

3 tier web application task

Region =ohio

Vpc = baby

Igw =1

Ec2 servers 4

Subnets=6 2=publi for web and 4=private for 2=app and 2=rds

Route tables= 2 for public web and 1 for app and 1 for rds total=4

web public 2 (rao and venkat)

app private 2 (potti and nirasam)

Rds2 = devil and rakashi

Subnet group in rds=1

ALb =2

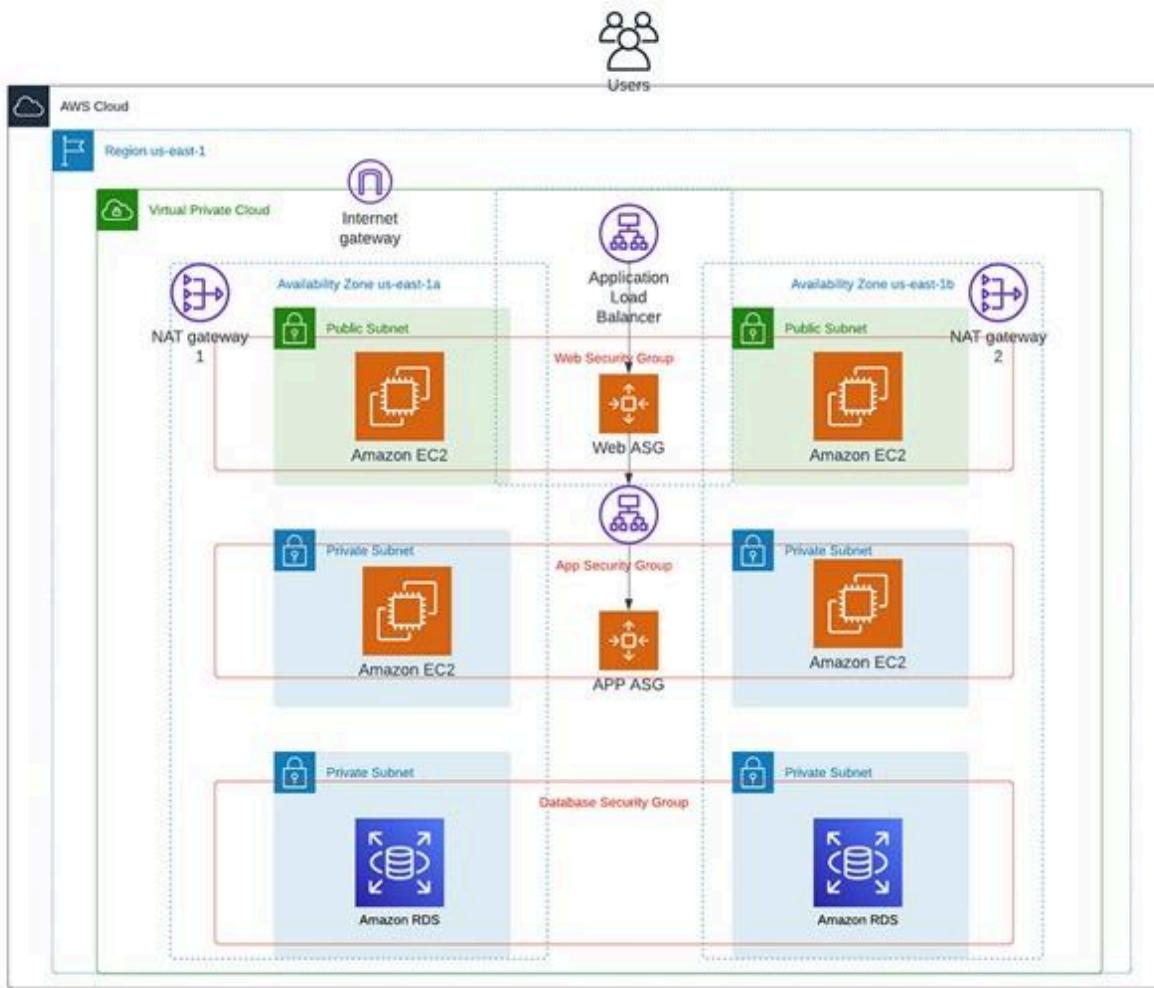
tg=2

sg = 5 (http ssh mysql)

web auto scaling = rao autoscaling

App auto scaling = potti autoscaling

Nat gateway 2 = potti-nat and rao-nat



1) Create vpc click create vpc name = baby

Your VPCs (1) Info

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
-	vpc-0a765047dc8eb3ddd	Available	172.31.0.0/16	-

Select a VPC above

2) Vpc name = baby-vpc

3) IPv4 CIDR= 10.0.0.0/16

4) Click create vpc

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

baby-vpc

IPv4 CIDR block Info
 IPv4 manual input IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.0.0.0/16

IPv6 CIDR block Info

Your VPCs (2) Info

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
-	vpc-0a765047dc8eb3ddd	Available	172.31.0.0/16	-
baby-vpc	vpc-0084435fb463bbf77	Available	10.0.0.0/16	-

Select a VPC above

5) Now create internet gateway go igw

6) Click create igw

Internet gateways (1) Info

Name	Internet gateway ID	State	VPC ID
-	igw-0dc374fd4f6e46945	Attached	vpc-0a765047dc8eb3ddd

Select an internet gateway above

Activities Google Chrome Jul 16 15:36

igws | VPC Console us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#igws:

aws Services Search [Alt+S]

VPC dashboard

EC2 Global View Filter by VPC

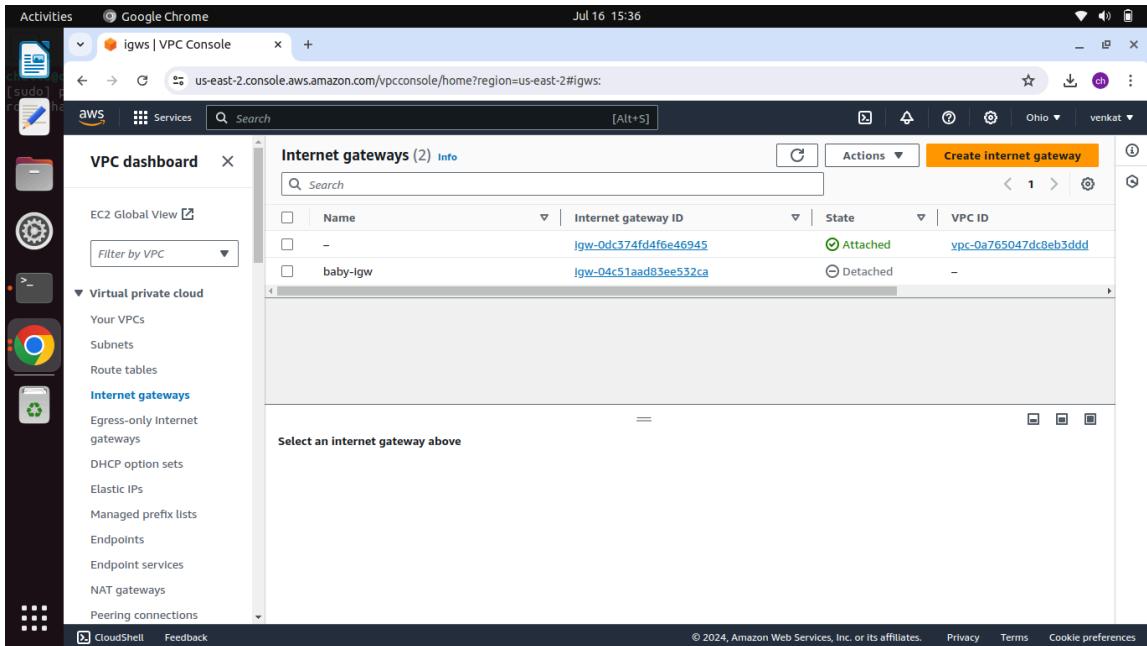
Virtual private cloud Your VPCs Subnets Route tables Internet gateways Egress-only Internet gateways DHCP option sets Elastic IPs Managed prefix lists Endpoints Endpoint services NAT gateways Peering connections

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

Internet gateways (2) Info

Name	Internet gateway ID	State	VPC ID
-	igw-0dc374fd4f6e46945	Attached	vpc-0a765047dc8eb5ddd
baby-igw	igw-04c51aad83ee532ca	Detached	-

Select an internet gateway above



Activities Google Chrome Jul 16 15:36

Create internet gateway us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#CreateInternetGateway:

VPC > Internet gateways > Create internet gateway

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

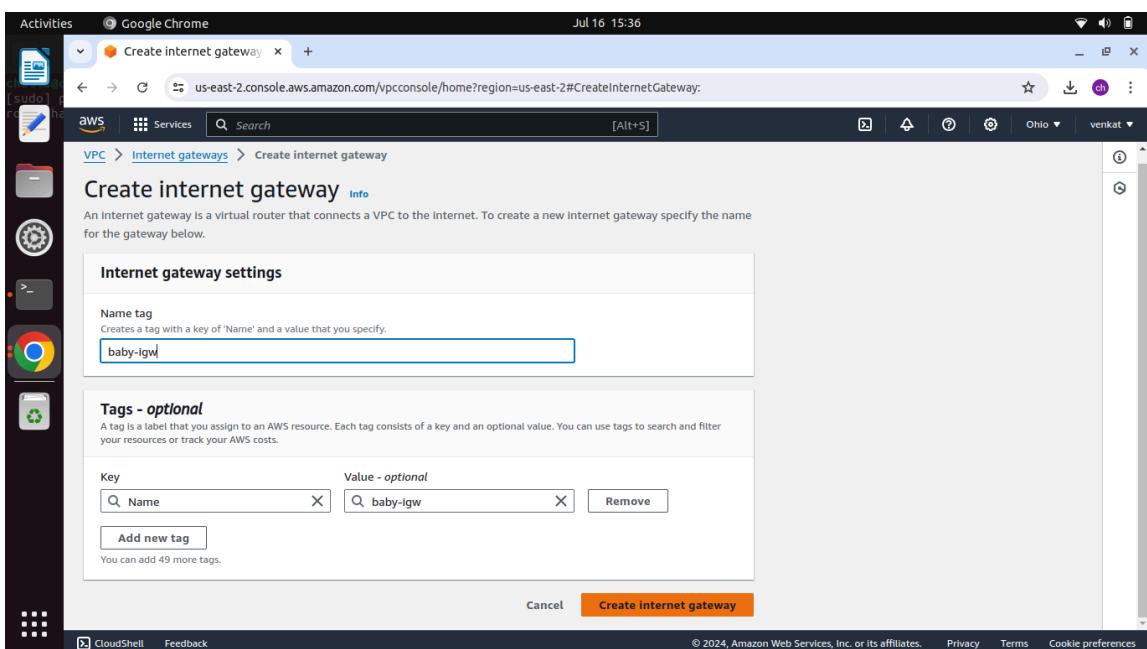
Name tag
Creates a tag with a key of 'Name' and a value that you specify.

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.

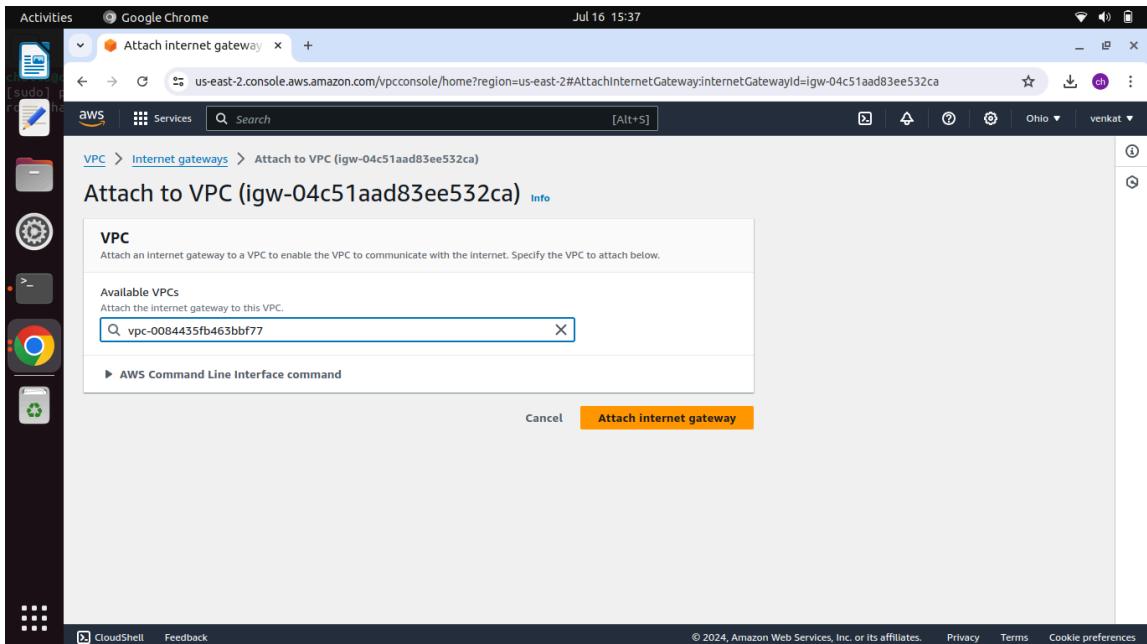
Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="baby-igw"/>

Add new tag
You can add 49 more tags.

Cancel **Create internet gateway** © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

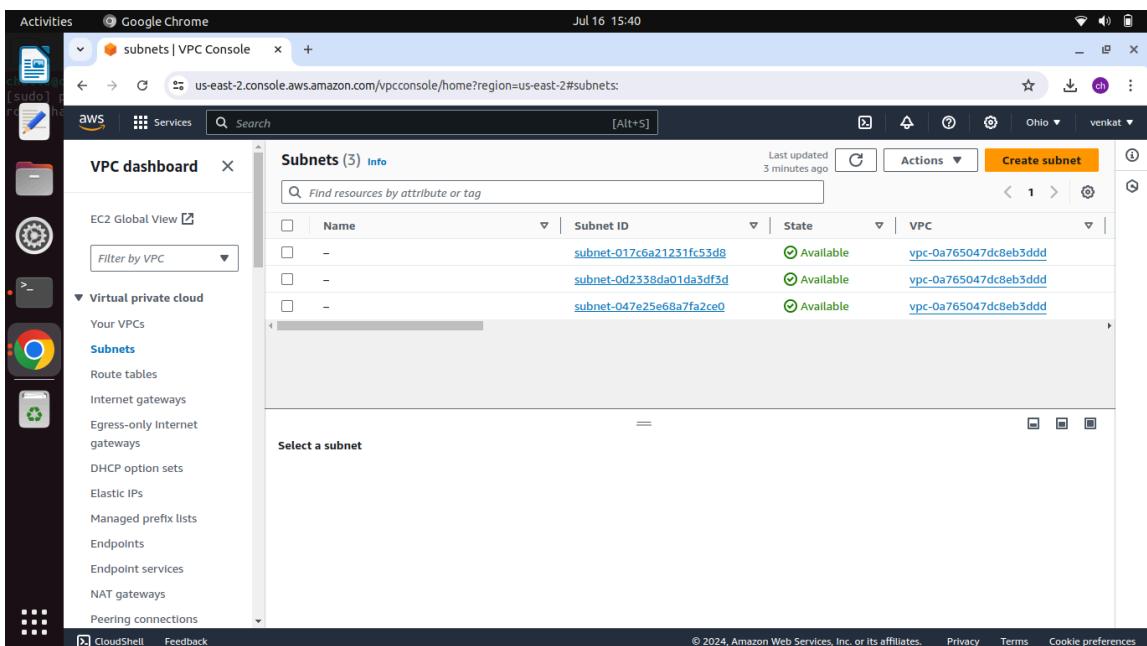


- 7) Select baby-igw click actions click attach to vpc and select baby-vpc click attach internet gate way



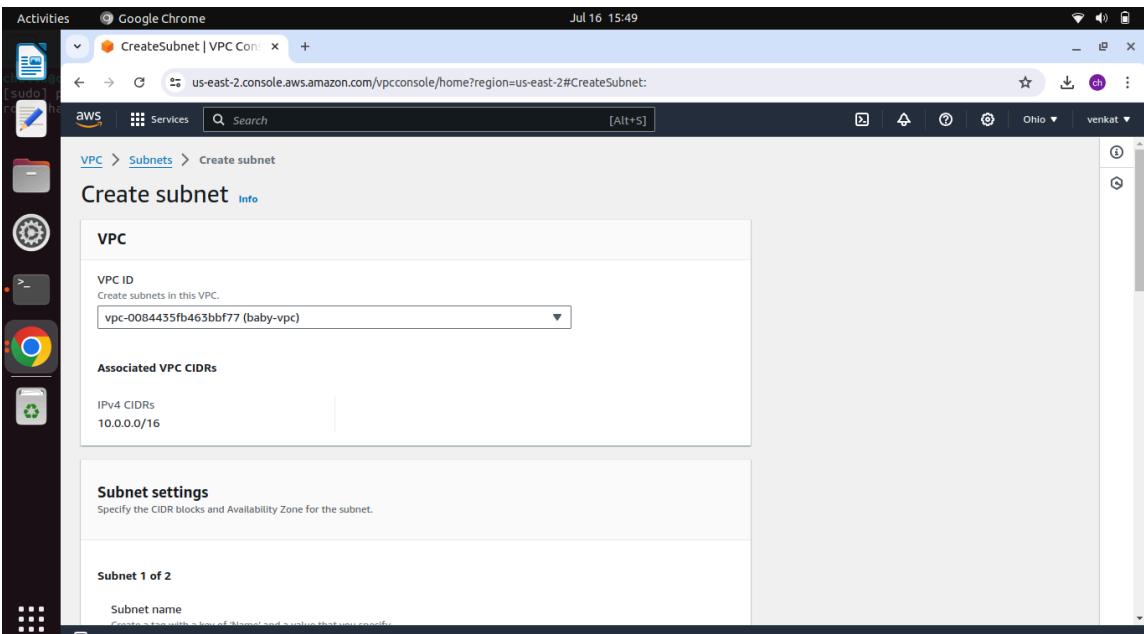
- 8) Create 2 public subnets for web application (rao and venkat)

- 9) Subnets click create subnet

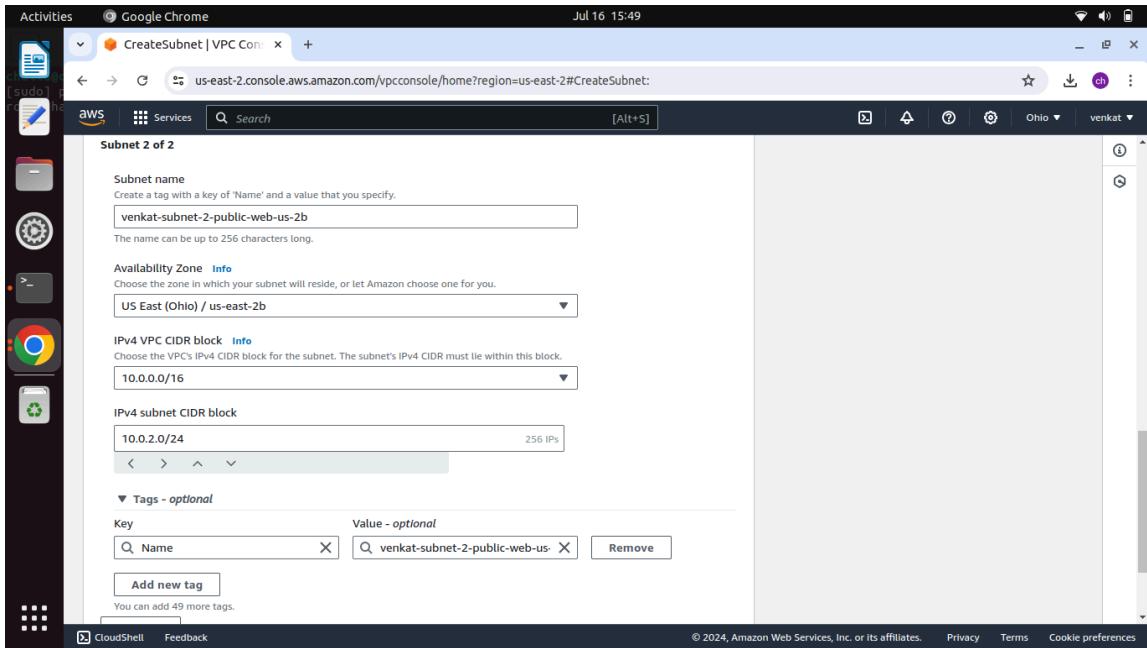


- 10) vpc= baby-vpc
- 11) Subnet name=rao-subnet-1-public-web-us-2a
- 12) Availability zone=us-east-2a

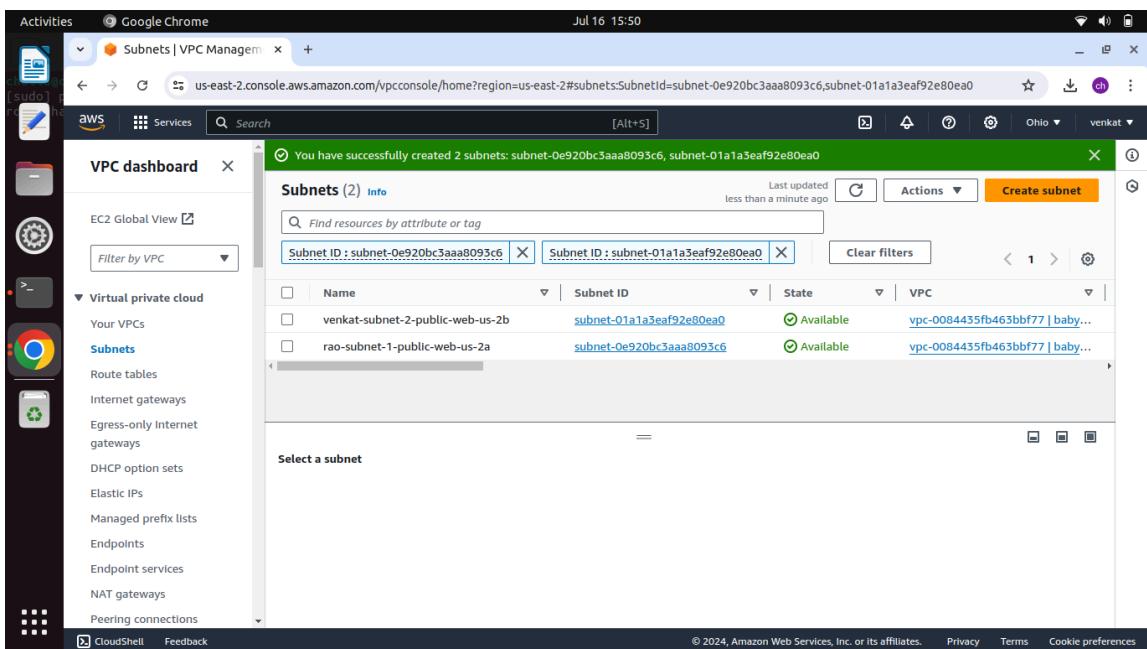
- 13) IPv4 subnet CIDR block=10.0.0.0/24
- 14) Click Add subnet
- 15) Subnet name=venkat-subnet-2-public-web-us-2b
- 16) Availability zone=us-east-2b
- 17) IPv4 subnet CIDR block=10.0.2.0/24
- 18) Click Add subnet



The screenshot shows the 'Create subnet' wizard in the AWS VPC console. In the 'VPC' section, the 'VPC ID' dropdown is set to 'vpc-0084435fb463bbf77 (baby-vpc)'. Under 'Associated VPC CIDRs', the 'IPv4 CIDRs' field contains '10.0.0.0/16'. The 'Subnet settings' section is titled 'Subnet 1 of 2'. It includes fields for 'Subnet name' (set to 'rao-subnet-1-public-web-us-2a'), 'Availability Zone' (set to 'US East (Ohio) / us-east-2a'), and 'IPv4 VPC CIDR block' (set to '10.0.0.0/16'). The 'IPv4 subnet CIDR block' field shows '10.0.0.0/24' with a note indicating it provides '256 IPs'. A 'Tags - optional' section at the bottom allows adding a tag with key 'Name' and value 'rao-subnet-1-public-web-us-2a'.



