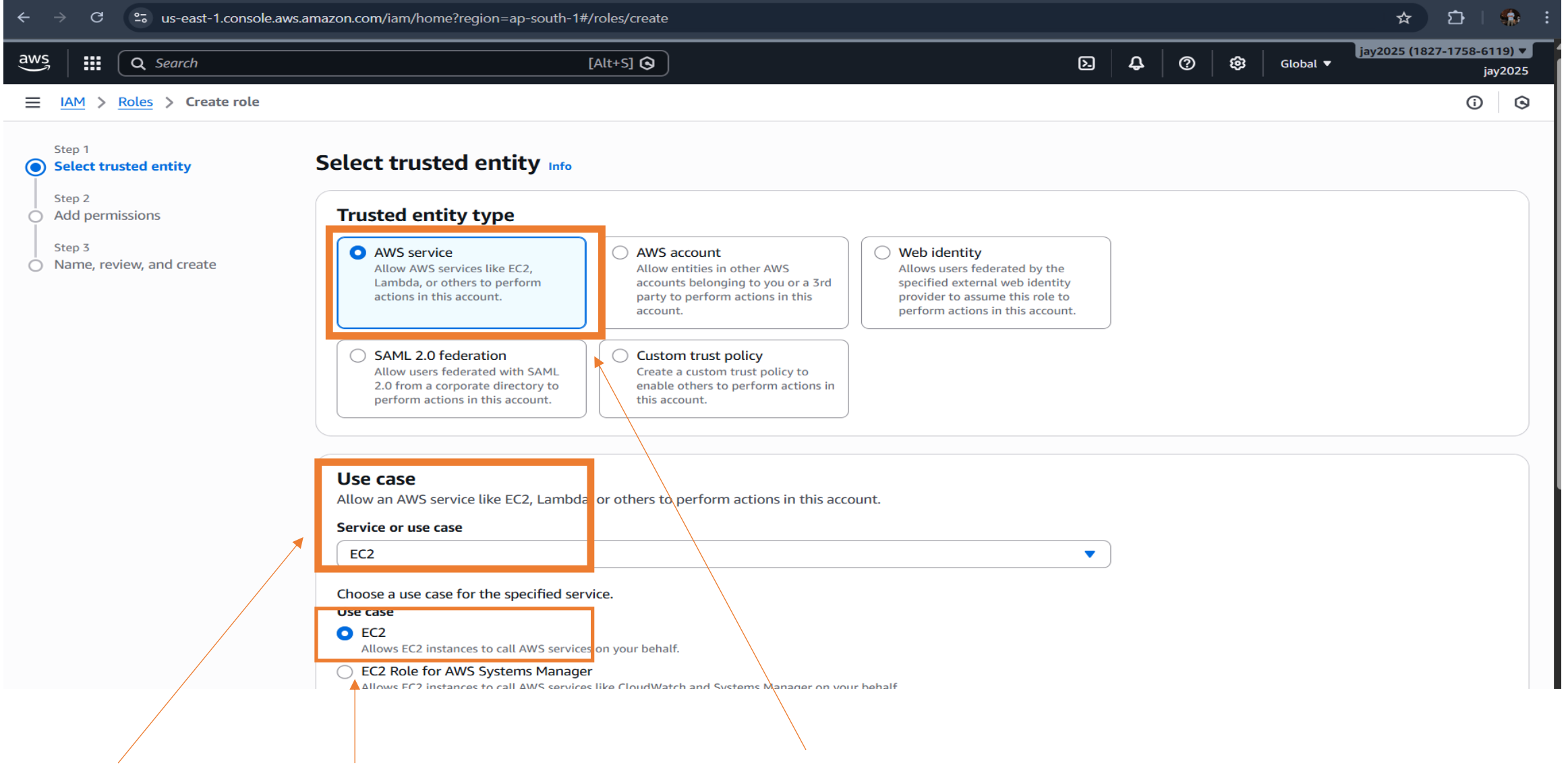
we create a new security group by providing the **security group name**, **description**, and selecting the correct **VPC**. Inbound rules are added to allow **SSH (22)**, **HTTP (80)**, and **Custom TCP (5000)** access from **0.0.0.0/0**.



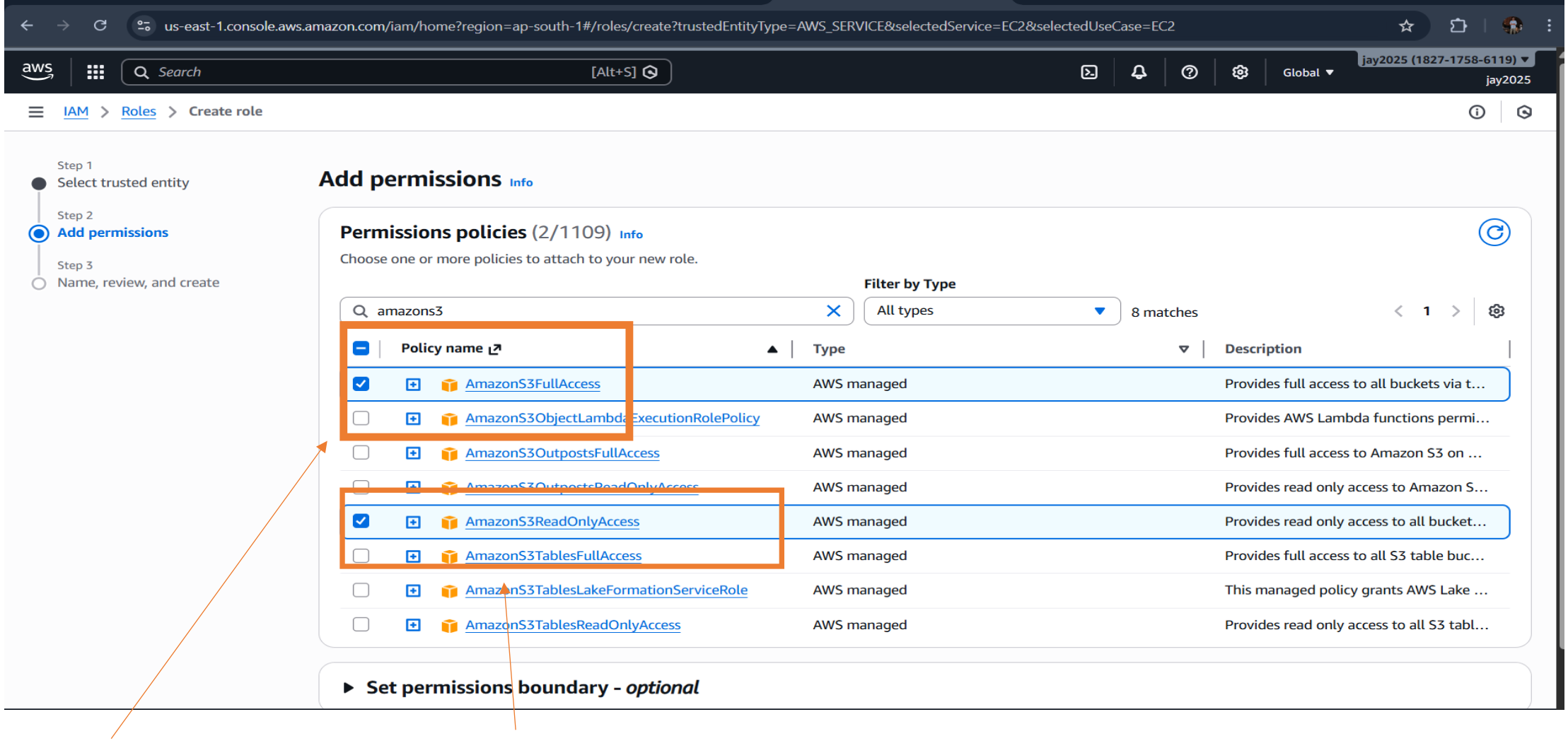
Outbound rules are configured to allow **all traffic** to **0.0.0.0/0**, enabling the EC2 instance to communicate with the internet and AWS services.



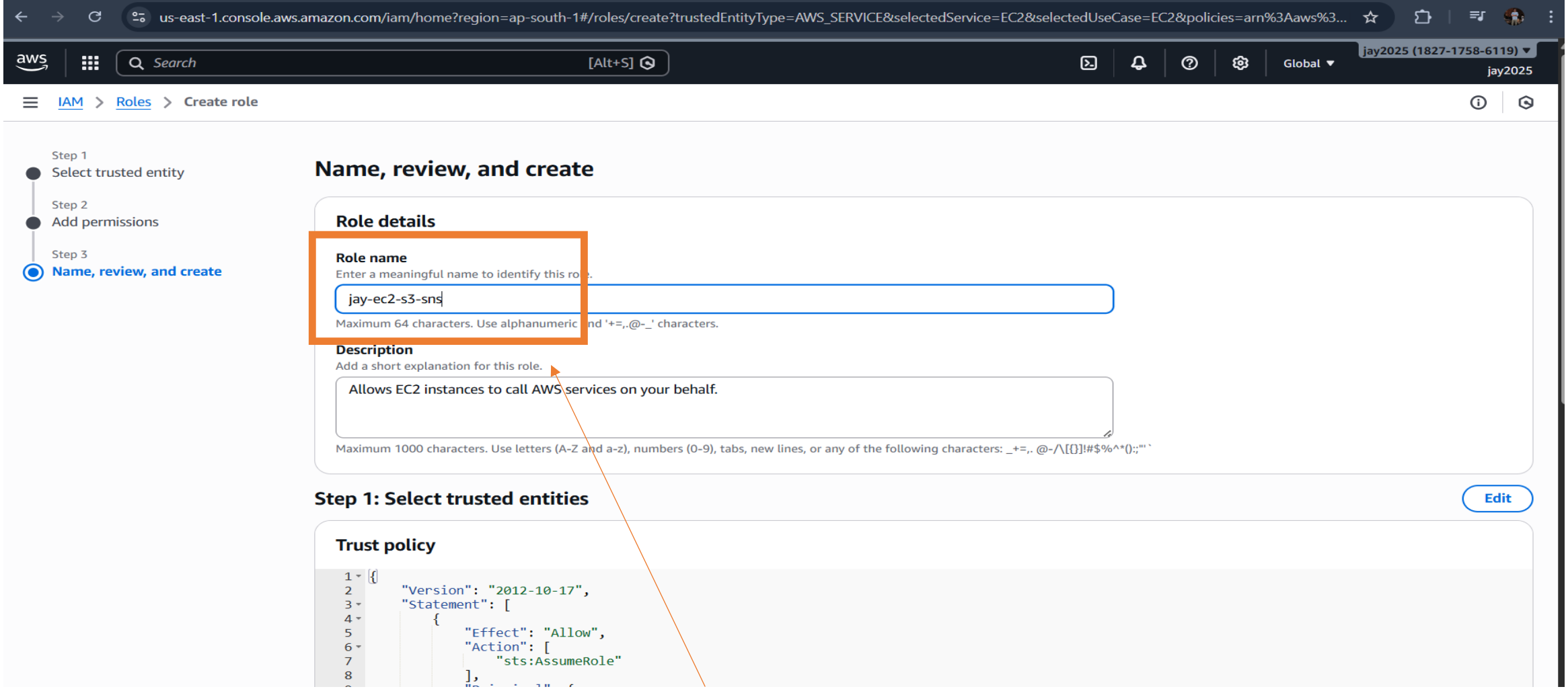
- Displays all IAM roles created in my AWS account.
- Click on create role button to create new role.



- Choose AWS service to allow EC2 to assume the role.
- Select EC2 as the use case for accessing other AWS services.
- Enables EC2 to interact with S3 and SNS securely.

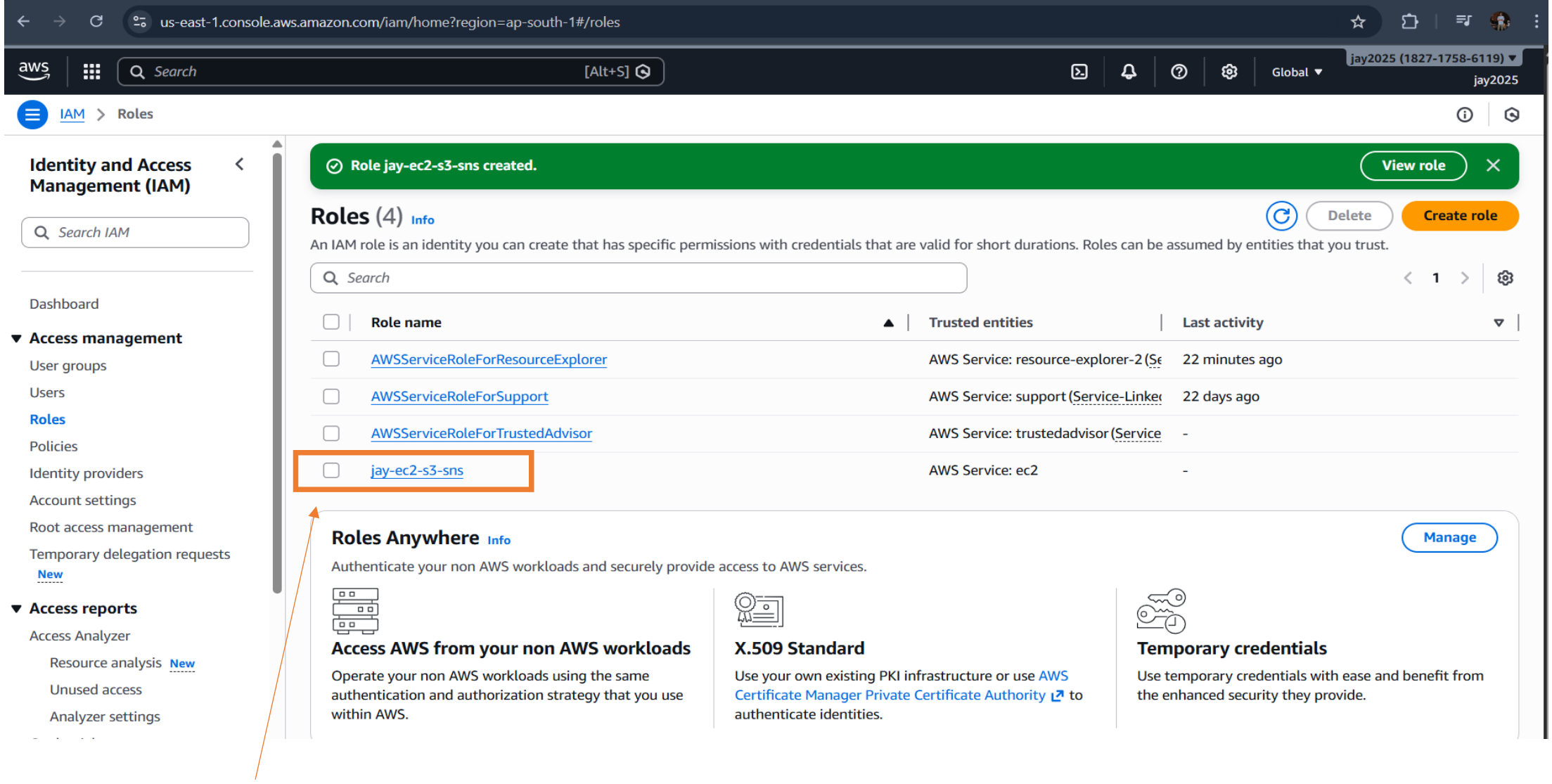


- Search for and select S3 access policies for the EC2 role.
- Attach **AmazoneS3FullAccess** and **AmazonS3ReadonlyAccess** as needed.
- These permissions allow EC2 to interact with S3 buckets securely.

