Building a Secure VPC Architecture on AWS: Public & Private Subnets, NAT, and Bastion Host

When it comes to designing a **secure and scalable network on AWS**, using a custom **Virtual Private Cloud (VPC)** with properly segmented **public and private subnets** is a must. In this blog, I'll walk you through how I built a secure VPC architecture using:

- Public & Private Subnets
- Bastion Host for secure access
- NAT Gateway for private instance internet access
- Internet Gateway and Route Tables
- Custom Security Groups and NACLs

Let's dive in! 🏊

🧠 Why This Architecture?

In production, not all resources should be exposed to the internet. For example:

- Web servers may need public access (public subnet)
- Databases must stay private (private subnet)

This project sets up **a VPC with both public and private subnets**, where:

- A **bastion host** in the public subnet is used to SSH into private EC2 instances
- A **NAT Gateway** allows private instances to access the internet (e.g., for software updates) **without exposing them publicly**

🧱 Project Architecture

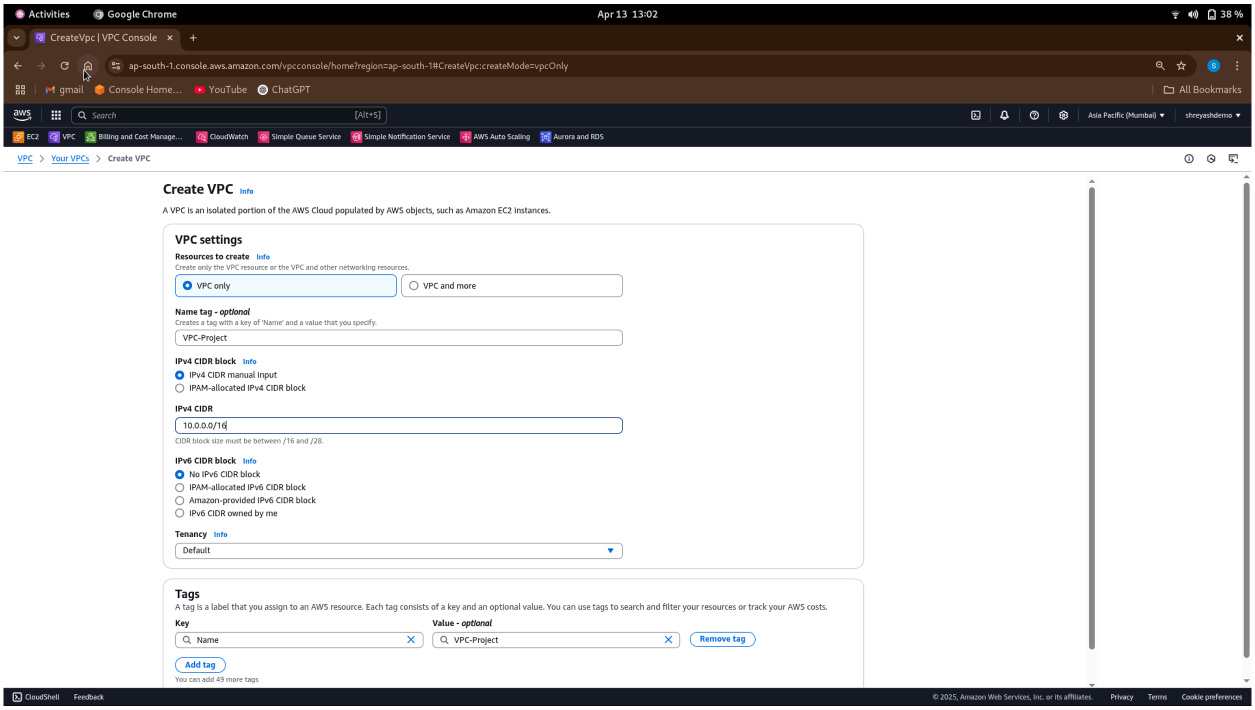
Here’s what the final architecture looks like:

```
                          +----------------------+
                         |    Internet Gateway  |
                         +----------+-----------+
                                    |
                         +----------v-----------+
                         |      Public Subnet  |
                         |                     |
                         |   Bastion Host (EC2) |
                         +----------+-----------+
                                           |
               +-----------------------+-----------------------+
               |                                               |
  +-----------v------------+          +-------------v-----------+
  |     Route Table        |          |     Private Subnet     |
  |  (with NAT gateway)    |              |                    |
  +-----------+------------+          |  Private EC2 Instance   |
               |                         +-------------+-----------+
  +-----------v------------+                                    |
  |      NAT Gateway       |          <---------------------------------+
  | (in public subnet)     |
  +------------------------+
```

🛠️ Step-by-Step Implementation

✅ Step 1: Create a Custom VPC

- Go to **VPC > Your VPCs > Create VPC**
  - Name: `MySecureVPC`
  - CIDR: `10.0.0.0/16`



ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#CreateVpc:createMode=vpcOnly

VPC > Your VPCs > Create VPC

Create VPC

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create
Create only the VPC resource or the VPC and other networking resources.
● VPC only ○ VPC and more

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

IPv4 CIDR block
● IPv4 CIDR manual input
○ IPAM-allocated IPv4 CIDR block

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

IPv6 CIDR block
● No IPv6 CIDR block
○ IPAM-allocated IPv6 CIDR block
○ Amazon-provided IPv6 CIDR block
○ IPv6 CIDR owned by me

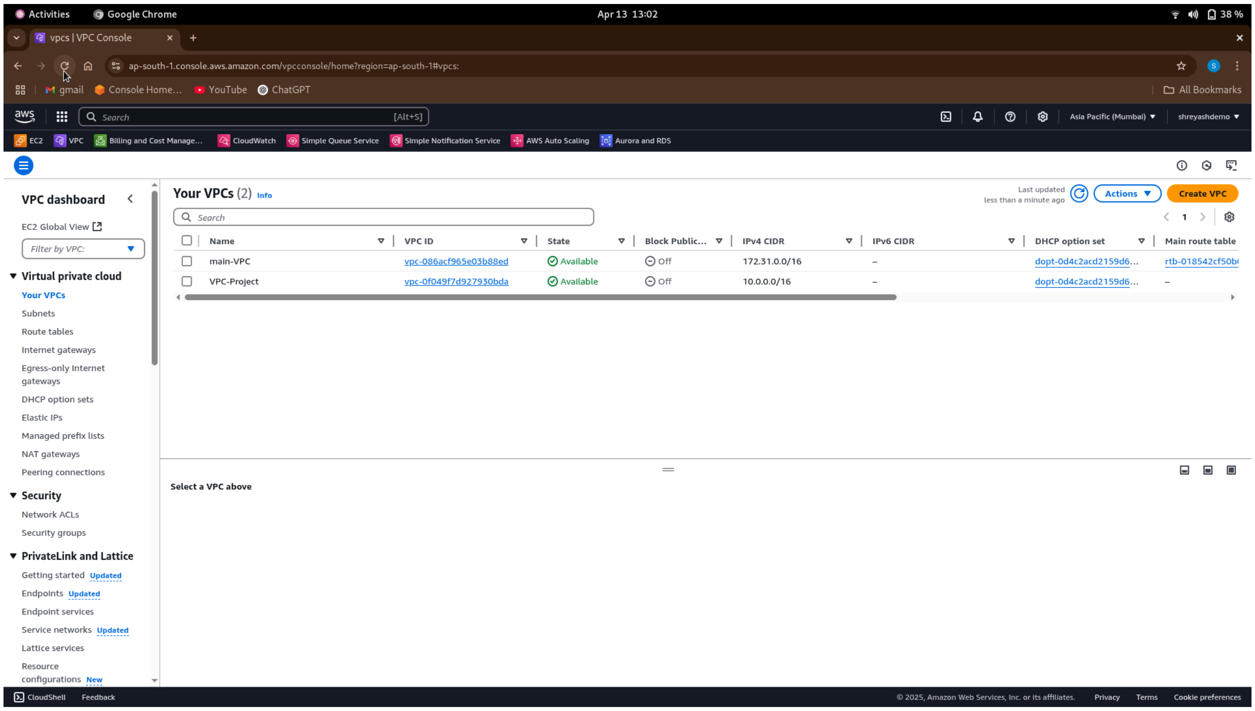
Tenancy
Default

Tags

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

Key	Value - *optional*
Name	VPC-Project

You can add 49 more tags



ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#vpcs:

Your VPCs (2)