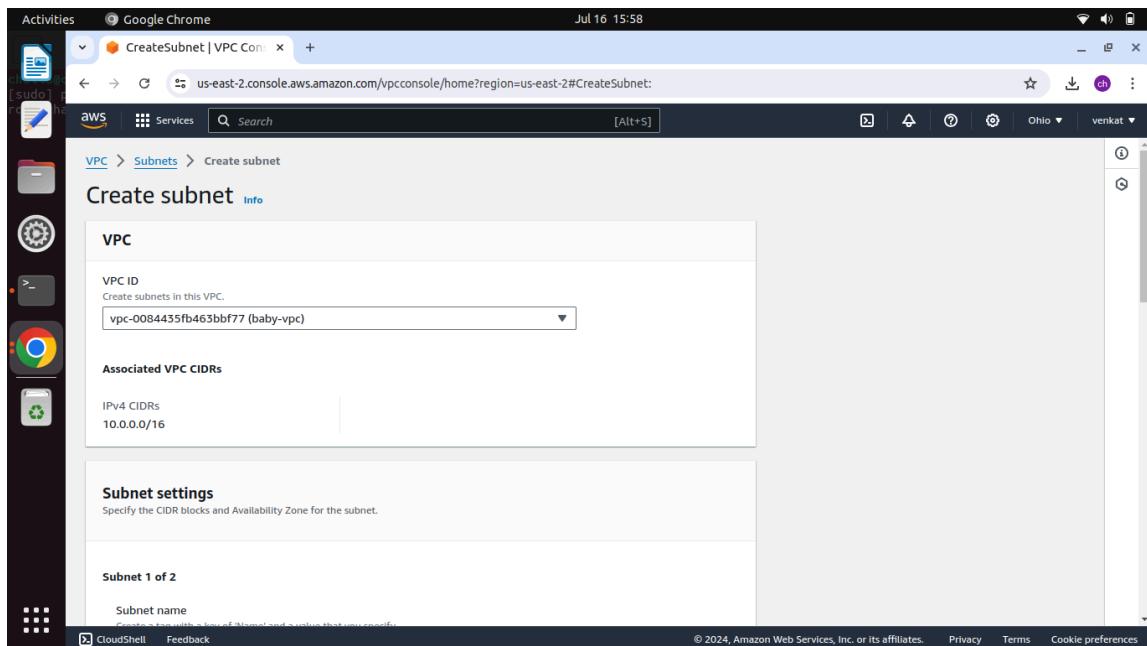
19) Click create subnet



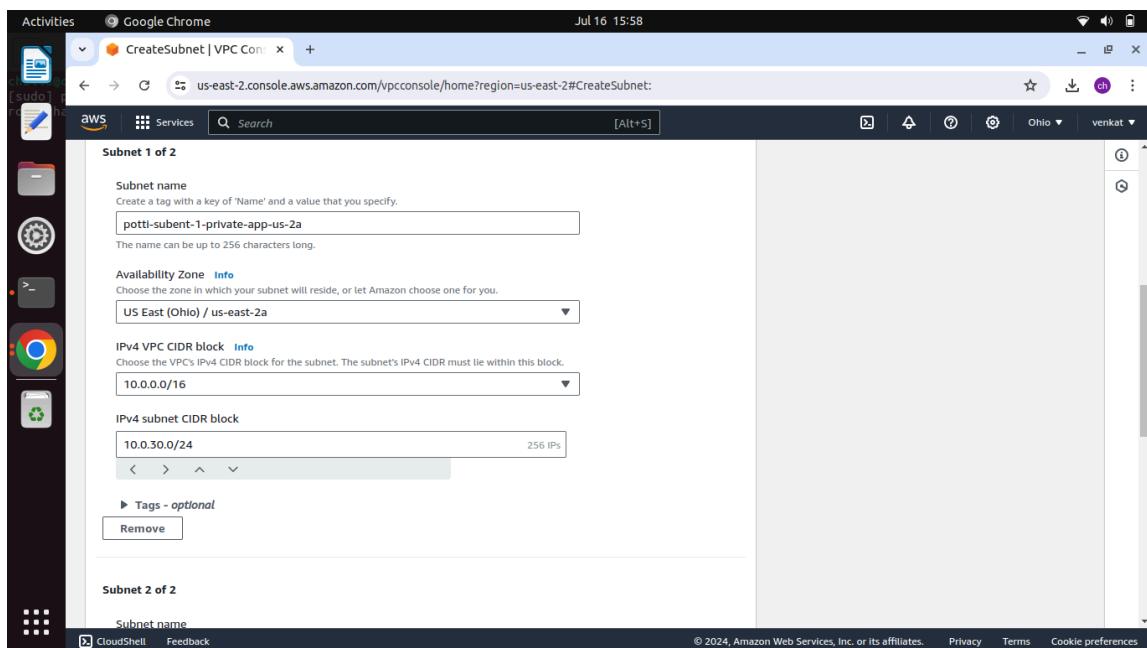
20) Now create 2 subnets in private subnets for app (potti and nirasam)

- 21) Click create subnet select vpc= baby-vpc
- 22) Subnet name= potti-subnet-1-private-app-us-2a
- 23) Availability zone=us-east-2a
- 24) IPv4 subnet CIDR block= 10.0.30.0/24

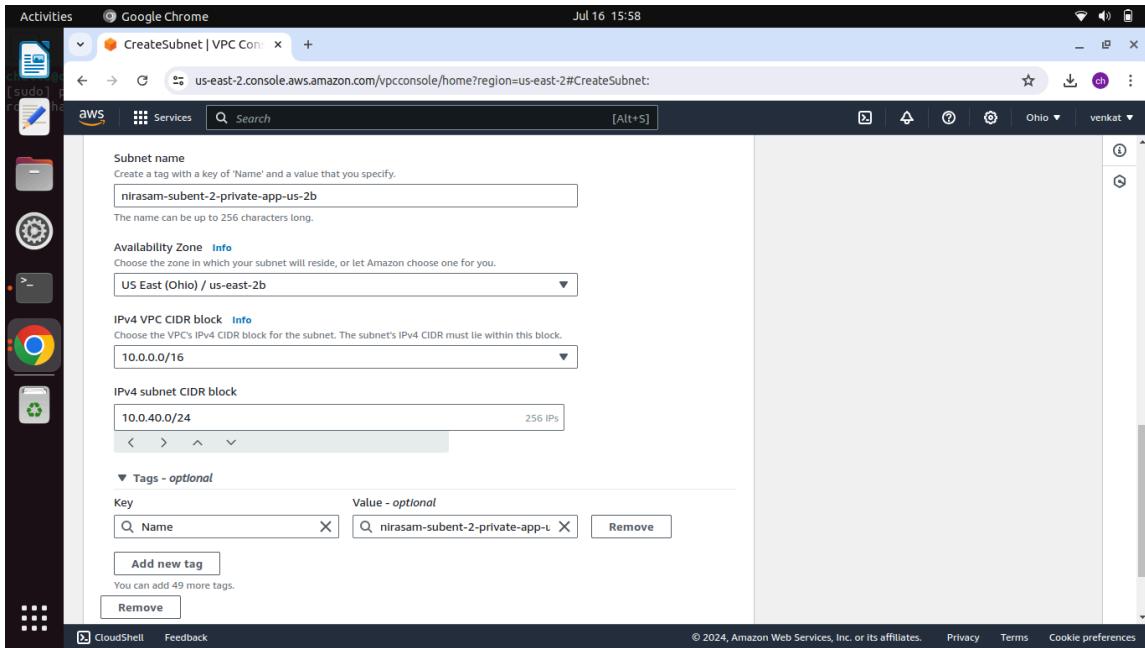
- 25) Click add subnet
- 26) Subnet name= nirasam-subnet-2-private-app-us-2b
- 27) Availability zone=us-east-2b
- 28) IPv4 subnet CIDR block= 10.0.40.0/24



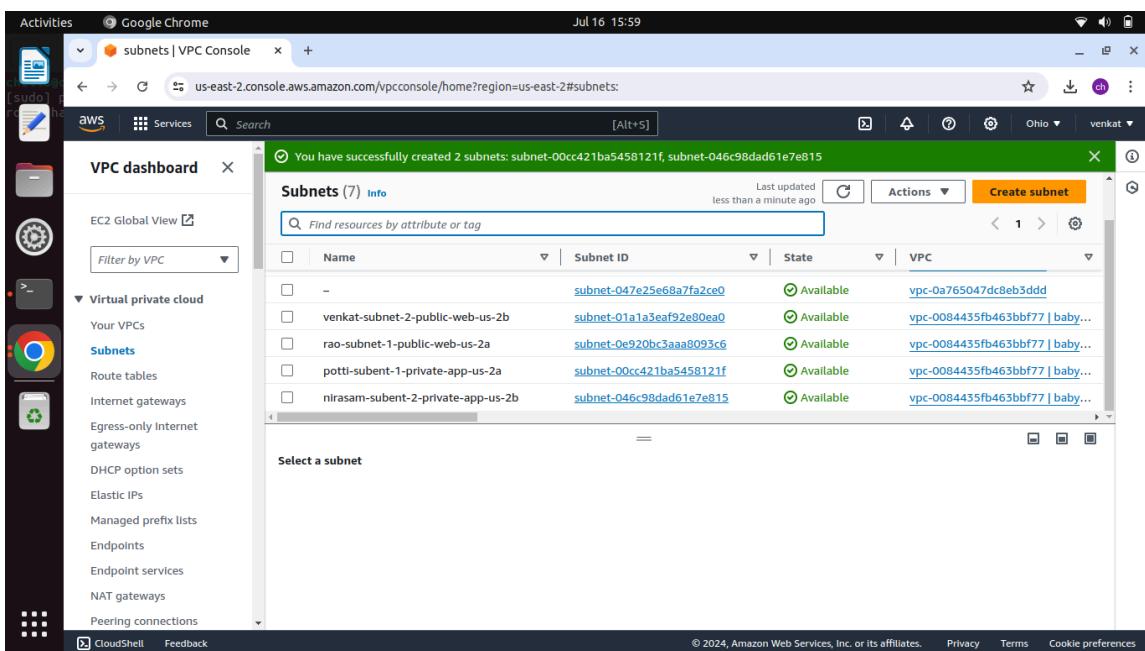
The screenshot shows the 'Create subnet' wizard in the AWS VPC console. Step 1: Set VPC ID. The VPC ID 'vpc-0084435fb463bbf77 (baby-vpc)' is selected. Step 2: Set Associated VPC CIDRs. The IPv4 CIDR '10.0.0.0/16' is listed. Step 3: Set Subnet settings. Subnet 1 of 2 is being configured. The Subnet name is 'potti-subnet-1-private-app-us-2a'. The Availability Zone is 'US East (Ohio) / us-east-2a'. The IPv4 VPC CIDR block is '10.0.0.0/16'. The IPv4 subnet CIDR block is '10.0.30.0/24'. A note indicates that the subnet's IP range must be within the VPC's CIDR.



The screenshot shows the 'Create subnet' wizard in the AWS VPC console. Step 4: Set Subnet 2 of 2. The Subnet name is 'Subnet 2 of 2'. A note indicates that the subnet's IP range must be within the VPC's CIDR.



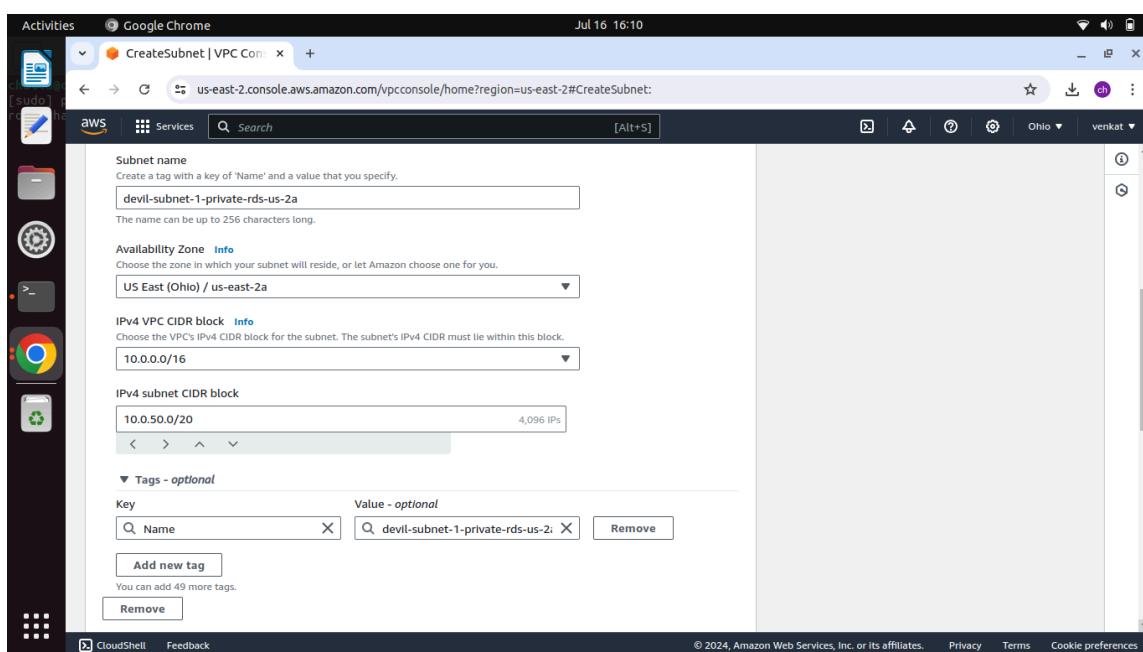
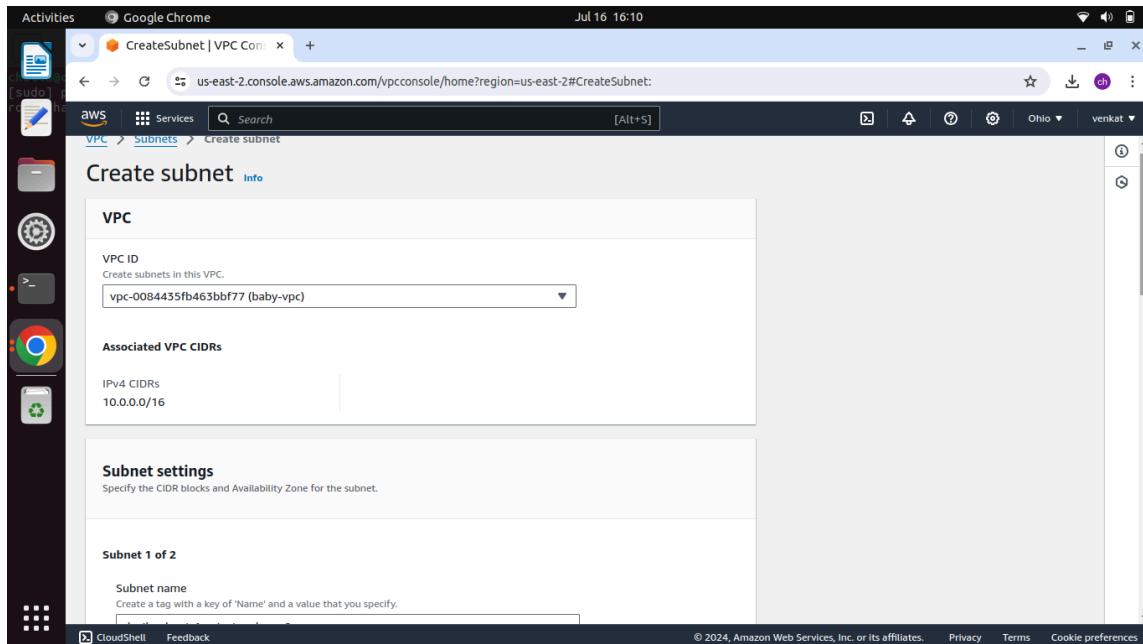
29) Click create subnet

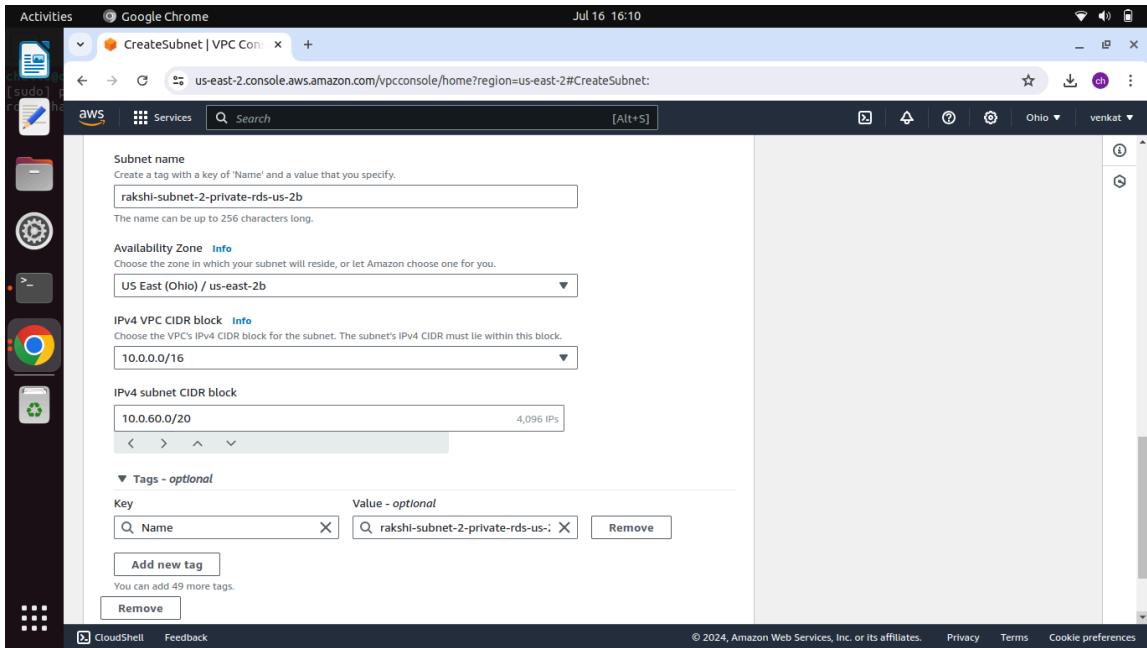


30) Again create 2 subnets in private for rds 2 (devil and rakshi)

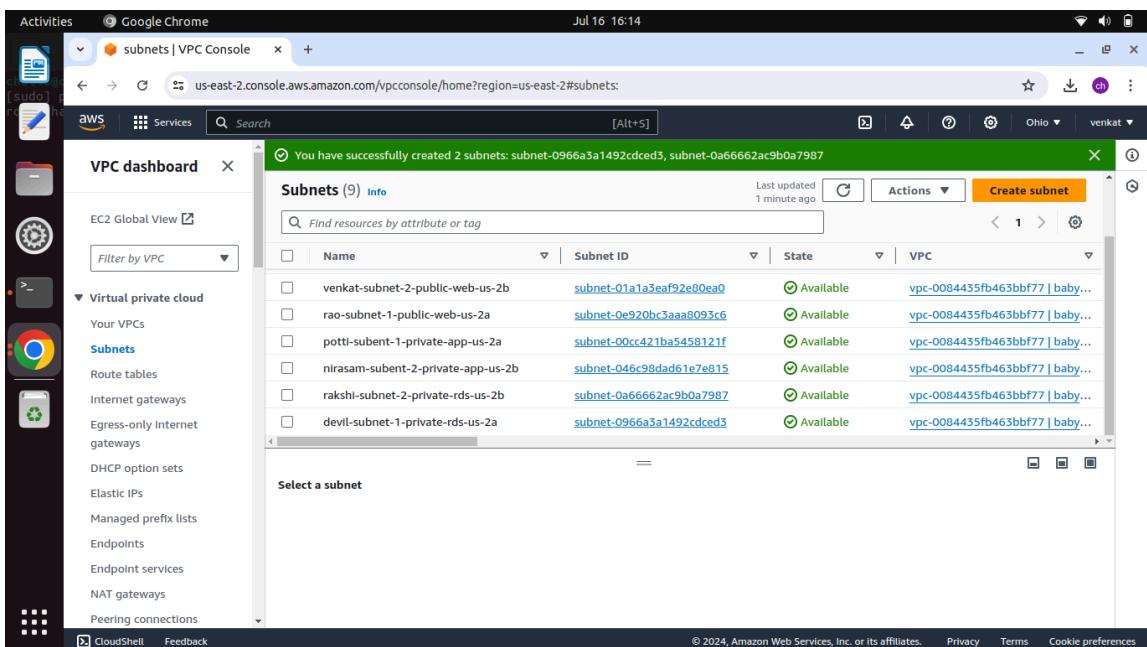
- 31) Subnets click create subnet
- 32) Click crete subnet select vpc= baby-vpc
- 33) Subnet name= devil-subnet-1-private-rds-us-2a
- 34) Availability zone=us-east-2a
- 35) IPv4 subnet CIDR block= 10.0.50.0/20
- 36) Click add subnet

- 37) Subnet name= rakshi-subnet-2-private-rds-us-2b
- 38) Availability zone=us-east-2b
- 39) IPv4 subnet CIDR block= 10.0.10.0/22





40) Click create subnet



41) Now create route tables

Route tables (2) info

Name	Route table ID	Explicit subnet associations	Main
-	rtb-05ddb63d8082e674b	-	Yes
-	rtb-086291f8ddb6afc7a	-	Yes

Select a route table

42) Route table click create route table

43) name= rao-route-1-web

44) vpc= baby-vpc

45) Click create route table

Route table settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.
rao-route-1-web

VPC
The VPC to use for this route table.
vpc-0084435fb463bbf77 (baby-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	Value - optional
Name	rao-route-1-web

Add new tag

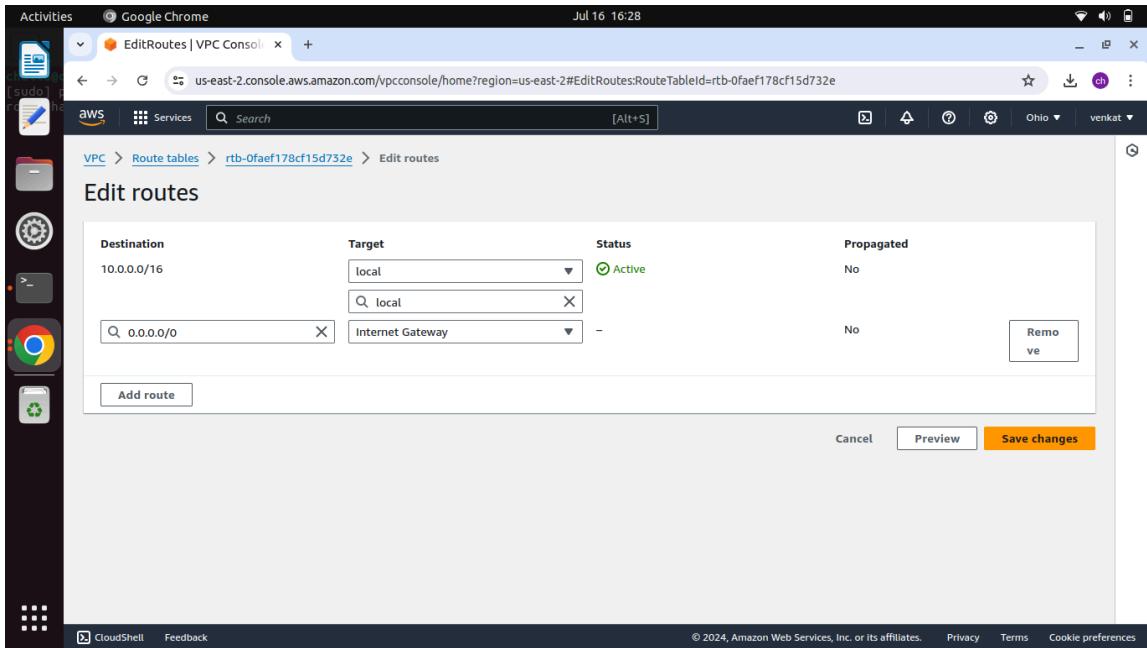
You can add 49 more tags.

Create route table

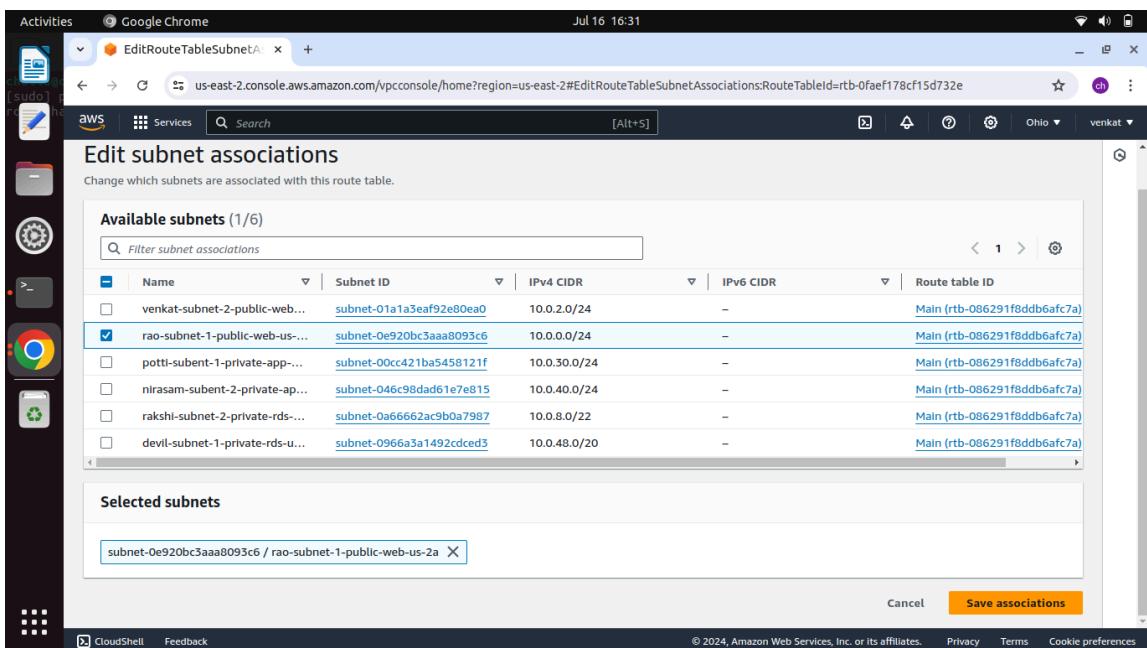
The screenshot shows the AWS VPC dashboard with the 'Route tables' section selected. A search bar at the top right contains the placeholder 'Find resources by attribute or tag'. Below it is a table titled 'Route tables (3) info' with columns for Name, Route table ID, Explicit subnet associations, Edge associations, and Main. The table lists three route tables: one unnamed (rtb-05ddb63d8082e674b), another unnamed (rtb-086291f8ddb6afc7a), and 'rao-route-1-web' (rtb-0faef178cf15d732e). The 'rao-route-1-web' row is highlighted with a blue selection bar. At the bottom of the table, there is a link 'Select a route table'.

46) Select rao-route-1-web click actions click edit routes add route destination=0.0.0.0/0 target= baby internet gateway click save changes

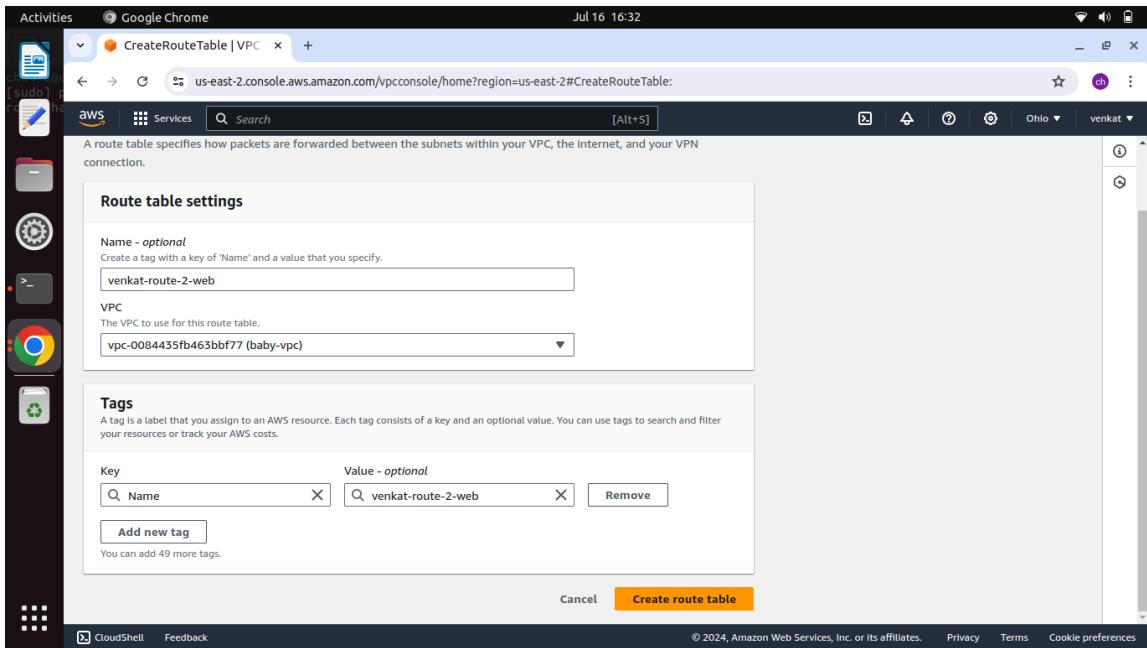
The screenshot shows the same VPC dashboard as before, but now the context menu for the 'rao-route-1-web' route table is open. The 'Edit routes' option is highlighted with a blue selection bar. The menu also includes options like 'View details', 'Set main route table', 'Edit subnet associations', 'Edit edge associations', 'Edit route propagation', 'Manage tags', and 'Delete route table'. Below the menu, the 'Details' tab of the route table configuration page is visible, showing the route table ID 'rtb-0faef178cf15d732e' and its association with the 'Main' edge.