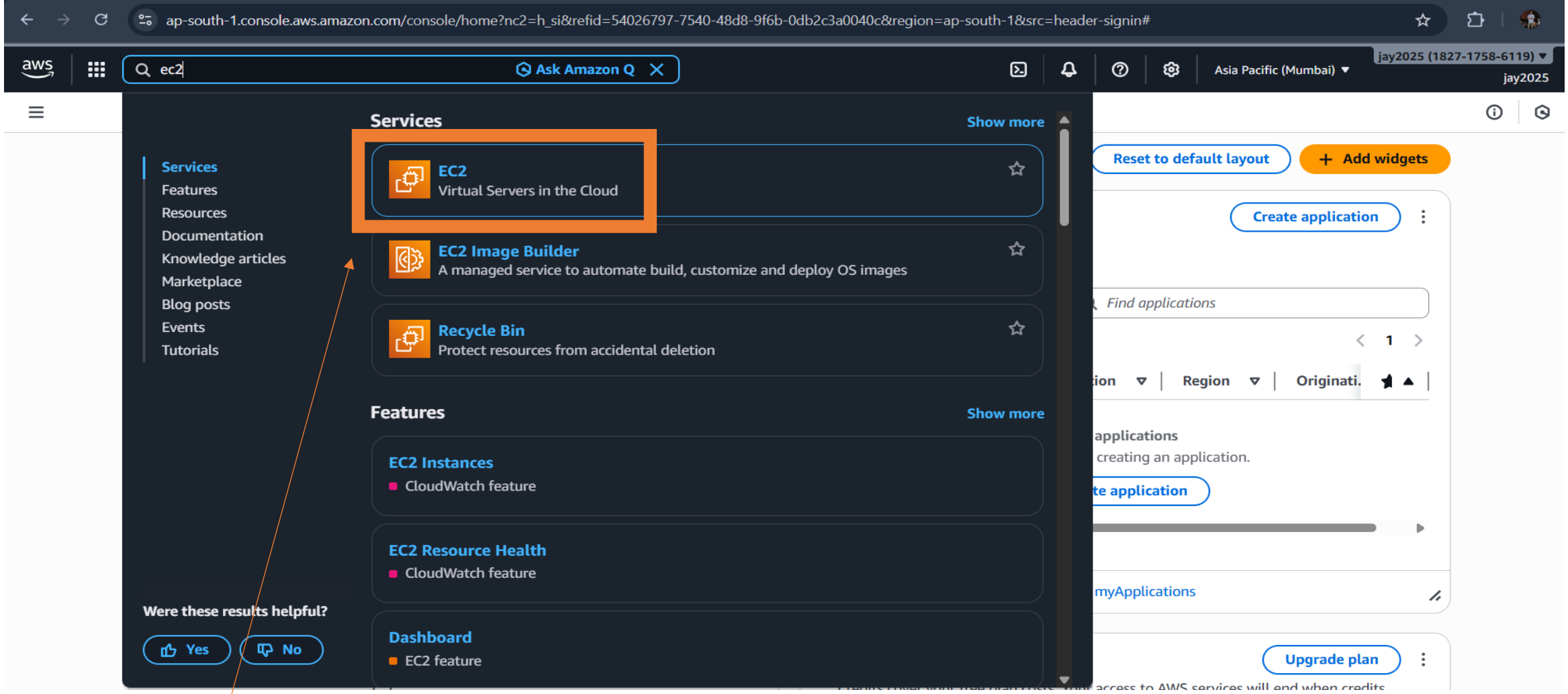


- Enter a clear role name like **jay-ec2-s3-sns** and add a description.
- Review the trust policy allowing EC2 to assume the role.
- Confirm and create the role for secure service access.

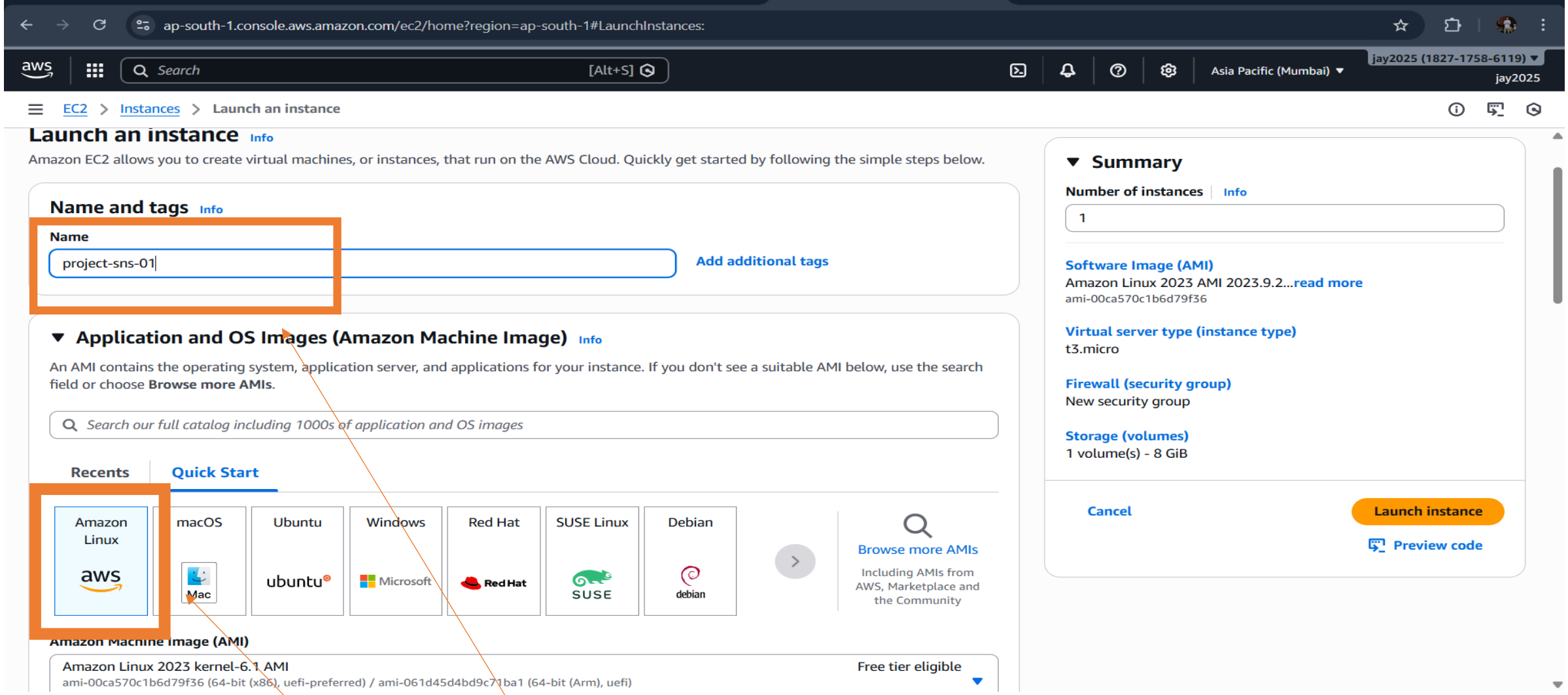
✅ This completes the IAM role setup for my project. The role now allows EC2 to interact with S3 and SNS, which is essential for my file upload notification system.



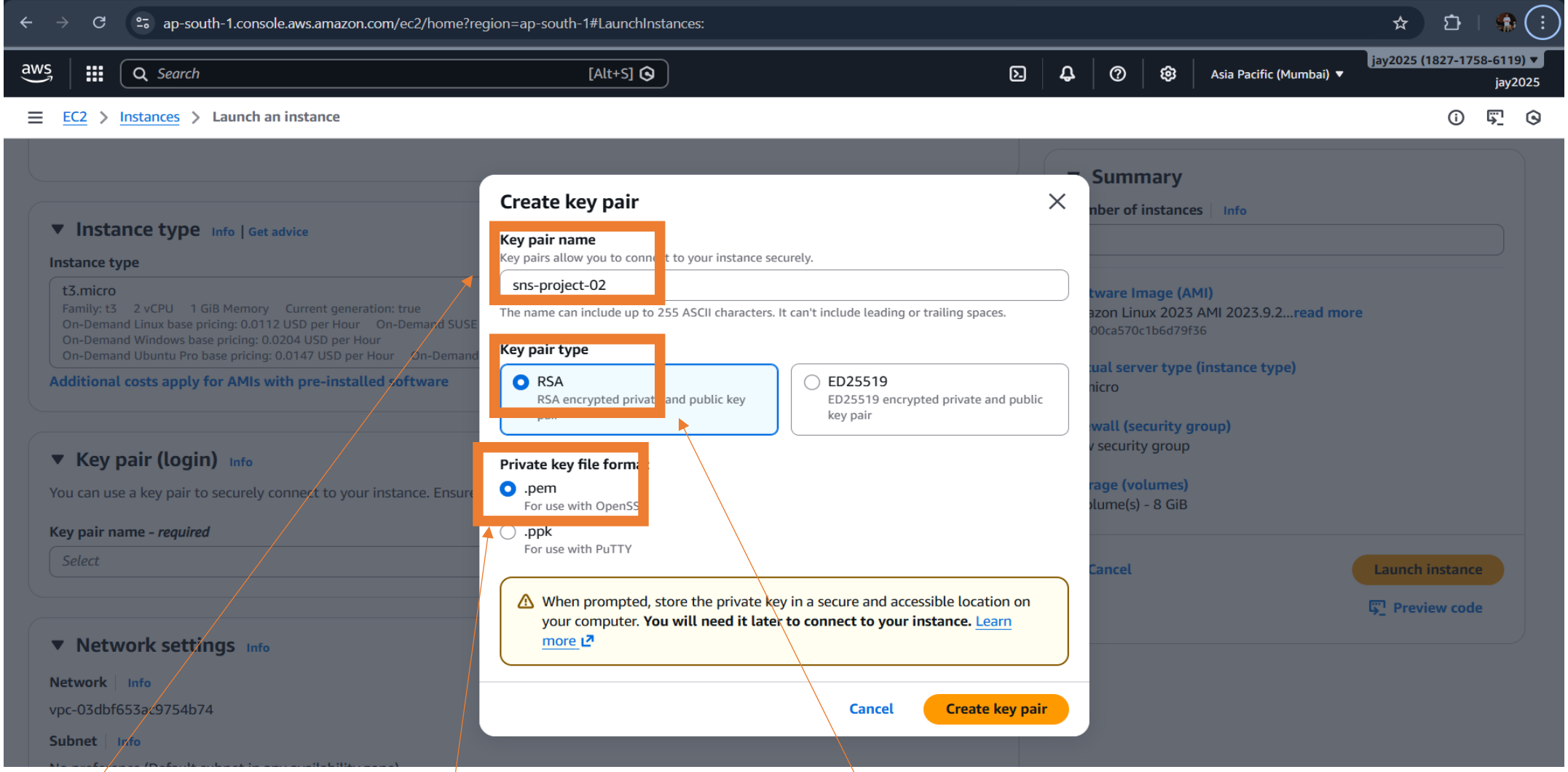
- Confirms successful creation of the **jay-ec2-s3-sns IAM role**.
- Role is now listed and ready for EC2 to assume.
- Enables secure access to S3 and SNS for my notification system.