Last updated
less than a minute ago

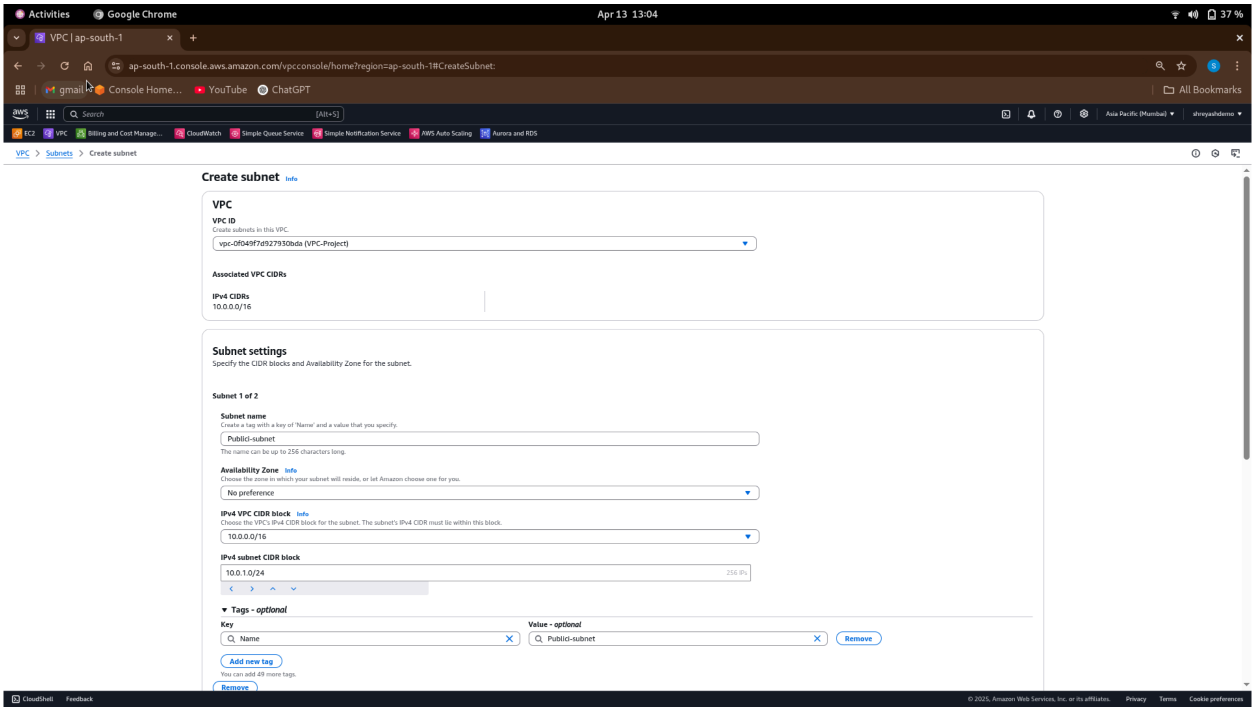
☐	Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main route table
☐	main-VPC	vpc-086acf965e03b88ed	Available	Off	172.31.0.0/16	–	dopt-0d4c2acd2159d6...	rtb-018542cf50b
☐	VPC-Project	vpc-0f049f7d927930bda	Available	Off	10.0.0.0/16	–	dopt-0d4c2acd2159d6...	–

Select a VPC above

✅ Step 2: Create Subnets

Public Subnet

- Name: `PublicSubnet`
- CIDR: `10.0.1.0/24`
- Availability Zone: `ap-south-1a`



Create subnet Info

VPC

VPC ID
Create subnets in this VPC.
vpc-0f049f7d927930bda (VPC-Project)

Associated VPC CIDRs

IPv4 CIDRs
10.0.0.0/16

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

Subnet 1 of 2

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
Publici-subnet
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.
No preference

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.
10.0.0.0/16

IPv4 subnet CIDR block
10.0.1.0/24 256 IPs

▼ Tags - *optional*

Key	Value - *optional*	
Name	Publici-subnet	Remove

Add new tag
You can add 49 more tags.

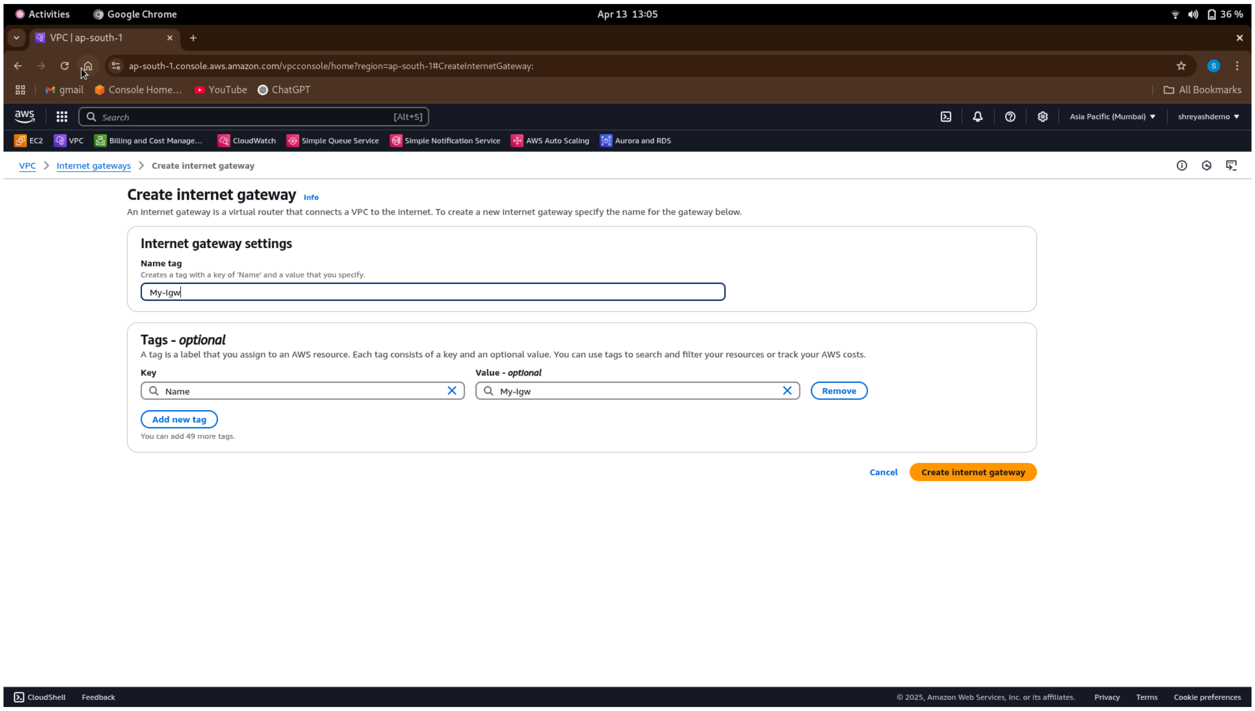
Remove

Private Subnet

- Name: `PrivateSubnet`
- CIDR: `10.0.2.0/24`
- Availability Zone: `ap-south-1a`

✅ Step 3: Internet Gateway (IGW)

- Create and **attach** it to your VPC



Create internet gateway Info

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

Internet gateway settings

Name tag: My-Igw

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

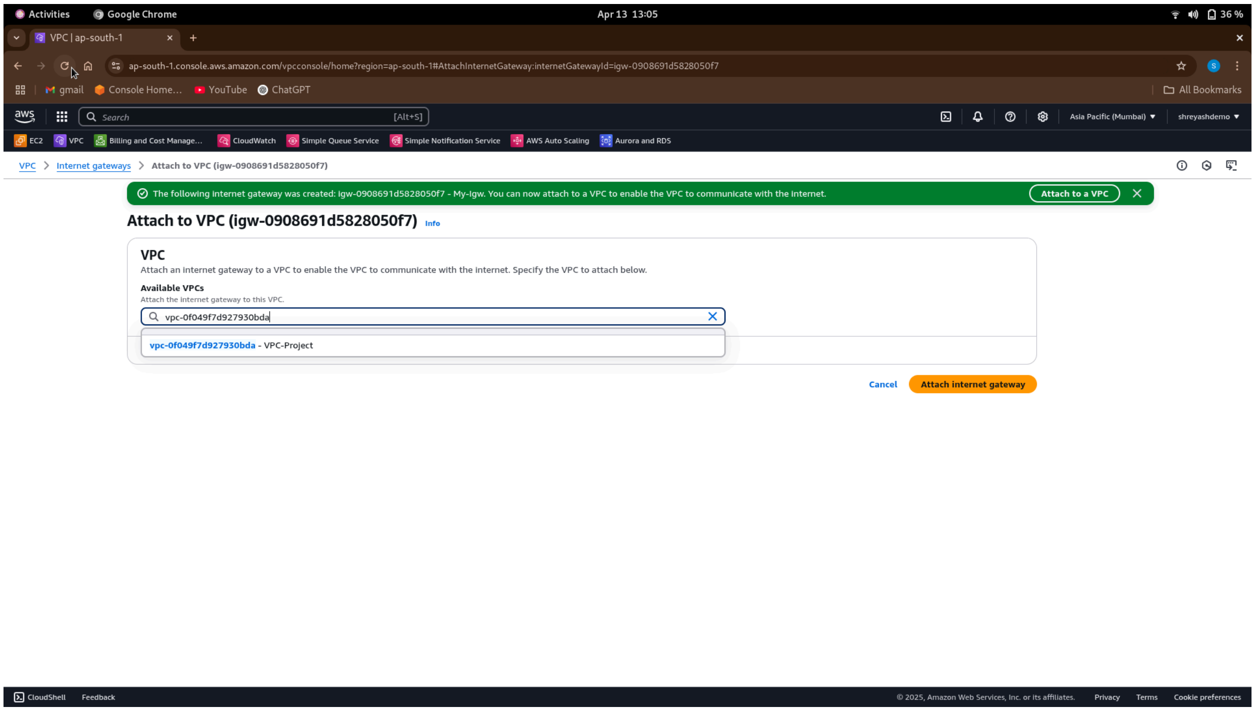
Tags - *optional*

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.