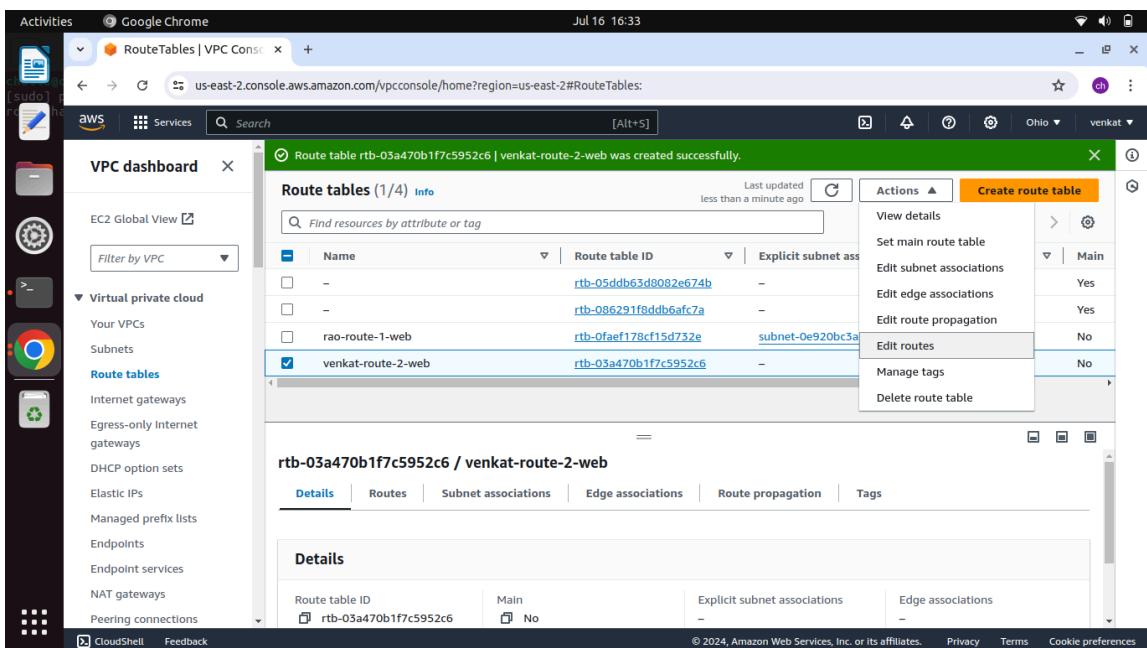
- 47) Again Select rao-route-1-web click actions click edit subnet associations select rao-subnet-1-public-web-us-2a and click save associations



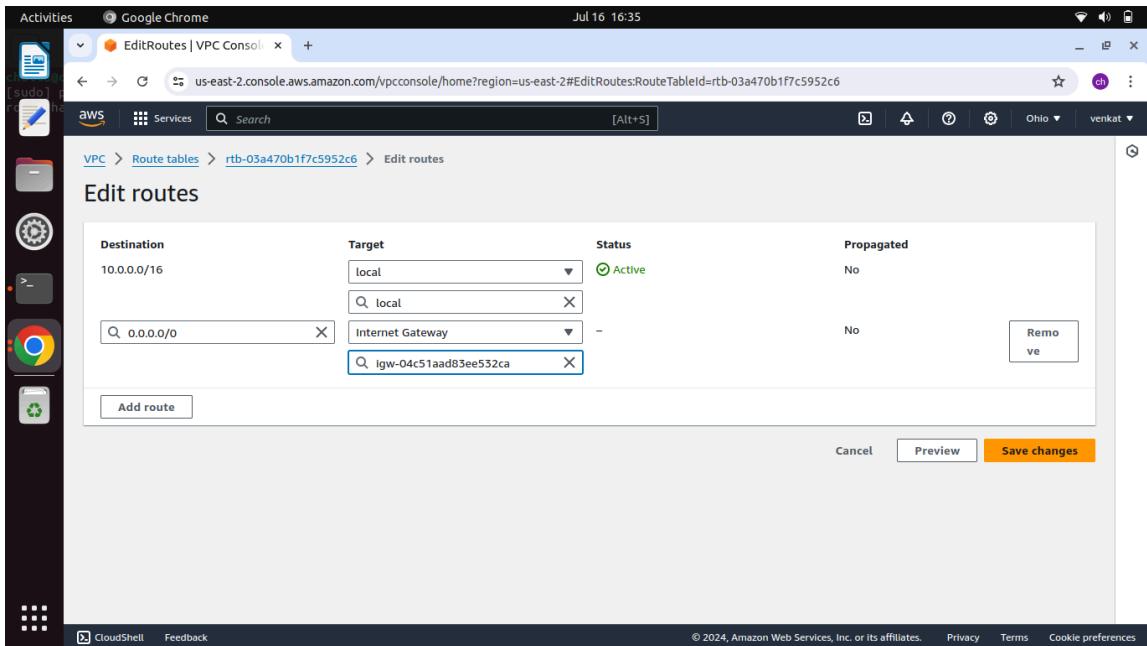
- 48) Route table click create route table
 49) name= venkat-route-2-web
 50) vpc= baby-vpc
 51) Click create route table



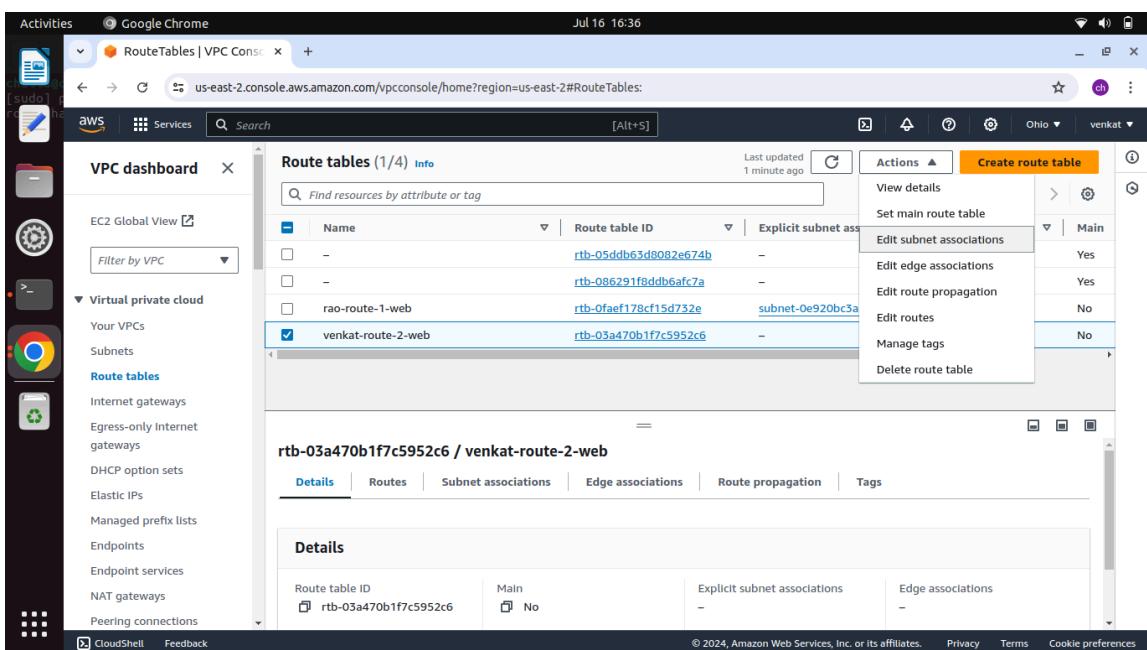
- 52) Select rao-route-1-web click actions click edit routes add route destination=0.0.0.0/0 target= baby internet gateway click save changes



- 53) Select venkat-route-2-web click actions click edit routes add route destination=0.0.0.0/0 target= baby internet gateway click save changes



- 54) Again Select venkat-route-2-web click actions click edit subnet associations select venkat-subnet-2-public-web-us-2b and click save associations



The screenshot shows the AWS VPC console with the URL us-east-2.console.aws.amazon.com/vpc/home?region=us-east-2#EditRouteTableSubnetAssociations:RouteTableId=rtb-03a470b1f7c5952c6. The page title is "Edit subnet associations". The "Available subnets" section contains 6 entries:

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/> venkat-subnet-2-public-web...	subnet-01a1a3eaf92e80ea0	10.0.2.0/24	-	Main (rtb-086291f8ddb6af7a)
<input type="checkbox"/> rao-subnet-1-public-web-us-...	subnet-0e920bc3aaa8093c6	10.0.0.0/24	-	rtb-0faef178cf15d732c / rao-7...
<input type="checkbox"/> potti-subnet-1-private-app...	subnet-0cc421ba5458121f	10.0.30.0/24	-	Main (rtb-086291f8ddb6af7a)
<input type="checkbox"/> nirasan-subnet-2-private-ap...	subnet-046c98dad61e7e815	10.0.40.0/24	-	Main (rtb-086291f8ddb6af7a)
<input type="checkbox"/> rakshi-subnet-2-private-rds...	subnet-0a66662ac9b0a7987	10.0.8.0/22	-	Main (rtb-086291f8ddb6af7a)
<input type="checkbox"/> devil-subnet-1-private-rds-u...	subnet-0966a3a1492cded3	10.0.48.0/20	-	Main (rtb-086291f8ddb6af7a)

The "Selected subnets" section contains one entry: "subnet-01a1a3eaf92e80ea0 / venkat-subnet-2-public-web-us-2b".

55) Now create route name=potti-nirasam-route-app

56) vpc=baby-vpc

The screenshot shows the AWS VPC console with the URL us-east-2.console.aws.amazon.com/vpc/home?region=us-east-2#CreateRouteTable. The page title is "CreateRouteTable | VPC".

Route table settings

- Name - optional**: potti-nirasam-route-app
- VPC**: vpc-0084435fb463bbf77 (baby-vpc)

Tags

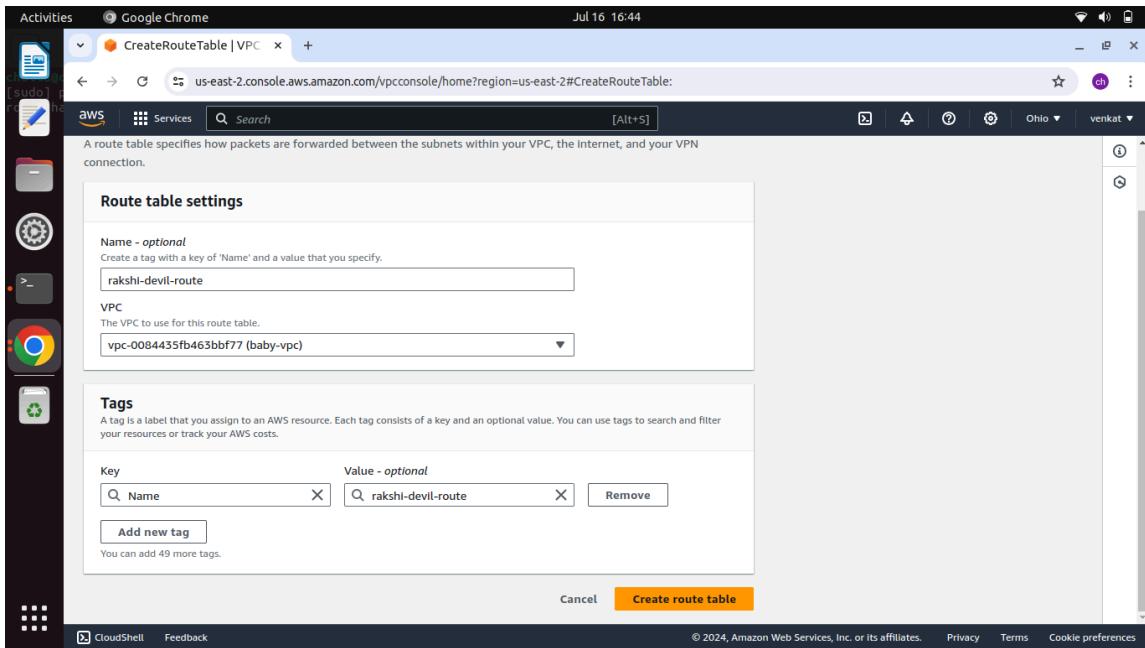
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
Name	potti-nirasam-route-app

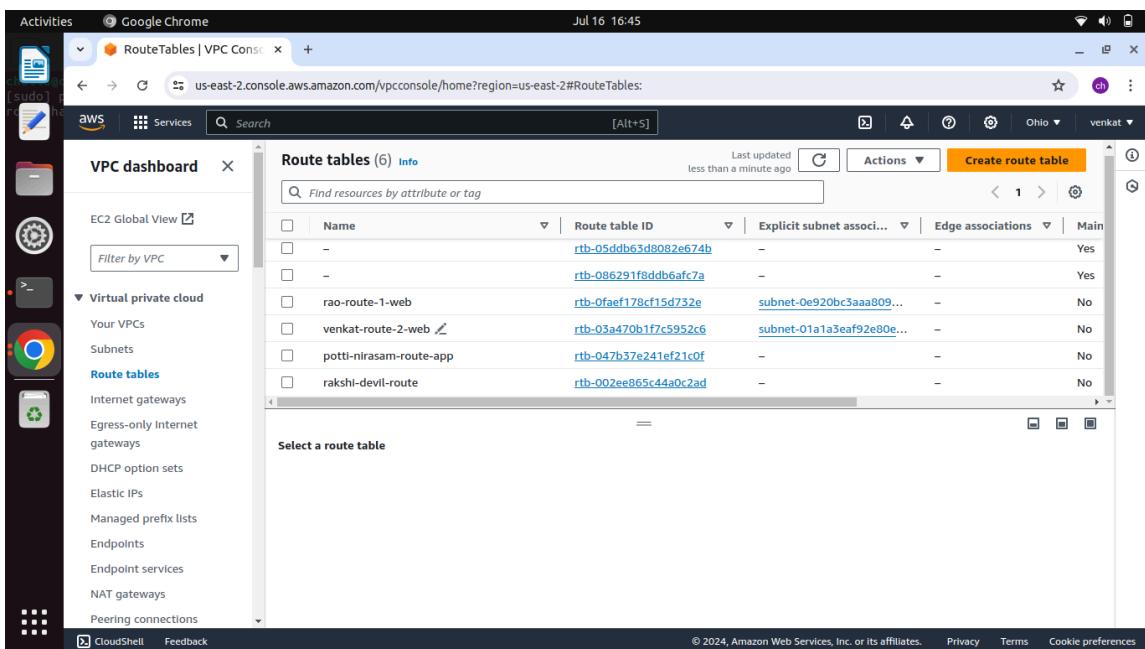
Create route table

57) Click create route table

58) Now gain create router table name=rakshi-devil-route select vpc= baby-vpc click create route table

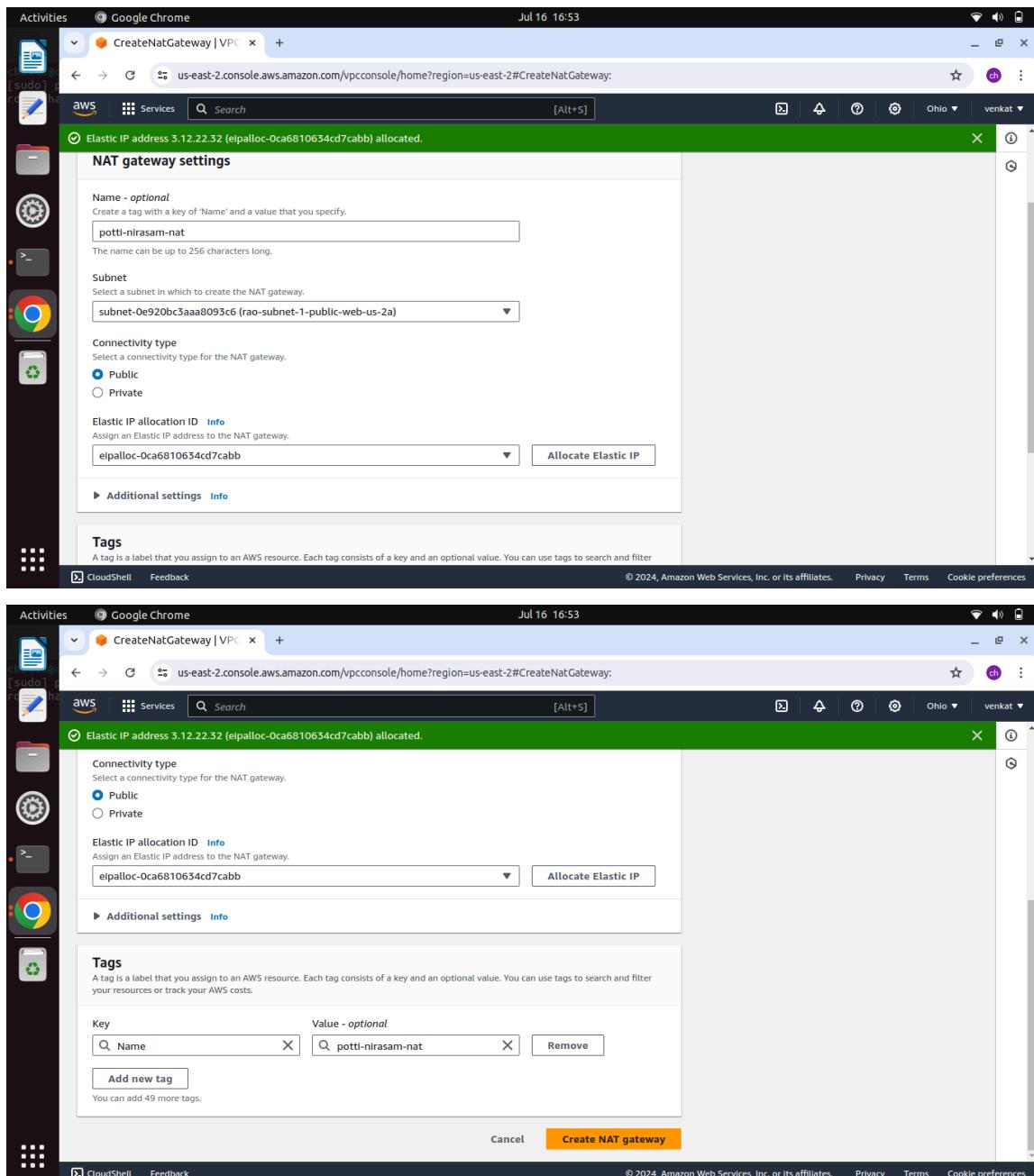


59) Click create route table



- 60) Now create nat-gateway and attach to route table
- 61) Go to NAT-gateway click create nat-gateway
- 62) name= potti-nirasam-nat
- 63) subnet= subnet-0e920bc3aaa8093c6 (rao-subnet-1-public-web-us-2a)
- 64) Connectivity = public
- 65) Click allocate elastic ip

66) Click create nat-gateway



67) Again creat 2nd nat-gateway

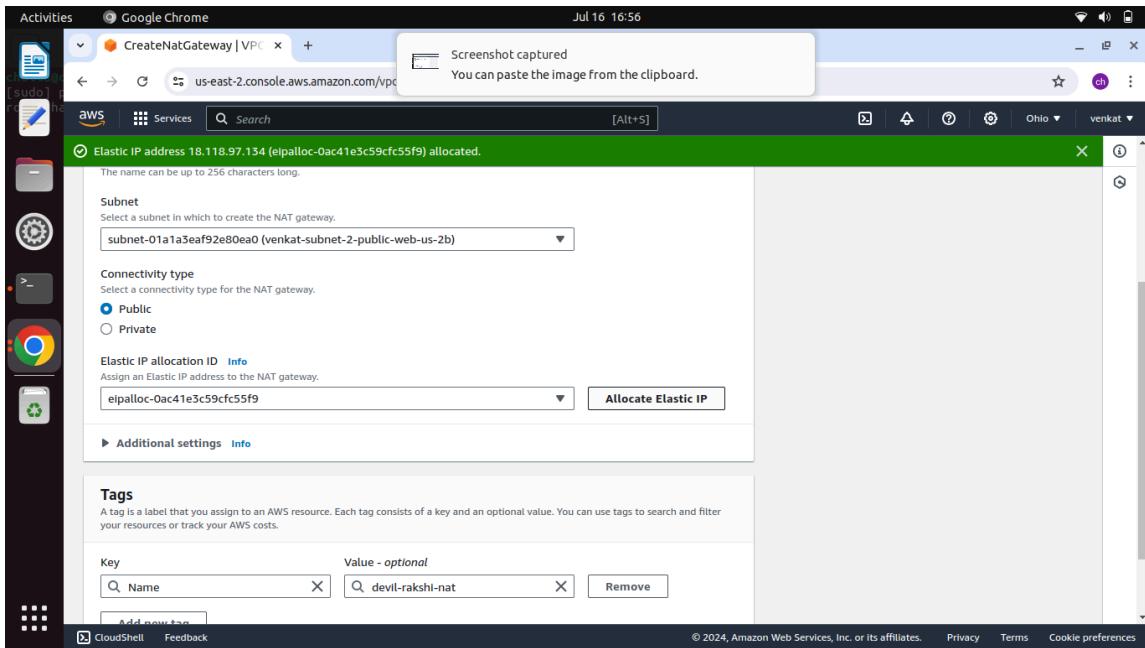
68) name= devil-rakshi-nat

69) subnet= subnet-01a1a3eaf92e80ea0 (venkat-subnet-2-public-web-us-2b)

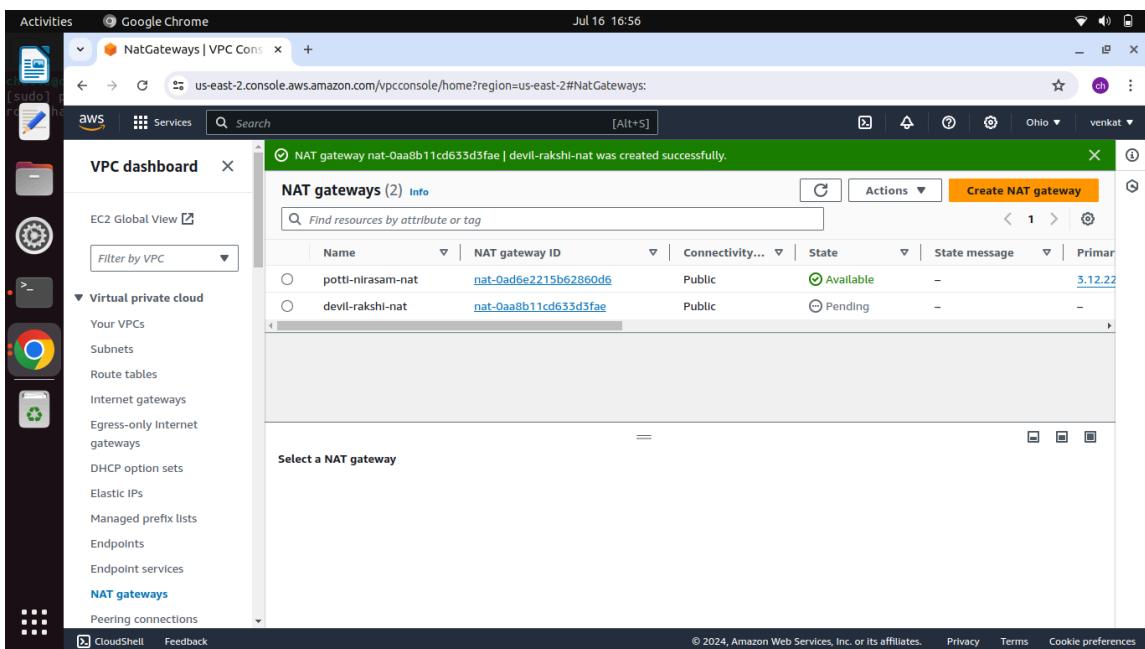
70) Connectivity = public

71) Click allocate elastic ip

72) Click create nat-gateway



73) Click create nat gateway



74) Now go to route table select potti-nirasam-route-app click actions click edit routes

The screenshot shows the AWS VPC console with the 'Route tables' page. A route table named 'potti-nirasam-route-app' is selected. The Actions menu is open, showing options like Set main route table, Edit subnet associations, and Edit routes.

75) Add route destination=0.0.0.0/0 and target= select potti-nirasam nat click save changes

The screenshot shows the 'Edit routes' dialog for the selected route table. A new route is being added with the following details:

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	NAT Gateway	-	No

The 'Save changes' button is highlighted at the bottom right of the dialog.