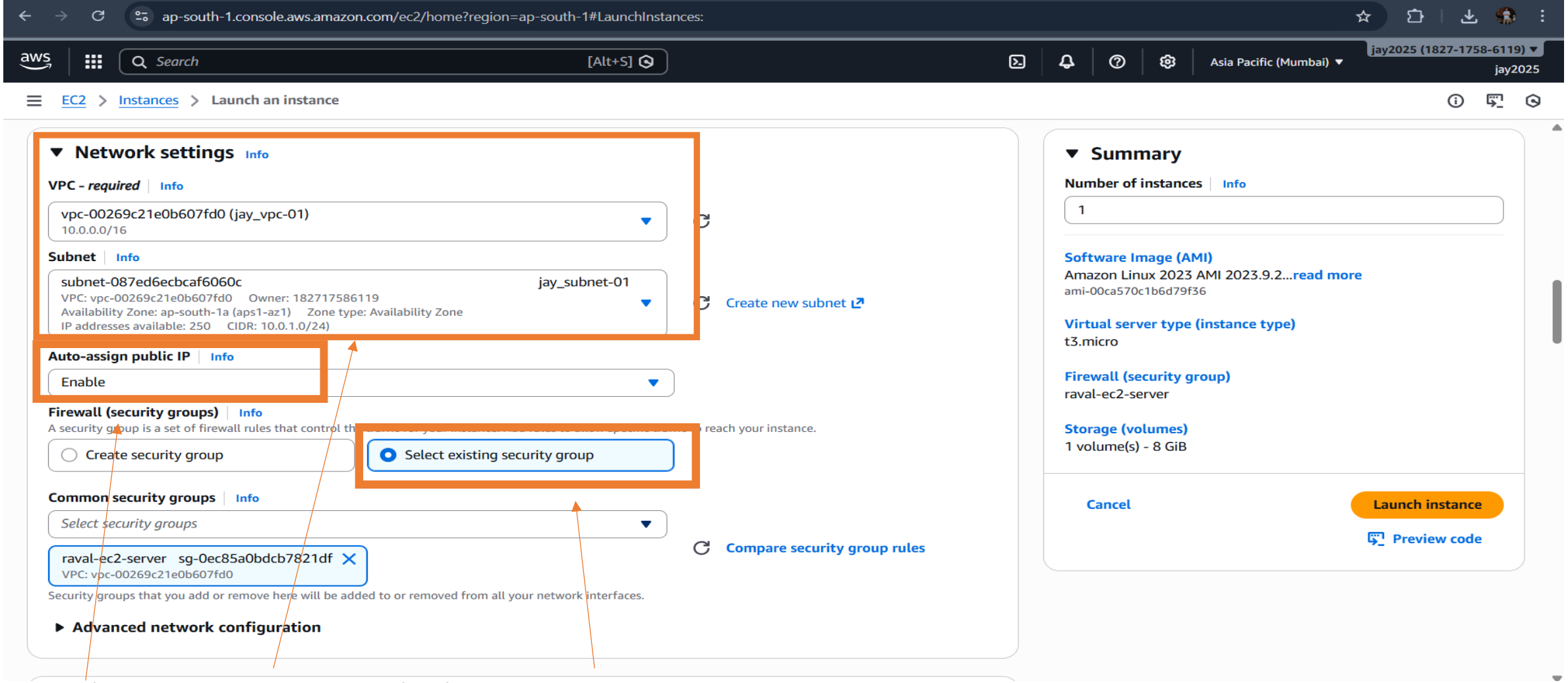
- Use the AWS search bar to find EC2 services



- Launch a new EC2 instance for file upload testing.
- Select Amazon Linux as the operating system.
- Name the instance **project-sns-01.**



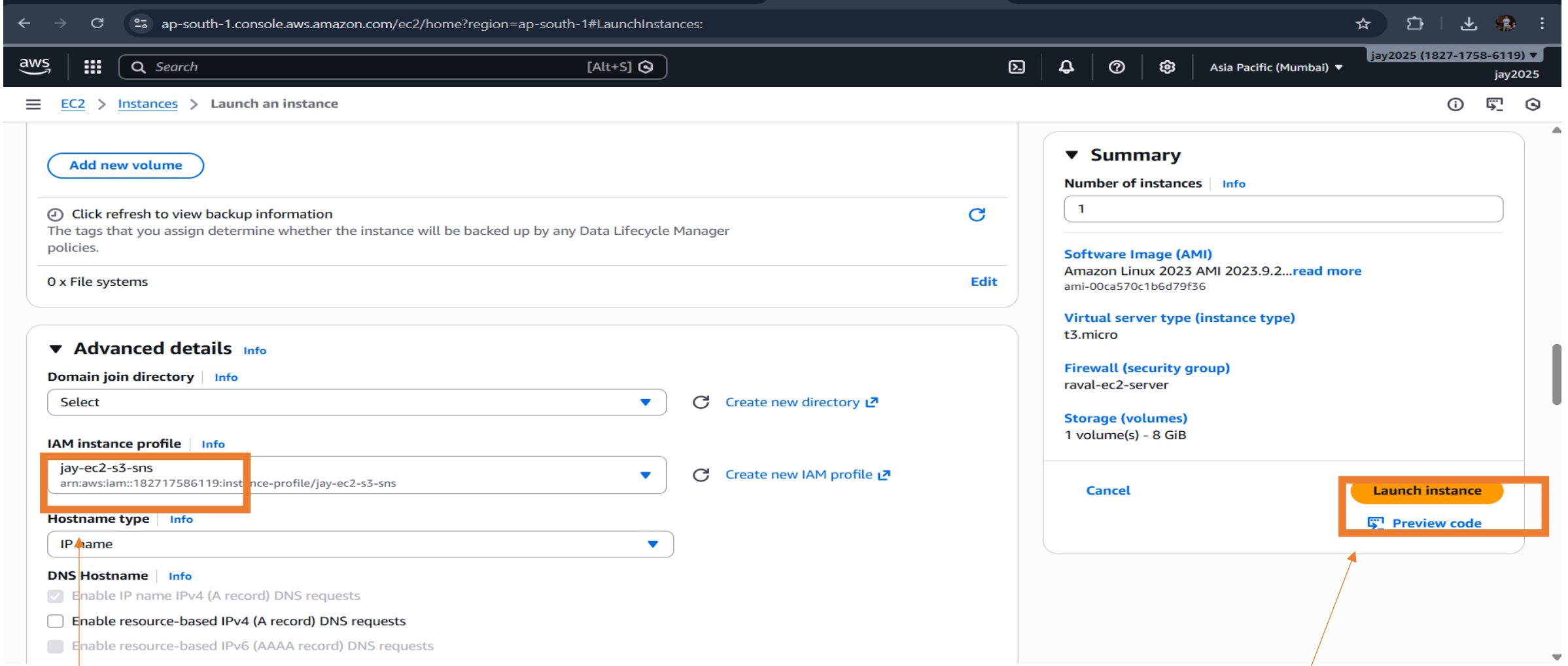
- Generate a key pair named **sns-project-02** for SSH access.
- Choose RSA encryption and format for OpenSSH.
- And creat key pair.



- Select my custom VPC and subnet .
- Enable auto-assign public IP for internet access.
- Apply the existing security group for firewall rules.

Why Use This Step

- Ensures the EC2 instance is placed inside a secure, isolated network.
- Public IP assignment allows external access for uploads and testing.
- Security group rules control inbound/outbound traffic for safe operations.

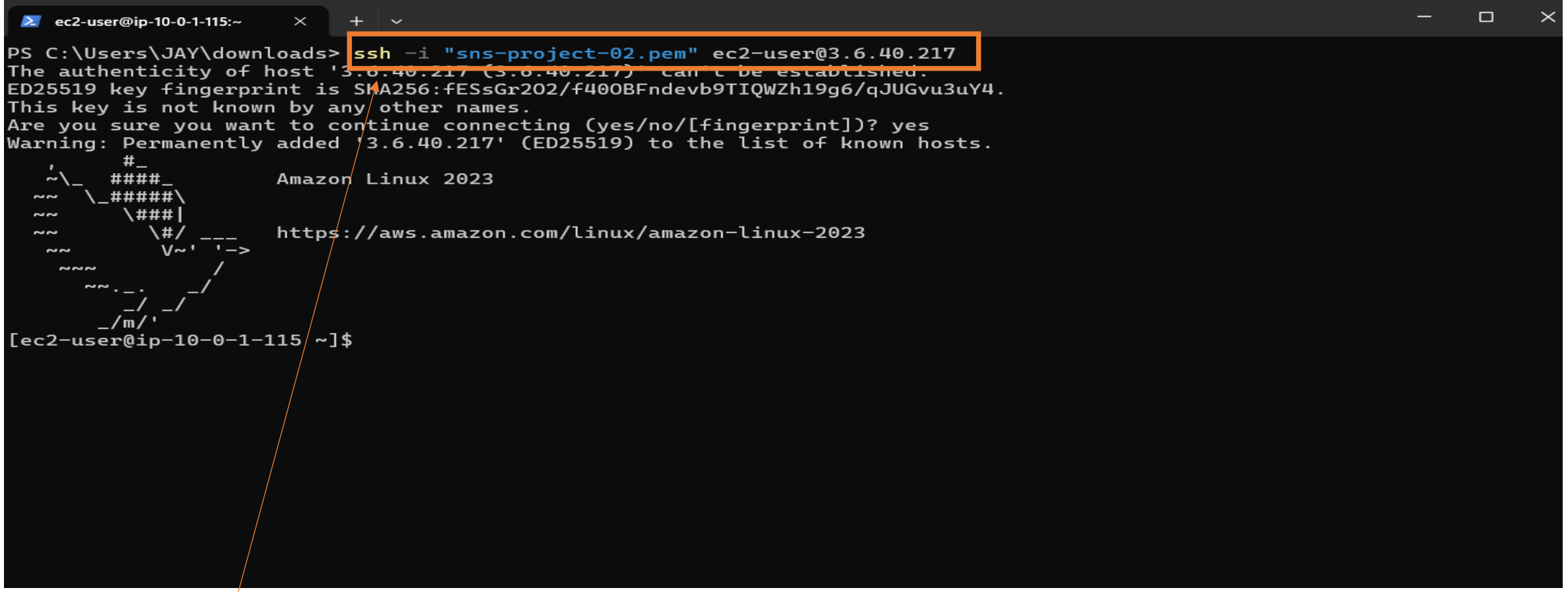


- Attach IAM role to the EC2 instance.

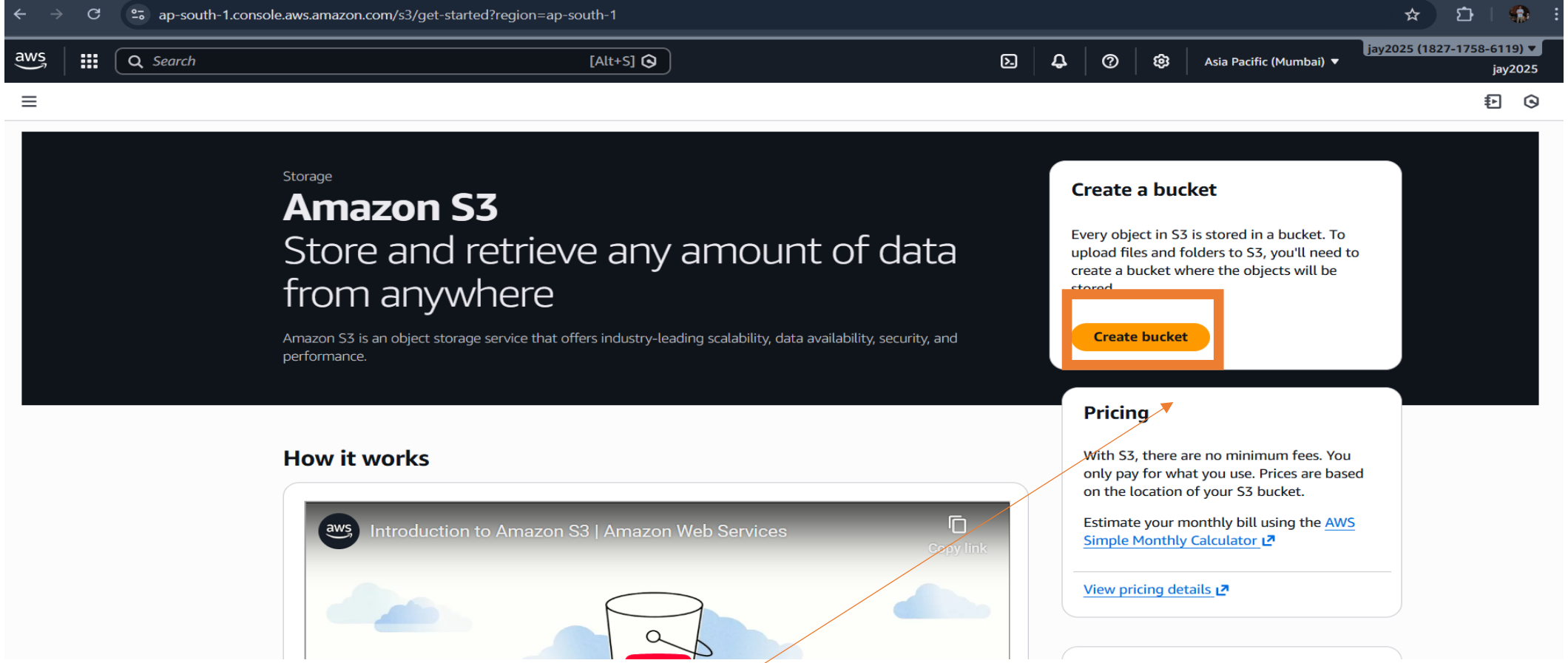
Why Use This Step

- IAM role gives EC2 permission to access S3 and publish to SNS.