Key	Value - *optional*
Name	My-Igw

You can add 49 more tags.

-


Activities
Google Chrome
Apr 13 13:05
36 %
VPC | ap-south-1
ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#AttachInternetGateway:internetGatewayId=igw-0908691d5828050f7
gmail
Console Home...
YouTube
ChatGPT
All Bookmarks
aws
Search
[Alt+S]
Asia Pacific (Mumbai) ▼
shreyashdemo ▼
EC2
VPC
Billing and Cost Manage...
CloudWatch
Simple Queue Service
Simple Notification Service
AWS Auto Scaling
Aurora and RDS
VPC > Internet gateways > Attach to VPC (igw-0908691d5828050f7)

The following internet gateway was created: igw-0908691d5828050f7 - My-Igw. You can now attach to a VPC to enable the VPC to communicate with the internet.

Attach to VPC (igw-0908691d5828050f7) 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: vpc-0f049f7d927930bda

Attach the internet gateway to this VPC.

vpc-0f049f7d927930bda - VPC-Project

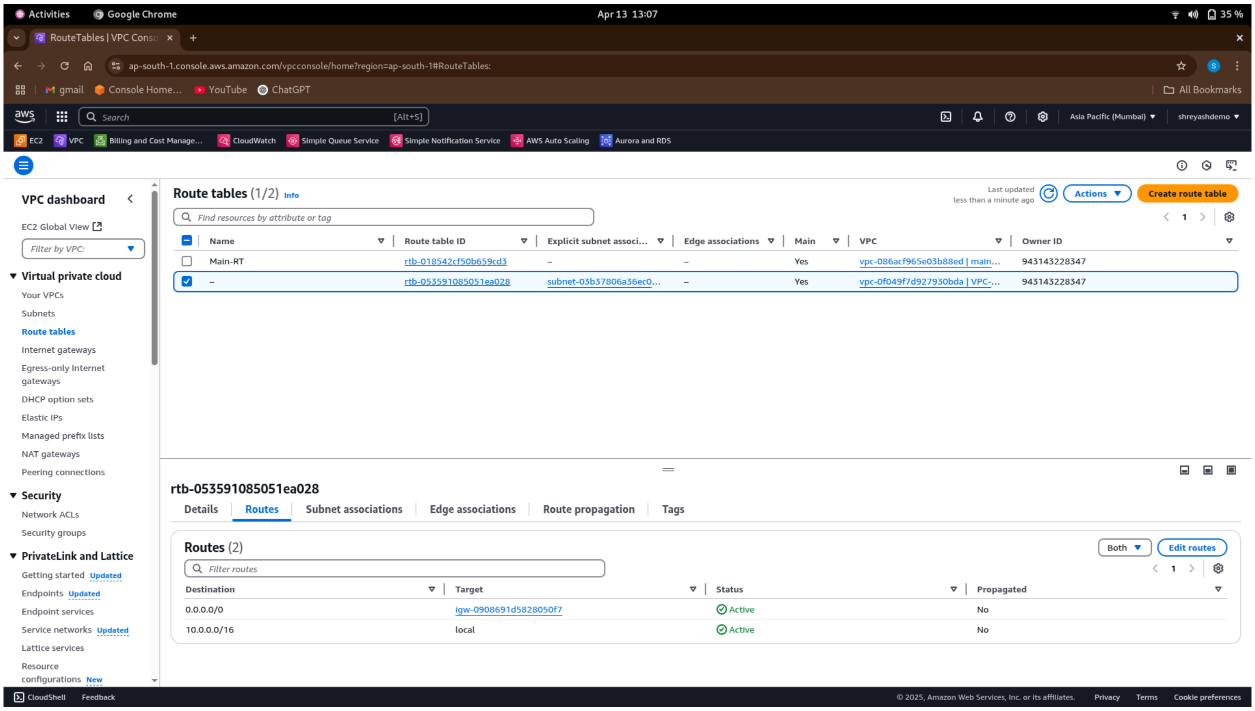
✅ Step 4: Route Tables

Public Route Table

- Associate it with `PublicSubnet`
- Add Route: `0.0.0.0/0` → Target: `Internet Gateway`

Private Route Table

- Associate it with `PrivateSubnet`
- Add Route: `0.0.0.0/0` → Target: `NAT Gateway` (later step)

- 

Route tables (1/2) Info

	Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC	Owner ID
□	Main-RT	rtb-018542cf50b659cd3	–	–	Yes	vpc-086acf965e03b88ed | main...	943143228347
☑	–	rtb-053591085051ea028	subnet-03b37806a36ec0...	–	Yes	vpc-0f049f7d927930bda | VPC-...	943143228347

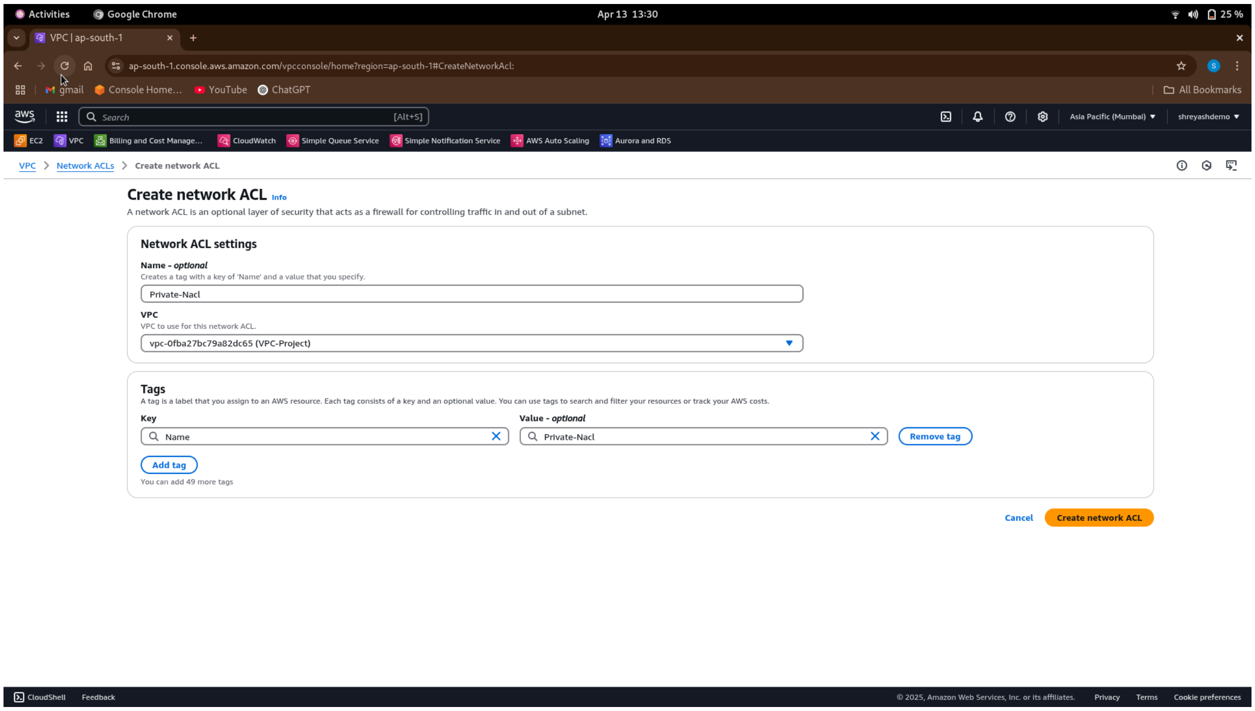
rtb-053591085051ea028

Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0908691d5828050f7	✓ Active	No
10.0.0.0/16	local	✓ Active	No

🧱 Step 5: Network ACLs (NACLs)

Creating a Network ACL for a Private Subnet



Create network ACL Info

A network ACL is an optional layer of security that acts as a firewall for controlling traffic in and out of a subnet.

