

# VPC project used in production

## VPC

Imagine you want to set up a private, secure, and isolated area in the cloud where you can run your applications and store your data. This is where a VPC comes into play.

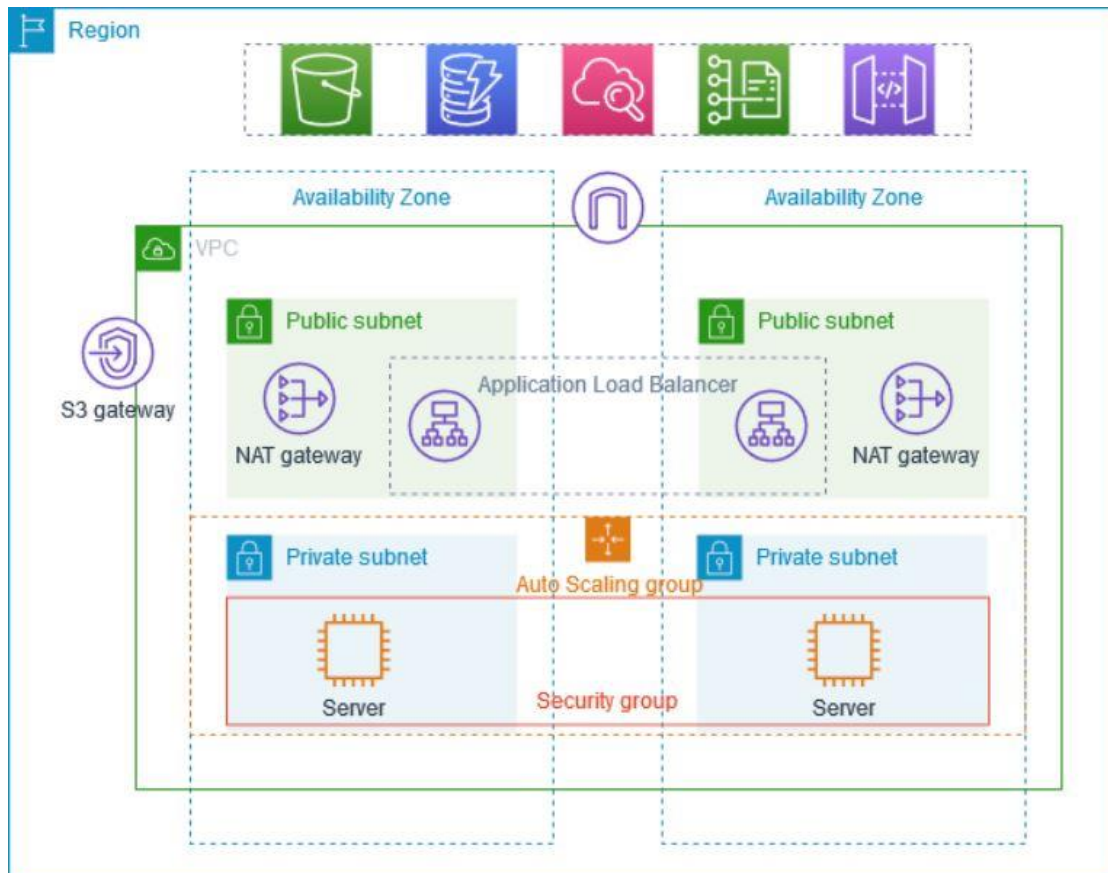
A VPC is a virtual network that you create in the cloud. It allows you to have your own private section of the internet, just like having your own network within a larger network. Within this VPC, you can create and manage various resources, such as servers, databases, and storage.

Think of it as having your own little "internet" within the bigger internet. This virtual network is completely isolated from other users' networks, so your data and applications are secure and protected.

Just like a physical network, a VPC has its own set of rules and configurations. You can define the IP address range for your VPC and create smaller subnetworks within it called subnets. These subnets help you organize your resources and control how they communicate with each other.

To connect your VPC to the internet or other networks, you can set up gateways or routers. These act as entry and exit points for traffic going in and out of your VPC. You can control the flow of traffic and set up security measures to protect your resources from unauthorized access.