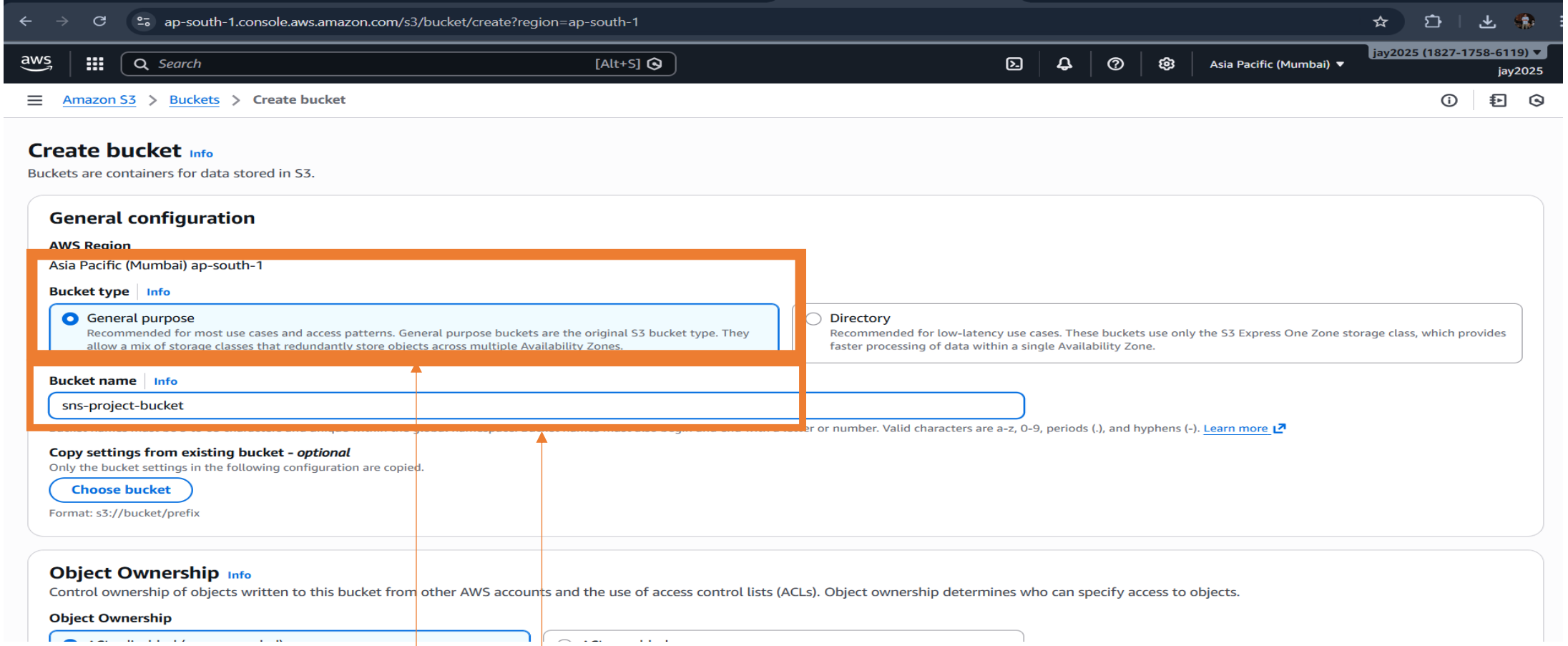
- And click on Lanch instance and connect to powershell



- Use the key file to securely connect to EC2.
- Run the SSH command with my public IP and username.
- Confirm login to Amazon Linux terminal for testing.



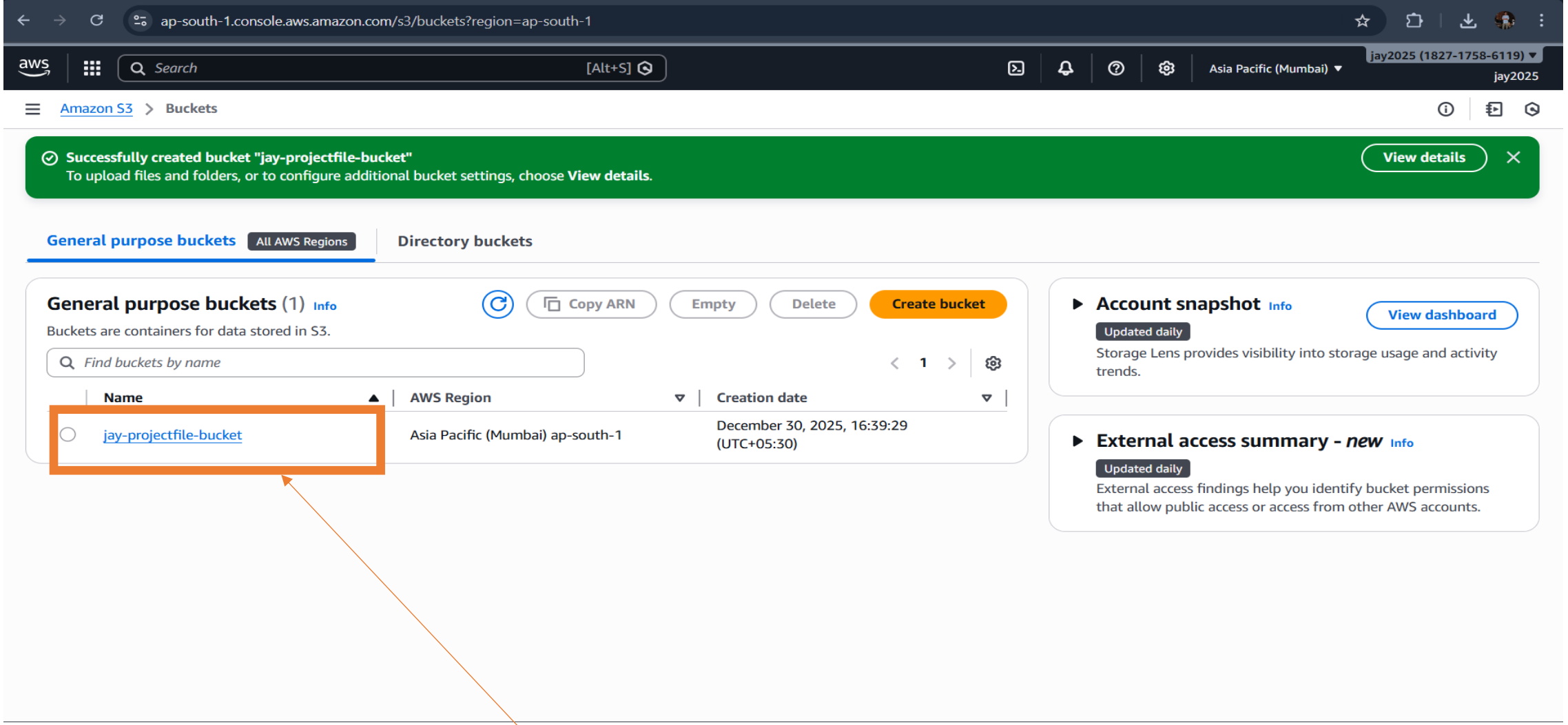
- Click on create bucket



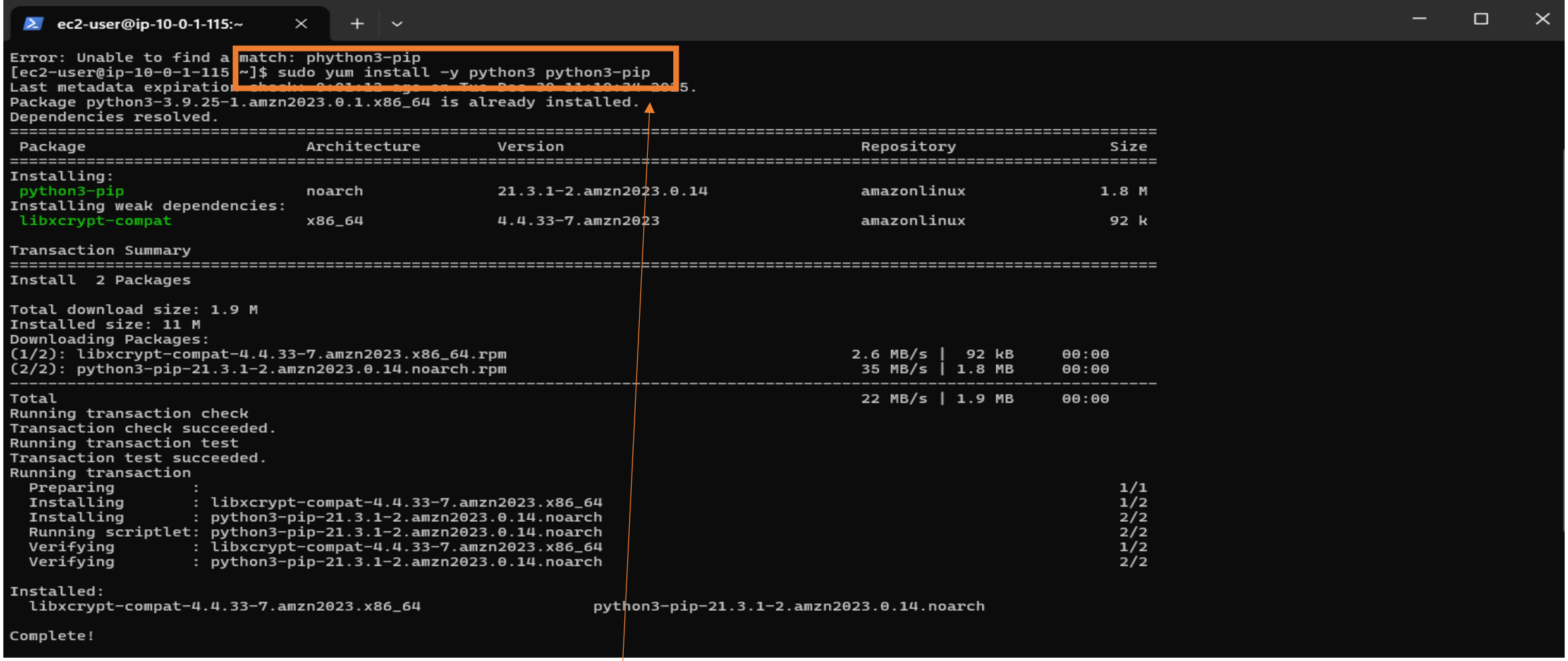
- Create a general-purpose bucket named .
- Select region for storage location.
- Use default settings for object ownership and access control.

Why Use This Step

- S3 stores uploaded files that trigger SNS notifications.
- Buckets are scalable, durable, and accessible from EC2.
- Region selection ensures low latency and compliance.



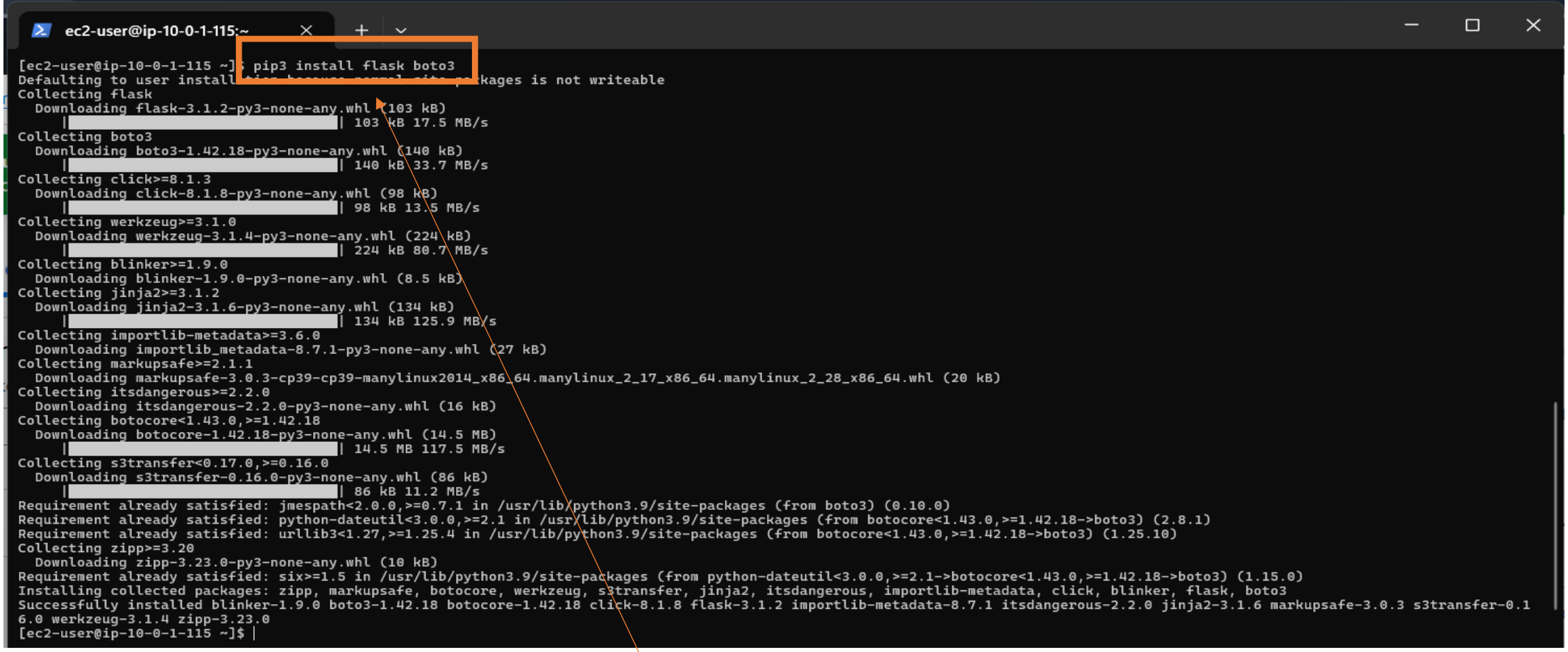
- Confirms successful creation of **jay-projectfile-bucket**
- Bucket is now ready to receive file uploads from EC2.



- Run **sudo yum install –y python3 python3-pip** to install pip.
- pip allows you to install Python packages like boto3.