Network ACL settings

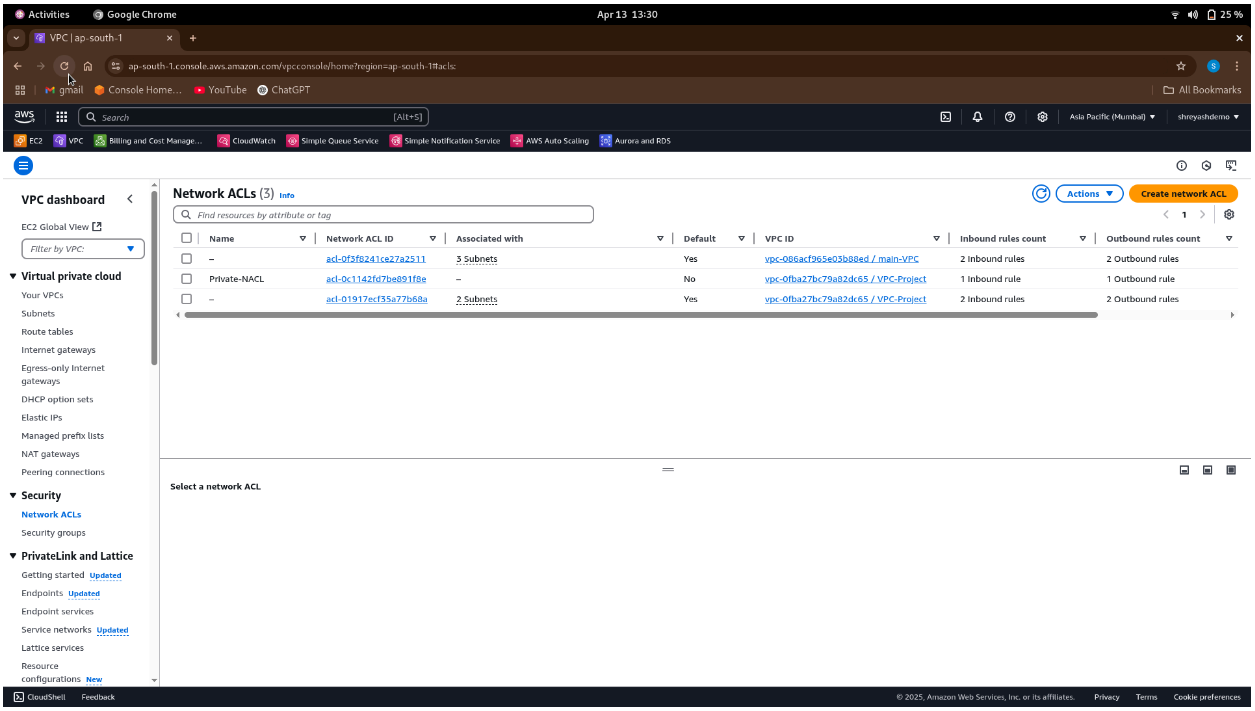
Name - *optional*
Creates a tag with a key of 'Name' and a value that you specify.
Private-Nacl

VPC
VPC to use for this network ACL.
vpc-0fba27bc79a82dc65 (VPC-Project)

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: Name
Value - *optional*: Private-Nacl

You can add 49 more tags



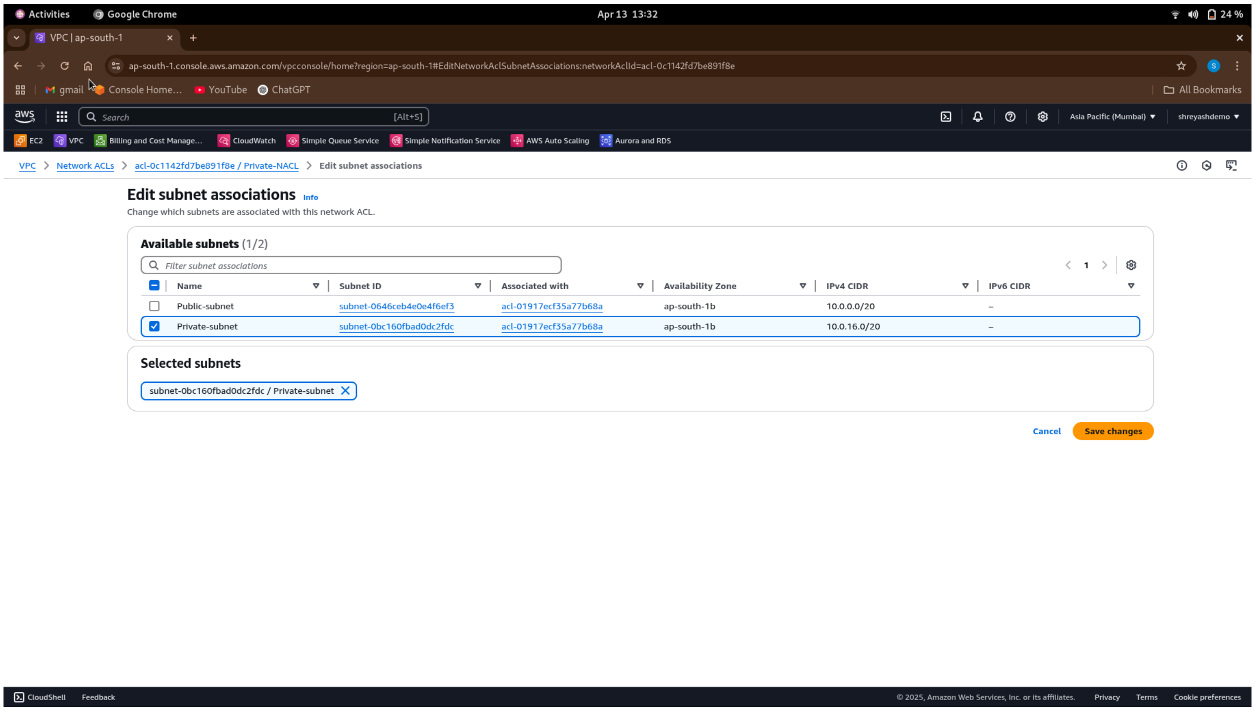
Network ACLs (3) Info

Name	Network ACL ID	Associated with	Default	VPC ID	Inbound rules count	Outbound rules count
–	acl-0f3f8241ce27a2511	3 Subnets	Yes	vpc-086acf965e03b88ed / main-VPC	2 Inbound rules	2 Outbound rules
Private-NACL	acl-0c1142fd7be891f8e	–	No	vpc-0fba27bc79a82dc65 / VPC-Project	1 Inbound rule	1 Outbound rule
–	acl-01917ecf35a77b68a	2 Subnets	Yes	vpc-0fba27bc79a82dc65 / VPC-Project	2 Inbound rules	2 Outbound rules

Select a network ACL

Private NACL

- Allow inbound SSH (22), HTTP (80), HTTPS (443)
- Allow all outbound traffic



ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#EditNetworkAclSubnetAssociations:networkAclId=acl-0c1142fd7be891f8e

VPC > Network ACLs > acl-0c1142fd7be891f8e / Private-NACL > Edit subnet associations

Edit subnet associations

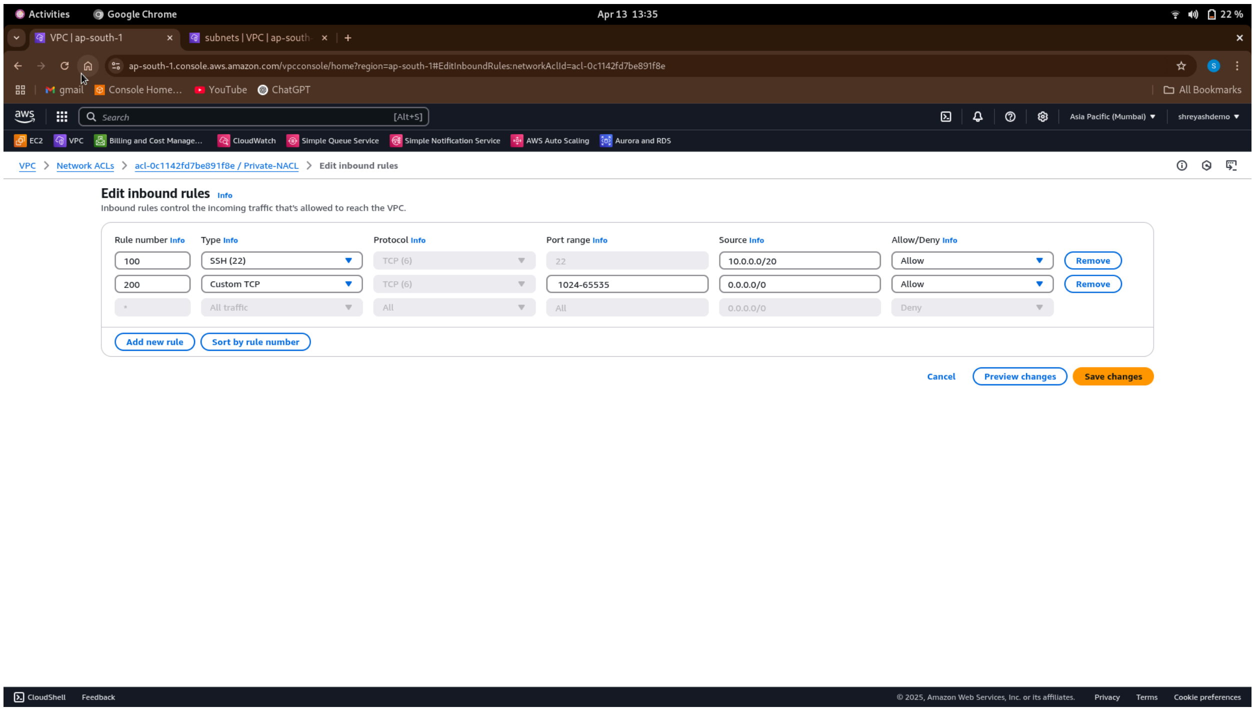
Change which subnets are associated with this network ACL.