76) Again select potti-nirasam-route-app click actions click edit subnet associations

Route tables (1/6) Info

Name	Route table ID	Explicit subnet associations
-	rtb-05ddb63d8082e674b	-
-	rtb-086291f8ddb6afc7a	-
rao-route-1-web	rtb-0faef178cf15d732e	subnet-0e920bc...
venkat-route-2-web	rtb-03a470b1f7c5952c6	subnet-01a1a3e...
potti-nirasam-route-app	rtb-047b37e241ef21c0f	-
rakshi-devil-route	rtb-002ee865c44a0c2ad	-

rtb-047b37e241ef21c0f / potti-nirasam-route-app

Details | Routes | Subnet associations | Edge associations | Route propagation | Tags

Details

Route table ID rtb-047b37e241ef21c0f	Main No	Explicit subnet associations -	Edge associations -
---	------------	-----------------------------------	------------------------

77) Select potti and nirasam subnets and click save associations

Edit subnet associations

Available subnets (2/6)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
venkat-subnet-2-public-web...	subnet-01a1a3eaf92e80ea0	10.0.2.0/24	-	rtb-03a470b1f7c5952c6 / venk...
rao-subnet-1-public-web-us...	subnet-0e920bc3aaa8093c6	10.0.0.0/24	-	rtb-0faef178cf15d732e / rao-...
potti-subnet-1-private-app...	subnet-00cc421ba5458121f	10.0.30.0/24	-	Main (rtb-086291f8ddb6afc7a)
nirasam-subnet-2-private-ap...	subnet-046c98dad61e7e815	10.0.40.0/24	-	Main (rtb-086291f8ddb6afc7a)
rakshi-subnet-2-private-rds...	subnet-0a66662ac9b0a7987	10.0.8.0/22	-	Main (rtb-086291f8ddb6afc7a)
devil-subnet-1-private-rds-u...	subnet-0966a3a1492cdced3	10.0.48.0/20	-	Main (rtb-086291f8ddb6afc7a)

Selected subnets

subnet-00cc421ba5458121f / potti-subnet-1-private-app-us-2a | subnet-046c98dad61e7e815 / nirasam-subnet-2-private-app-us-2b

Cancel | **Save associations**

78) Now select rakshi-devil-route click actions click edit routes

79) Click add route destination= 0.0.0.0/0 and target= devil-rakshi-nat select click save changes

80) Now again select rakshi-devil-route click actions click edit subnet associations

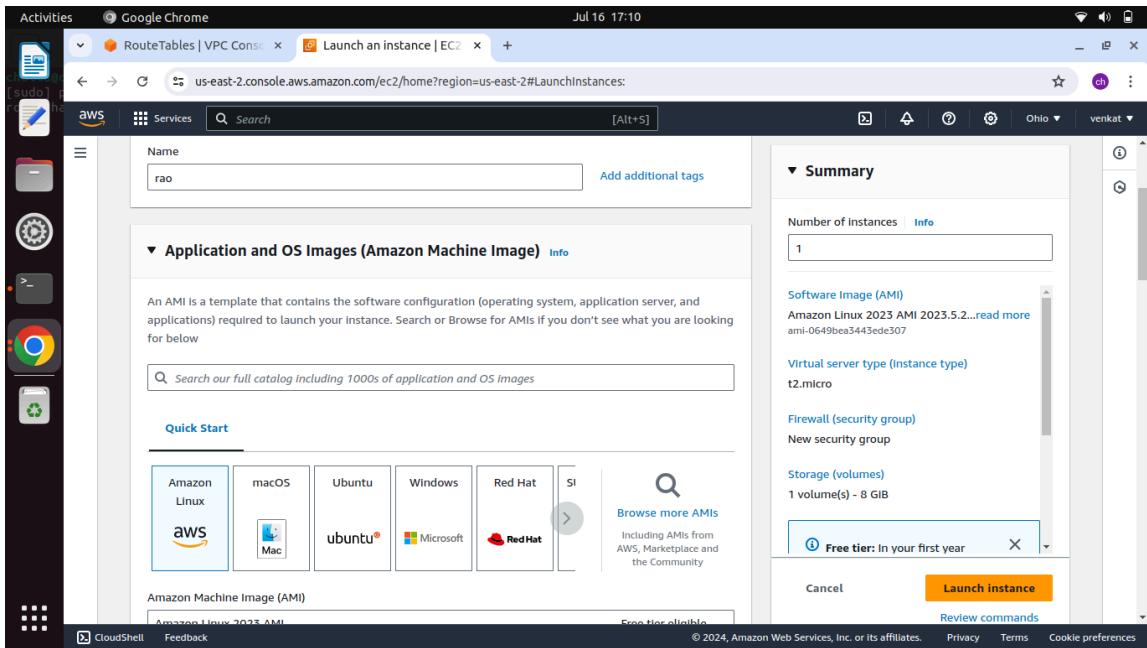
The screenshot shows the AWS VPC dashboard with the 'Route tables' section selected. A route table named 'rakshi-devil-route' is highlighted. A context menu is open over this table, with the option 'Edit subnet associations' highlighted in orange. The table list shows several other route tables like 'rao-route-1-web', 'venkat-route-2-web', etc.

81) Select rakshi and devil subnets click save associations

The screenshot shows the 'Edit subnet associations' dialog for the route table 'rtb-002ee865c44a0c2ad'. In the 'Available subnets' list, two subnets are selected: 'rakshi-subnet-2-private-rds-us-2b' and 'devil-subnet-1-private-rds-us-2a'. These are listed under the 'Selected subnets' section. At the bottom right, there is a 'Save associations' button.

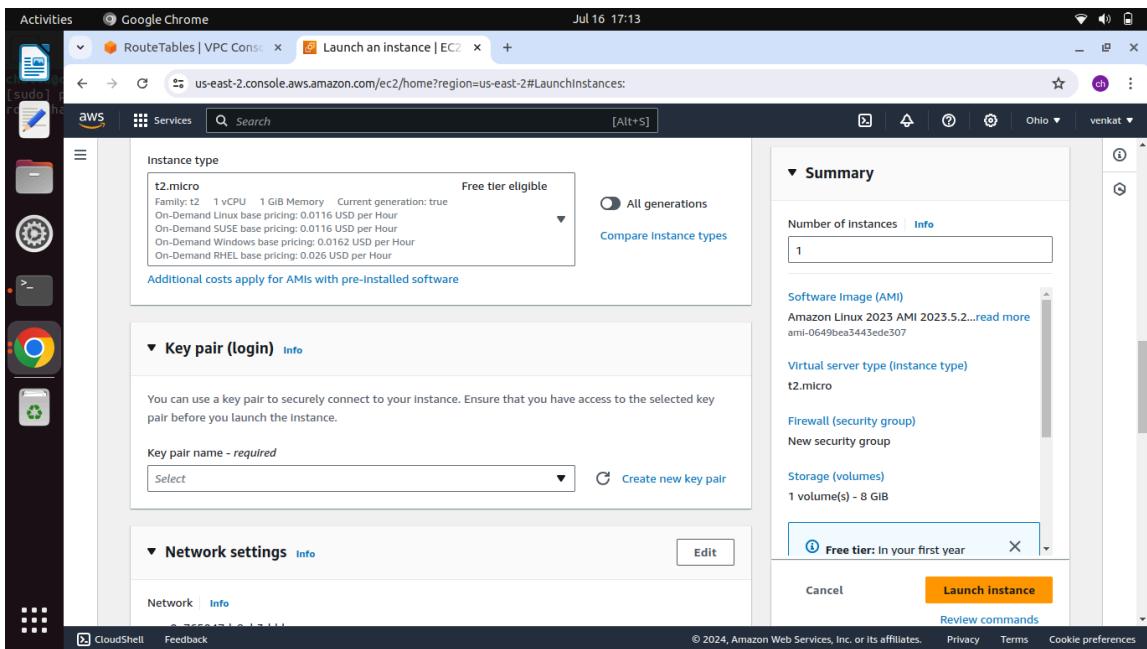
82) Now go ec2

83) Click launch instance name= rao

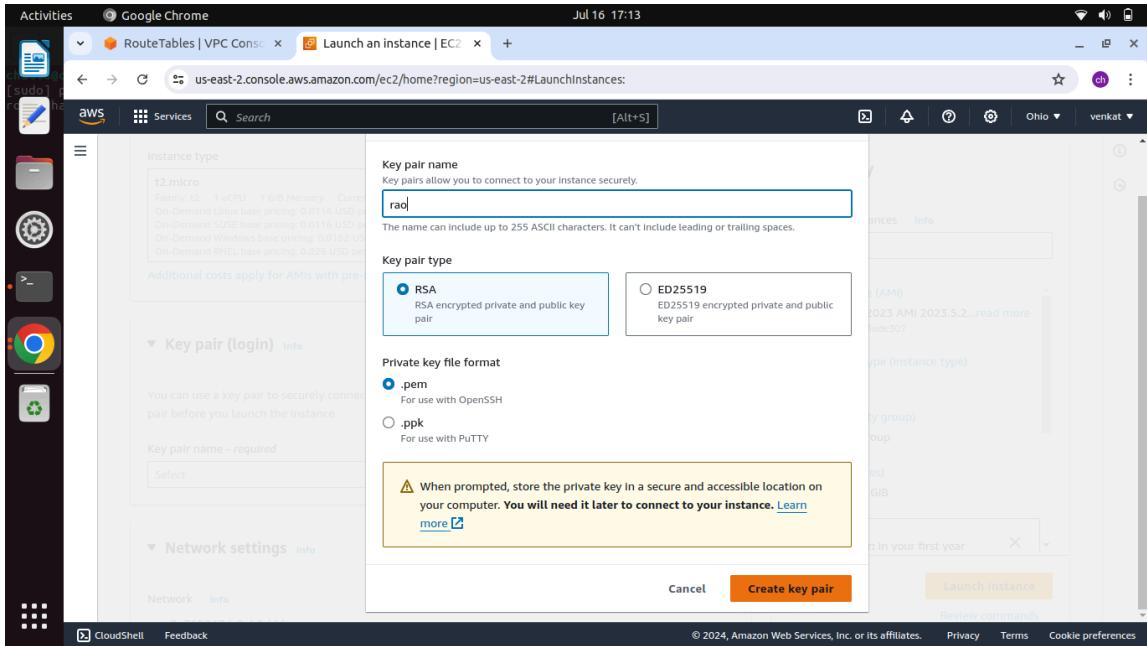


84) os= amazon linux

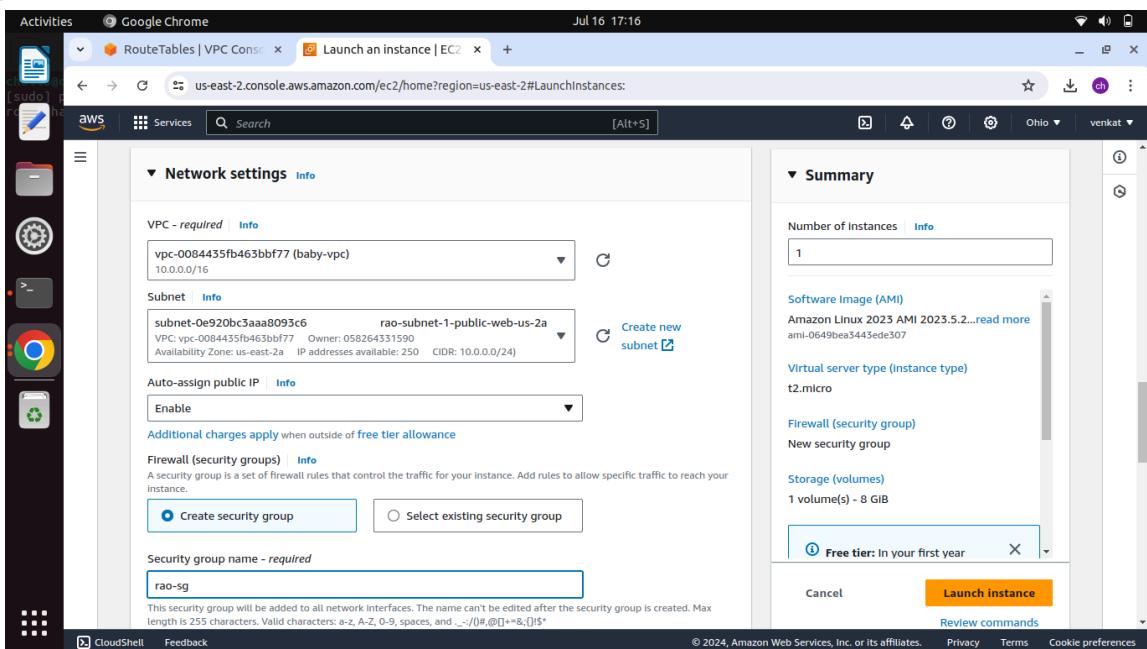
85) Instance =t2.micro



86) Click create key pair name=rao click create key pair



- 87) Network settings click edit
- 88) vpc= baby-vpc
- 89) subnet= select rao subnet
- 90) Auto assign public ip= enable
- 91) Click create security group name= rao-sg
- 92) Click launch instance



The image shows two screenshots of the AWS EC2 console. The top screenshot is the 'Launch an instance' wizard, step 3: 'Configure security group'. It shows a summary of the configuration: 1 instance, Amazon Linux 2023 AMI 2023.5.2..., t2.micro instance type, and 1 volume (8 GiB). A security group rule for SSH (TCP port 22) from Anywhere is listed. A warning message states: 'Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' The bottom screenshot shows the 'Instances' page with one instance named 'rao' (ID: i-00004c997d475d087) running. The sidebar on the left includes links for EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, and Elastic Block Store.

93) Again create 2nd server = venkat

Activities Google Chrome Jul 16 17:19

RouteTables | VPC Console Launch an instance | EC2

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#LaunchInstances:

aws Services Search [Alt+S]

Name: venkat 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 SI

Mac ubuntu Microsoft RedHat

Browse more AMIs Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI

Free tier eligible

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

This screenshot shows the AWS EC2 console interface for launching a new instance. The 'Quick Start' tab is selected, showing a grid of popular AMI icons: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and others. A search bar is available to find specific AMIs. The 'Summary' section on the right indicates 1 instance will be launched, using the 'Amazon Linux 2023 AMI'. The 'Virtual server type (instance type)' is set to 't2.micro'. Other settings include a 'New security group' and '1 volume(s) - 8 GB'. A 'Launch instance' button is prominently displayed.

Activities Google Chrome Jul 16 17:19

RouteTables | VPC Console Launch an instance | EC2

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#LaunchInstances:

aws Services Search [Alt+S]

Create key pair

Key pair name: venkat

Key pairs allow you to connect to your instance securely. The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type: RSA

RSA encrypted private and public key pair

ED25519

ED25519 encrypted private and public key pair

Private key file format: .pem

.pem For use with OpenSSH

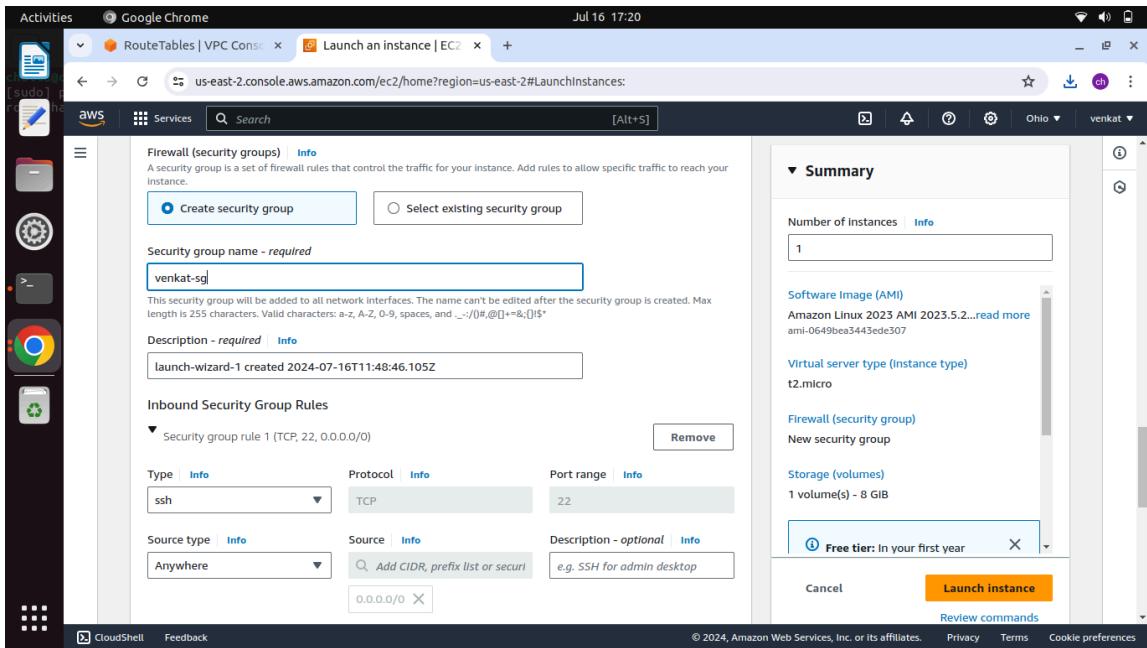
.ppk For use with PuTTY

When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. Learn

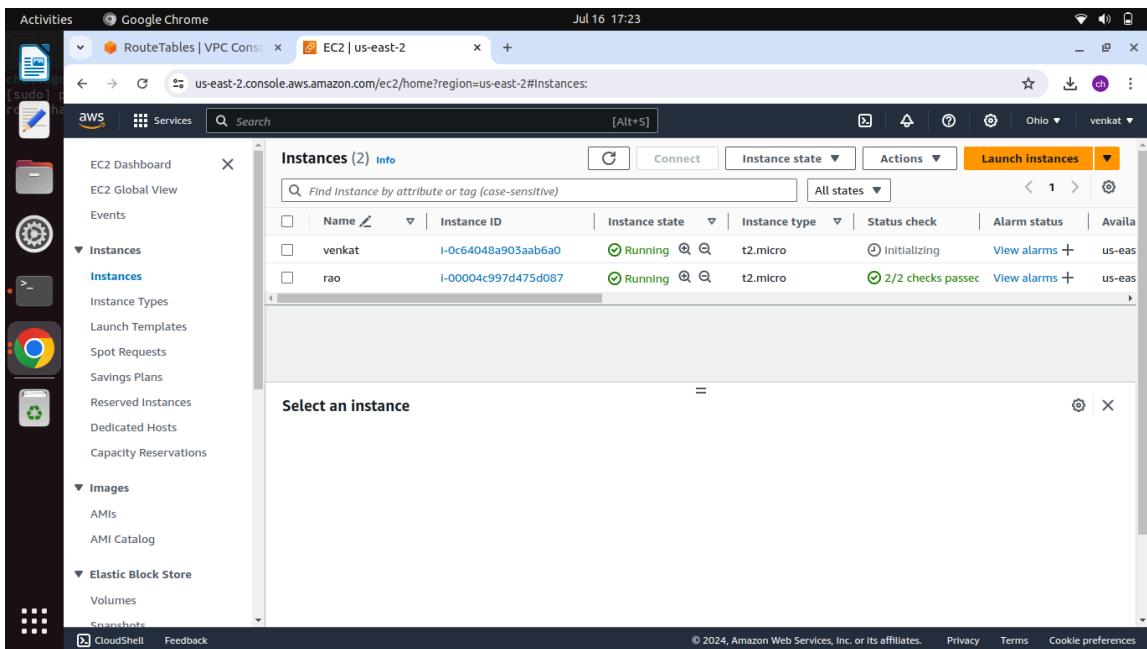
Cancel Create key pair

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

This screenshot shows a modal dialog titled 'Create key pair'. It asks for a 'Key pair name' (set to 'venkat') and provides options for 'Key pair type': RSA (selected) and ED25519. It also specifies the 'Private key file format' as '.pem'. A note at the bottom instructs the user to store the private key securely for future connection. The 'Create key pair' button is highlighted.



- 94) Keypair = venkat
- 95) Network settings click edit
- 96) vpc= baby-vpc
- 97) subnet= select venkat subnet
- 98) Auto assign public ip= enable
- 99) Click create security group name= venkat-sg
- 100) Click launch instance



- 101) Now create for app 2 servers
- 102) Name= potti-app
- 103) Keypair =potti-app

Activities Google Chrome Jul 16 17:29

RouteTables | VPC Console | Launch an instance | EC2

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#LaunchInstances:

EC2 Instances Launch an instance

Launch an instance

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

Name: potti-app

Application and OS Images (Amazon Machine Image)

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

Summary

Number of Instances: 1

Software Image (AMI): Amazon Linux 2023.5.2...read more
ami-0649bea54435ede307

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

Cancel Launch instance Review commands

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

Activities Google Chrome Jul 16 17:29

RouteTables | VPC Console | Launch an instance | EC2

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#LaunchInstances:

EC2 Instances Launch an instance

Instance type

12.micro Family: 12 vCPU, 7.68 Memory, Compute On-Demand Linux base pricing: \$0.11/Hr USD per instance per hour

Additional costs apply for AMIs with pre-installed software.

Key pair name

Key pairs allow you to connect to your instance securely.

potti-app

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

RSA RSA encrypted private and public key pair

ED25519 ED25519 encrypted private and public key pair

Private key file format

.pem For use with OpenSSH

.ppk For use with PuTTY

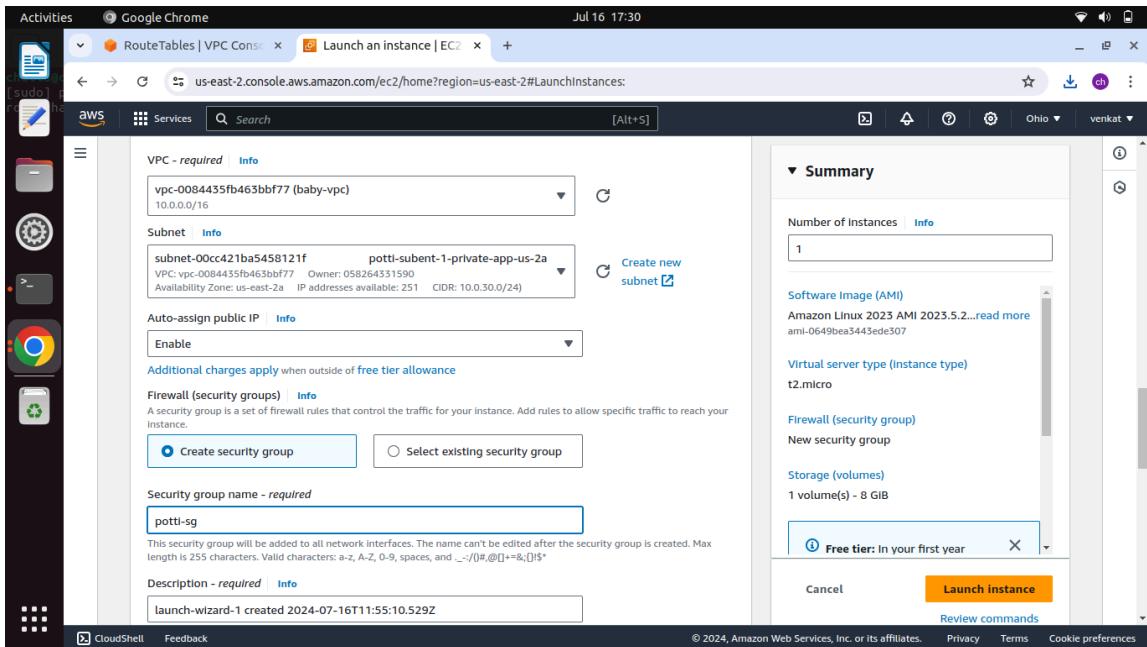
⚠ When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

Cancel Create key pair

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

- 104) Network settings click edit
- 105) vpc= baby-vpc
- 106) subnet= select potti subnet
- 107) Auto assign public ip= enable

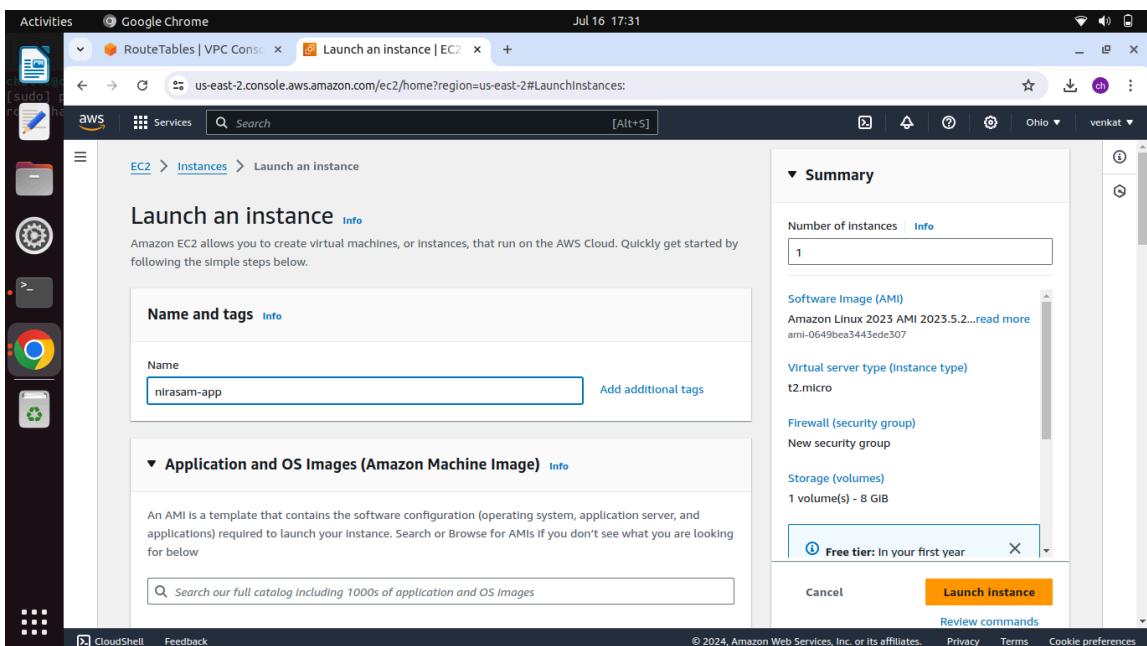
108) Click create security group name= potti-sg



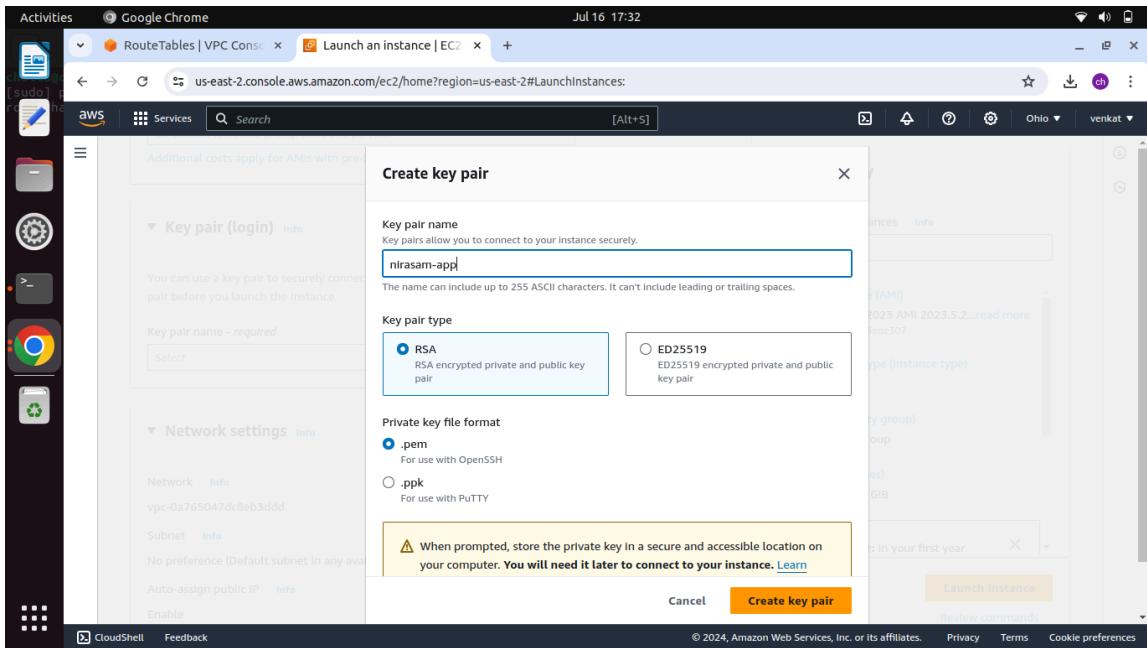
109) Click launch instance

110) Again crete nirasam server

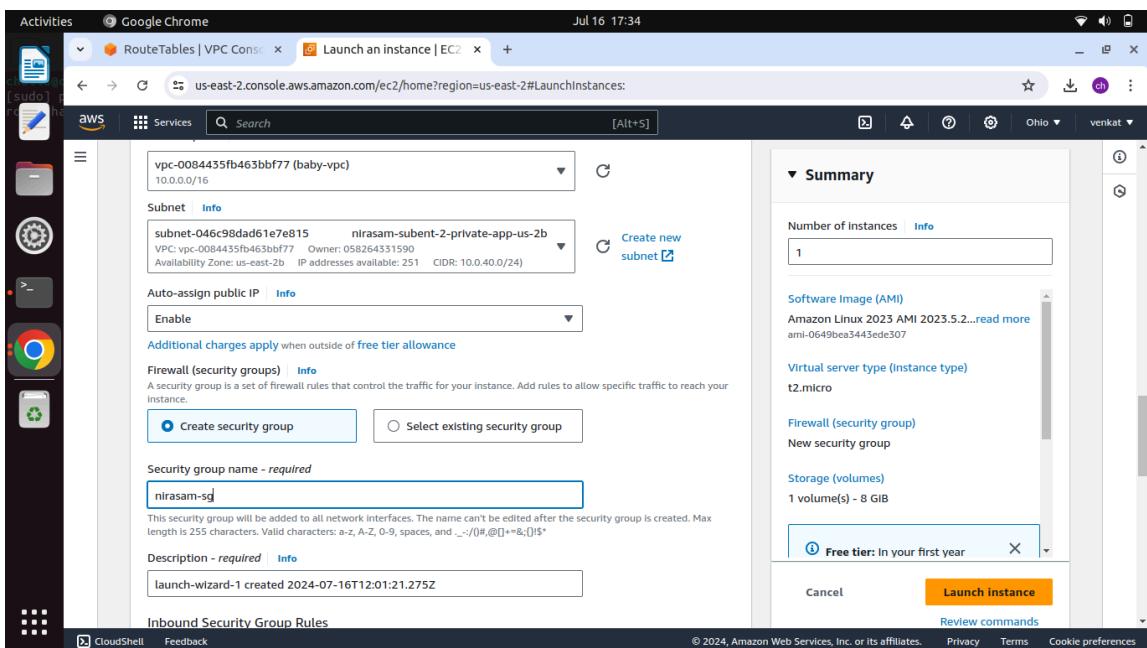
111) name= nirasam-app



112) Click create key pair = nirasam-app



- 113) Network settings click edit
- 114) vpc= baby-vpc
- 115) subnet= select nirasam subnet
- 116) Auto assign public ip= enable
- 117) Click create security group name= nirasam-sg



- 118) Click launch instance

The screenshot shows the AWS EC2 Instances page. The left sidebar includes links for EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instances Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, and Elastic Block Store. The main content area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
nirasam-app	i-0c3ffbd1f82ad84d2	Running	t2.micro	-	2/2 checks passed	us-eas
venkat	i-0c64048a903aab6a0	Running	t2.micro	2/2 checks passed	View alarms +	us-eas
potti-app	i-0b8f5606fb0d01614c	Running	t2.micro	2/2 checks passed	View alarms +	us-eas
rao	i-00004c997d475d087	Running	t2.micro	2/2 checks passed	View alarms +	us-eas

A modal window titled "Select an instance" is open at the bottom, showing the same list of instances.

