

NACL_and_SecurityGroups

Create a VPC → demovpc

Create VPC [Info](#)

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

VPC settings

Resources to create [Info](#)
Create only the VPC resource or the VPC and other networking resources.

☒ VPC only ☐ VPC and more

Name tag - optional [Info](#)
Creates a tag with a key of 'Name' and a value that you specify.

demovpc

IPv4 CIDR block [Info](#)
☒ 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 [Info](#)
☒ No IPv6 CIDR block
☐ IPAM-allocated IPv6 CIDR block
☐ Amazon-provided IPv6 CIDR block
☐ IPv6 CIDR owned by me

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

You successfully created vpc-04f7283b6a27a737f / demovpc

Last updated less than a minute ago [Actions](#) [Create VPC](#)

<input type="checkbox"/>	Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR	DHCP option se
<input type="checkbox"/>	-	vpc-074321e9b52b24a31	Available	Off	172.31.0.0/16	-	dopt-0509027c
<input type="checkbox"/>	demovpc	vpc-04f7283b6a27a737f	Available	Off	10.0.0.0/16	-	dopt-0509027c

Select a VPC above

Create two subnet one private subnet and one public subnet in demovpc

aws [Search] [Alt+S] Mumbai Rohan Borate

VPC > Subnets > Create subnet

Create subnet [Info](#)

VPC
Create subnets in this VPC:
vpc-04f7283b6a27a737f (demovpc)

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 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
subnetA
The name can be up to 256 characters long.

Availability Zone [Info](#)

aws [Search] [Alt+S] Mumbai Rohan Borate

VPC > Subnets > Create subnet

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
subnetA
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.
Asia Pacific (Mumbai) / ap-south-1a

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
10.0.0.0/17 32,768 IPs
< > ^ v

Tags - optional

Key	Value - optional	
Name	subnetA	Remove

[Add new tag](#)
You can add 49 more tags.

[Remove](#)

aws [Search] [Alt+S] Mumbai Rohan Borate

VPC > Subnets > Create subnet

[Remove](#)

Subnet 2 of 2

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
subnetB
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.
Asia Pacific (Mumbai) / ap-south-1b

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
10.0.128.0/17 32,768 IPs
< > ^ v

Tags - optional

[Remove](#)

[Add new subnet](#)

AWS IAM EC2

Search [Alt+S]

Mumbai Rohan Borate

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

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

You have successfully created 2 subnets: subnet-01e2bc0b32dab8c22, subnet-078c1012d25df8841

Last updated less than a minute ago

Actions Create subnet

Find resources by attribute or tag

Subnet ID: subnet-01e2bc0b32dab8c22 Subnet ID: subnet-078c1012d25df8841 Clear filters

<input type="checkbox"/>	Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR
<input type="checkbox"/>	subnetB	subnet-078c1012d25df8841	Available	vpc-04f7283b6a27a737f dem...	Off	10.0.128.0/17	-
<input type="checkbox"/>	subnetA	subnet-01e2bc0b32dab8c22	Available	vpc-04f7283b6a27a737f dem...	Off	10.0.0.0/17	-

Select a subnet

Create internet gateway

AWS IAM EC2

Search [Alt+S]

Mumbai Rohan Borate

VPC > Internet gateways > Create internet gateway

Create internet gateway

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

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

internetgateway

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

Q Name X

Value - optional

Q internetgateway X Remove

Add new tag

You can add 49 more tags.

Cancel Create internet gateway

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Attach internet gateway to vpc

AWS IAM EC2

Search [Alt+S]

Mumbai Rohan Borate

VPC > Internet gateways > Attach to VPC (igw-0d1bc687a076c70fc)

The following internet gateway was created: igw-0d1bc687a076c70fc - internetgateway. You can now attach to a VPC to enable the VPC to communicate with the internet.

Attach to a VPC

Attach to VPC (igw-0d1bc687a076c70fc)

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.

Q vpc-04f7283b6a27a737f X

AWS Command Line Interface command

Cancel Attach internet gateway

AWS IAM EC2

[Alt+S]

Mumbai Rohan Borate

VPC > Route tables > Create route table

Create route table info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.

VPC
The VPC to use for this route table.
 vpc-04f7283b6a27a7537f (demo-vpc)

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 X

Value - optional

routetable X Remove

Add new tag

You can add 49 more tags.