Available subnets (1/2)

	Name	Subnet ID	Associated with	Availability Zone	IPv4 CIDR	IPv6 CIDR
□	Public-subnet	subnet-0646ceb4e0e4f6ef3	acl-01917ecf35a77b68a	ap-south-1b	10.0.0.0/20	–
☑	Private-subnet	subnet-0bc160fbad0dc2fdc	acl-01917ecf35a77b68a	ap-south-1b	10.0.16.0/20	–

Selected subnets

subnet-0bc160fbad0dc2fdc / Private-subnet

- 

ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#EditInboundRules:networkAclId=acl-0c1142fd7be891f8e

VPC > Network ACLs > acl-0c1142fd7be891f8e / Private-NACL > Edit inbound rules

Edit inbound rules

Inbound rules control the incoming traffic that's allowed to reach the VPC.

Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	SSH (22)	TCP (6)	22	10.0.0.0/20	Allow
200	Custom TCP	TCP (6)	1024-65535	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

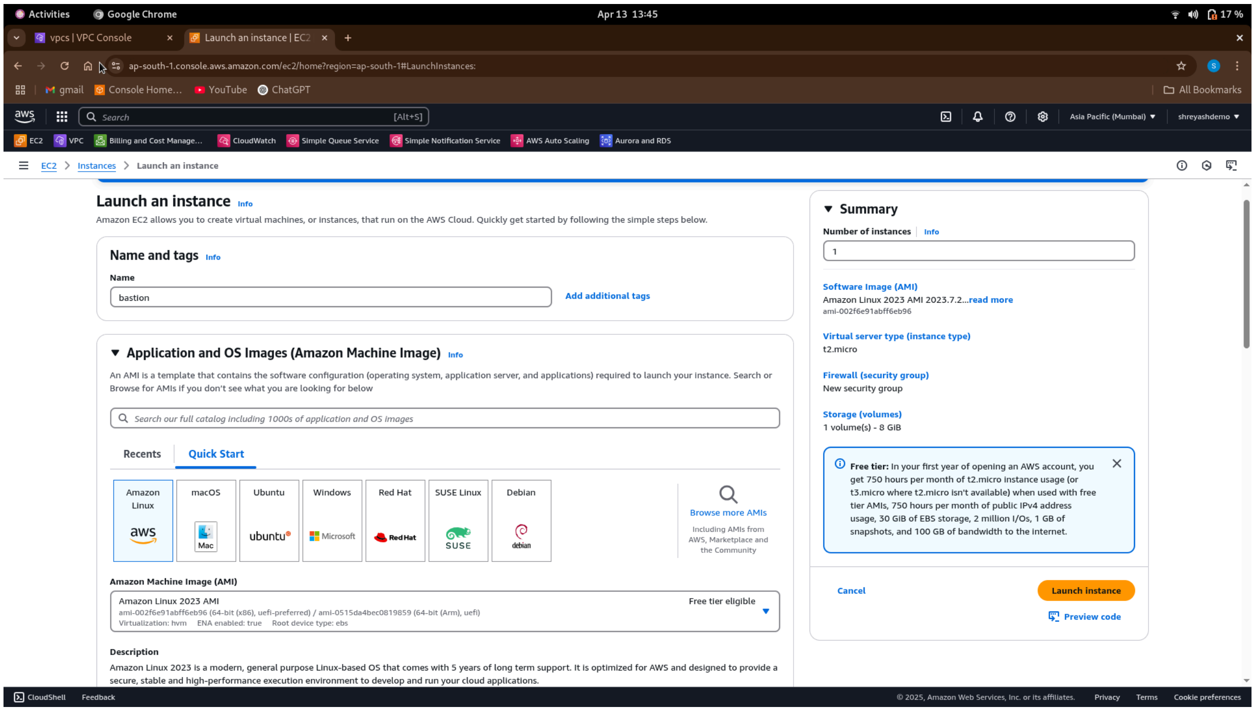
🖥️ Step 6: EC2 Instances

Bastion Host

- Launch in **PublicSubnet**
- Assign **Elastic IP**
- Attach **Bastion SG**
- Add SSH key pair

Private EC2

- Launch in **PrivateSubnet**
- Attach **Private SG**
- No public IP



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: bastion

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 | Debian

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI — Free tier eligible
ami-002f6e91abff6eb96 (64-bit (x86), uefi-preferred) / ami-0515da4bec0819859 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

▼ Summary

Number of instances | Info: 1

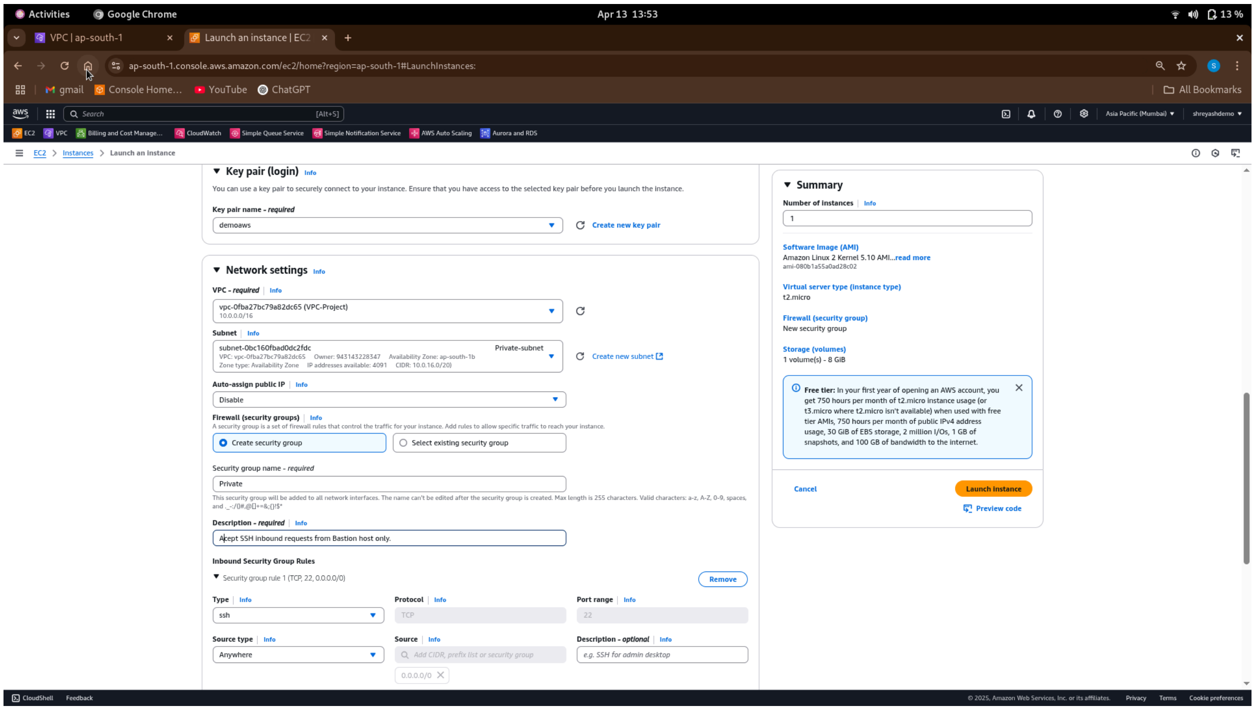
Software Image (AMI)
Amazon Linux 2023 AMI 2023.7.2...read more
ami-002f6e91abff6eb96

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

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 t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.



Key pair (login) Info

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*: demoaws

Create new key pair

Network settings Info

VPC - *required*: vpc-0fba27bc79a82dc65 (VPC-Project) 10.0.0.0/16

Subnet: subnet-0bc160fbad0dc2fdc Private-subnet
VPC: vpc-0fba27bc79a82dc65 Owner: 943143228347 Availability Zone: ap-south-1b
Zone type: Availability Zone IP addresses available: 4091 CIDR: 10.0.16.0/20)

Create new subnet

Auto-assign public IP: Disable

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

● Create security group ○ Select existing security group

Security group name - *required*: Private

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 ._-:/()#,@[]+=&;{}!$*

Description - *required*: Acept SSH inbound requests from Bastion host only.