Why Use This Step

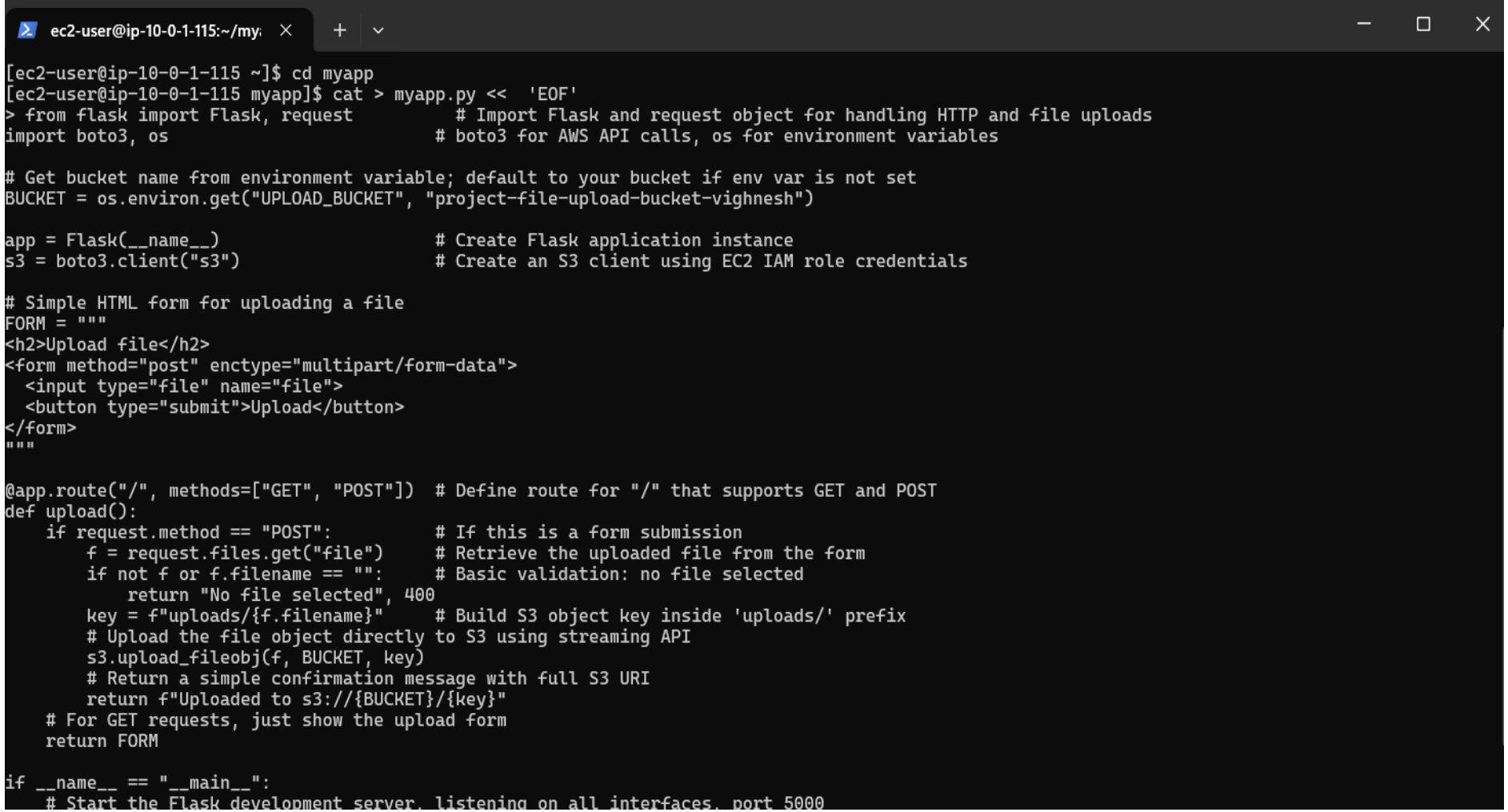
- Python and pip are needed to write and run upload scripts.



- Run **pip3 install flask boto3** on EC2 terminal.
- Flask is used to build a simple web interface.
- boto3 is the AWS SDK for Python to access S3 and SNS.

Why Use This Step

- Flask lets you create a lightweight upload form or API.
- boto3 enables your Python code to interact with AWS services.



- Create a Python script using Flask and boto3.
- Build a simple web form to upload files.

Why Use This Step

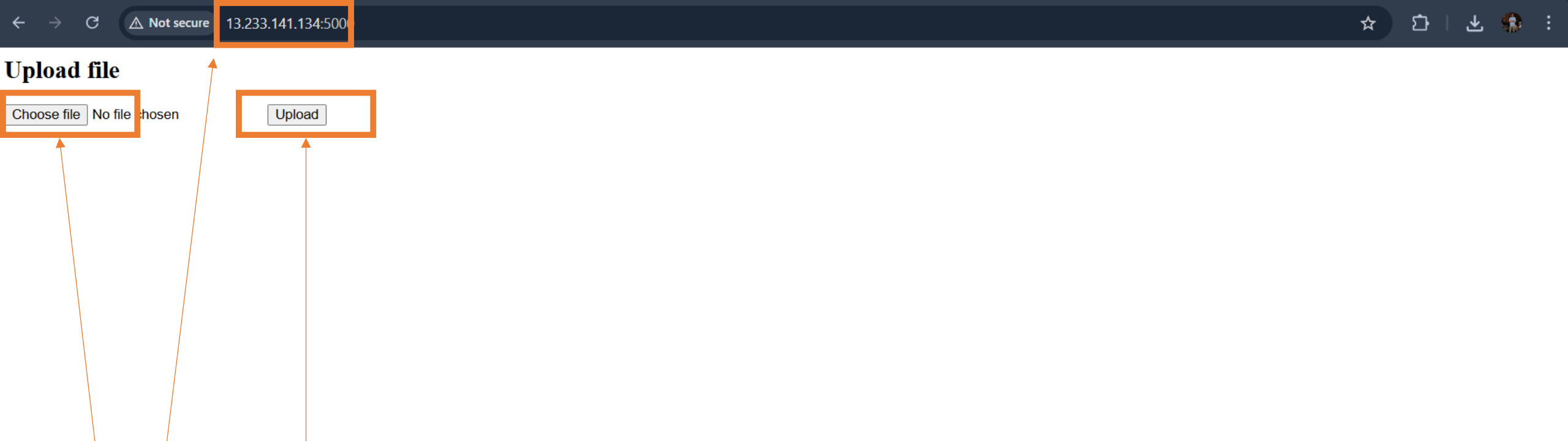
- Flask provides a clean interface for file uploads.
- boto3 handles secure S3 uploads using EC2 IAM role.

```
[ec2-user@ip-10-0-1-115 myapp]$ python3 myapp.py &
[1] 2741
[ec2-user@ip-10-0-1-115 myapp]$ /home/ec2-user/.local/lib/python3.9/site-packages/boto3/compat.py:89: PythonDeprecationWarning: Boto3 will no lo
nger support Python 3.9 starting April 29, 2026. To continue receiving service updates, bug fixes, and security updates please upgrade to Python
 3.10 or later. More information can be found here: https://aws.amazon.com/blogs/developer/python-support-policy-updates-for-aws-sdks-and-tools/
  warnings.warn(warning, PythonDeprecationWarning)
 * Serving Flask app 'myapp'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://10.0.1.115:5000
Press CTRL+C to quit
```

- Start the Flask app using .
- App runs on port 5000 and listens on all interfaces.
- Accessible via EC2 public IP for file uploads.

Why Use This Step

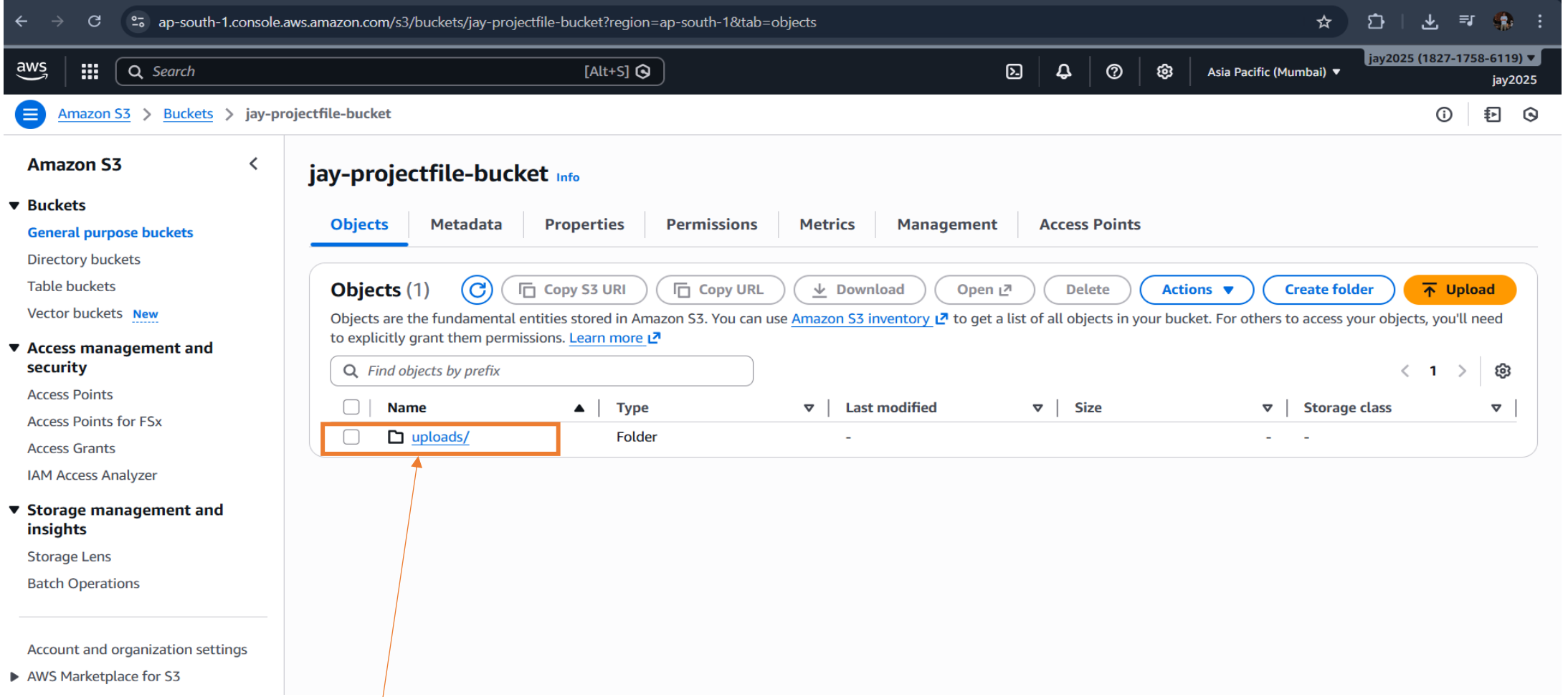
- Launches the upload interface for testing S3 integration.



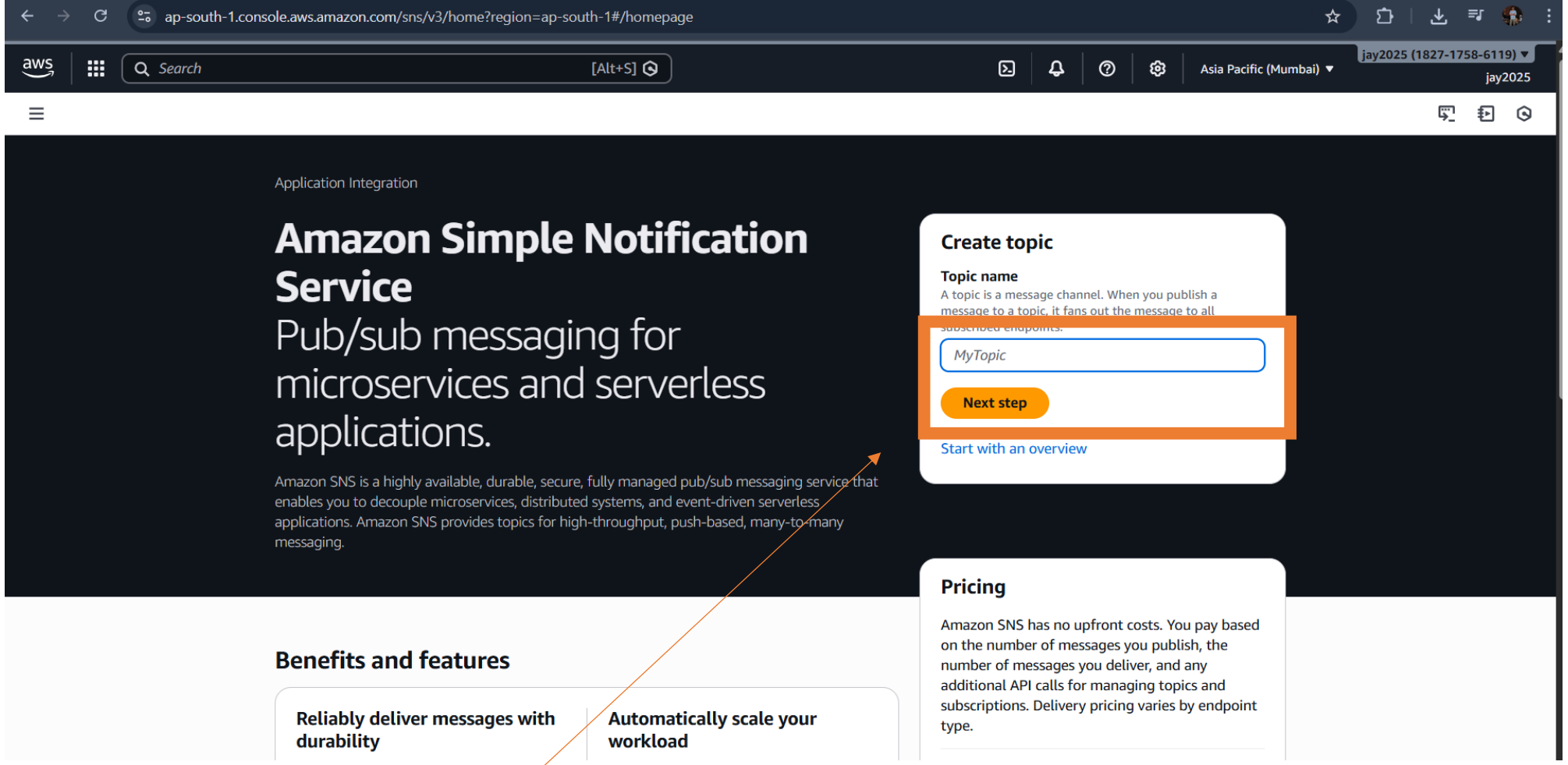
- Open my EC2 public ip with :5000 port no in browser.
- Use the form to select and upload a file.
- The file will be sent to S3 bucket via Flask + boto3.