Cancel Create route table

aws

Search

[Alt+S]

iam

ec2

VPC

Route tables

rtb-03adf2e0d0070409

Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/2)

Q Filter subnet associations

	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input type="checkbox"/>	subnetB	subnet-078c1012d25df8841	10.0.128.0/17	–	Main (rtb-0524732735c0141e7)
<input checked="" type="checkbox"/>	subnetA	subnet-01e2bc0b32dab8c22	10.0.0.0/17	–	Main (rtb-0524732735c0141e7)

Selected subnets

subnet-01e2bc0b32dab8c22 / subnetA

Cancel Save associations

CloudShellFeedback

© 2024, Amazon Web Services, Inc. or its affiliates. PrivacyTermsCookie preferences

aws

Search [Alt+S]

Mumbai

Rohan Borate

IAM

EC2

VPC

>

Route tables

>

rtb-034adf2e0d0070409

>

Edit routes

Destination

10.0.0.0/16

0.0.0.0/0

Add route

Target

local

local

Internet Gateway

igw-

Status

Active

-

Propagated

No

No

Remove

Cancel

Preview

Save changes

Create an instance in that VPC in subnetA which is public subnet .because we have associate it routetable where internet gateway is also added

The screenshot shows the 'Launch an instance' wizard in the AWS Management Console, specifically the 'Network settings' step. The 'VPC' is set to 'vpc-04f7283b6a27a737f (demovpc)' and the 'Subnet' is 'subnet-01e2bc0b32dab8c22 (subnetA)'. The 'Auto-assign public IP' is set to 'Disable'. The 'Firewall (security groups)' section shows 'Create security group' selected. The 'Summary' panel on the right shows 'Number of instances' as 1, 'Software Image (AMI)' as 'Amazon Linux 2023 AMI 2023.6.2', 'Virtual server type (instance type)' as 't2.micro', and 'Storage (volumes)' as '1 volume(s) - 8 GiB'. A 'Free tier' badge indicates 750 hours of t2.micro usage in the first year.

Network settings Info

VPC - required Info
vpc-04f7283b6a27a737f (demovpc)
10.0.0.0/16

Subnet Info
subnet-01e2bc0b32dab8c22 subnetA
VPC: vpc-04f7283b6a27a737f Owner: 555786028785
Availability Zone: ap-south-1a Zone type: Availability Zone
IP addresses available: 32763 CIDR: 10.0.0.0/17

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-7

Description - required Info
launch-wizard-7 created 2024-12-28T11:48:26.490Z

Summary Info

Number of instances Info
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.6.2...read more
ami-0f405997b4d0ff7aac

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 includes 750 hours of t2.micro (or t3.micro in

Cancel Launch Instance Preview code

The screenshot shows the 'Instances' page in the AWS Management Console. It displays a table with two instances: 'demosever' (Running) and 'elasticIP' (Terminated). The table columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, and Public. The 'demosever' instance is a t2.micro type, initialized, and located in ap-south-1a. The 'elasticIP' instance is also a t2.micro type, terminated, and located in ap-south-1b.

Instances (2) Info

Find Instance by attribute or tag (case-sensitive) All states

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public
<input type="checkbox"/>	demosever	i-0d2a38d636edc3cce	Running	t2.micro	Initializing	View alarms +	ap-south-1a	-	3.108.
<input type="checkbox"/>	elasticIP	i-0e00ec39d0a70a3cd	Terminated	t2.micro	-	View alarms +	ap-south-1b	-	-