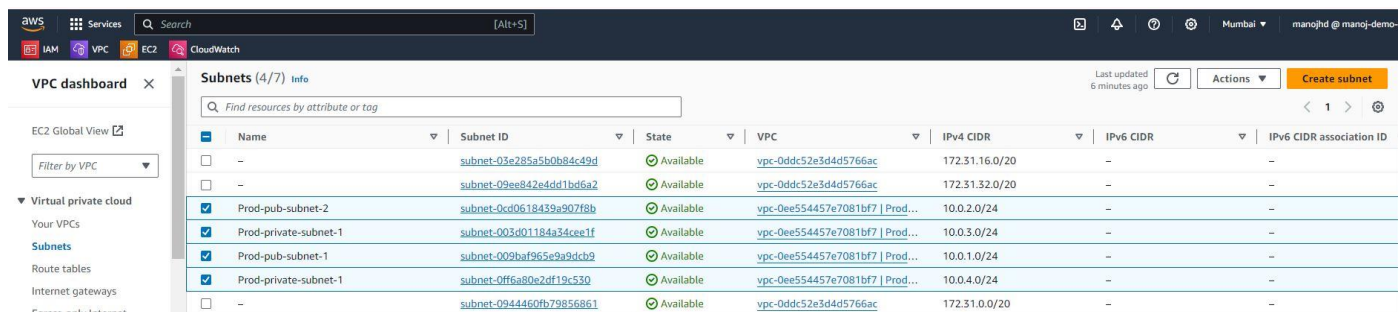
With a VPC, you have control over your network environment. You can define access rules, set up firewalls, and configure security groups to regulate who can access your resources and how they can communicate.



## Subnets

A subnet is a range of IP addresses in your VPC. A subnet must reside in a single Availability Zone. After you add subnets, you can deploy AWS resources in your VPC.

You can assign IP addresses, both IPv4 and IPv6, to your VPCs and subnets. You can also bring your public IPv4 and IPv6 GUA addresses to AWS and allocate them to resources in your VPC, such as EC2 instances, NAT gateways, and Network Load Balancers.



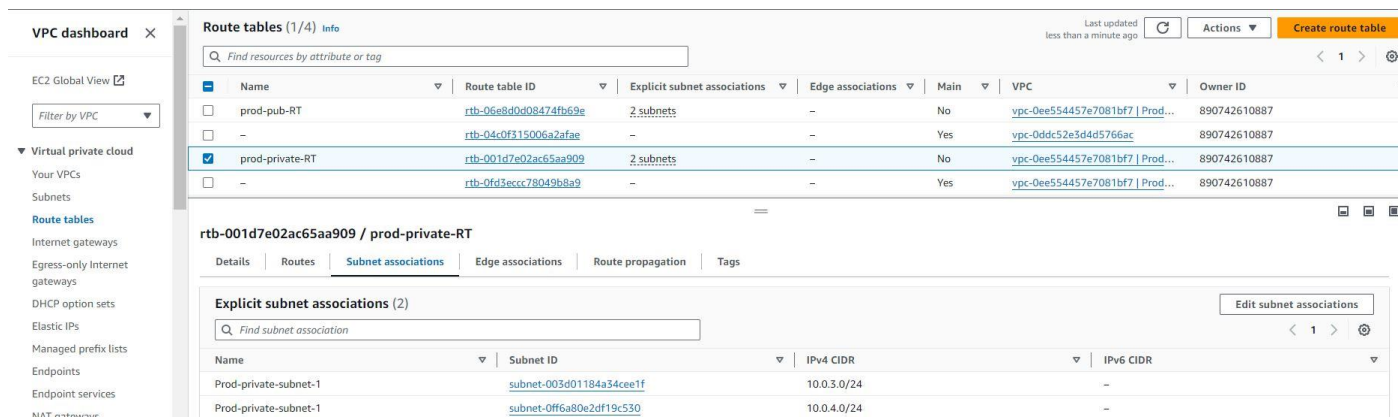
The screenshot shows the AWS VPC console 'Subnets (4/7)' page. It displays a table of subnets with columns for Name, Subnet ID, State, VPC, IPv4 CIDR, IPv6 CIDR, and IPv6 CIDR association ID. The subnets listed are:

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	IPv6 CIDR association ID
-	subnet-03e285a5b0b84c49d	Available	vpc-0ddc52e3d4d5766ac	172.31.16.0/20	-	-
-	subnet-09ee842e4dd1bd6a2	Available	vpc-0ddc52e3d4d5766ac	172.31.32.0/20	-	-
Prod-pub-subnet-2	subnet-0cd0618439a907f8b	Available	vpc-0ee554457e7081bf7   Prod...	10.0.2.0/24	-	-
Prod-private-subnet-1	subnet-003d01184a34cee1f	Available	vpc-0ee554457e7081bf7   Prod...	10.0.3.0/24	-	-
Prod-pub-subnet-1	subnet-009ba965e9a9dc9	Available	vpc-0ee554457e7081bf7   Prod...	10.0.1.0/24	-	-
Prod-private-subnet-1	subnet-0ff6a80e2df19c530	Available	vpc-0ee554457e7081bf7   Prod...	10.0.4.0/24	-	-
-	subnet-0944460fb79856861	Available	vpc-0ddc52e3d4d5766ac	172.31.0.0/20	-	-

## Route table

Use route tables to determine where network traffic from your subnet or gateway is directed.