119) Now connect rao server set hostname=rao and using rao server connect nirasam and potti private servers

120) Select rao-web click connect

The screenshot shows the AWS EC2 Instances page with the rao-web instance selected. The left sidebar is identical to the previous screenshot. The main content area shows the selected instance in the table:

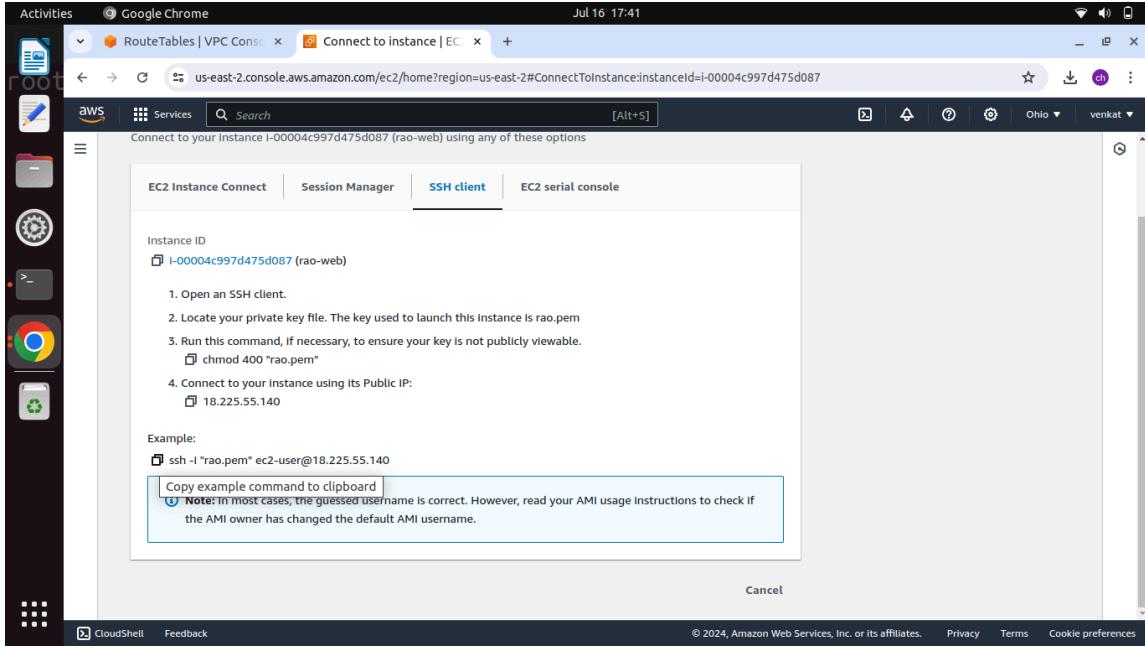
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
nirasam-app	i-0c3ffbd1f82ad84d2	Running	t2.micro	-	View alarms +	us-eas
venkat-web	i-0c64048a903aab6a0	Running	t2.micro	2/2 checks passed	View alarms +	us-eas
potti-app	i-0b8f5606fb0d01614c	Running	t2.micro	2/2 checks passed	View alarms +	us-eas
rao-web	i-00004c997d475d087	Running	t2.micro	2/2 checks passed	View alarms +	us-eas

A detailed view of the rao-web instance is shown below, with the "Details" tab selected. The "Instance summary" section contains the following information:

- Instance ID: i-00004c997d475d087 (rao-web)
- Public IPv4 address: 18.225.55.140 | [open address](#)
- Private IPv4 addresses: 10.0.0.26
- IPv6 address: -
- Instance state: Running
- Public IP DNS name (IPv4 only): -
- Hostname type: -

121) Click ssh client

122) Select example blow ssh command



123) Go to key-pair location open set rao keypair file permission
400

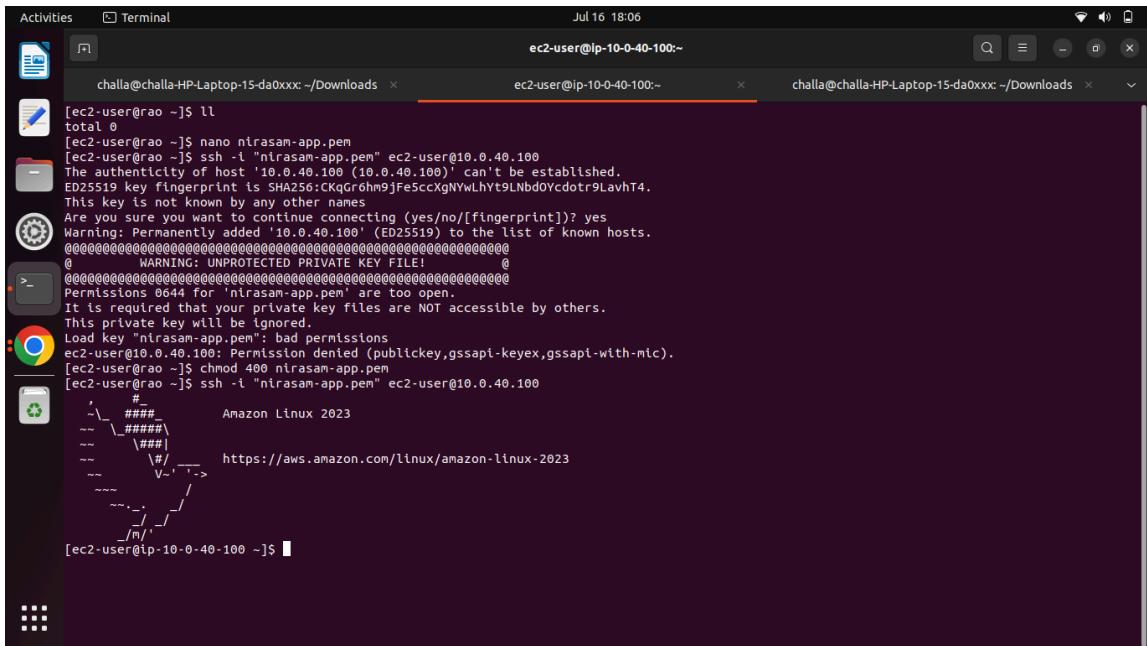
124) Now connect ssh copy cmd

```
Activities Terminal Jul 16 17:44
ec2-user@ip-10-0-0-26:~$ cd Downloads/
ec2-user@ip-10-0-0-26:~/Downloads$ ll
total 106320
drwxr-x 2 challa challa 4096 Jul 16 17:32 .
drwxr-x--- 19 challa challa 4096 Jul 16 13:13 ../
drwxr-- 1 challa challa 108773084 Jul 12 20:44 google-chrome-stable_current_amd64.deb
drwxr-- 1 challa challa 1674 Jul 16 17:32 nirasam-app.pem
drwxr-- 1 challa challa 1674 Jul 16 17:25 pilli-app.pem
drwxr-- 1 challa challa 1674 Jul 16 17:29 potti-app.pem
drwxr-- 1 challa challa 1678 Jul 16 17:13 rao.pem
drwxr-- 1 challa challa 1674 Jul 16 17:20 venkat.pem
drwxr-- 1 challa challa 66853 Jul 16 15:28 'WhatsApp Image 2024-07-16 at 3.28.41 PM.jpeg'
challa@ip-10-0-0-26:~/Downloads$ chmod 400 rao.pem
challa@ip-10-0-0-26:~/Downloads$ ssh -l "rao.pem" ec2-user@18.225.55.140
Warning: authenticity of host '18.225.55.140 (18.225.55.140)' can't be established.
ECDSA key fingerprint is SHA256:xaktTBswDT3D6vsHT4uvlzXp1iqPhcBa5HTLDoZdSc.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '18.225.55.140' (ED25519) to the list of known hosts.
#_ 
### 
\## 
\#/ https://aws.amazon.com/linux/amazon-linux-2023
/ 
/ 
/ 
[ec2-user@ip-10-0-0-26 ~]$
```

125) Set hostname rao

126) Now copy nirasam-app.pem file to rao server using nano
command set nirasam pem also 400 file permissions

- 127) Now connect nirasam-app server using ssh -i "nirasam-app.pem" ec2-user@10.0.40.100
128) 10.0.40.100 = nirasam server private ip



The screenshot shows a terminal window titled "Terminal" with three tabs. The active tab shows the command:

```
challa@challa-HP-Laptop-15-da0xxx: ~/Downloads ~
```

Output from the terminal:

```
[ec2-user@rao ~]$ ll
total 0
[ec2-user@rao ~]$ nano nirasam-app.pem
[ec2-user@rao ~]$ ssh -l "nirasam-app.pem" ec2-user@10.0.40.100
The authenticity of host '10.0.40.100 (10.0.40.100)' can't be established.
ED25519 key fingerprint is SHA256:CKqGr6hm9jFe5ccXgNYwLhYt9LNbd0Ycdotr9LavhT4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.40.100' (ED25519) to the list of known hosts.
@@@@@@@ WARNING: UNPROTECTED PRIVATE KEY FILE! @@@@
Permissions 0644 for 'nirasam-app.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "nirasam-app.pem": bad permissions
ec2-user@10.0.40.100: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[ec2-user@rao ~]$ chmod 400 nirasam-app.pem
[ec2-user@rao ~]$ ssh -l "nirasam-app.pem" ec2-user@10.0.40.100
,
  #
  \_ ####_
  -- \####\ Amazon Linux 2023
  -- \###\ https://aws.amazon.com/linux/amazon-linux-2023
  -- \#/  V,->
  -- /  /
  ~~ /  /
  /n/  /
```

[ec2-user@ip-10-0-40-100 ~]\$

- 129) Now create load balancing 2 app servers
130) Go to security group create security group
131) Name= rao-venkat-app-sg
132) vpc= baby vpc
133) Add inbound rules= ssh and http click create security group

The screenshot shows two consecutive screenshots of the AWS EC2 Create Security Group interface.

Screenshot 1 (Top): Basic Details

- Security group name:** rao-venkat-app-sg
- Description:** allow
- VPC:** vpc-0084435fb463bbf77 (baby-vpc)

Screenshot 2 (Bottom): Inbound Rules

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	Cus... <input type="button" value="Delete"/>	
HTTP	TCP	80	Cus... <input type="button" value="Delete"/>	

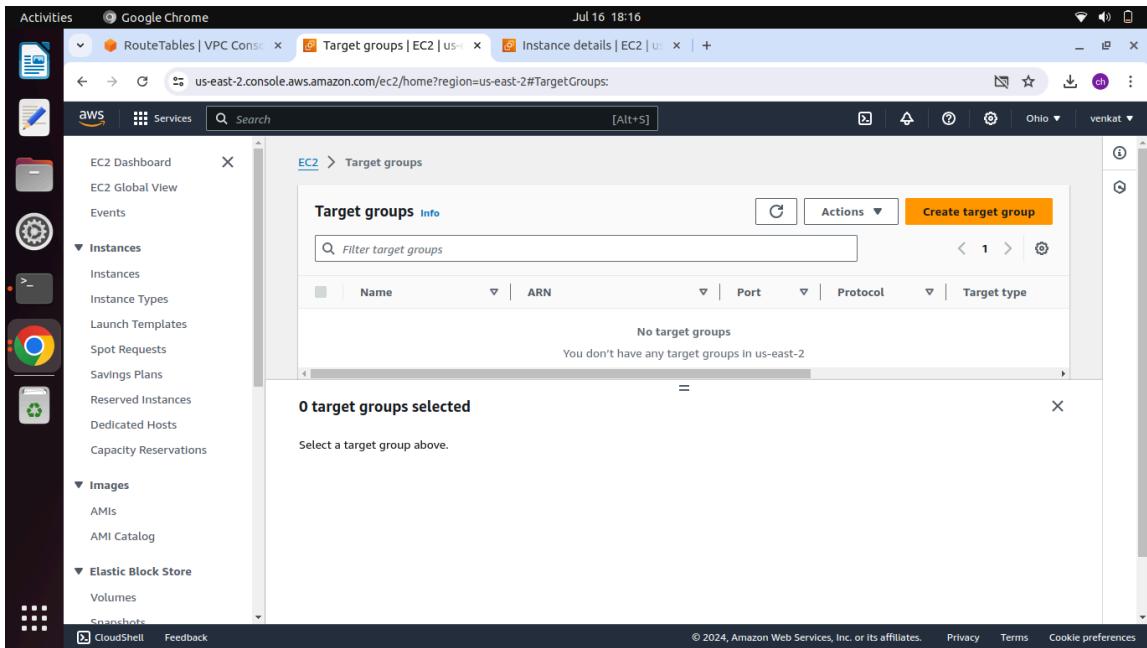
Inbound Rules (Summary):

- SSH (TCP, Port 22) from Custom source
- HTTP (TCP, Port 80) from Custom source

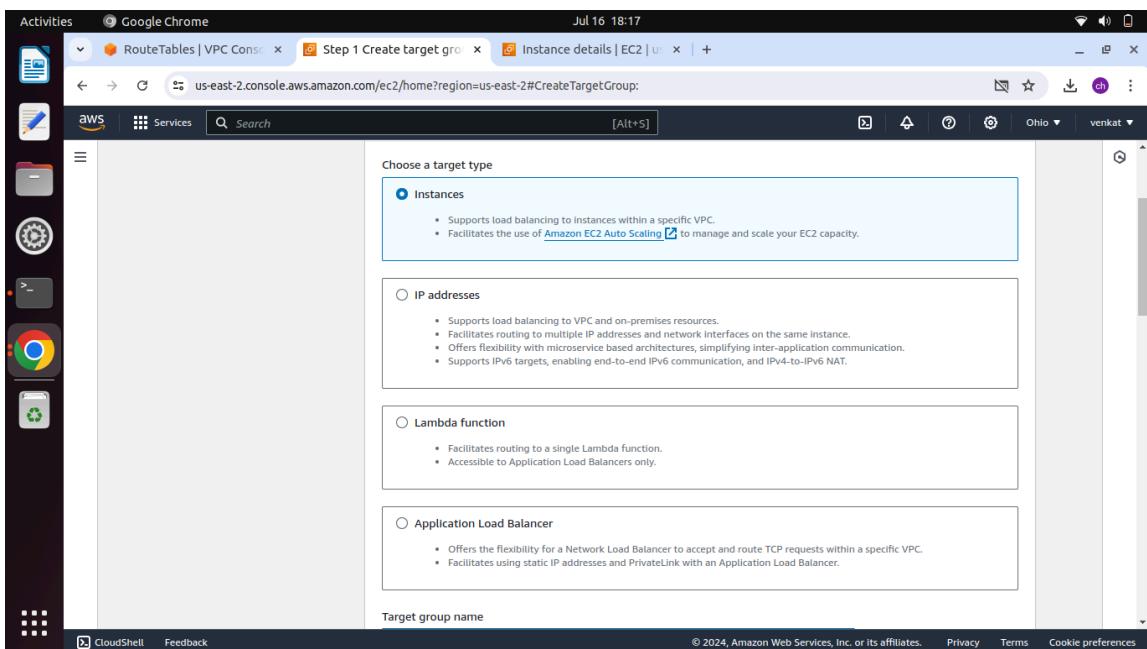
Outbound Rules (Summary):

- All traffic (All, All) to 0.0.0.0/0

134) Go to target group click crete target group



- 135) Select instances
- 136) Target group name=venkat-rao-tg
- 137) vpc= baby-vpc
- 138) Click =next
- 139) Select rao and venkat server and click include as pending below



Activities Google Chrome Jul 16 18:18

RouteTables | VPC Console Step 1 Create target group Instance details | EC2 | ...

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#CreateTargetGroup:

aws Services Search [Alt+S]

Target group name: rao-venkat-tg

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol : Port

Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation.

HTTP 80 1-65535

IP address type

Only targets with the indicated IP address type can be registered to this target group.

IPv4

Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

IPv6

Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). [Learn more](#)

VPC

Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

vpc-0a765047dc8eb3ddd
IPv4 VPC CIDR: 172.31.0.0/16

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

Activities Google Chrome Jul 16 18:19

RouteTables | VPC Console Step 1 Create target group Instance details | EC2 | ...

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#CreateTargetGroup:

aws Services Search [Alt+S]

Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

baby-vpc
vpc-0084435fb463bbf77
IPv4 VPC CIDR: 10.0.0.0/16

Protocol version

HTTP1

Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

HTTP2

Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

gRPC

Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

HTTP

Health check path

Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred.

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

The screenshot shows the AWS Lambda console interface. At the top, there's a navigation bar with tabs for 'Functions', 'Events', 'Logs', and 'Metrics'. Below the navigation bar, a search bar is present. The main content area displays a table with two rows of data. The first row contains the function name 'HelloWorld' and the ARN 'arn:aws:lambda:us-east-2:123456789012:function:HelloWorld'. The second row contains the function name 'HelloWorld' and the ARN 'arn:aws:lambda:us-east-2:123456789012:function:HelloWorld'. Both rows have a status column showing 'Running' with a green circle icon. The table also includes columns for 'Last modified', 'Code size', 'Memory', and 'Timeout'. At the bottom of the page, there are buttons for 'Create function' and 'Edit function'.

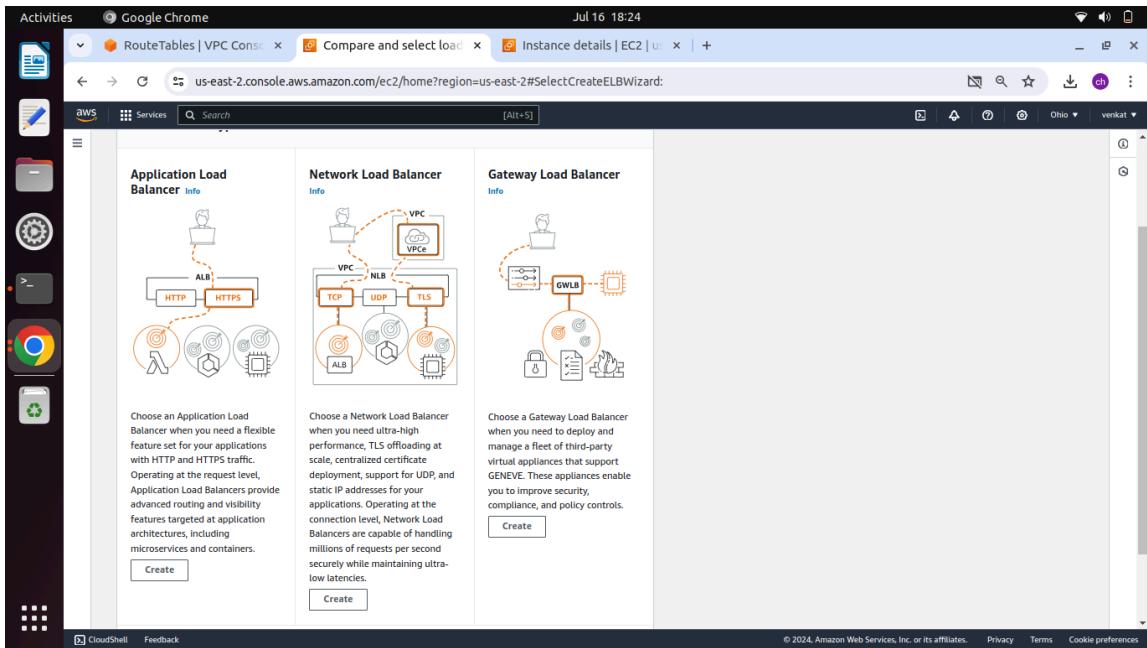
140) Click create target group

The screenshot shows the AWS EC2 Target groups console. The left sidebar includes options like EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, and Elastic Block Store (Volumes, Snapshots). The main pane displays 'Target groups (1/1)' with a table. One row is selected for 'rao-venkat-tg'. The details for this target group are shown in a modal, including its ARN, port (80), protocol (HTTP), and target type (Instance). The VPC ID listed is 'vpc-0084435fb463bbf77'.

141) Now click load balancer

The screenshot shows the AWS EC2 Load balancers console. The left sidebar includes options like Elastic Block Store (Volumes, Snapshots), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (selected), Load Balancers (selected), Target Groups, Trust Stores (New), Auto Scaling, and Auto Scaling Groups. The main pane displays 'Load balancers' with a table showing 'No load balancers'. A message at the bottom says 'Select a load balancer above.'

142) Click create load balancer



- 143) Select application load click create
- 144) Load balancer name= rao-venkat-loadbalancer
- 145) scheme=internet facing
- 146) Select ipv4
- 147) vpc= baby-vpc
- 148) Availability zones select both 2a and 2b and select public subnets
- 149) sg= rao-venkat-sg
- 150) Listener routing= rao-venkat-tg

Activities Google Chrome Jul 16 18:27

RouteTables | VPC Cons... Create application load balancer Instance details | EC2 | ... +

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#CreateALBWizard:

Basic configuration

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme [Info](#)
Scheme can't be changed after the load balancer is created.

Internet-facing
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

Internal
An internal load balancer routes requests from clients to targets using private IP addresses. Compatible with the IPv4 and Dualstack IP address types.

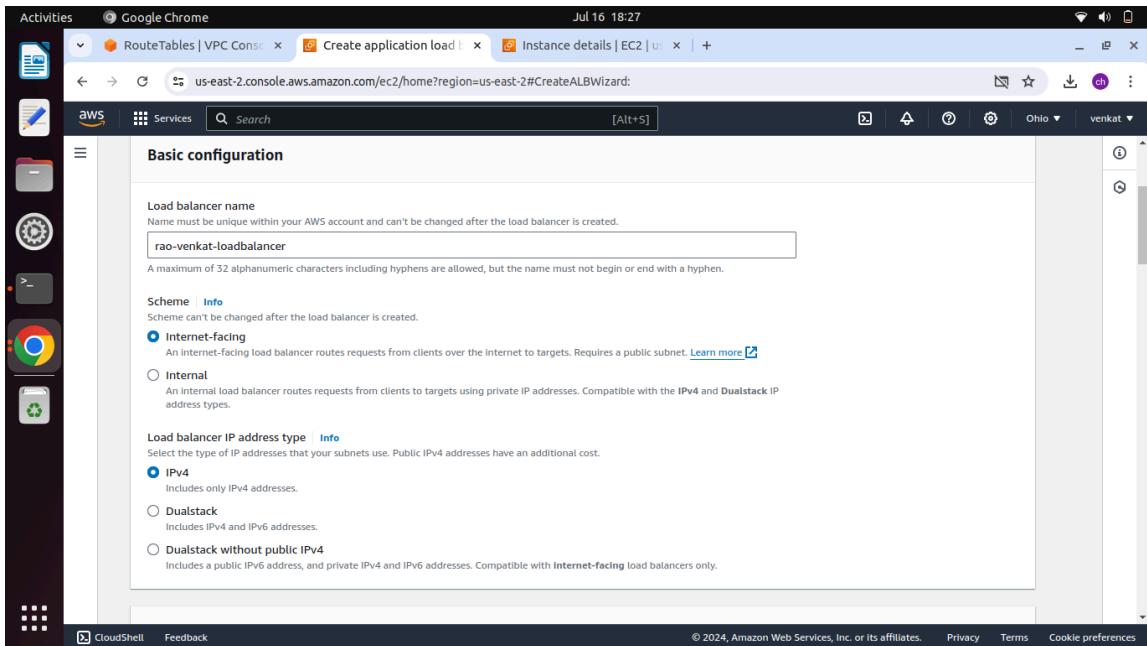
Load balancer IP address type [Info](#)
Select the type of IP addresses that your subnets use. Public IPv4 addresses have an additional cost.

IPv4
Includes only IPv4 addresses.