Why Use This Step

- Confirms your EC2 app is publicly accessible.
- Allows manual testing of file uploads to S3.



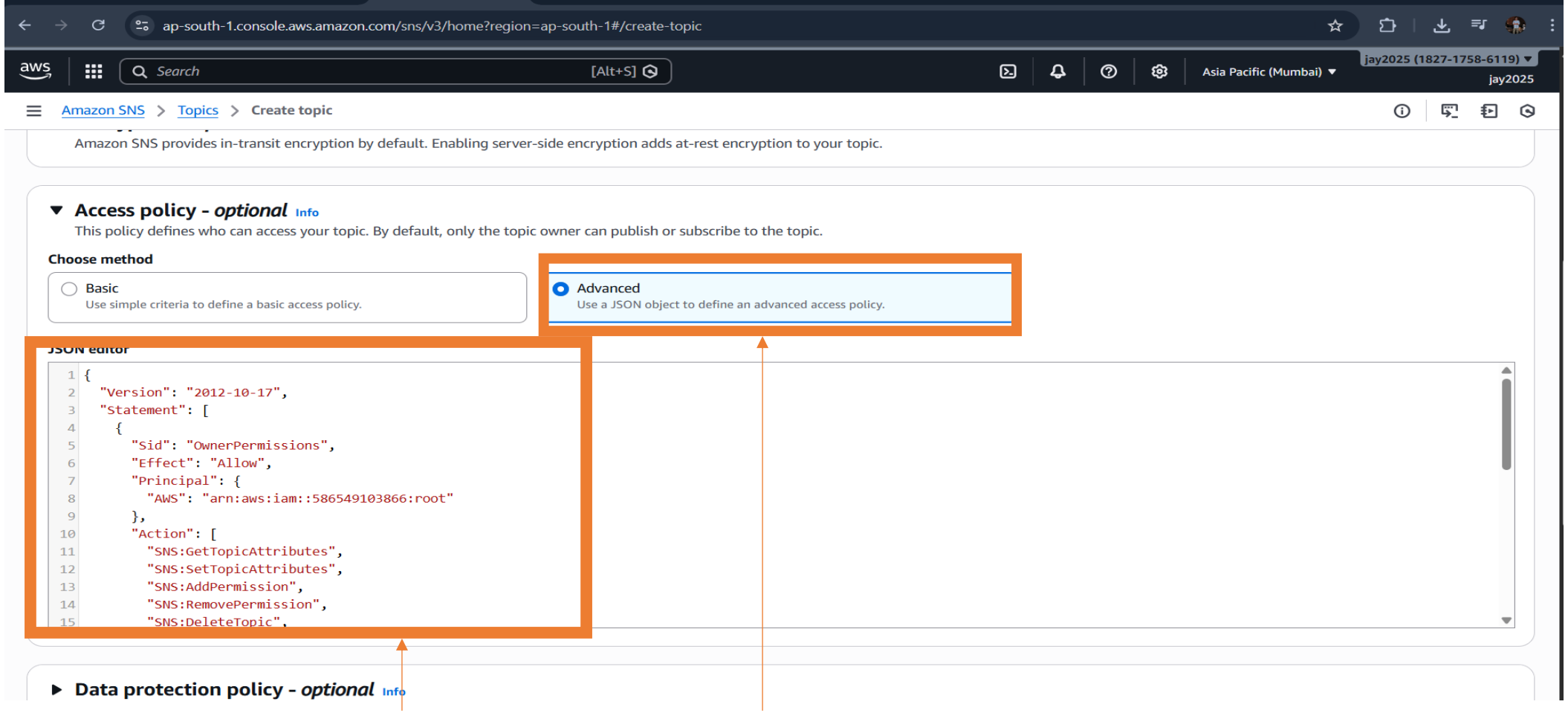
- Open the S3 **bucket jay-projectfile-bucket** in AWS Console.
- Confirm the **uploads/** folder was created.



- Go to Amazon SNS and click “Create topic.”
- Enter a topic name **[jay-upload-alert]**

Why Use This Step

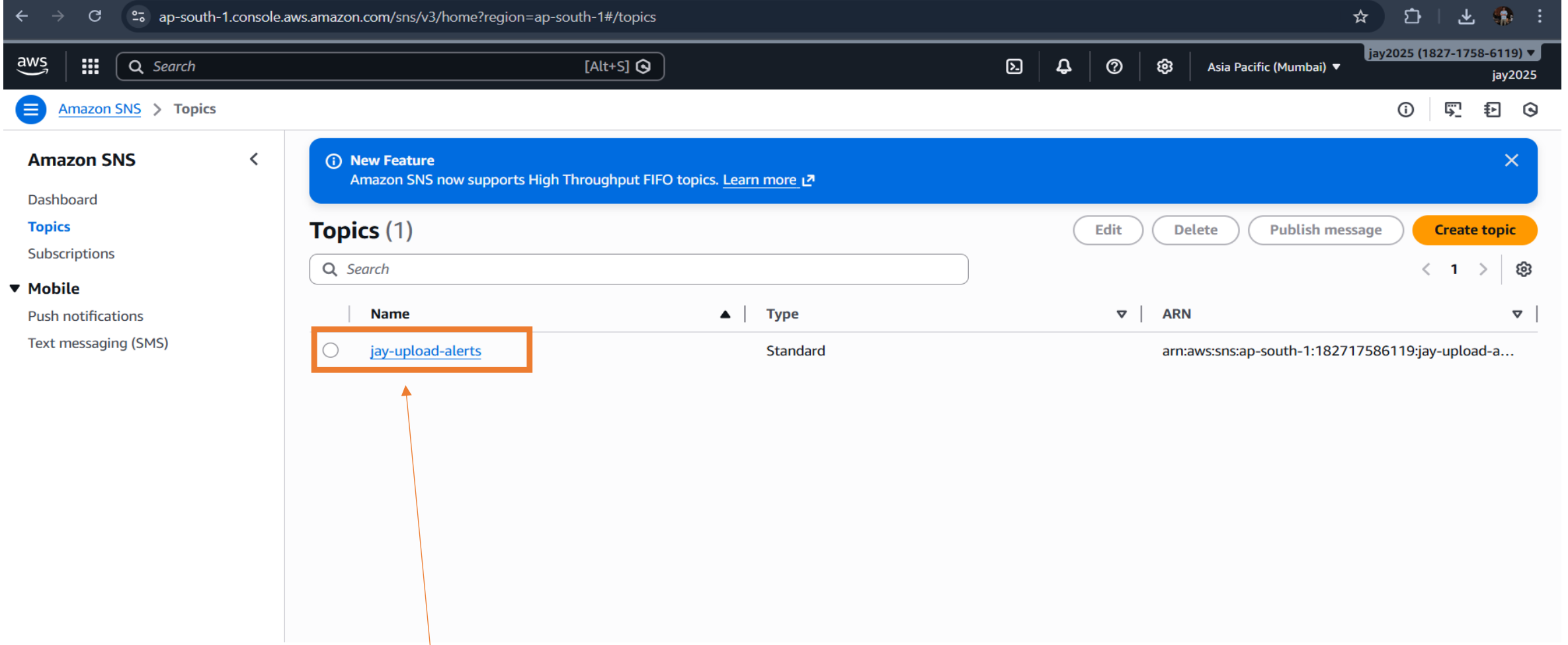
- SNS enables real-time alerts when files are uploaded to S3.



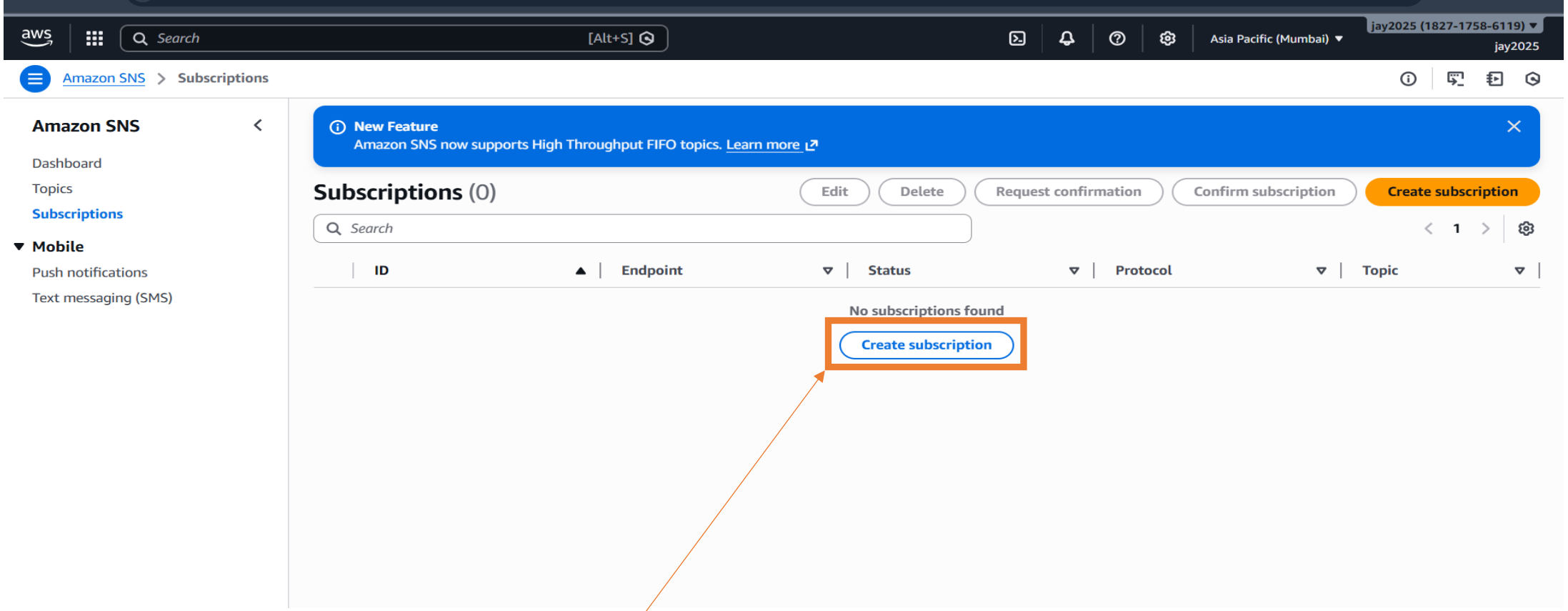
- Use the “Advanced” option to write a custom JSON policy.
- Grant permissions to IAM user or root account.
- Allow actions like GetTopicAttributes, SetTopicAttributes, and DeleteTopic.

Why Use This Step

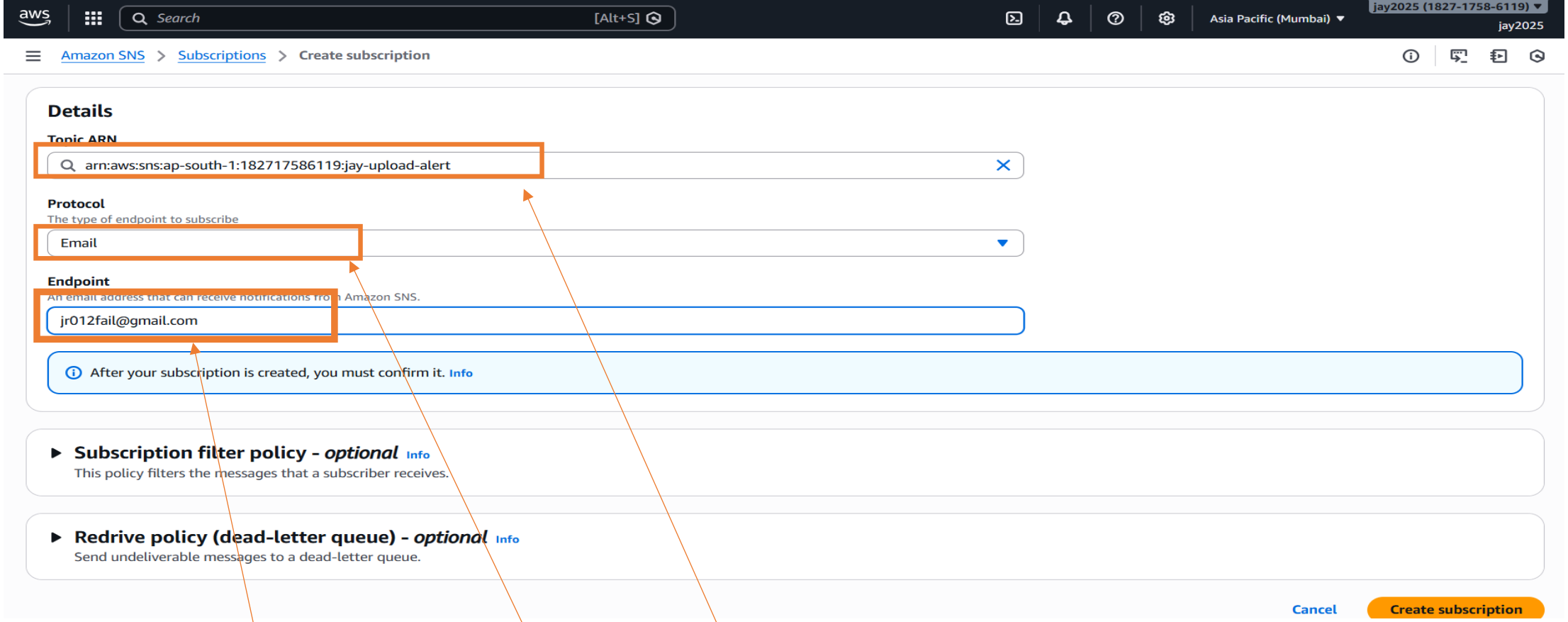
- Controls who can manage and interact with SNS topic.
- Ensures only trusted identities can modify or delete the topic.
- Prepares the topic for secure integration with S3 notifications.