### Private route table



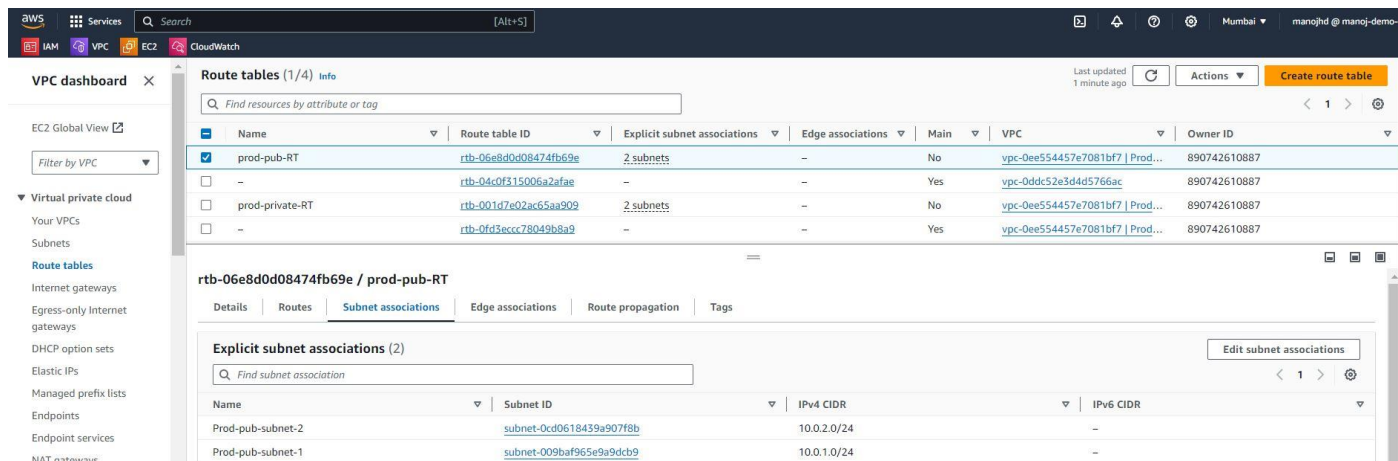
The screenshot shows the AWS VPC console 'Route tables (1/4)' page. It displays a table of route tables with columns for Name, Route table ID, Explicit subnet associations, Edge associations, Main, VPC, and Owner ID. The route tables listed are:

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC	Owner ID
prod-pub-RT	rtb-06e8d0d08474fb69e	2 subnets	-	No	vpc-0ee554457e7081bf7   Prod...	890742610887
-	rtb-04c0f315006a2afae	-	-	Yes	vpc-0ddc52e3d4d5766ac	890742610887
prod-private-RT	rtb-001d7e02ac65aa909	2 subnets	-	No	vpc-0ee554457e7081bf7   Prod...	890742610887
-	rtb-0fd3eccc78049b8a9	-	-	Yes	vpc-0ee554457e7081bf7   Prod...	890742610887

Below the table, the 'Subnet associations' tab for 'rtb-001d7e02ac65aa909 / prod-private-RT' is shown, displaying a table of explicit subnet associations:

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
Prod-private-subnet-1	subnet-003d01184a34cee1f	10.0.3.0/24	-
Prod-private-subnet-1	subnet-0ff6a80e2df19c530	10.0.4.0/24	-

### Public route table



The screenshot shows the AWS VPC console 'Route tables (1/4)' page. It displays a table of route tables with columns for Name, Route table ID, Explicit subnet associations, Edge associations, Main, VPC, and Owner ID. The route tables listed are:

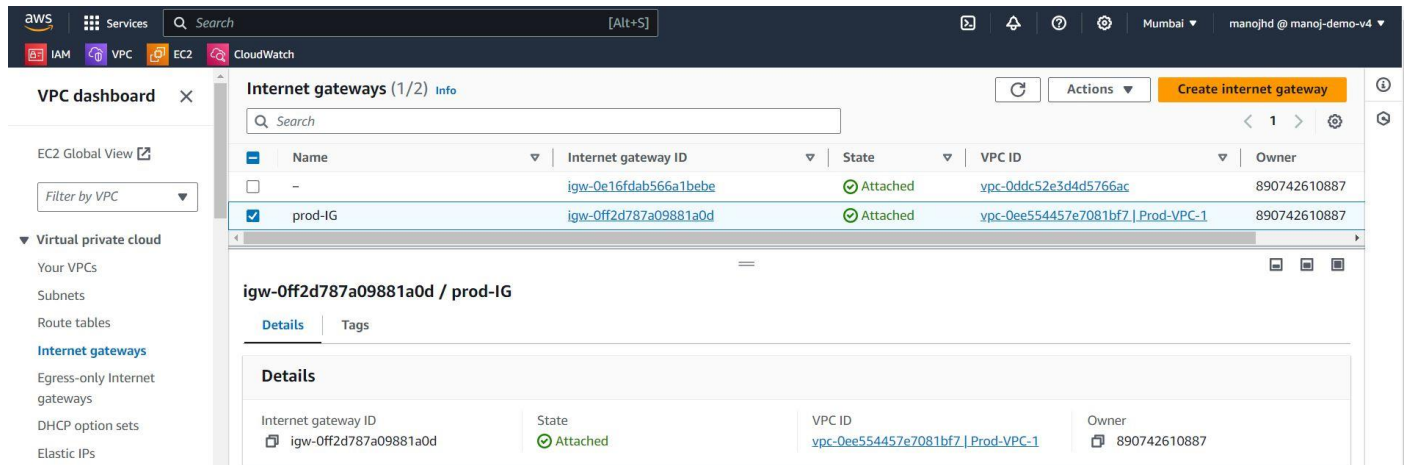
Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC	Owner ID
prod-pub-RT	rtb-06e8d0d08474fb69e	2 subnets	-	No	vpc-0ee554457e7081bf7   Prod...	890742610887
-	rtb-04c0f315006a2afae	-	-	Yes	vpc-0ddc52e3d4d5766ac	890742610887
prod-private-RT	rtb-001d7e02ac65aa909	2 subnets	-	No	vpc-0ee554457e7081bf7   Prod...	890742610887
-	rtb-0fd3eccc78049b8a9	-	-	Yes	vpc-0ee554457e7081bf7   Prod...	890742610887

Below the table, the 'Subnet associations' tab for 'rtb-06e8d0d08474fb69e / prod-pub-RT' is shown, displaying a table of explicit subnet associations:

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
Prod-pub-subnet-2	subnet-0cd0618439a907f8b	10.0.2.0/24	-
Prod-pub-subnet-1	subnet-009ba965e9a9dc9	10.0.1.0/24	-

## Internet Gateway

Internet Gateway has been created and attached to VPC and route table.



The screenshot shows the AWS Management Console interface for Internet Gateways. The left sidebar contains navigation links for VPC, EC2, and CloudWatch. The main content area displays a table of Internet Gateways. The 'prod-IG' gateway is highlighted, showing its details.

Name	Internet gateway ID	State	VPC ID	Owner
prod-IG	igw-0ff2d787a09881a0d	Attached	vpc-0ee554457e7081bf7   Prod-VPC-1	890742610887