Dualstack
Includes IPv4 and IPv6 addresses.

Dualstack without public IPv4
Includes a public IPv6 address, and private IPv4 and IPv6 addresses. Compatible with [Internet-facing](#) load balancers only.

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



Activities Google Chrome Jul 16 18:30

RouteTables | VPC Cons... Create application load balancer Instance details | EC2 | ... +

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#CreateALBWizard:

VPC [Info](#)
Select the virtual private cloud (VPC) for your targets or you can [create a new VPC](#). Only VPCs with an internet gateway are enabled for selection. The selected VPC can't be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

baby-vpc
vpc-0084435fb463bbf77
IPv4 VPC CIDR: 10.0.0.0/16

Mappings [Info](#)
Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

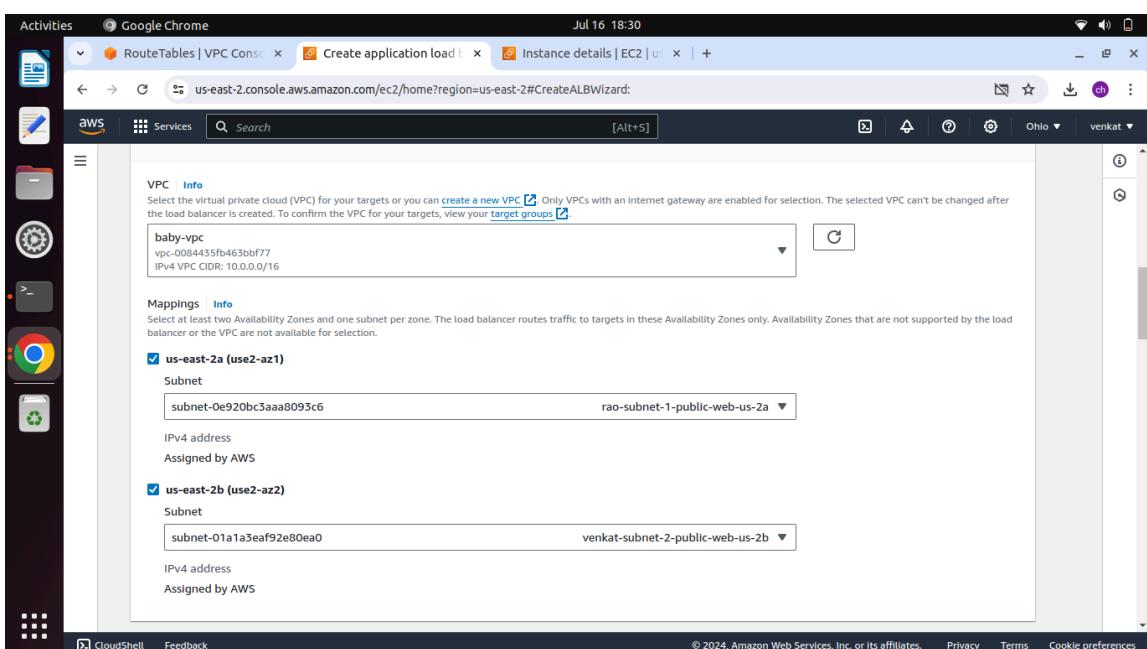
us-east-2a (use2-az1)
Subnet
 rao-subnet-1-public-web-us-2a

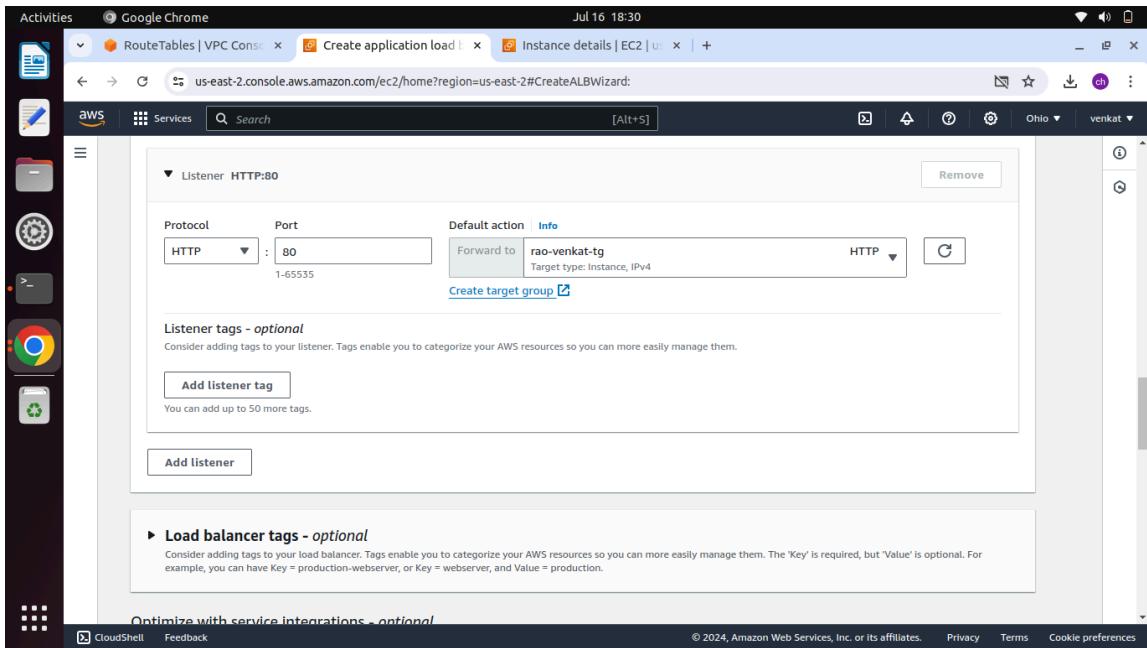
IPv4 address
Assigned by AWS

us-east-2b (use2-az2)
Subnet
 venkat-subnet-2-public-web-us-2b

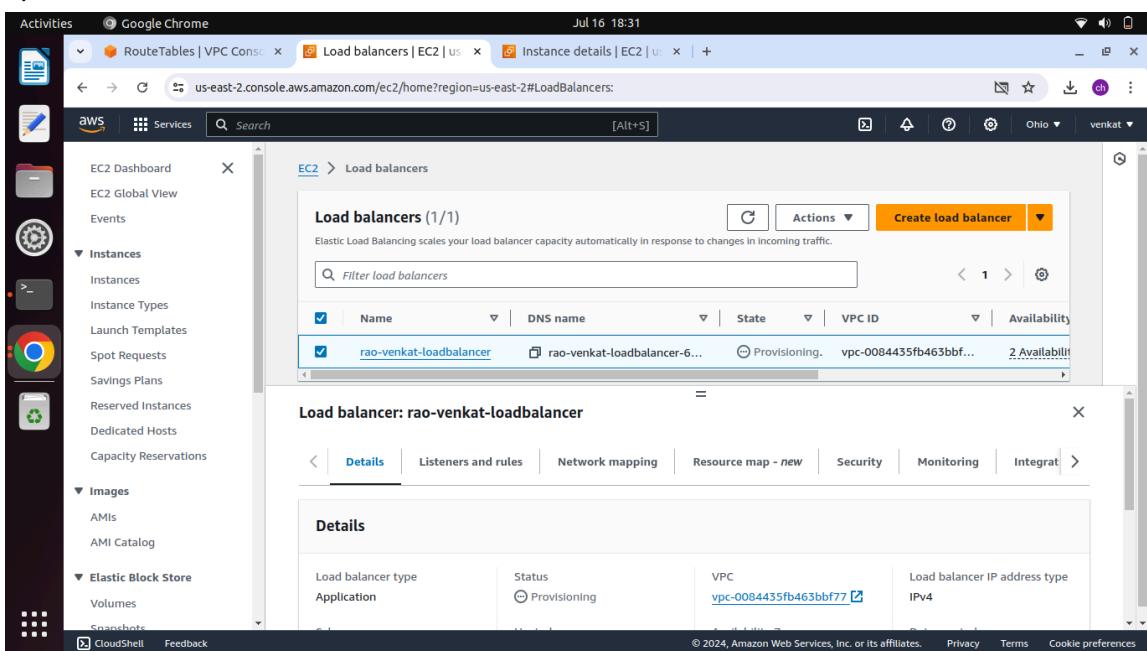
IPv4 address
Assigned by AWS

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





151) Click create load balancer



152) Now check load balancer working or not

153) Connect rao terminal install nginx

```

Activities Terminal Jul 16 18:35
root@rao:~-
challa@challa-HP-Laptop-15-da0xxx: ~/Downloads x root@rao:~ challa@challa-HP-Laptop-15-da0xxx: ~/Downloads x
total 4
-r----- 1 ec2-user ec2-user 1675 Jul 16 12:29 nirasm-app.pem
[ec2-user@rao ~]$ sudo -i
[root@rao ~]# yum install nginx
Last metadata expiration check: 1:14:44 ago on Tue Jul 16 11:47:59 2024.
Dependencies resolved.
=====
Package           Architecture      Version       Repository      Size
=====
Installing:
nginx            x86_64          1:1.24.0-1.amzn2023.0.2   amazonlinux    32 k
Installing dependencies:
generic-logos-httd 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          1:1.24.0-1.amzn2023.0.2   amazonlinux  586 k
nginx-filesystem  noarch          1:1.24.0-1.amzn2023.0.2   amazonlinux  9.1 k
nginx-nimetypes   noarch          2.1.49-3.amzn2023.0.3   amazonlinux  21 k
Transaction Summary
=====
Install 7 Packages
Total download size: 1.0 M
Installed size: 3.4 M
Is this ok [y/N]: y
Downloading Packages:
(1/7): generic-logos-httd-18.0.0-12.amzn2023.0.3.noarch.rpm           325 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): nginx-1.24.0-1.amzn2023.0.2.x86_64.rpm                         1.8 MB/s | 32 kB  00:00
(4/7): gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64.rpm                  3.6 MB/s | 308 kB  00:00
(5/7): nginx-core-1.24.0-1.amzn2023.0.2.x86_64.rpm                     17 MB/s | 586 kB  00:00
(6/7): nginx-filesystem-1.24.0-1.amzn2023.0.2.noarch.rpm                 423 kB/s | 9.1 kB  00:00
(7/7): nginx-nimetypes-2.1.49-3.amzn2023.0.3.noarch.rpm                  1.0 MB/s | 21 kB  00:00
Total                                         6.2 MB/s | 1.0 MB  00:00

```

154) Go to cd /usr/share/nginx/html

155) Rm -f index.html

156) Nano index.html and write inside this is rao sever

157) Systemctl restart nginx

```

Activities Terminal Jul 16 18:41
root@rao:/usr/share/nginx/html x root@rao:/usr/share/nginx/html x challa@challa-HP-Laptop-15-da0xxx: ~/Downloads x
drwxr-xr-x. 3 root root 112 Jul 16 13:02 html
drwxr-xr-x. 2 root root 6 Oct 13 2023 modules
[root@rao nginx]# cd html
[root@rao html]# rm -f index.html
[root@rao html]# nano index.html
[root@rao html]# history
 1 hostnamectl set-hostname rao
 2 exec bash
 3 nano nirasm-app.pem
 4 chmod 400 nirasm-app.pem
 5 ssh -t "nirasm-app.pem" ec2-user@10.0.40.100
 6 ll
 7 rm nirasm-app.pem
 8 vi nirasm-app.pem
 9 ll
10 vi nirasm-app.pem
11 yum install nginx
12 cd /usr
13 ll
14 cd
15 cd /var/
16 ll
17 cd
18 ll
19 cd /usr/
20 ll
21 cd share/
22 ll
23 cd nginx/
24 ll
25 cd html
26 rm -f index.html
27 nano index.html
28 history
[root@rao html]# systemctl restart nginx
[root@rao html]#

```

158) Add rao-sg to port number 80 http

The screenshot shows the AWS EC2 ModifyInboundSecurityGroupRules interface. The URL is us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#ModifyInboundSecurityGroupRules:securityGroupId=sg-03ce4c41c323cf591. The page displays two inbound rules:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-05c483dc3eef79e3	SSH	TCP	22	Cus... <input type="text" value="0.0.0.0/0"/>	<input type="button" value="Delete"/>
-	HTTP	TCP	80	An... <input type="text" value="0.0.0.0/0"/>	<input type="button" value="Delete"/>

A warning message at the bottom states: "⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your Instance. We recommend setting security group rules to allow access from known IP addresses only." Buttons for "Cancel", "Preview changes", and "Save rules" are visible.

159) Now take rao server public ip paste google

The screenshot shows a web browser window with the URL 18.225.55.140. The page content is "this is rao server". The browser toolbar shows tabs for RouteTables | VPC Consol, SecurityGroup | EC2 | us-east-2, Instances | EC2 | us-east-2, and 18.225.55.140. The status bar indicates the date and time as Jul 16 18:48.

160) Now install in same as nginx in venkat server

161) Take load balancer dns paste google refresh change names
venkat rao servers add venkat sg also http

The screenshot shows the AWS CloudWatch Metrics interface. A metric named "rao-venkat-loadbalancer" is displayed with a value of 1. The metric has a timestamp of Jul 16 19:01. The interface includes a navigation bar with tabs for RouteTables, SecurityGroup, Load balancers, and CloudWatch Metrics.

Activities Google Chrome Jul 16 19:01

RouteTables | VPC C... SecurityGroup | EC2 | Load balancers | EC2 | rao-venkat-loadbalan... rao-venkat-loadbalan...

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#LoadBalancers

aws Services Search [Alt+5]

EC2 > Load balancers

Load balancers (1/1)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Filter load balancers

Name DNS name State VPC ID Availability

rao-venkat-loadbalancer rao-venkat-loadbalancer-6... Active vpc-0084435fb463bbf... 2 Available

Load balancer: rao-venkat-loadbalancer

east-2b (use2-az2)
subnet-0e920bc3aaa8093c6 us- az1

DNS name copied

Load balancer ARN arn:aws:elasticloadbalancing:us-east-2:058264351590:loadbalancer/app/rao-venkat-loadbalancer/6d939ab30e12b675

rao-venkat-loadbalancer-627940136.us-east-2.elb.amazonaws.com (A Record)

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

Activities Google Chrome Jul 16 19:00

RouteTables | VPC C... SecurityGroup | EC2 | Load balancers | EC2 | rao-venkat-loadbalan... rao-venkat-loadbalan...

Not secure rao-venkat-loadbalancer-627940136.us-east-2.elb.amazonaws.com

this venkat server

162) Create ami using rao server

Activities Google Chrome Jul 16 19:06

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#InstancesinstanceState=running

aws Services Search [Alt+S]

EC2 Dashboard EC2 Global View Events Instances Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Capacity Reservations Images AMIs AMI Catalog Elastic Block Store Volumes Snapshots CloudShell Feedback

Instances (1/4) Info Actions ▾ Launch instances ▾

Instance state = running X Clear filters All

Name	Instance ID	Instance state	Instance type
nirasm-app	i-0c3ffbd1f82ad84d2	Running	t2.micro
venkat-web	i-0c64048a903aab6a0	Running	t2.micro
potti-app	i-0b8f5606fb0d1614c	Running	t2.micro
rao-web	i-00004c997d475d087	Running	t2.micro

Create image Create template from instance Launch more like this

i-00004c997d475d087 (rao-web)

Details Status and alarms Monitoring Security Networking Storage Tags

Instance summary Info

Instance ID	I-00004c997d475d087 (rao-web)	Public IPv4 address	18.225.55.140 open address	Private IPv4 addresses	10.0.0.26
IPv6 address	-	Instance state	Running	Public IPv4 DNS	-
Hostname type		Private IP DNS name (IPv4 only)			

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

Activities Google Chrome Jul 16 19:06

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#CreateImage:instanceId=i-00004c997d475d087

aws Services Search [Alt+S]

configuration of an existing instance.

Instance ID I-00004c997d475d087 (rao-web)

Image name rao-image

Maximum 127 characters. Can't be modified after creation.

Image description - optional allow

Maximum 255 characters

No reboot Enable

Instance volumes

Storage type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/...	Create new sna...	8	EBS General Pu...	3000		<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable

CloudShell Feedback

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

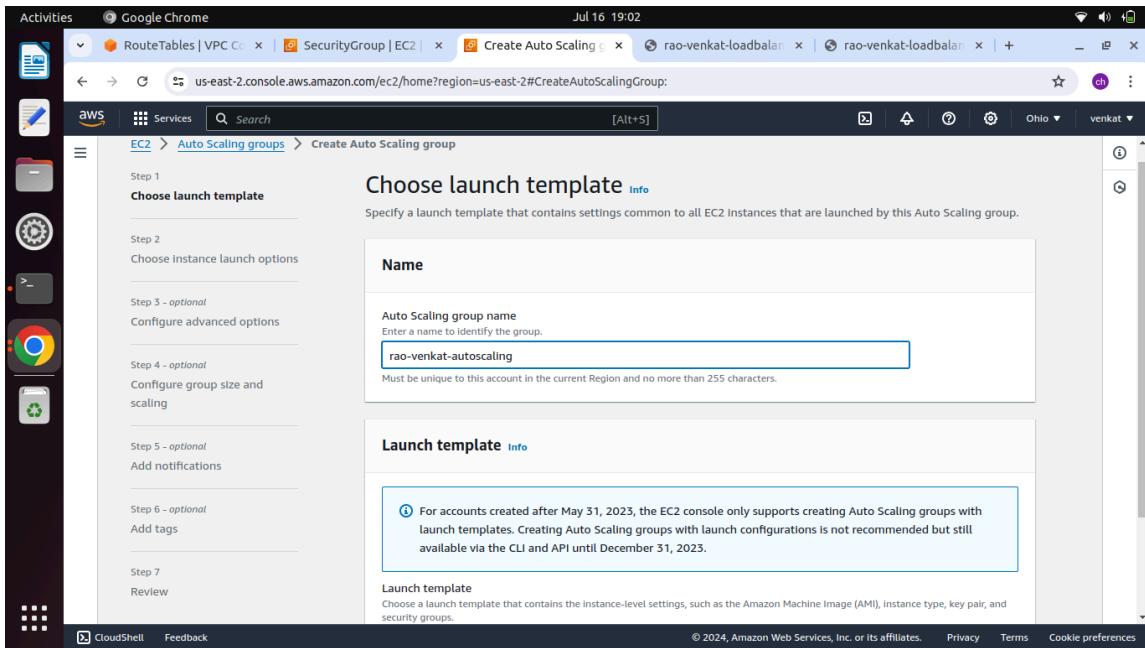
The screenshot displays two browser tabs within the AWS Cloud9 IDE:

- Top Tab:** Shows the 'Create Image' dialog for an AMI. It includes fields for Storage type (EBS), Volume size (8), IOPS (3000), Throughput (3000), and Encryption (checked). A note states: "During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes." There are options for 'Tag image and snapshots together' or 'Tag image and snapshots separately'. A 'Create image' button is at the bottom.
- Bottom Tab:** Shows the 'Amazon Machine Images (AMIs)' list. The table has columns: Name, AMI ID, and Source. One row is selected, showing:

Name	AMI ID	Source
rao-image	ami-076867e4e775f4719	058264331590/rao-image

163) Go to auto scaling

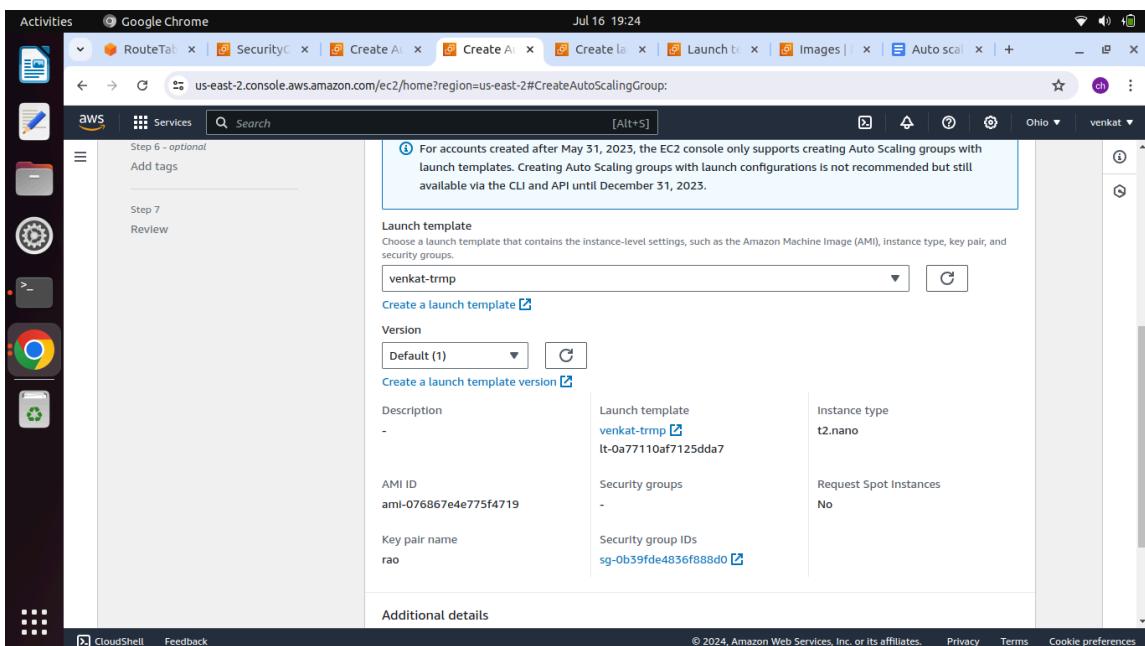
164) Create auto scaling group



165) Click create launch template

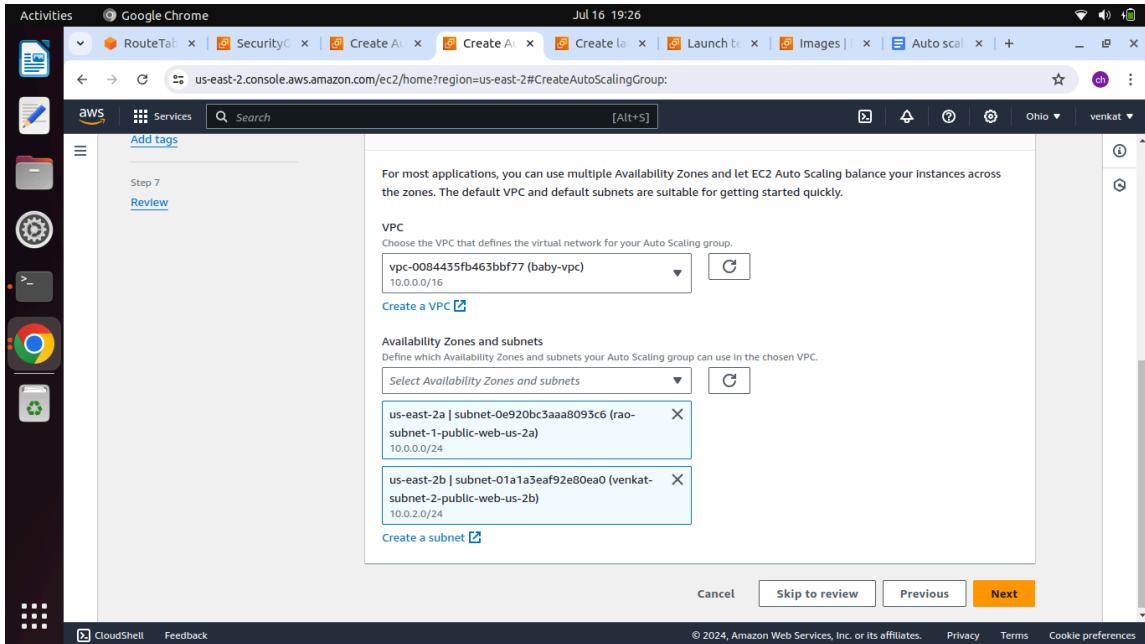
166) Set name venkat-temp click template here select sg=rao-sg and select instance type= t2.micro and select keypair=rao.pem and don't select subnet click create template

167) Now select reo-venkat-temp



168) Select vpc=baby-vpc

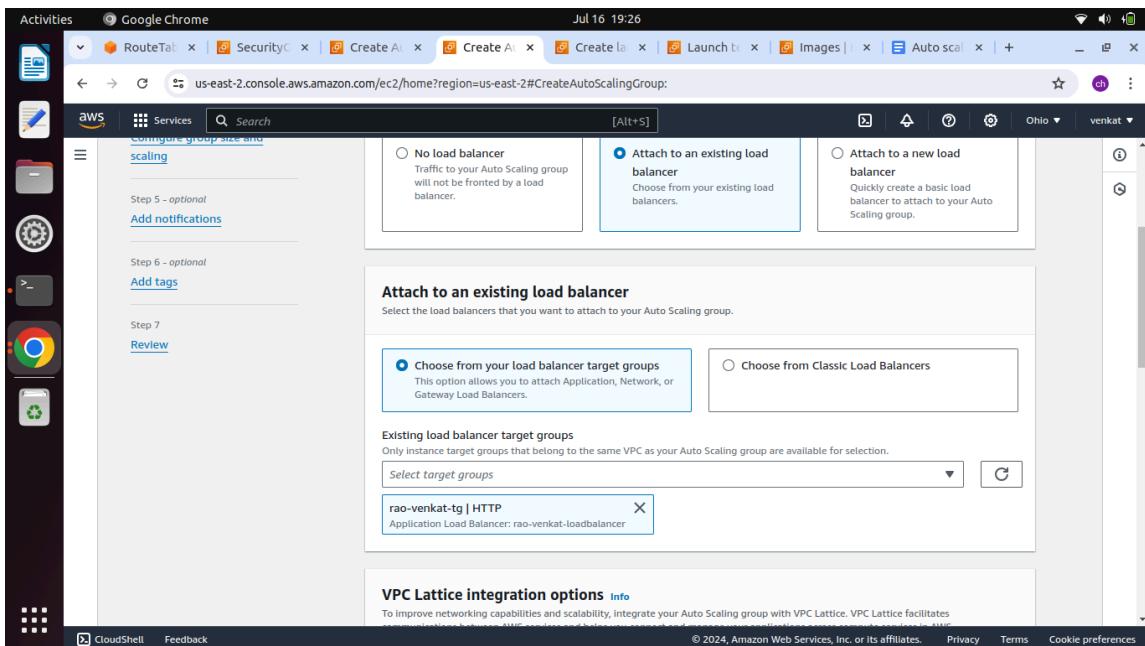
169) Availability zones= rao-subnet-public-1-web-us-2a and venkat-subnet-2-public-web-us-2b and click next