Inbound Security Group Rules

Security group rule 1 (TCP, 22, 0.0.0.0/0)

Remove

Type	Protocol	Port range
ssh	TCP	22

Source type	Source	Description - *optional*
Anywhere	Add CIDR, prefix list or security group 0.0.0.0/0	e.g. SSH for admin desktop

Summary

Number of instances: 1

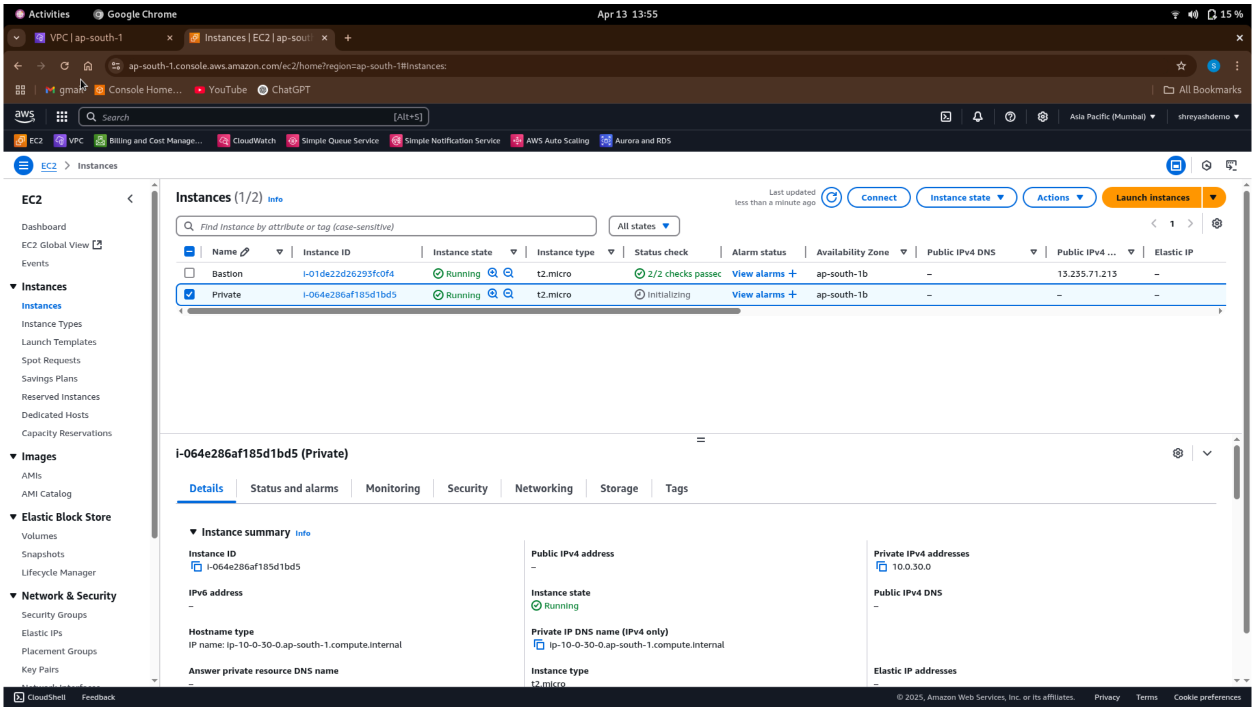
Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI...read more
ami-080b1a55a0ad28c02

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

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 t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.



Instances (1/2) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
□	Bastion	i-01de22d26293fc0f4	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	–	13.235.71.213	–
☑	Private	i-064e286af185d1bd5	Running	t2.micro	Initializing	View alarms +	ap-south-1b	–	–	–

i-064e286af185d1bd5 (Private)

Instance summary Info

Instance ID: i-064e286af185d1bd5

Public IPv4 address: –

Private IPv4 addresses: 10.0.30.0

IPv6 address: –

Instance state: Running

Public IPv4 DNS: –

Hostname type: IP name: ip-10-0-30-0.ap-south-1.compute.internal

Private IP DNS name (IPv4 only): ip-10-0-30-0.ap-south-1.compute.internal

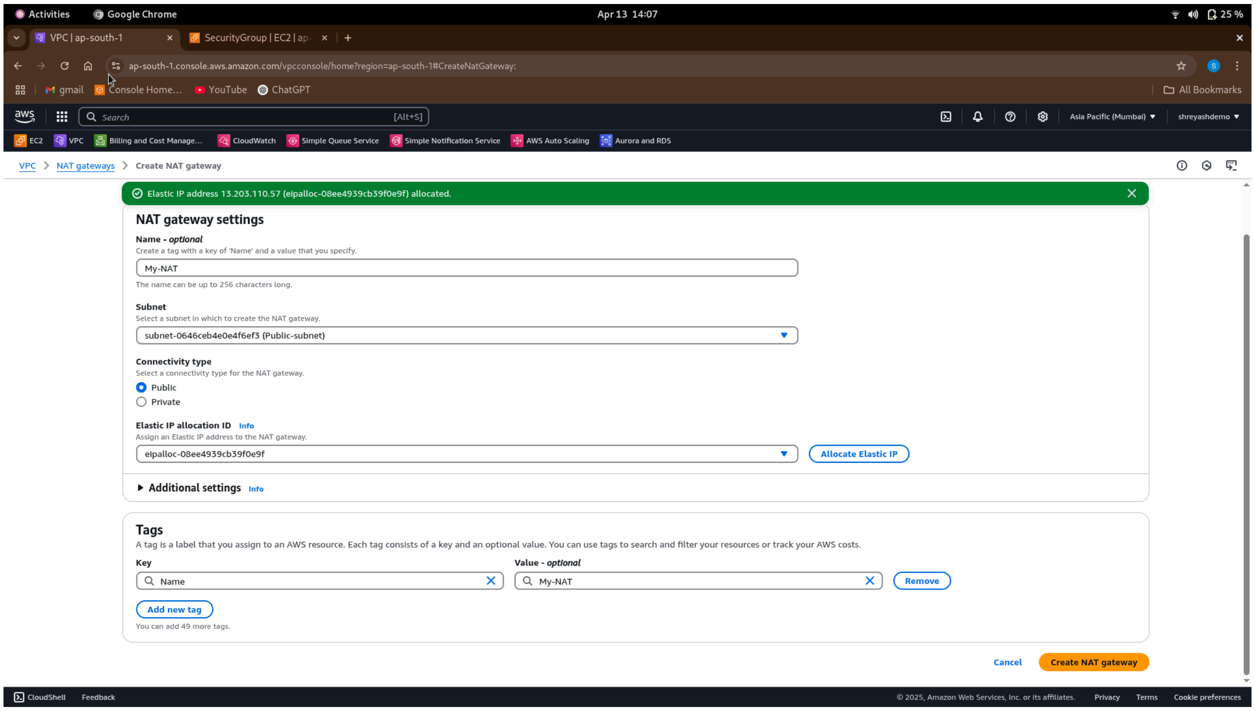
Answer private resource DNS name: –

Instance type: t2.micro

Elastic IP addresses: –

🌐 Step 7: NAT Gateway

- Create an Elastic IP
- Create NAT Gateway in `PublicSubnet`
- Assign it to the **Private Route Table**



ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#CreateNatGateway:

VPC > NAT gateways > Create NAT gateway

Elastic IP address 13.203.110.57 (eipalloc-08ee4939cb39f0e9f) allocated.

NAT gateway settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.
My-NAT
The name can be up to 256 characters long.

Subnet
Select a subnet in which to create the NAT gateway.
subnet-0646ceb4e0e4f6ef3 (Public-subnet)

Connectivity type
Select a connectivity type for the NAT gateway.
● Public
○ Private

Elastic IP allocation ID Info
Assign an Elastic IP address to the NAT gateway.
eipalloc-08ee4939cb39f0e9f
Allocate Elastic IP