- SNS topic **jay-upload-alerts** created successfully.



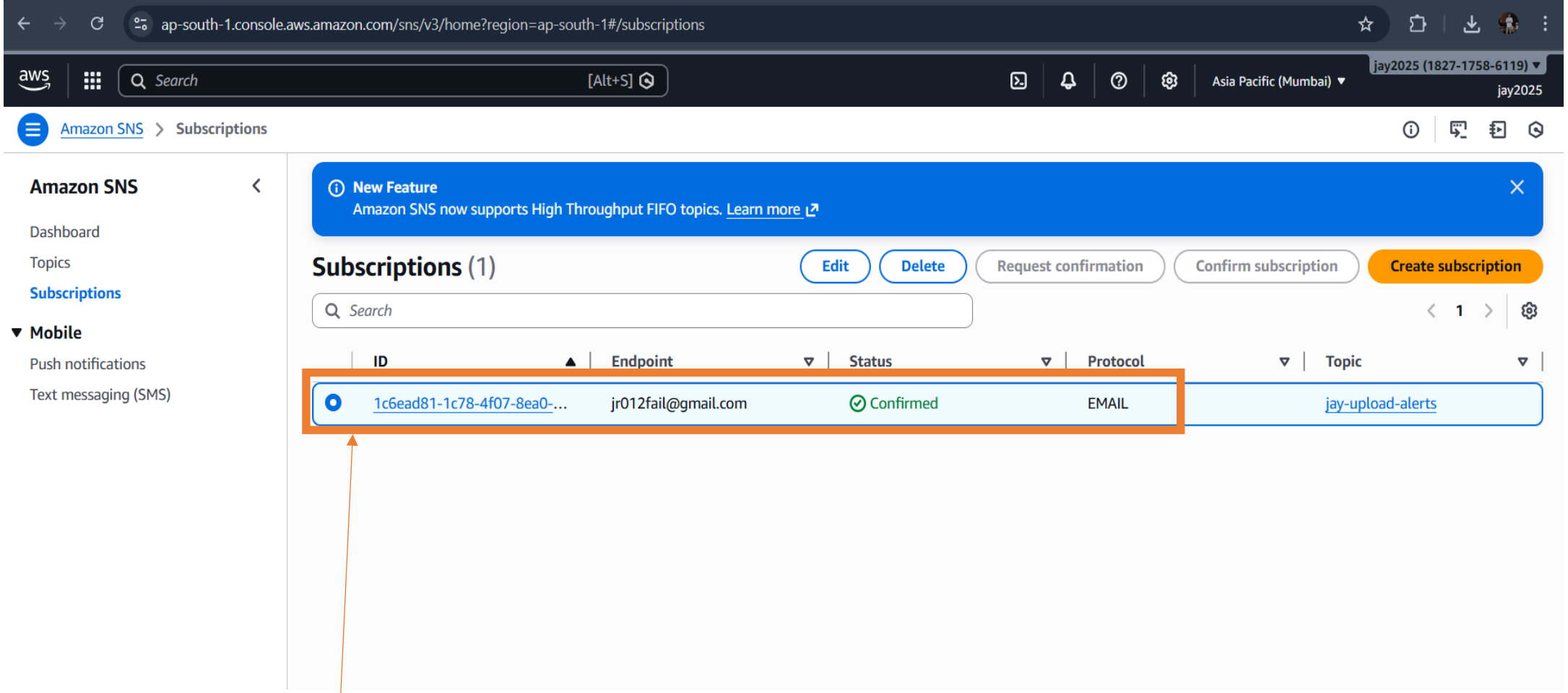
Click Creat subscription



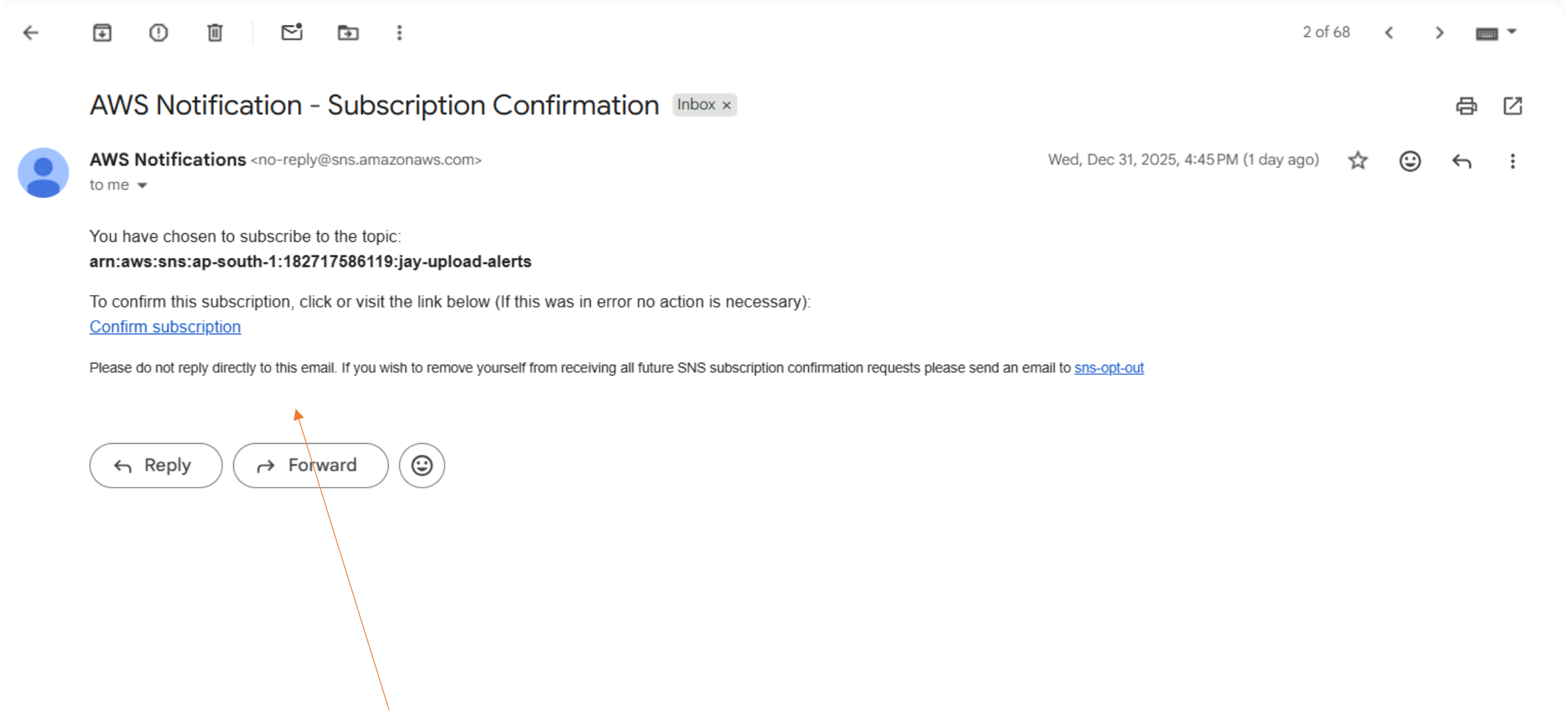
- Subscribe to topic **jay-upload-alert** using protocol: Email.
- Enter email address as the endpoint.
- Confirm the subscription via the email link sent by AWS.

Why Use This Step

- Enables real-time email alerts when files are uploaded to S3.



- Email subscription to topic **jay-upload-alert** is now confirmed.



- AWS sends a confirmation email to my email.

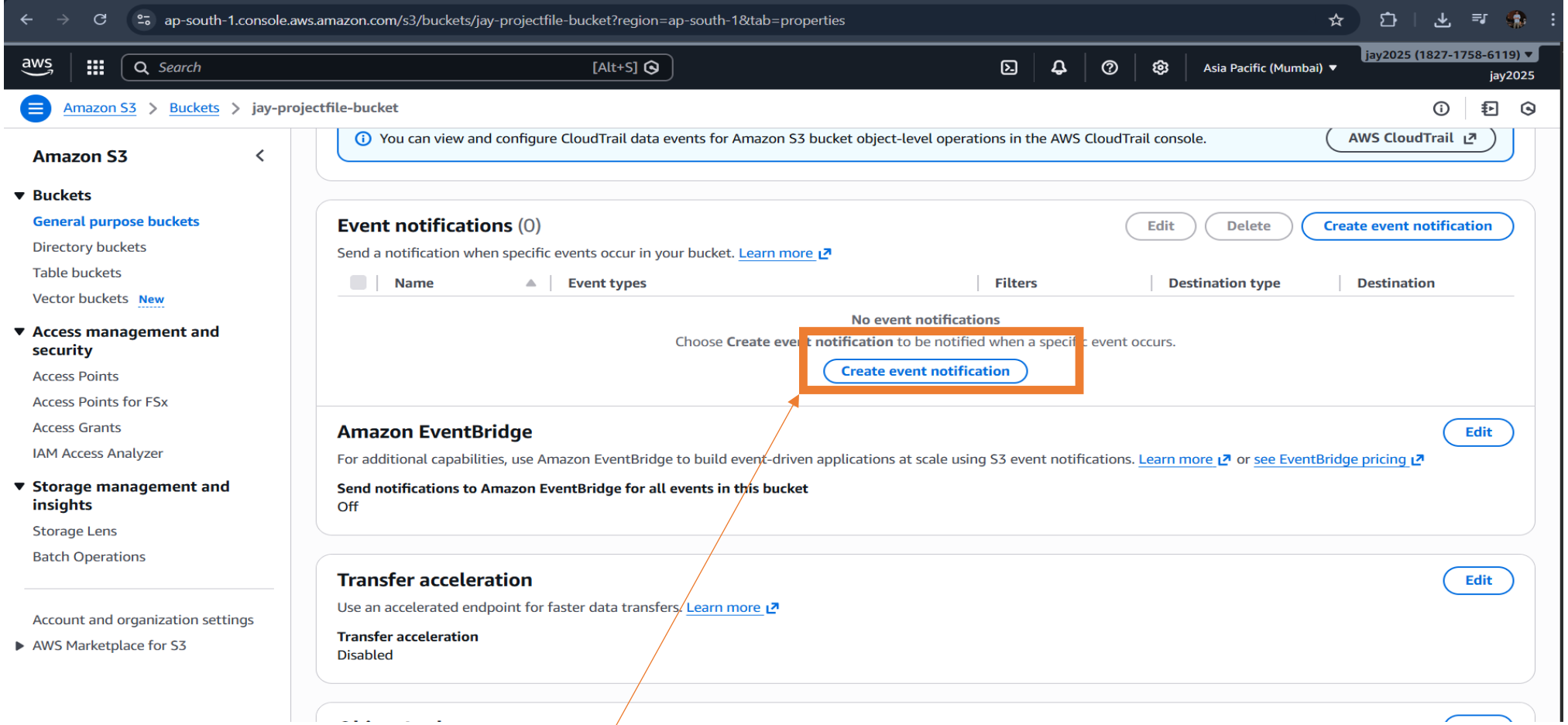
Simple Notification Service

Subscription confirmed!

You have successfully subscribed.

Your subscription's id is:
`arn:aws:sns:ap-south-1:182717586119:jay-upload-alert:aa972abb-649c-4395-a64d-d4a0dec4acdc`

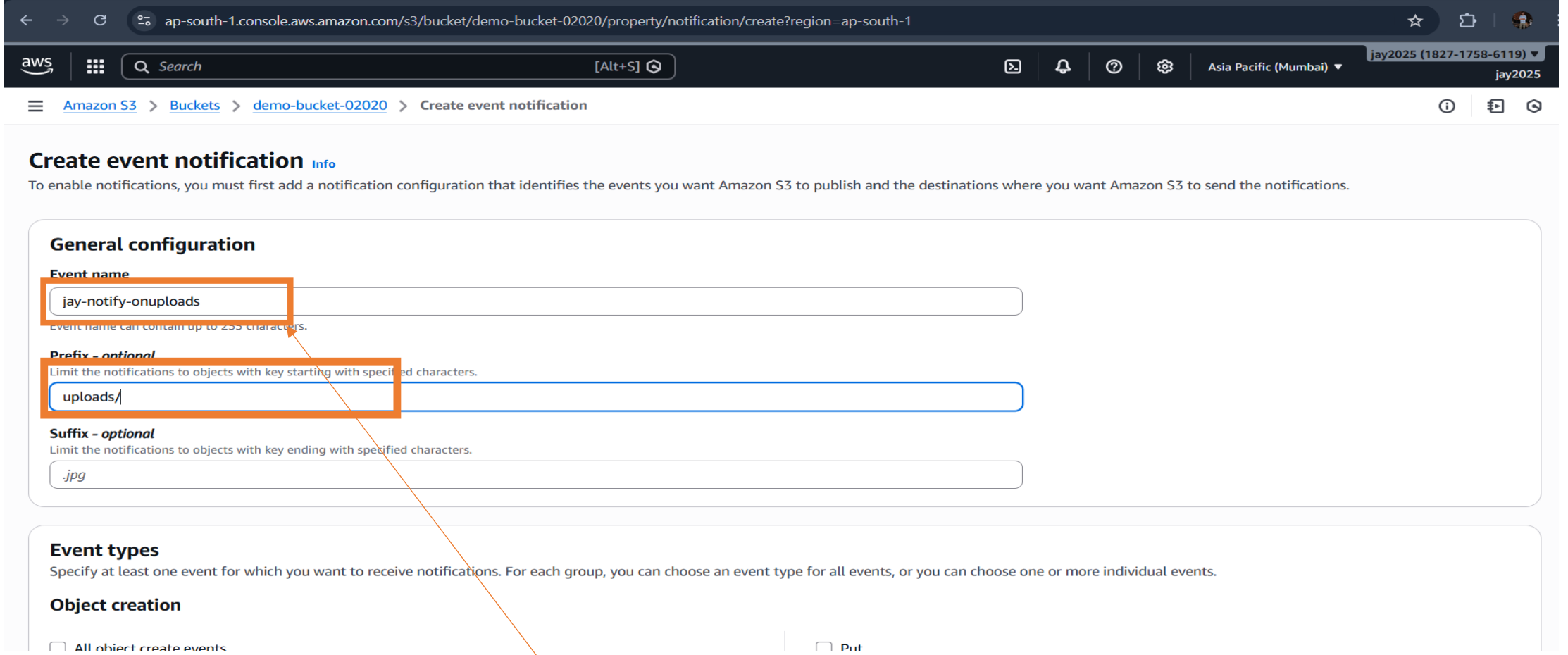
If it was not your intention to subscribe, click here to unsubscribe.