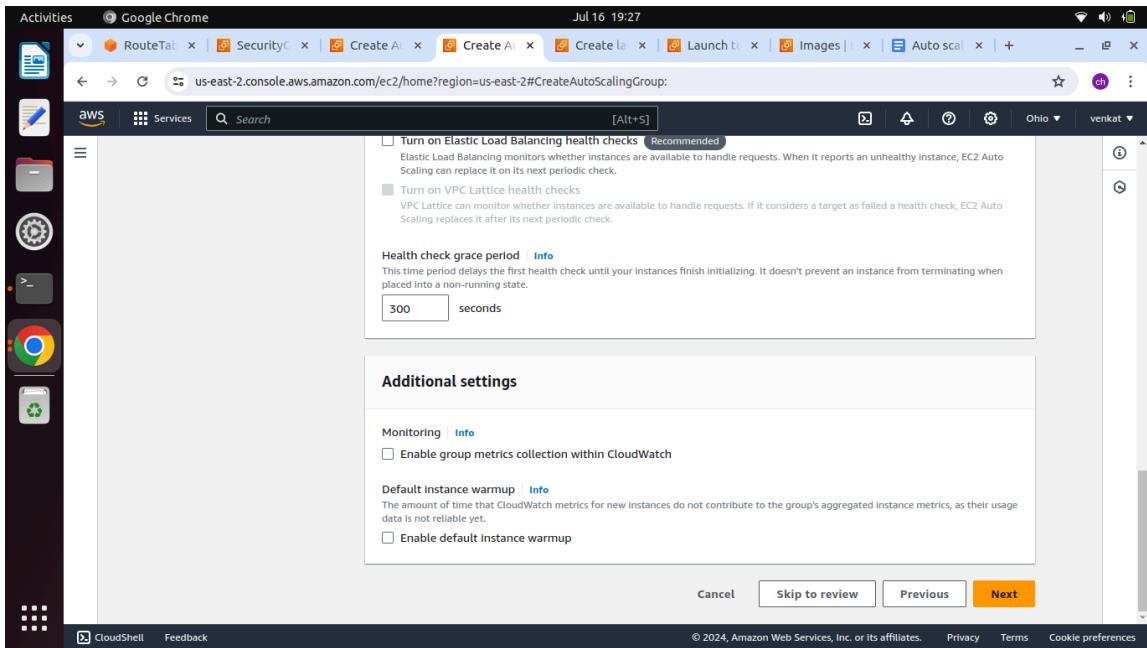
170) Select attach to an existing load balancer

171) Select choose from your load balancer target groups

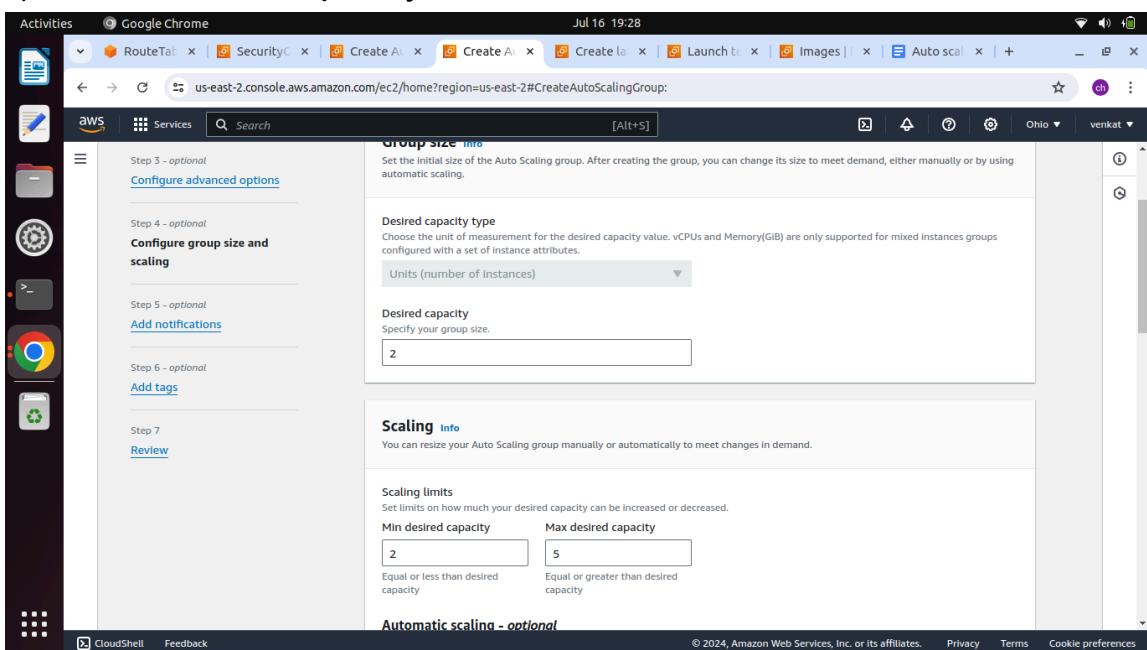
172) Target group= rao-venkat-tg



173) Click next



- 174) Desired capacity=2
- 175) Min desired capacity=2
- 176) Max desired capacity=5



- 177) Auto scaling option= target tracking scaling policy
- 178) Target value= 50
- 179) Instance warmp=100
- 180) Click next

Activities Google Chrome Jul 16 19:29

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#CreateAutoScalingGroup:

Automatic scaling - optional

Choose whether to use a target tracking policy [Info](#)
You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

No scaling policies
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

Target tracking scaling policy
Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

Scaling policy name: Target Tracking Policy

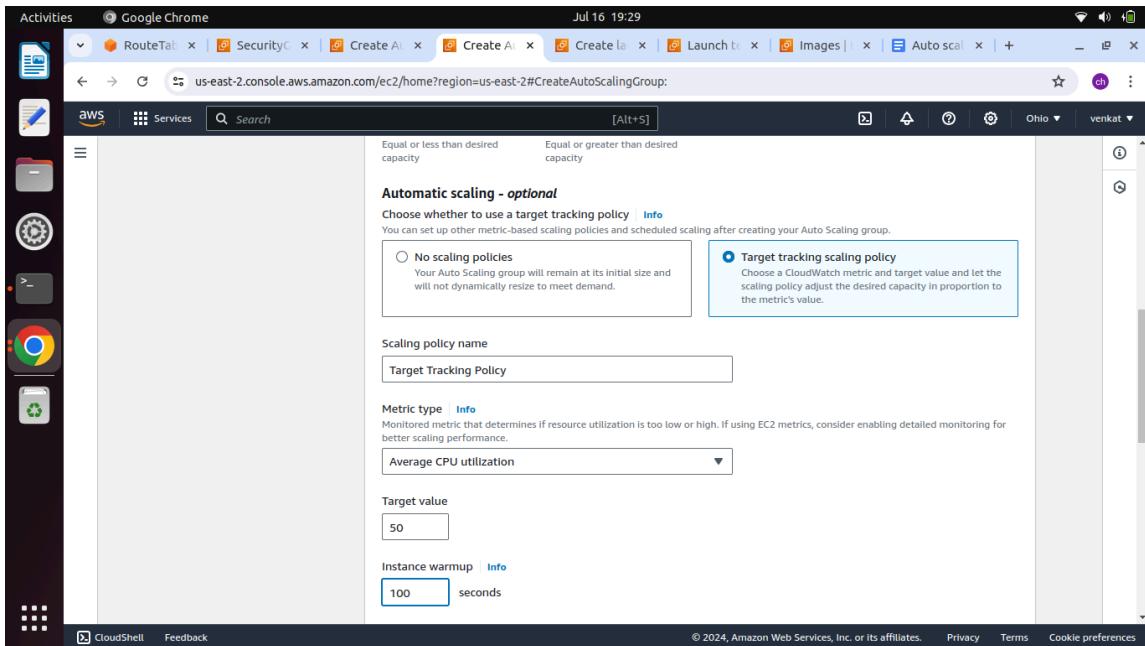
Metric type [Info](#)
Monitored metric that determines if resource utilization is too low or high. If using EC2 metrics, consider enabling detailed monitoring for better scaling performance.

Average CPU utilization

Target value: 50

Instance warmup [Info](#)
100 seconds

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



Activities Google Chrome Jul 16 19:29

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#CreateAutoScalingGroup:

Choose a replacement behavior depending on your availability requirements

Mixed behavior
For rebalancing events, new instances will launch before terminating others. For all other events, instances terminate and launch at the same time.

Prioritize availability
Launch before terminating
Launch new instances and wait for them to be ready before terminating others. This allows you to go above your desired capacity by a given percentage and may temporarily increase costs.

Control costs
Terminate and launch
Terminate and launch instances at the same time. This allows you to go below your desired capacity by a given percentage and may temporarily reduce availability.

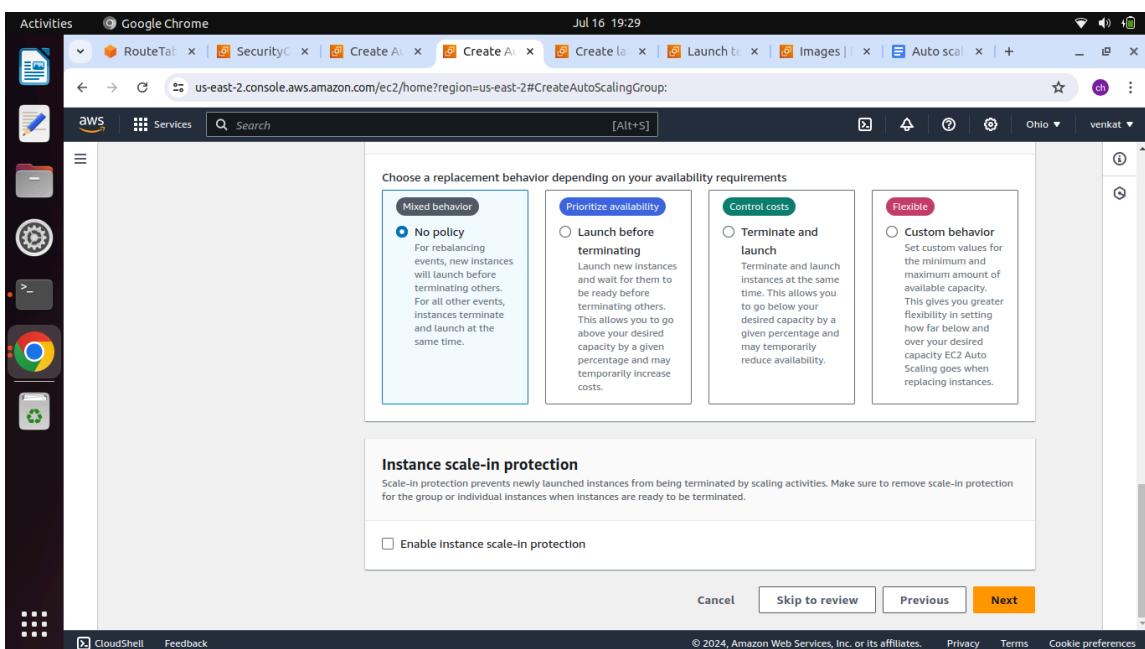
Flexible
Custom behavior
Set custom values for the minimum and maximum amount of available capacity. This gives you greater flexibility in setting how far below and over your desired capacity EC2 Auto Scaling goes when replacing instances.

Instance scale-in protection
Scale-In protection prevents newly launched instances from being terminated by scaling activities. Make sure to remove scale-in protection for the group or individual instances when instances are ready to be terminated.

Enable Instance scale-in protection

Cancel Skip to review Previous Next

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



Activities Google Chrome Jul 16 19:29

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#CreateAutoScalingGroup:

aws Services Search [Alt+S]

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1 Choose launch template

Step 2 Choose instance launch options

Step 3 - optional Configure advanced options

Step 4 - optional Configure group size and scaling

Step 5 - optional Add notifications

Step 6 - optional Add tags

Step 7 Review

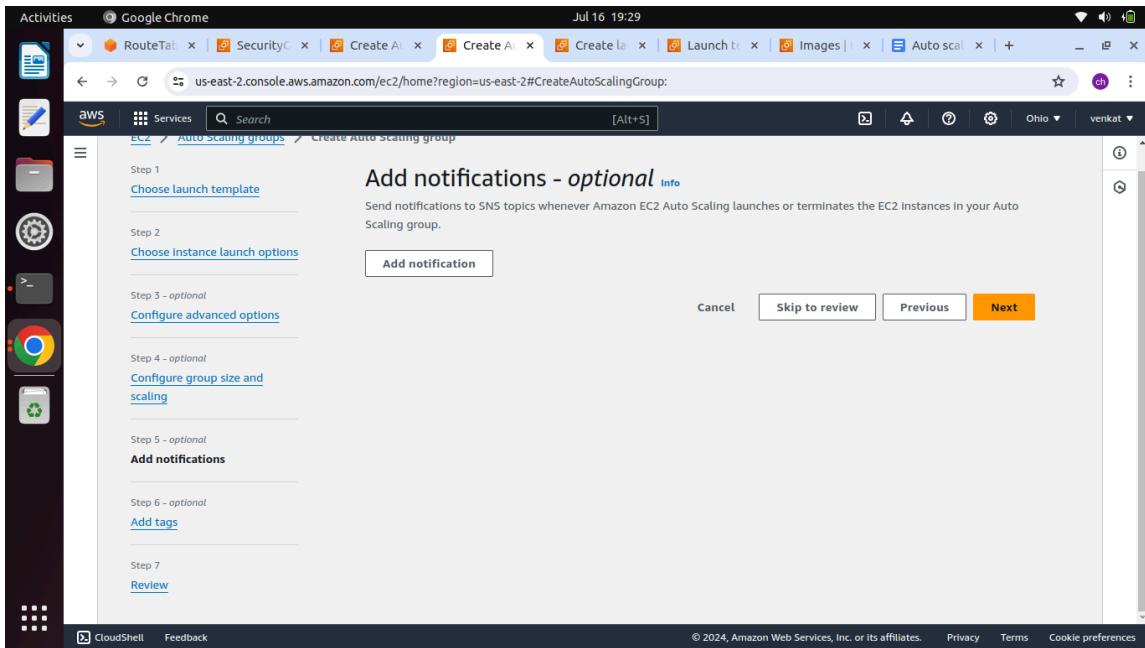
Add notifications - optional Info

Send notifications to SNS topics whenever Amazon EC2 Auto Scaling launches or terminates the EC2 instances in your Auto Scaling group.

Add notification

Cancel Skip to review Previous Next

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



Activities Google Chrome Jul 16 19:29

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#CreateAutoScalingGroup:

aws Services Search [Alt+S]

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1 Choose launch template

Step 2 Choose instance launch options

Step 3 - optional Configure advanced options

Step 4 - optional Configure group size and scaling

Step 5 - optional Add notifications

Step 6 - optional Add tags

Step 7 Review

Add tags - optional Info

Add tags to help you search, filter, and track your Auto Scaling group across AWS. You can also choose to automatically add these tags to instances when they are launched.

You can optionally choose to add tags to instances (and their attached EBS volumes) by specifying tags in your launch template. We recommend caution, however, because the tag values for instances from your launch template will be overridden if there are any duplicate keys specified for the Auto Scaling group.

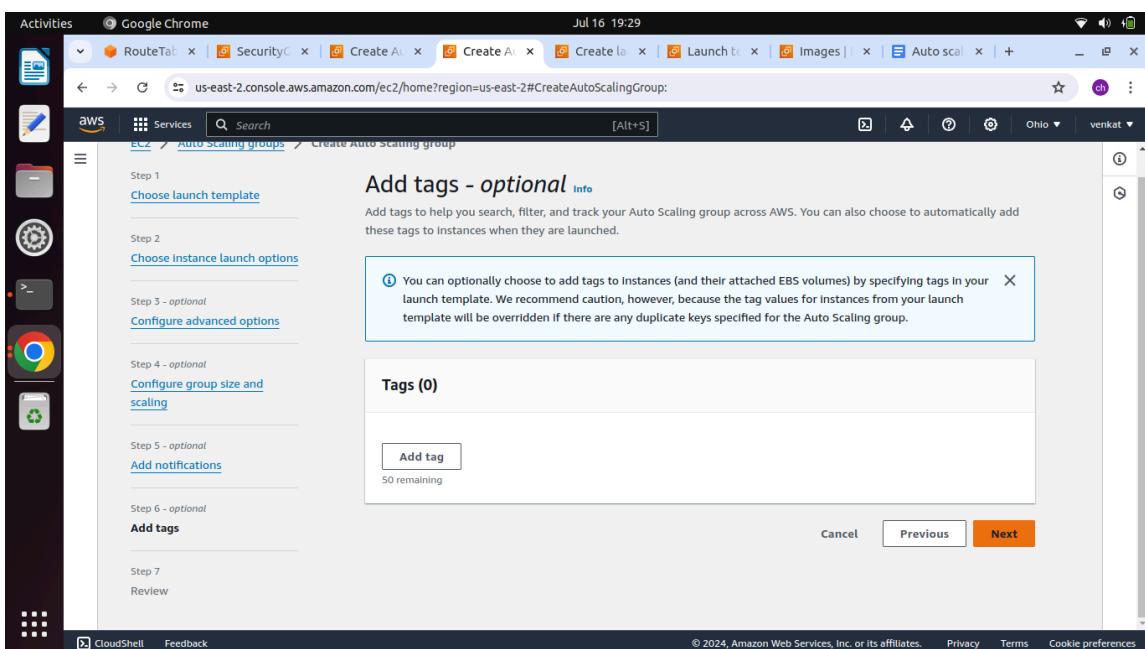
Tags (0)

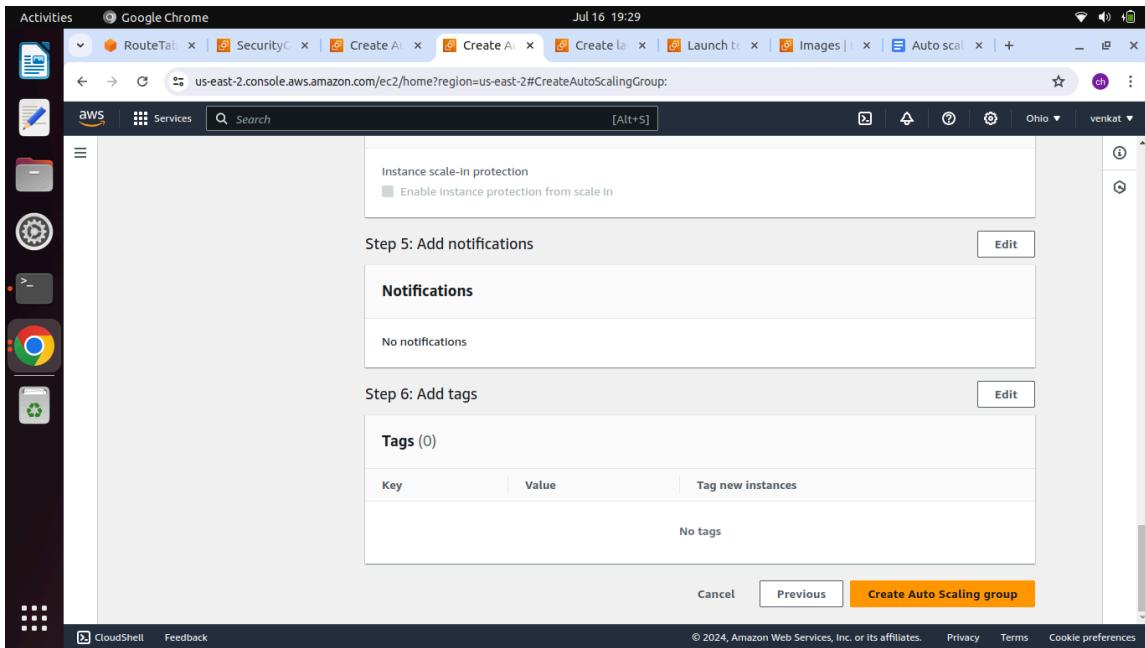
Add tag

50 remaining

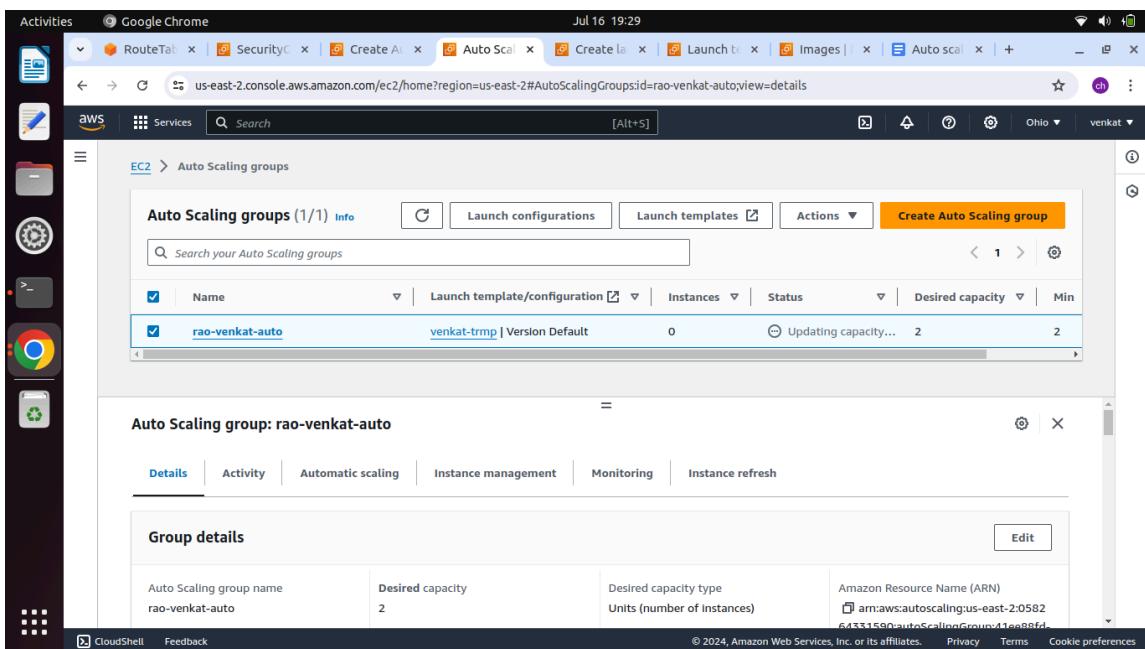
Cancel Previous Next

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





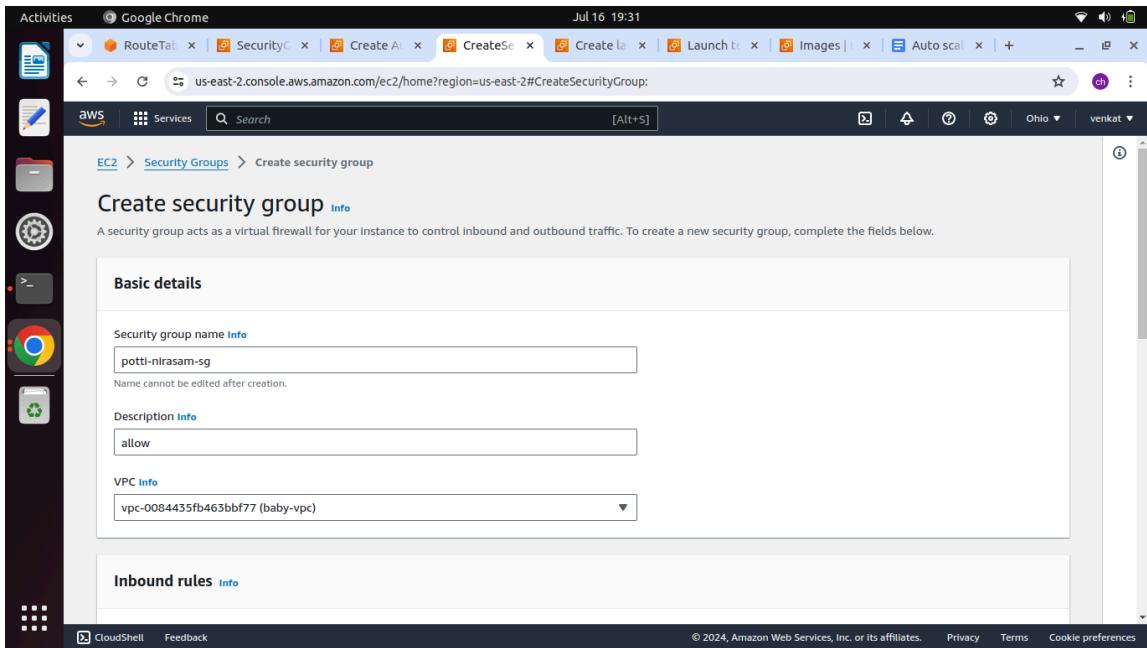
181) Click create auto scaling group



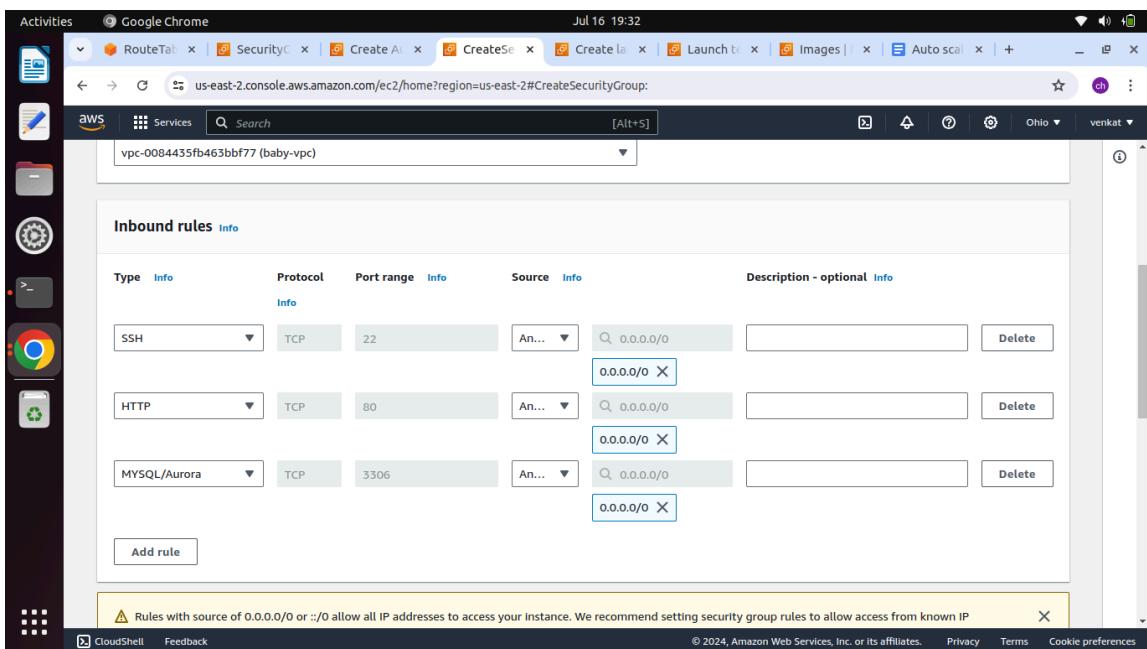
182) Now go to ec2 instances check instances creating automatically