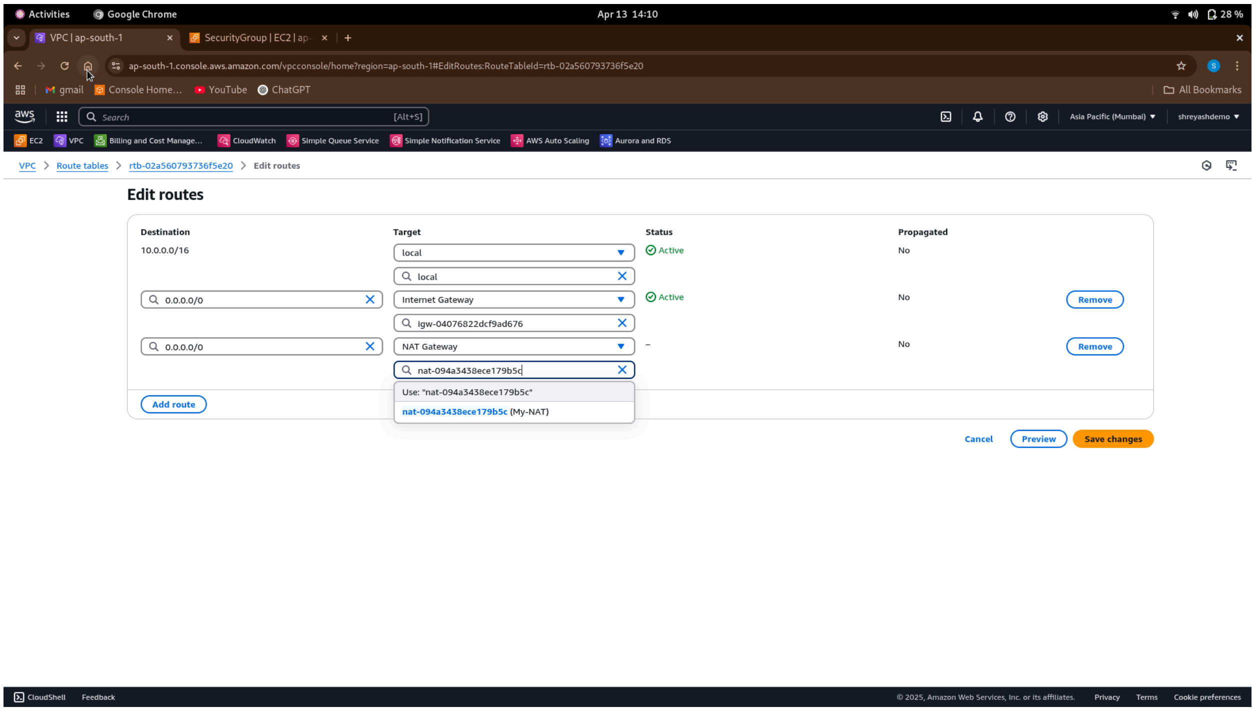
▶ **Additional settings** Info

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

Key	Value - *optional*	
Name	My-NAT	Remove

Add new tag
You can add 49 more tags.

Cancel Create NAT gateway



ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#EditRoutes:RouteTableId=rtb-02a560793736f5e20

VPC > Route tables > rtb-02a560793736f5e20 > Edit routes

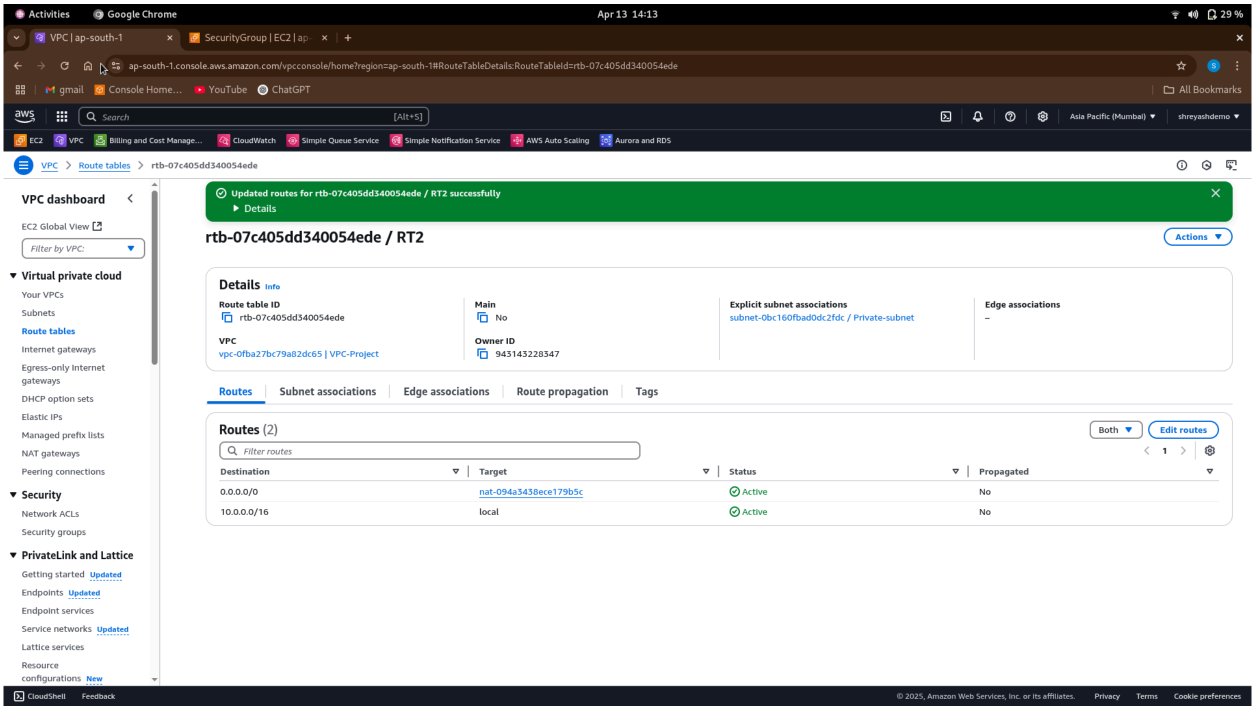
Edit routes

Destination	Target	Status	Propagated	
10.0.0.0/16	local local	Active	No	
0.0.0.0/0	Internet Gateway igw-04076822dcf9ad676	Active	No	Remove
0.0.0.0/0	NAT Gateway nat-094a3438ece179b5c	–	No	Remove

Use: "nat-094a3438ece179b5c"
nat-094a3438ece179b5c (My-NAT)

Add route

Cancel Preview Save changes



Updated routes for rtb-07c405dd340054ede / RT2 successfully

rtb-07c405dd340054ede / RT2

Details

Route table ID: rtb-07c405dd340054ede

Main: No

Explicit subnet associations: subnet-0bc160fbad0dc2fdc / Private-subnet

Edge associations: –

VPC: vpc-0fba27bc79a82dc65 | VPC-Project

Owner ID: 943143228347

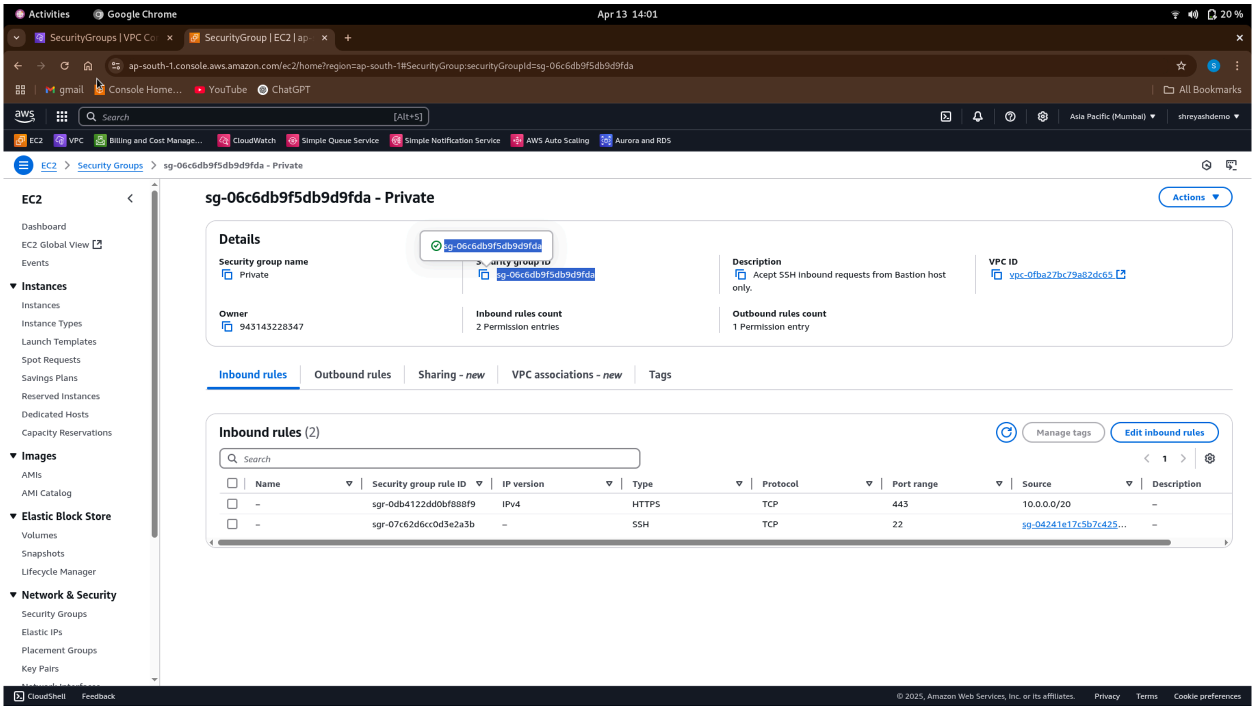
Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	nat-094a3438ece179b5c	Active	No
10.0.0.0/16	local	Active	No

🖥️ Step *: Rule of SG-bastion:

1. Select the private instance. In the Security tab, click the actionable security group link (for example, SG-Private).
2. From the VPC Dashboard, click Security Groups. Make note of the Group ID of the SG-Private security group.
3. Select the SG-bastion security group, switch to the Outbound rules tab, and click Edit outbound rules. Now that you have a private security group, you can restrict Outbound rules to instances using SG-Private. Configure the following:

- Type: SSH
- Protocol: TCP
- Port: 22
- Destination: Select Custom and then enter the security Security group ID of SG-Private



ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#SecurityGroup:securityGroupId=sg-06c6db9f5db9d9fda

EC2 > Security Groups > sg-06c6db9f5db9d9fda - Private

sg-06c6db9f5db9d9fda - Private

Details

sg-06c6db9f5db9d9fda

Security group name: Private

Security group ID: sg-06c6db9f5db9d9fda

Description: Acept SSH inbound requests from Bastion host only.