- Go to the S3 bucket **jay-projectfile-bucket> Properties > Event notifications.**
- Click “Create event notification.”

Why Use This Step

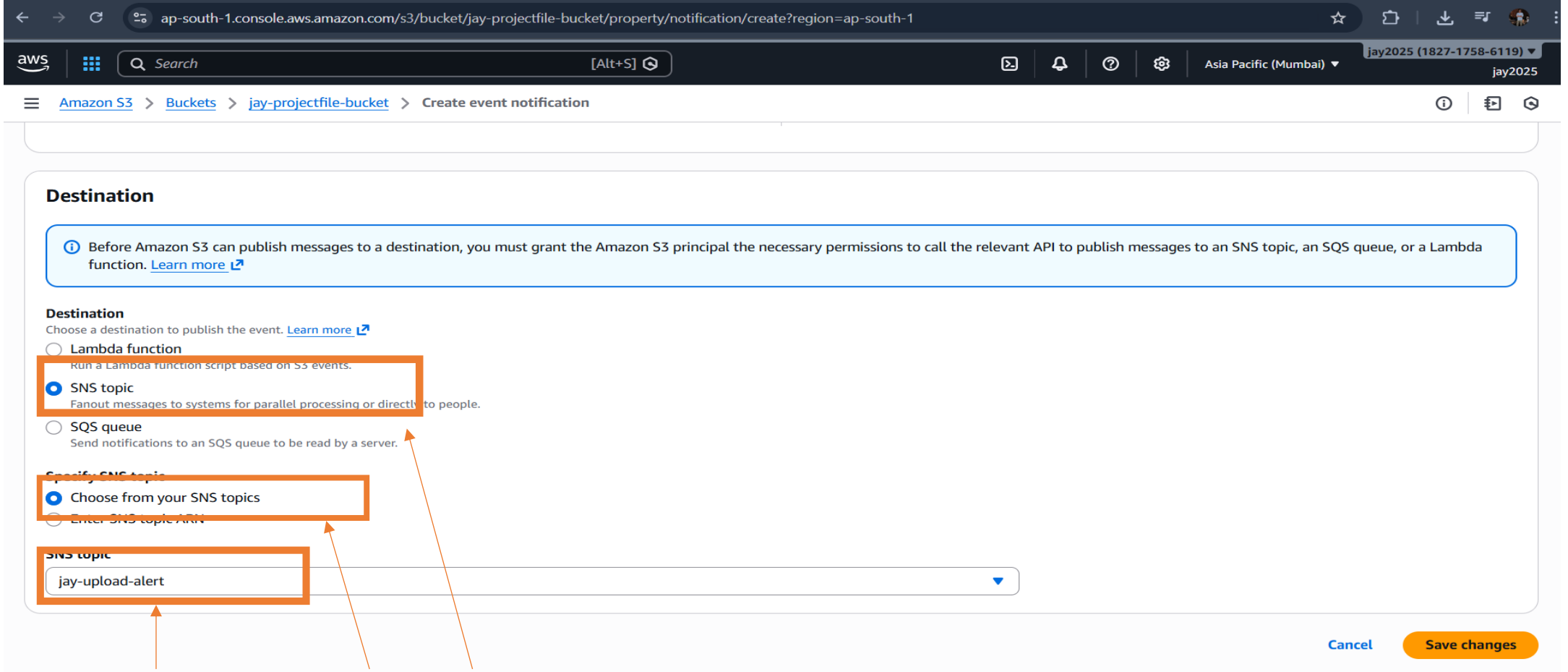
- Automatically triggers SNS alerts when a new file is uploaded.



- Event name: jay-notify-onuploads
- Prefix: uploads/ — triggers only for files in the uploads folder
- Event type: All object create events
- Destination: SNS topic

Why Use This Step

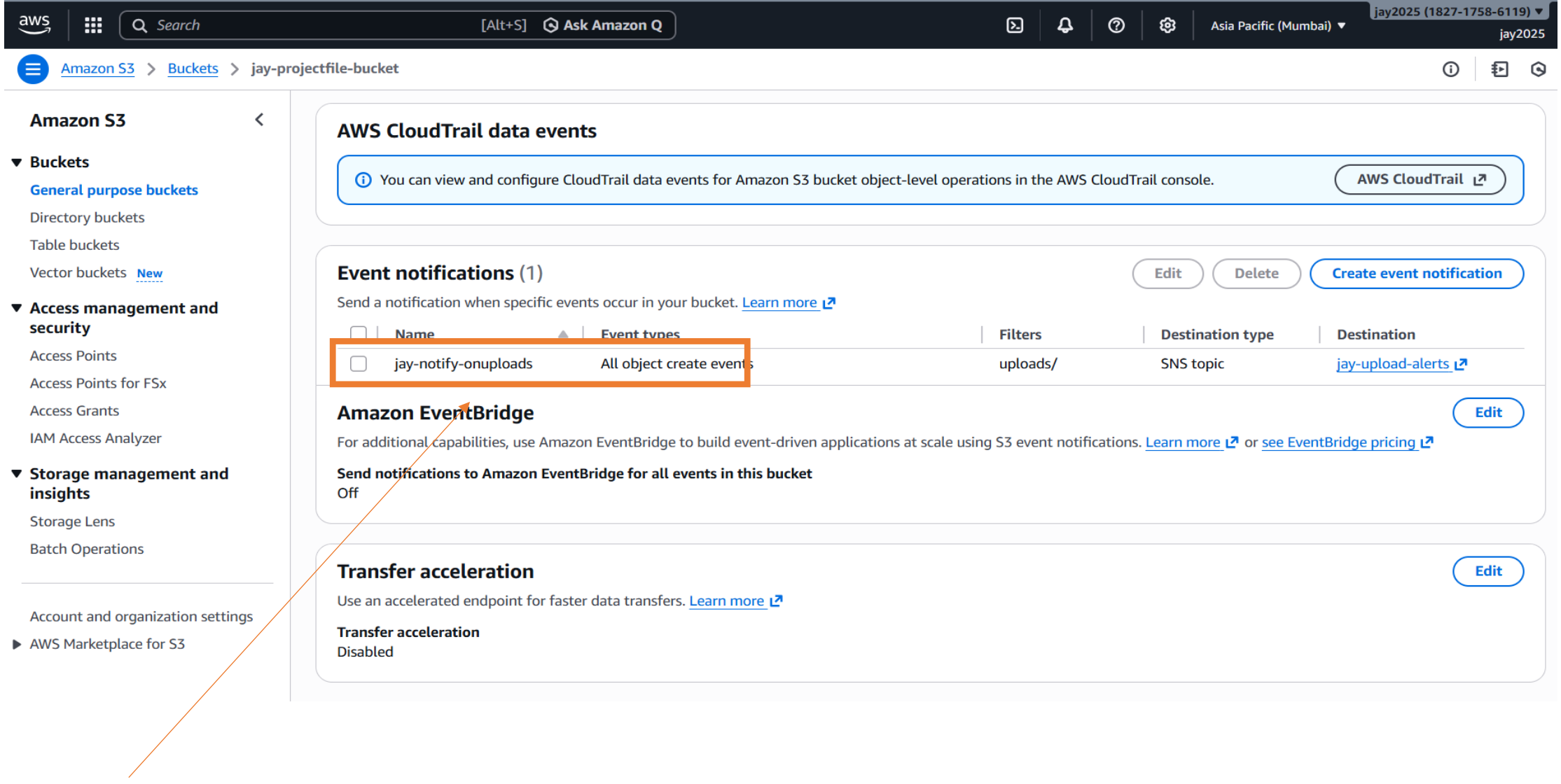
- Filters notifications to match specific file types and locations.
- Completes the automation: S3 → SNS → Email alert.



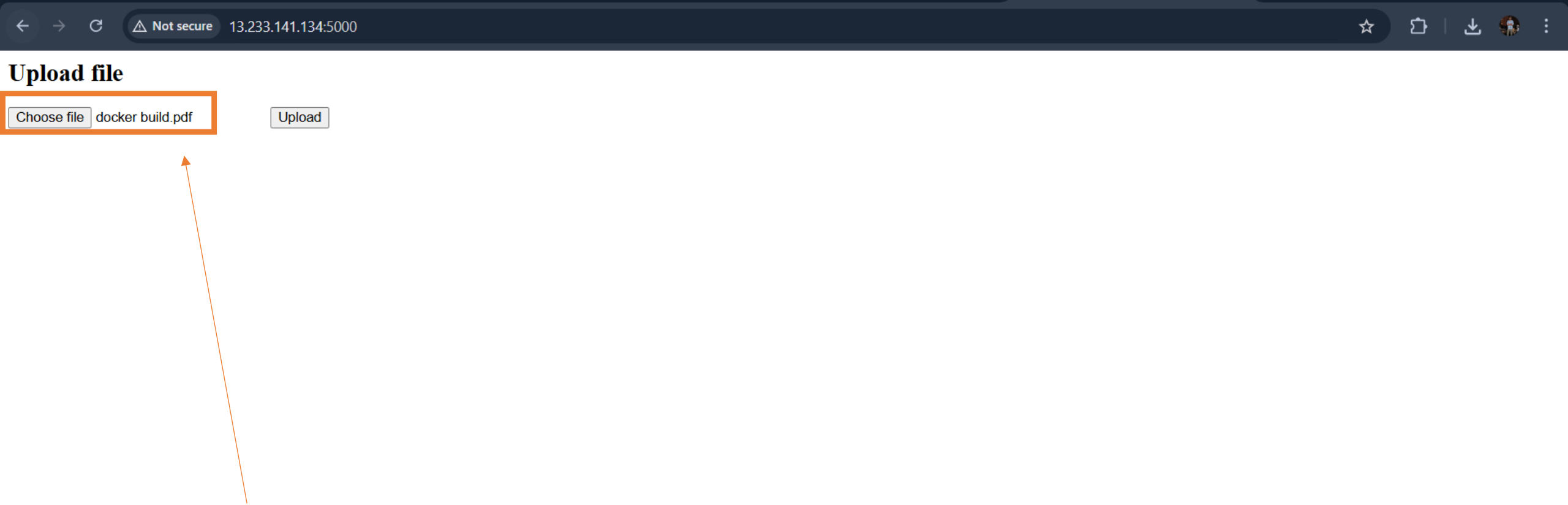
- Destination type: SNS topic
- Selected topic: **jay-upload-alert**
- Region: Asia Pacific (Mumbai)
- Ensure S3 has permission to publish to the SNS topic

Why Use This Step

- Connects S3 uploads directly to your SNS alert system


aws
Search
[Alt+S]
Ask Amazon Q
Asia Pacific (Mumbai)
jay2025 (1827-1758-6119)
jay2025
Amazon S3 > Buckets > jay-projectfile-bucket
Amazon S3
Buckets
General purpose buckets
Directory buckets
Table buckets
Vector buckets New
Access management and security
Access Points
Access Points for FSx
Access Grants
IAM Access Analyzer
Storage management and insights
Storage Lens
Batch Operations
Account and organization settings
AWS Marketplace for S3
AWS CloudTrail data events
You can view and configure CloudTrail data events for Amazon S3 bucket object-level operations in the AWS CloudTrail console.
AWS CloudTrail
Event notifications (1)
Edit
Delete
Create event notification
Send a notification when specific events occur in your bucket. Learn more
Name
Event types
Filters
Destination type
Destination
jay-notify-onuploads
All object create event
uploads/
SNS topic
jay-upload-alerts
Amazon EventBridge
Edit
For additional capabilities, use Amazon EventBridge to build event-driven applications at scale using S3 event notifications. Learn more or see EventBridge pricing
Send notifications to Amazon EventBridge for all events in this bucket
Off
Transfer acceleration
Edit
Use an accelerated endpoint for faster data transfers. Learn more
Transfer acceleration
Disabled

Done



File select and upload to s3

AWS Notifications <no-reply@sns.amazonaws.com>

Wed, Dec 31, 2025, 4:49 PM (1 day ago)

to me

{"Records":[{"eventVersion":"2.1","eventSource":"aws:s3","awsRegion":"ap-south-1","eventTime":"2025-12-31T11:19:26.525Z","eventName":"ObjectCreated:Put","userIdentity":{"principalId":"AWS:AROASVCWRDLDVHUJPX6DQ:i-000baed15445bf0d1"},"requestParameters":{"sourceIPAddress":"13.233.99.175"},"responseElements":{"x-amz-request-id":"SETKV9ZDKX2WSG6X","x-amz-id-2":"0mCko17VhRldW2qlK0NBrRJB5LktdwRY/Ct/Ga5Eq28f+yvNyLNDXpJFFGjirryytKPYf8lURViYdcafBskM0LHzNr8kAgZz"},"s3":{"s3SchemaVersion":"1.0","configurationId":"jay-notify-onuploads","bucket":{"name":"jay-projectfile-bucket","ownerIdentity":{"principalId":"ACTB0XIJLYM1B"},"arn":"arn:aws:s3:::jay-projectfile-bucket"},"object":{"key":"uploads/docker+build.pdf","size":307077,"eTag":"e5ca5817d8f5d047833e7c5b56c91f73","sequencer":"00695506BE74A27B4A"}}}]}

•••

Reply Forward

An email notification is received from AWS SNS.

Thank you.

Presented By
JAY