Select an instance

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

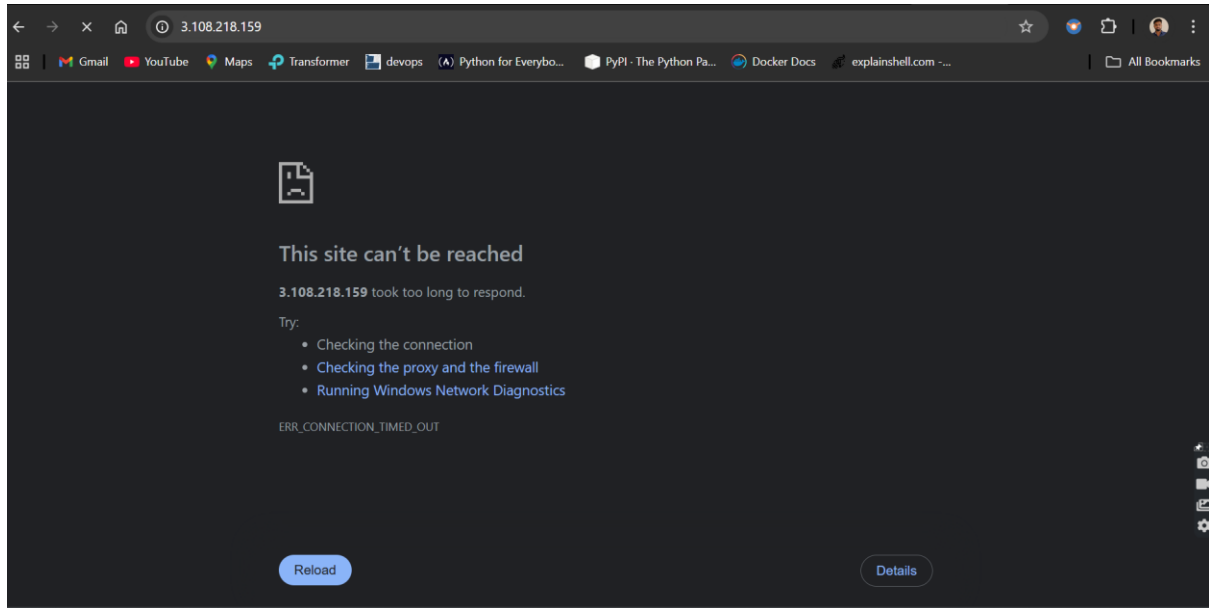
Connect to that create EC2 instance (demoserver) and install nginx on that instance

```
[ec2-user@ip-10-0-8-181 ~]$ yum search nginx
Amazon Linux 2023 repository                    52 MB/s | 30 MB   00:00
Amazon Linux 2023 Kernel Livepatch repository  58 kB/s | 11 kB   00:00
===== Name Exactly Matched: nginx =====
nginx.x86_64 : A high performance web server and reverse proxy server
===== Name & Summary Matched: nginx =====
collectd-nginx.x86_64 : Nginx plugin for collectd
nginx-all-modules.noarch : A meta package that installs all available Nginx modules
nginx-core.x86_64 : nginx minimal core
nginx-filesystem.noarch : The basic directory layout for the Nginx server
nginx-mimetypes.noarch : MIME type mappings for nginx
nginx-mod-devel.x86_64 : Nginx module development files
nginx-mod-http-image-filter.x86_64 : Nginx HTTP image filter module
nginx-mod-http-perl.x86_64 : Nginx HTTP perl module
nginx-mod-http-xslt-filter.x86_64 : Nginx XSLT module
nginx-mod-mail.x86_64 : Nginx mail modules
nginx-mod-stream.x86_64 : Nginx stream modules
python3-certbot-nginx.noarch : The nginx plugin for certbot
[ec2-user@ip-10-0-8-181 ~]$
```