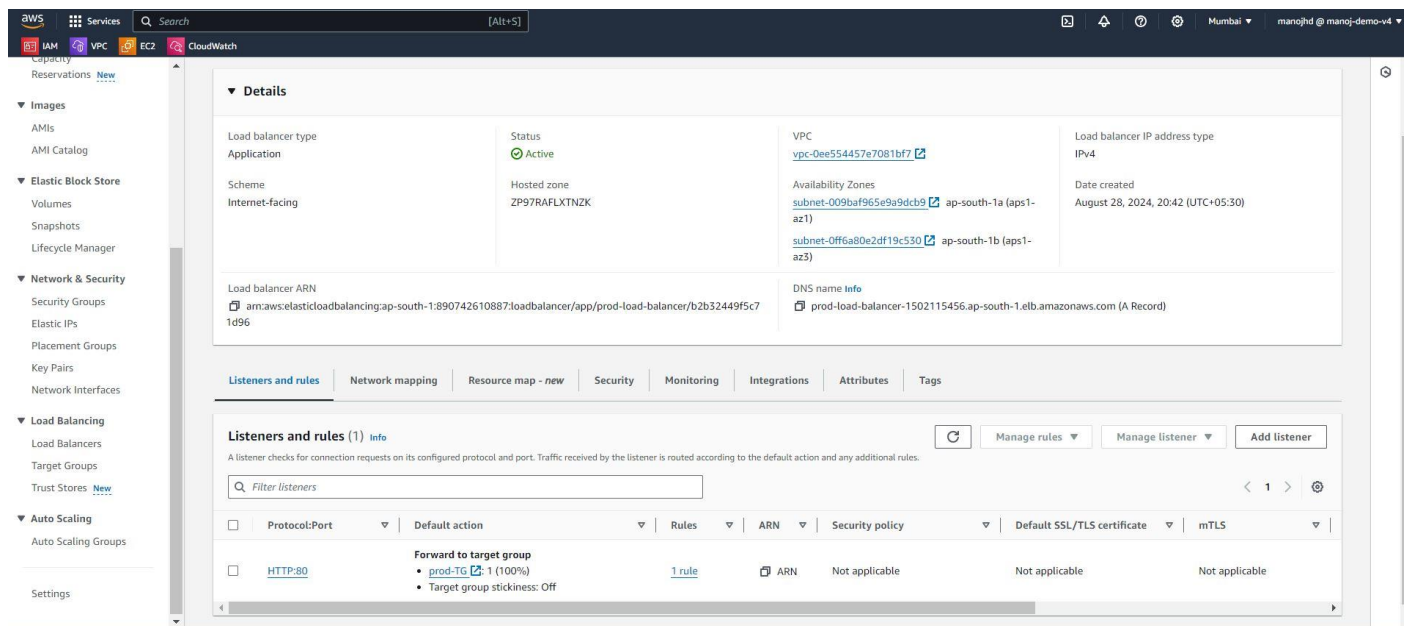
**igw-0ff2d787a09881a0d / prod-IG**

**Details**

Internet gateway ID	State	VPC ID	Owner
igw-0ff2d787a09881a0d	Attached	vpc-0ee554457e7081bf7   Prod-VPC-1	890742610887

## Load balancer

Load balancer has been created inside the VPC and attached target group where instances has attached.



The screenshot shows the AWS Management Console interface for a Load Balancer. The left sidebar contains navigation links for VPC, EC2, and CloudWatch. The main content area displays the details of the load balancer, including its status, VPC, and target group.

**Details**

Load balancer type	Status	VPC	Load balancer IP address type
Application	Active	vpc-0ee554457e7081bf7	IPv4

**Listeners and rules (1)**

Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL/TLS certificate	mTLS
HTTP:80	Forward to target group • prod-TG 1 (100%) • Target group stickiness: Off	1 rule	ARN	Not applicable	Not applicable	Not applicable

## Target group

The screenshot displays the AWS Management Console interface for Target Groups. The left sidebar shows the navigation menu with categories like Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main content area is titled 'Target groups (1/1)' and includes a search bar and a table of target groups. A modal window titled 'Target group: prod-TG' is open, showing 'Registered targets (2)' with a table of instance details.

**Target groups (1/1)**

Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
prod-TG	arn:aws:elasticloadbalancing...	8000	HTTP	Instance	prod-load-balancer	vpc-0ee554457e7081bf7

**Target group: prod-TG**

**Registered targets (2)**

Instance ID	Name	Port	Zone	Health status	Health status details	Launch...	Anomaly detection...
i-0bdf82419180d1287	Instance 2	8000	ap-south-1b	Healthy	-	August 28...	Normal
i-0f8e683548197ba88	Instance 1	8000	ap-south-1a	Healthy	-	August 28...	Normal

## Auto scaling group

Created ASG to scale up or scale down the number of instances based on the traffic.

The screenshot displays the AWS Management Console interface for Auto Scaling Groups. The left sidebar shows the navigation menu. The main content area is titled 'Auto Scaling groups (1/1)' and includes a search bar and a table of auto scaling groups. A modal window titled 'Auto Scaling group: prod-ASG' is open, showing 'Instances (2/2)' with a table of instance details.

**Auto Scaling groups (1/1)**

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	A...
prod-ASG	prod-template   Version Default	2	-	2	1	4	ap...