VPC ID: vpc-0fba27bc79a82dc65

Owner: 943143228347

Inbound rules count: 2 Permission entries

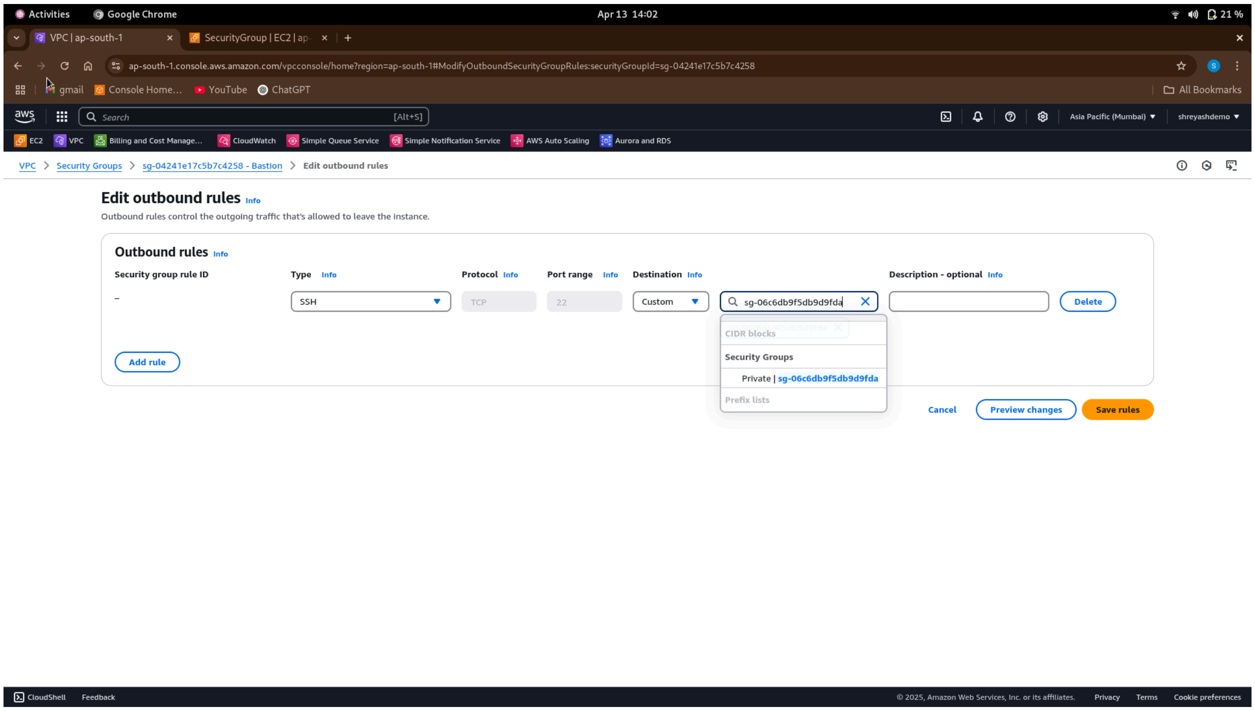
Outbound rules count: 1 Permission entry

Inbound rules | Outbound rules | Sharing - *new* | VPC associations - *new* | Tags

Inbound rules (2)

Manage tags | Edit inbound rules

□	Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
□	–	sgr-0db4122dd0bf888f9	IPv4	HTTPS	TCP	443	10.0.0.0/20	–
□	–	sgr-07c62d6cc0d3e2a3b	–	SSH	TCP	22	sg-04241e17c5b7c425...	–



ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#ModifyOutboundSecurityGroupRules:securityGroupId=sg-04241e17c5b7c4258

VPC > Security Groups > sg-04241e17c5b7c4258 - Bastion > Edit outbound rules

Edit outbound rules Info

Outbound rules control the outgoing traffic that's allowed to leave the instance.

Outbound rules Info

Security group rule ID	Type	Protocol	Port range	Destination		Description - optional	
–	SSH	TCP	22	Custom	sg-06c6db9f5db9d9fda		Delete

CIDR blocks

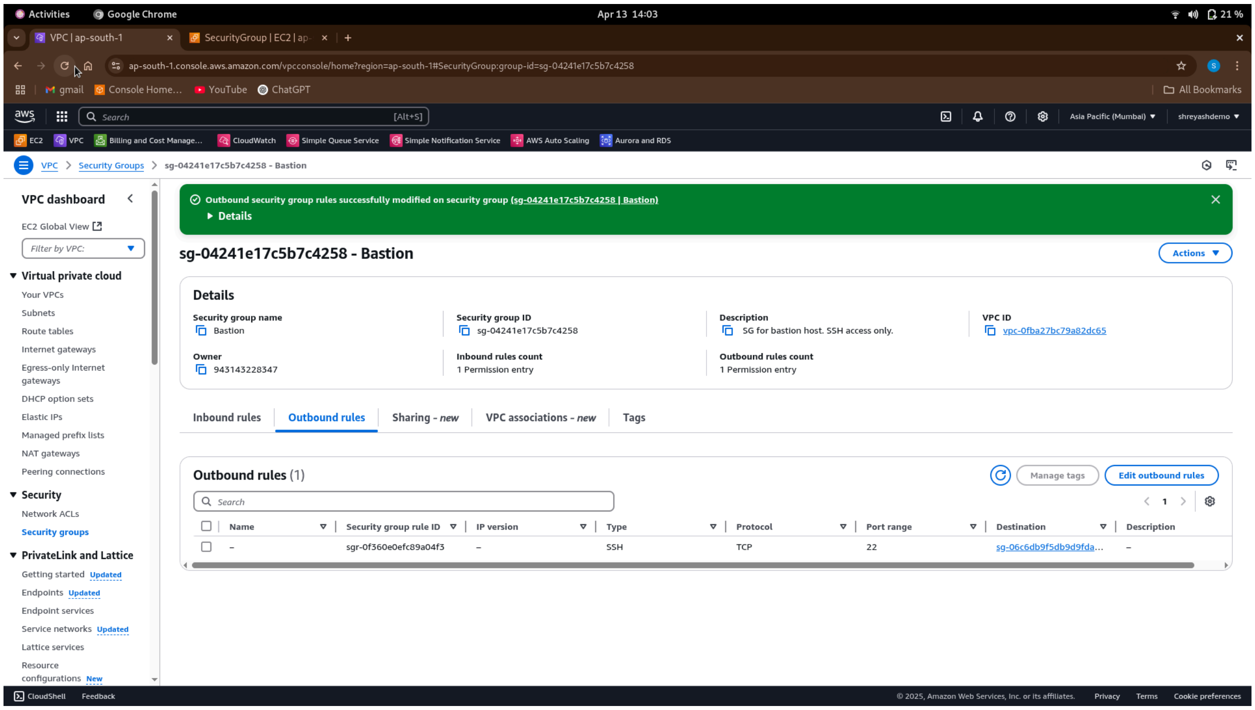
Security Groups

Private | sg-06c6db9f5db9d9fda

Prefix lists

Add rule

Cancel | Preview changes | Save rules



Outbound security group rules successfully modified on security group (sg-04241e17c5b7c4258 | Bastion)

Details

sg-04241e17c5b7c4258 - Bastion

Details

Security group name Bastion	Security group ID sg-04241e17c5b7c4258	Description SG for bastion host. SSH access only.	VPC ID vpc-0fba27bc79a82dc65
Owner 943143228347	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

Outbound rules (1)

☐	Name	Security group rule ID	IP version	Type	Protocol	Port range	Destination	Description
☐	–	sgr-0f360e0efc89a04f3	–	SSH	TCP	22	sg-06c6db9f5db9d9fda...	–

🔐 Step 9: SSH Access to Private EC2 (via Bastion)

1. SSH into the Bastion Host:

```
bashCopyEditssh -i bastion-key.pem ec2-user@<Elastic-IP>
```

2. From there, SSH into private instance:

```
bashCopyEditssh -i private-key.pem ec2-user@<Private-IP>
```

🧪 Test the Setup

✅ Can access Bastion via SSH
✅ Can’t access private instance directly
✅ Can SSH into private instance **only** via Bastion
✅ Private EC2 can ping google.com (thanks to NAT)

‘

🧠 Key Learnings

- **Bastion Hosts** are critical for secure SSH access
- **NAT Gateways** enable private instances to update software securely
- Route Tables and NACLs define traffic flow—plan them carefully
- **Least privilege** in security groups is always the best practice

---

🚀 What's Next?

I plan to extend this with:

- Load Balancers
- Auto Scaling Groups
- RDS in private subnets
- VPC Peering between environments

📌 Conclusion

This secure VPC setup is a strong foundation for production-ready cloud infrastructure. By isolating public and private workloads and routing traffic intelligently, you can build resilient and secure systems.