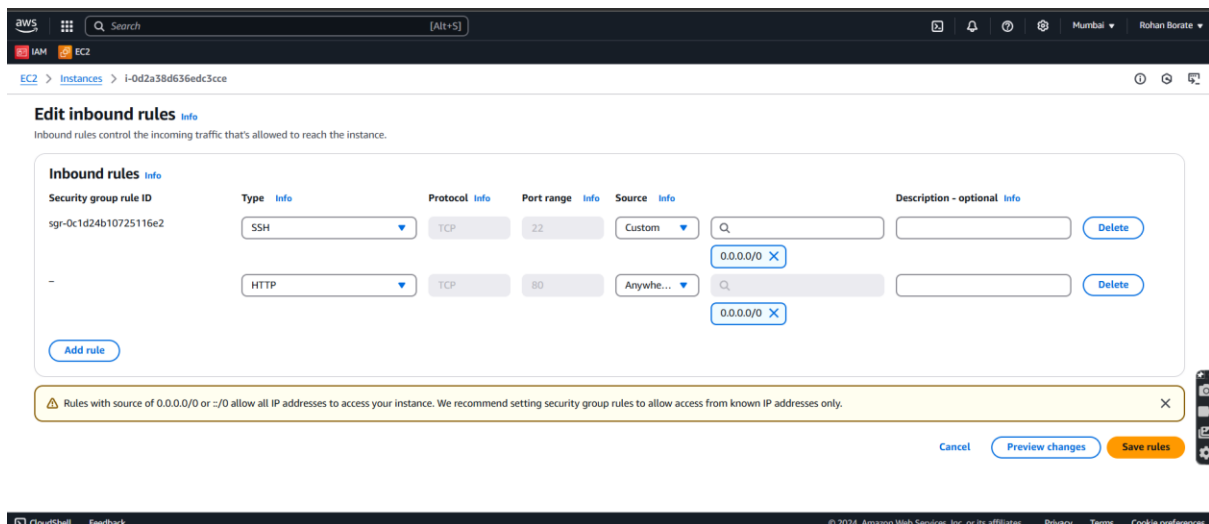
```
[ec2-user@ip-10-0-8-181 ~]$ yum install nginx
Error: This command has to be run with superuser privileges (under the root user on most systems).
[ec2-user@ip-10-0-8-181 ~]$ sudo -i
[root@ip-10-0-8-181 ~]# yum install nginx
Last metadata expiration check: 0:01:37 ago on Sat Dec 28 12:05:45 2024.
Dependencies resolved.
=====
Package                                Architecture      Version                                Repository          Size
=====
Installing:
nginx                                  x86_64            1:1.26.2-1.amzn2023.0.1              amazonlinux          33 k
Installing dependencies:
generic-logos-httpd                   noarch            18.0.0-12.amzn2023.0.3              amazonlinux          19 k
gperftools-libs                       x86_64            2.9.1-1.amzn2023.0.3                amazonlinux          308 k
libunwind                             x86_64            1.4.0-5.amzn2023.0.2                amazonlinux          66 k
nginx-core                            x86_64            1:1.26.2-1.amzn2023.0.1              amazonlinux          670 k
nginx-filesystem                      noarch            1:1.26.2-1.amzn2023.0.1              amazonlinux          9.9 k
nginx-mimetypes                       noarch            2.1.49-3.amzn2023.0.3                amazonlinux          21 k
Transaction Summary
=====
Install 7 Packages

Total download size: 1.1 M
Installed size: 3.6 M
Is this ok [y/N]: y
Downloading Packages:
(1/7): generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch.rpm              321 kB/s | 19 kB   00:00
(2/7): libunwind-1.4.0-5.amzn2023.0.2.x86_64.rpm                        1.0 MB/s | 66 kB   00:00
(3/7): gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64.rpm                  4.3 MB/s | 308 kB  00:00
(4/7): nginx-1.26.2-1.amzn2023.0.1.x86_64.rpm                           1.9 MB/s | 33 kB   00:00
(5/7): nginx-core-1.26.2-1.amzn2023.0.1.x86_64.rpm                       22 MB/s | 670 kB   00:00
(6/7): nginx-filesystem-1.26.2-1.amzn2023.0.1.noarch.rpm                 417 kB/s | 9.9 kB   00:00
```