**Auto Scaling group: prod-ASG**

**Instances (2/2)**

Instance ID	Lifecycle	Instanc...	Weight...	Launch ...	Availab...	Health ...	Protected from
i-0bdf82419180d1287	InService	t2.micro	-	prod-template	ap-south-1b	Healthy	
i-0f8e683548197ba88	InService	t2.micro	-	prod-template	ap-south-1a	Healthy	

## NAT gateway

NAT gateway which is created inside the public subnet and assigned static IP address and connect to private subnets through private route table, which hides the private IP address of private instance and send to traffic to collect necessary information to hide and secure the IP address of private instance.

The screenshot displays the AWS Management Console interface for a NAT gateway. The left sidebar shows the navigation menu with categories like 'Your VPCs', 'Security', and 'DNS firewall'. The main content area is titled 'NAT gateways (1/1) Info'. It features a table with columns: Name, NAT gateway ID, Connectivity..., State, State message, Primary public IP..., and Primary private IP... The table lists one entry: 'prod-NAT-1' with ID 'nat-061fda8aa3f09361e', Public connectivity, Available state, and IP addresses 43.205.87.240 and 10.0.1.239. Below the table, the 'Details' tab is selected, showing a grid of key-value pairs for the NAT gateway's configuration, including its ID, ARN, connectivity type, public and private IP addresses, VPC, and creation/deletion timestamps.

Name	NAT gateway ID	Connectivity...	State	State message	Primary public IP...	Primary private IP...
prod-NAT-1	nat-061fda8aa3f09361e	Public	Available	-	43.205.87.240	10.0.1.239

**Details**

NAT gateway ID nat-061fda8aa3f09361e	Connectivity type Public	State Available	State message -
NAT gateway ARN arn:aws:ec2:ap-south-1:890742610887:natgateway/nat-061fda8aa3f09361e	Primary public IPv4 address 43.205.87.240	Primary private IPv4 address 10.0.1.239	Primary network interface ID eni-0ba57970494f3cb5c
VPC vpc-0ee554457e7081bf7 / Prod-VPC-1	Subnet subnet-009baf965e9a9dcb9 / Prod-pub-subnet-1	Created Wednesday 28 August 2024 at 20:05:25 GMT+5:30	Deleted -

## Network ACL

NACL work on subnet level, which allow or deny the traffic to instance on subnet level.

The screenshot shows the AWS Management Console for Network ACLs. The left sidebar is the same as the previous screenshot. The main content area is titled 'Network ACLs (1/2) Info'. It contains a table with columns: Name, Network ACL ID, Associated with, Default, and VPC ID. Two entries are listed: 'prod-NACL-1' (ID: acl-0e1c80501c27edd5c) associated with 4 subnets, and another ACL (ID: acl-0015cc785967ca7c5) associated with 3 subnets. Below the table, the 'Inbound rules' tab is selected for 'acl-0e1c80501c27edd5c / prod-NACL-1'. It shows a list of inbound rules with columns: Rule number, Type, Protocol, Port range, Source, and Allow/Deny. Two rules are shown: Rule 100 (All traffic, All, All, 0.0.0.0/0, Allow) and Rule \* (All traffic, All, All, 0.0.0.0/0, Deny).

Name	Network ACL ID	Associated with	Default	VPC ID
prod-NACL-1	acl-0e1c80501c27edd5c	4 Subnets	Yes	vpc-0ee554457e7081bf7 / Prod-VPC-1
-	acl-0015cc785967ca7c5	3 Subnets	Yes	vpc-0ddc52e3d4d5766ac

**acl-0e1c80501c27edd5c / prod-NACL-1**

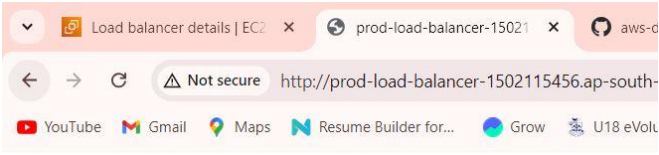
**Inbound rules (2)**

Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	All traffic	All	All	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

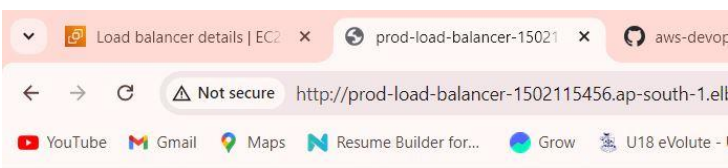


Result

Traffic flowing on both instance 1 and Instance 2 based on number of requests received.



Traffic flowing through Instance 2



Traffic flowing through Instance 1