Start the nginx and check It is running , with port 80

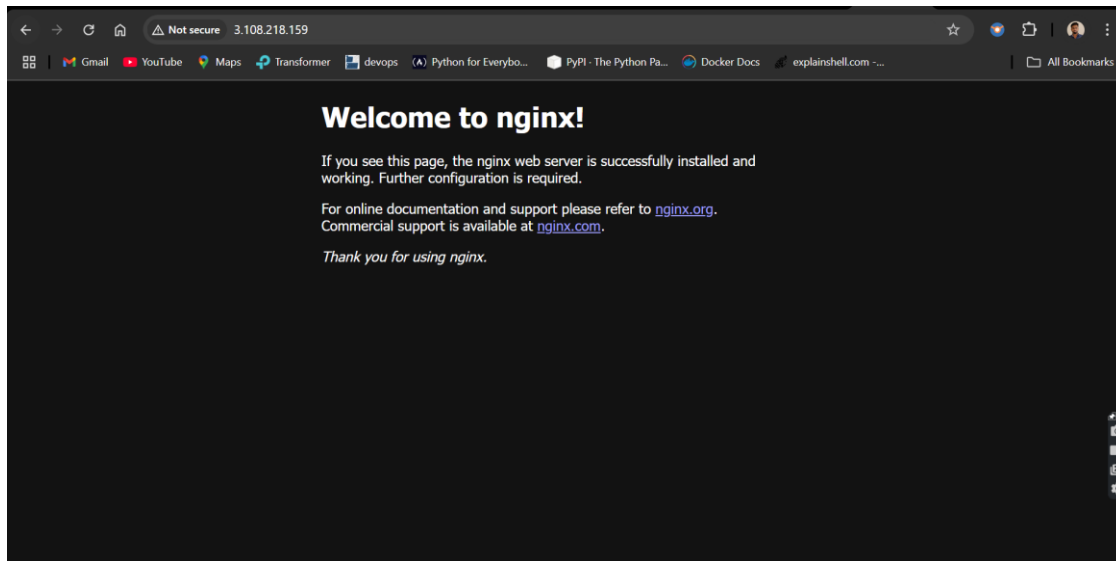


Enable port 80 in security group inbound rules



Now check nginx is accessible as we are enable port 80 , through the security group

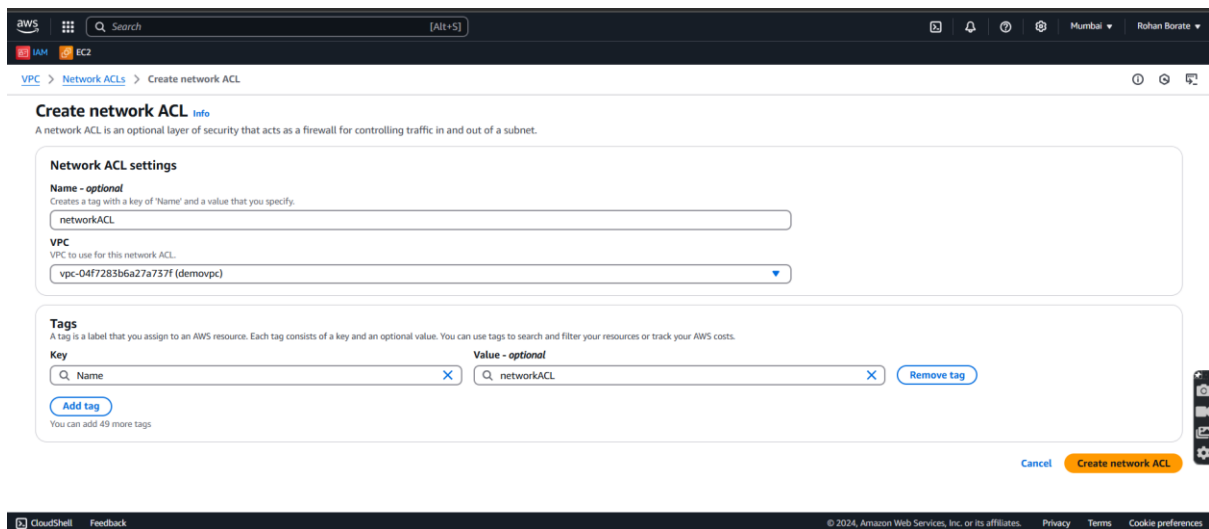
Here we are enabling the security through the security group



Now go to the NACL and denied port 80 ,

Basically NACL used to allow and denied specific ports and IP

Create NACL



Associate subnetA to NACL

The screenshot shows the AWS Management Console interface for editing subnet associations of a Network ACL. The breadcrumb trail is VPC > Network ACLs > acl-0eb0c94b86d2051d8 / networkACL > Edit subnet associations. The page title is 'Edit subnet associations' with an info icon. Below the title is a subtitle: 'Change which subnets are associated with this network ACL.' The main content area is divided into two sections: 'Available subnets (1/2)' and 'Selected subnets'. The 'Available subnets' section contains a table with columns: Name, Subnet ID, Associated with, Availability Zone, IPv4 CIDR, and IPv6 CIDR. The table lists two subnets: 'subnetB' and 'subnetA'. 'subnetA' is selected, indicated by a blue highlight and a checkmark in the selection column. The 'Selected subnets' section shows 'subnet-01e2bc0b32dab8c22 / subnetA' with a close button. At the bottom right, there are 'Cancel' and 'Save changes' buttons. A vertical toolbar with various icons is visible on the right side of the console.

Name	Subnet ID	Associated with	Availability Zone	IPv4 CIDR	IPv6 CIDR
subnetB	subnet-078c1012d25df8841	acl-0b4909fe4a2192583	ap-south-1b	10.0.128.0/17	--
<input checked="" type="checkbox"/> subnetA	subnet-01e2bc0b32dab8c22	acl-0b4909fe4a2192583	ap-south-1a	10.0.0.0/17	--

Denied port 80 in NACL inbound rule

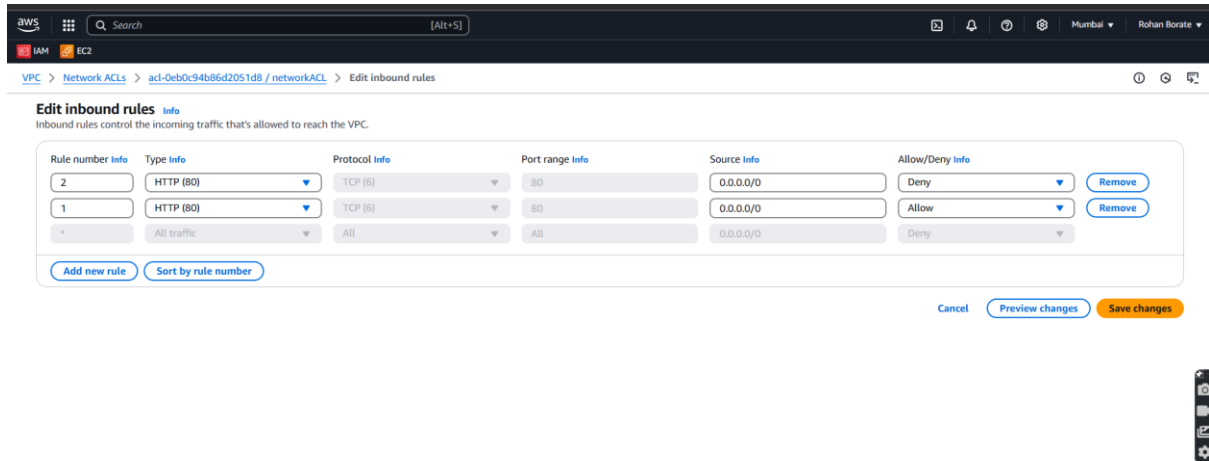
The screenshot shows the AWS Management Console interface for editing inbound rules of a Network ACL. The breadcrumb trail is VPC > Network ACLs > acl-0eb0c94b86d2051d8 / networkACL > Edit inbound rules. The page title is 'Edit inbound rules' with an info icon. Below the title is a subtitle: 'Inbound rules control the incoming traffic that's allowed to reach the VPC.' The main content area is a form for creating or editing inbound rules. It includes fields for Rule number (1), Type (HTTP (80)), Protocol (TCP (6)), Port range (80), Source (0.0.0.0/0), and Allow/Deny (Deny). There are also buttons for 'Add new rule', 'Sort by rule number', and 'Remove'. At the bottom right, there are 'Cancel', 'Preview changes', and 'Save changes' buttons. A vertical toolbar with various icons is visible on the right side of the console.

Now check nginx is accessible through port 80 , in security groups port 80 is enabled , but we have denied port 80 in NACL , NACL works at subnet level , so nginx should not be accessible

The screenshot shows a web browser window with the address bar displaying '3.108.218.159'. The browser's address bar shows several tabs: Gmail, YouTube, Maps, Transformer, devops, Python for Everybo..., PyPI - The Python Pa..., Docker Docs, and explainshell.com. The main content area of the browser displays a 'This site can't be reached' error message. The message states: '3.108.218.159 took too long to respond.' Below this, it says 'Try:' followed by a list of suggestions: 'Checking the connection', 'Checking the proxy and the firewall', and 'Running Windows Network Diagnostics'. At the bottom of the error message, it says 'ERR_CONNECTION_TIMED_OUT'. There are 'Reload' and 'Details' buttons at the bottom of the error message. A vertical toolbar with various icons is visible on the right side of the browser window.

NACL works on the rule , means priority from low to high , 1 is highest priority

We can check now this by adding new rule which is allowing port 80 with giving high priority than other rule which is denying this port 80



Check nginx is available , it should be accessible