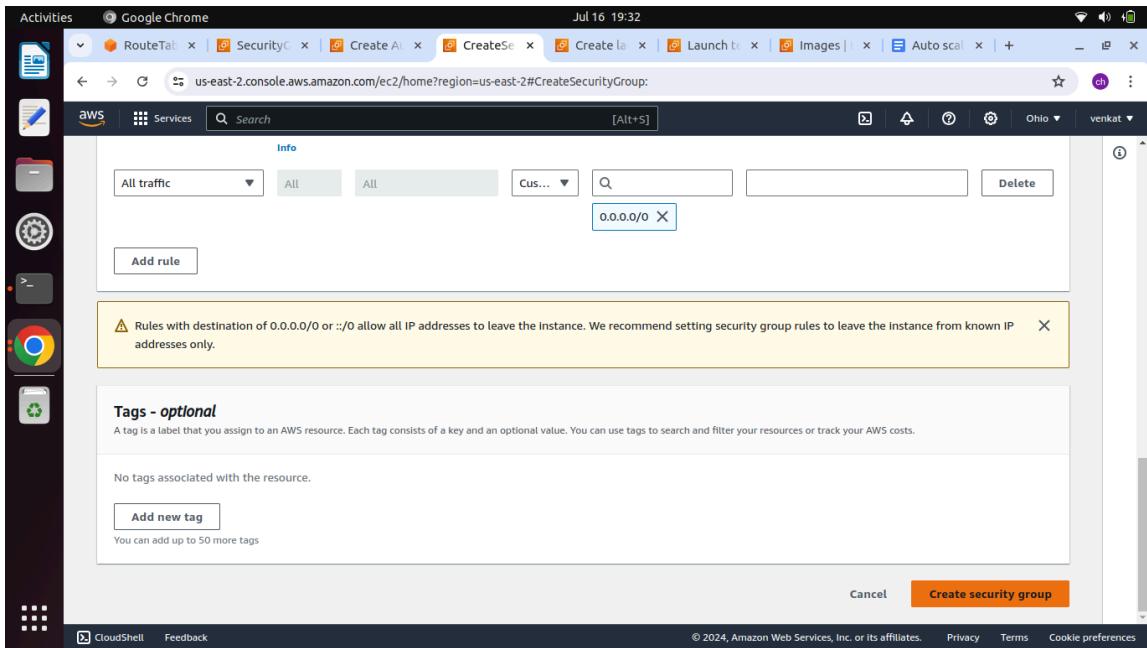
183) Now go to security group crete sg name= potti-nirasam-sg

184) vpc=baby-vpc



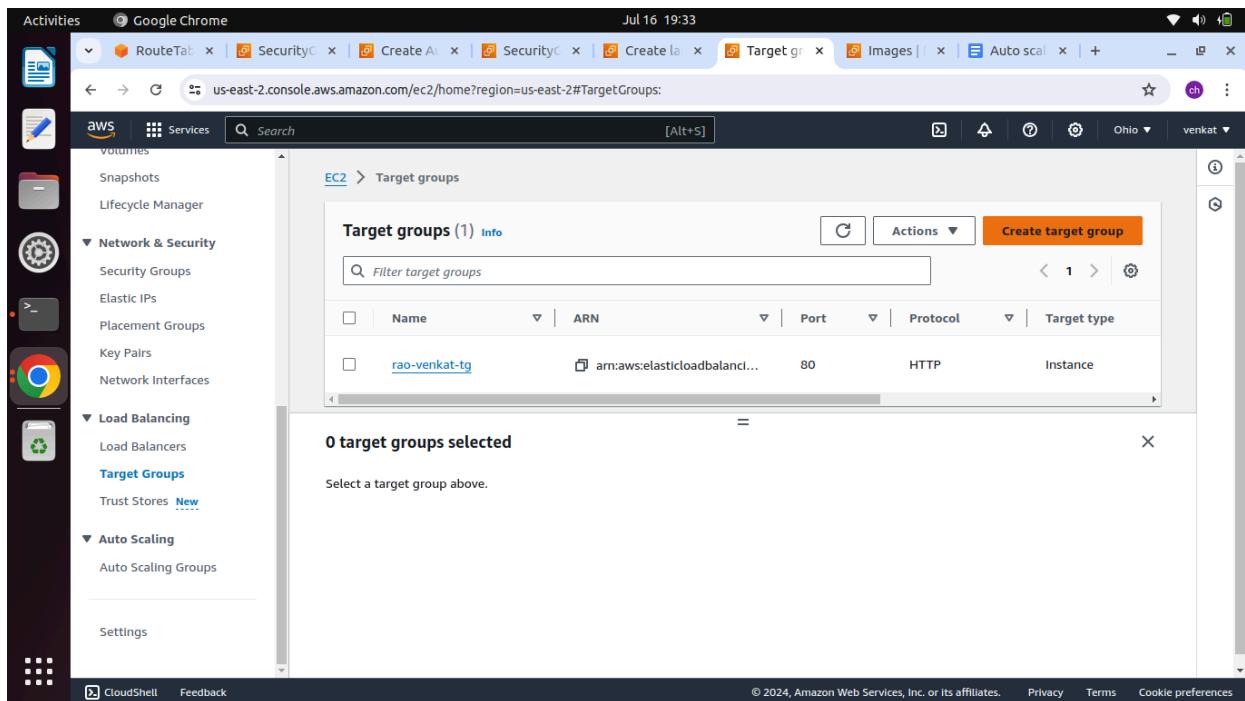
185) Inbound rules add http ssh and mysql/aurora rules



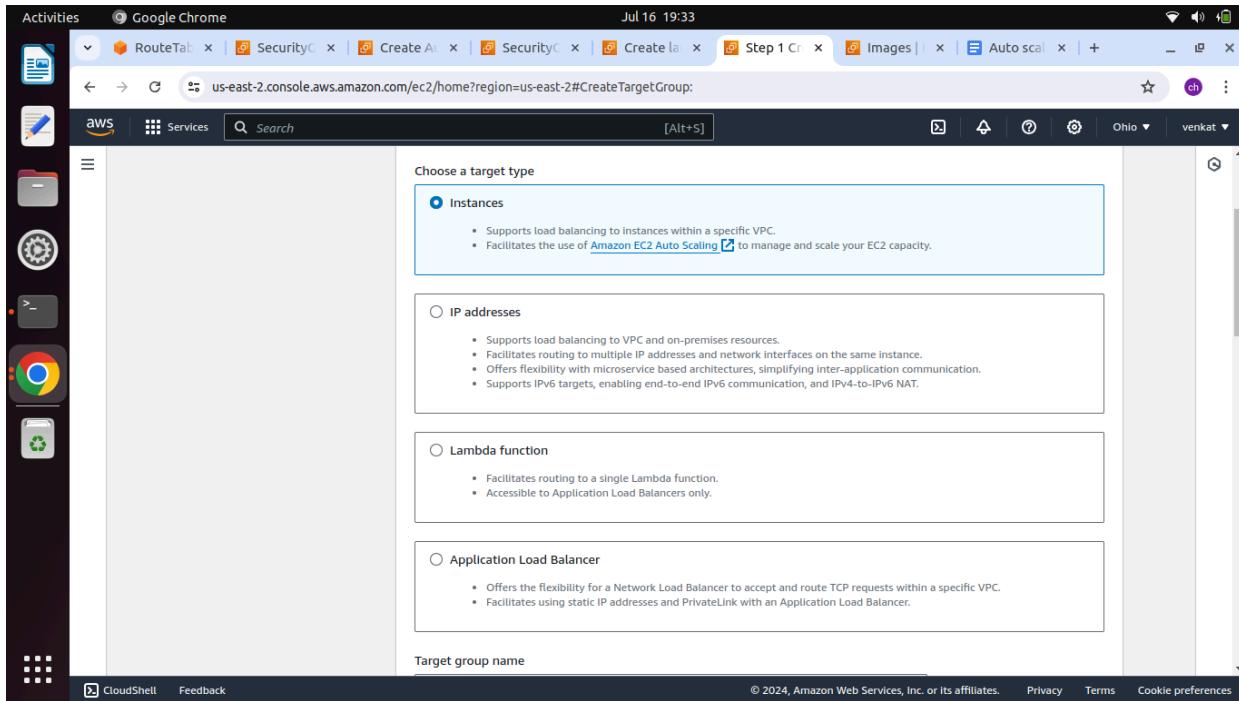


186) Click create security group

187) Go to target group click crete target group



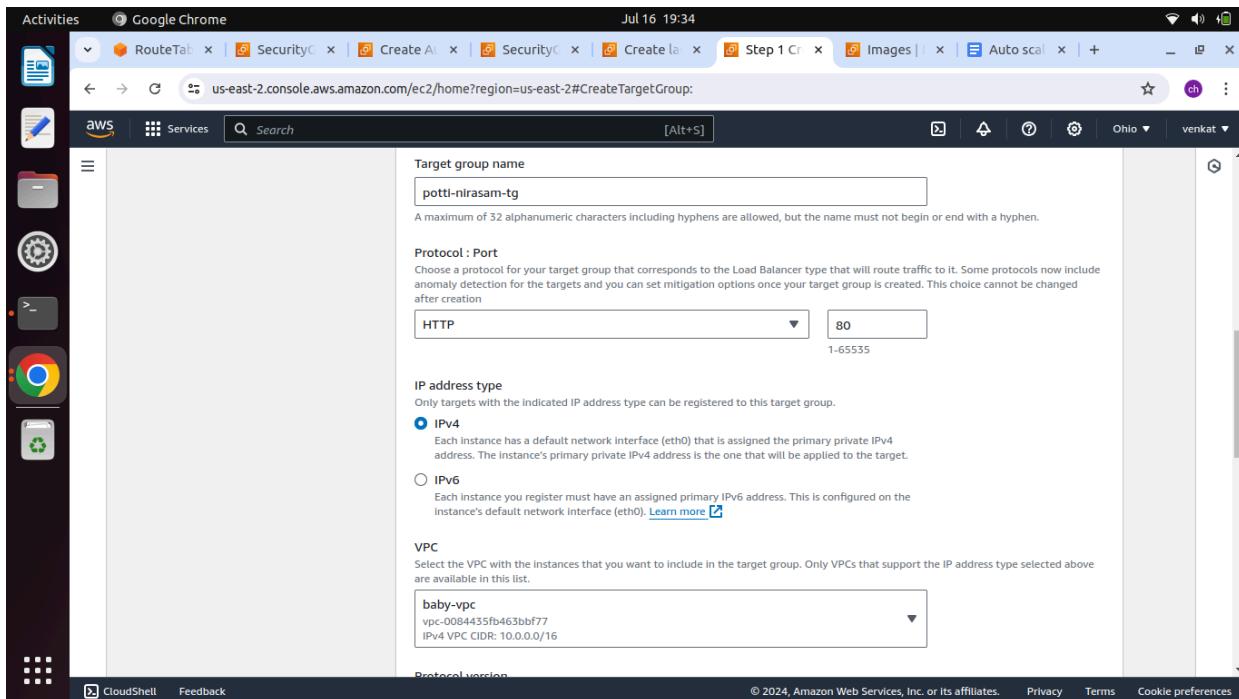
188) Select instances



189) Target group name= potti-nirasam-tg

190) Select ipv4

191) vpc=baby-vpc



192) Select nirasam-app and potti-app instance and click include as pending

Activities Google Chrome Jul 16 19:34

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#CreateTargetGroup:

AWS Services Search [Alt+S]

Filter instances

Instance ID	Name	State	Security groups
I-09be1c0ab5f4a11e8		Running	rao-venkat
I-0c3ffbd1f82ad84d2	nirasam-app	Running	nirasam-sg
I-0b8f5606fb0d01614c	potti-app	Running	potti-sg
I-0c64048a903aab6a0	venkat-web	Running	venkat-sg
I-00004c997d475d087	rao-web	Running	rao-sg

2 selected

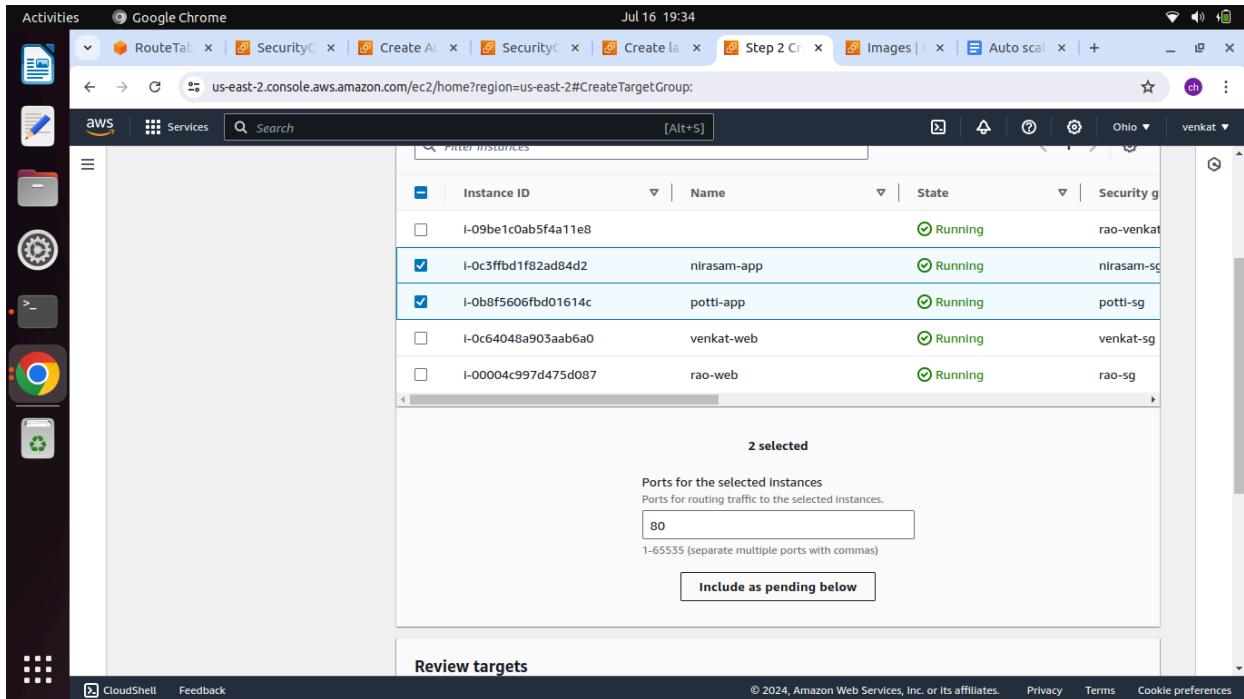
Ports for the selected instances
Ports for routing traffic to the selected instances.

80
1-65535 (separate multiple ports with commas)

Include as pending below

Review targets

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



Activities Google Chrome Jul 16 19:34

us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#CreateTargetGroup:

AWS Services Search [Alt+S]

1-65535 (separate multiple ports with commas)

Include as pending below

2 selections are now pending below. Include more or register targets when ready.

Review targets

Targets (2)

Remove all pending

Show only pending

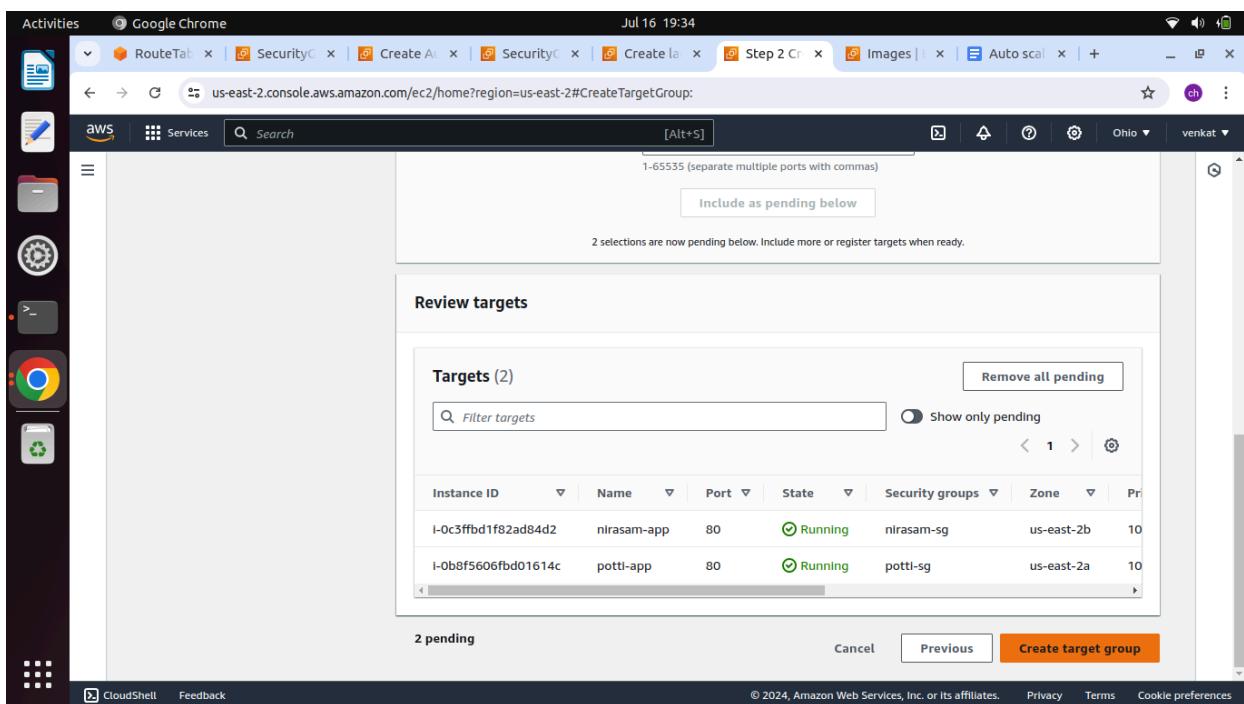
Filter targets

Instance ID	Name	Port	State	Security groups	Zone	Pr
I-0c3ffbd1f82ad84d2	nirasam-app	80	Running	nirasam-sg	us-east-2b	10
I-0b8f5606fb0d01614c	potti-app	80	Running	potti-sg	us-east-2a	10

2 pending

Cancel Previous Create target group

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



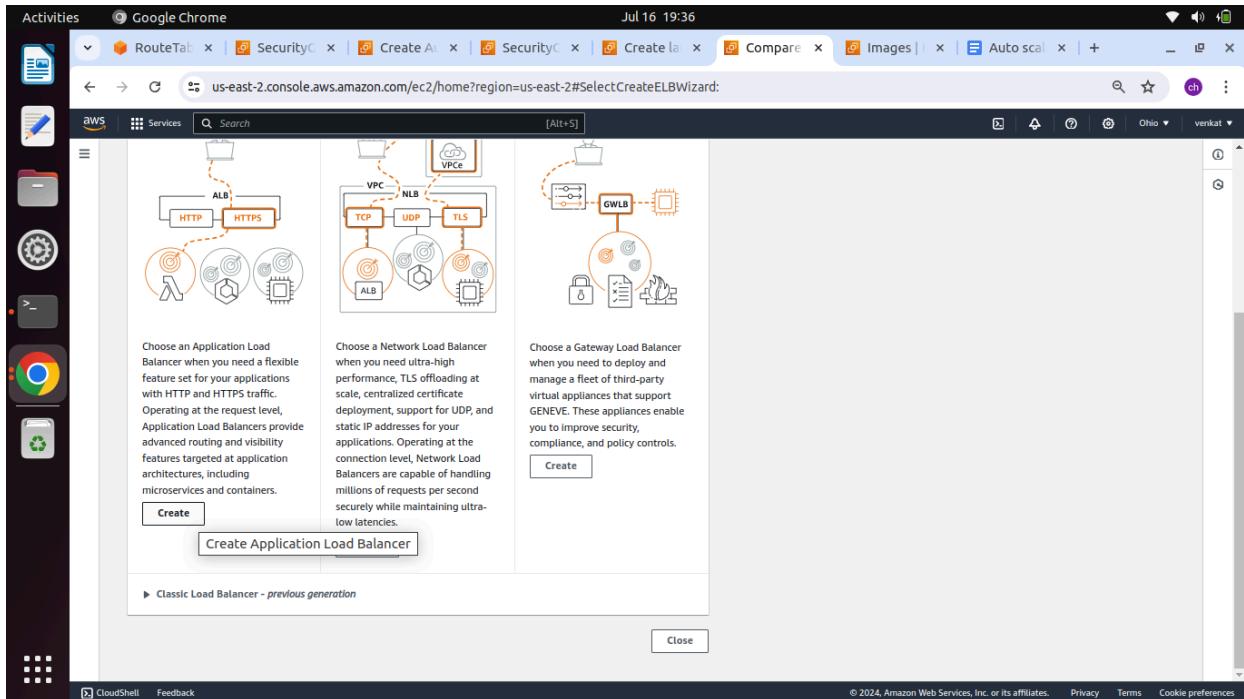
193) Click create target group

The screenshot shows the AWS EC2 Target groups page. The left sidebar lists various services like EC2 Dashboard, Instances, Images, and Elastic Block Store. The main content area shows a table titled 'Target groups (2)'. One row is visible for 'potti-nirasmam-tg' with ARN 'arn:aws:elasticloadbalancing:us-east-2:058264331590:targetgroup/potti-nirasmam-tg', Port 80, Protocol HTTP, and Target type Instance. A message below the table says '0 target groups selected' and 'Select a target group above.'

194) Go to load balancer click crete load balancer

The screenshot shows the AWS EC2 Load balancers page. The left sidebar includes Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (Load Balancers, Target Groups, Trust Stores), and Auto Scaling (Auto Scaling Groups). The main content area shows a table titled 'Load balancers (1)'. One row is visible for 'rao-venkat-loadbalancer' with DNS name 'rao-venkat-loadbalancer-6...', State Active, VPC ID 'vpc-0084435fb463bbff...', and Availability 2. A message below the table says '0 load balancers selected' and 'Select a load balancer above.'

195) Select application load balancer click create

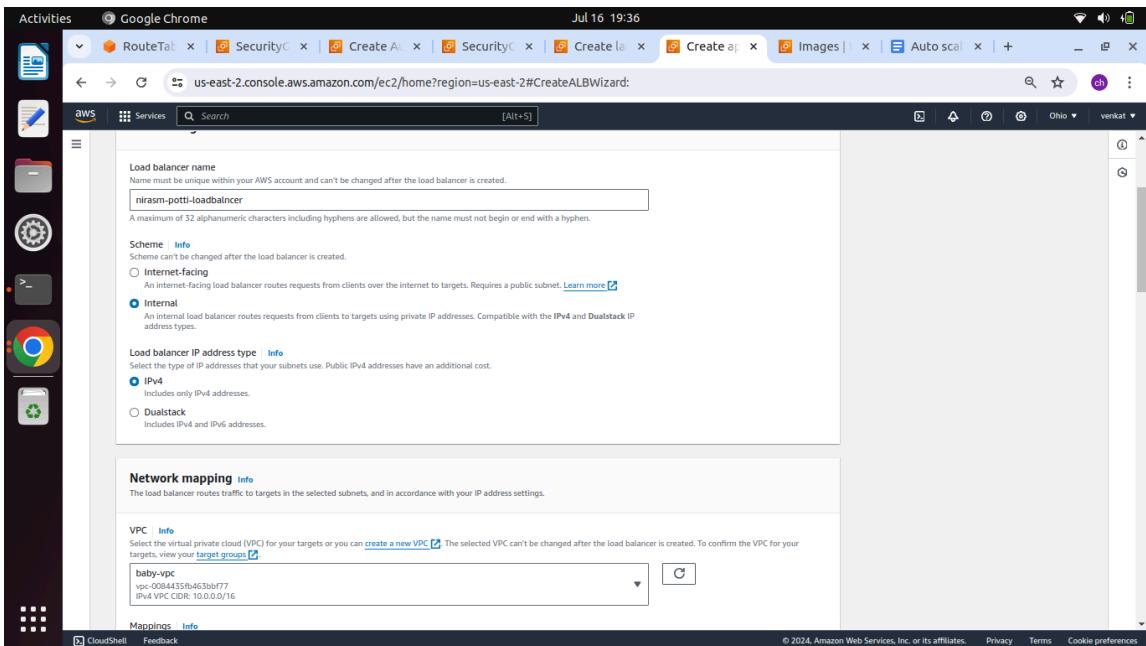


196) Load balancer name= nirasm-potti-load-balancer

197) scheme=internal

198) Ipv4

199) vpc=baby-vpc



200) Select both availability zones 2a and 2b subnets potti and nirasm select

201) sg= potti-nirasam-sg

The screenshot shows the AWS Lambda console with the URL <https://us-east-2.console.aws.amazon.com/lambda/home?region=us-east-2#CreateLambda:>. The page is titled "Create Lambda Function". The "Function name" field contains "potti-nirasam-tg". The "Runtime" dropdown is set to "Node.js 18.x". Under "Handler", "Index file" is selected. The "Code" section shows a "Create new" button. The "VPC" section has "Add VPC" checked. The "Security groups" dropdown lists "potti-nirasam-sg". The "Listeners and routing" section is collapsed.

202) Select target group=potti-nirasam-sg

The screenshot shows the AWS Lambda console with the URL <https://us-east-2.console.aws.amazon.com/lambda/home?region=us-east-2#CreateLambda:>. The page is titled "Create Lambda Function". The "Function name" field contains "potti-nirasam-tg". The "Runtime" dropdown is set to "Node.js 18.x". Under "Handler", "Index file" is selected. The "Code" section shows a "Create new" button. The "VPC" section has "Add VPC" checked. The "Security groups" dropdown lists "potti-nirasam-sg". The "Listeners and routing" section is expanded, showing a single listener for port 80. The "Protocol" is set to "HTTP" and the "Port" is "80". The "Default action" dropdown is set to "Forward to" and the "Target" is "potti-nirasam-tg". The "Listener tags - optional" section is collapsed.

203) Click create load balancer

204) Go to ec2 instance select potti-app click actions click image and template click create image

205) Image name= potti-image

206) No reboot= enable click crete image

Create image Info

An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.

Instance ID
i-0b8f5606fb0d01614c (potti-app)

Image name
potti-image

Maximum 127 characters. Can't be modified after creation.

Image description - optional
allow

Maximum 255 characters

No reboot
 Enable

Instance volumes

Storage type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev/sda1	Create new snapshot	8	EBS General Purpose (SSD)	3000	1000 MiB/s	<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable

Add volume

Amazon Machine Images (AMIs) (2) Info

Name	AMI name	AMI ID	Source	Owner	Visibility
potti-image	ami-03644615bf698ca10	058264331590/potti-image	058264331590	Private	
rao-image	ami-076867e4c775f4719	058264331590/rao-image	058264331590	Private	

Select an AMI

207) Go to auto scaling group click create auto scaling group

The screenshot shows the AWS Management Console with the URL us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#AutoScalingGroups. The left sidebar is collapsed. The main content area shows the 'Auto Scaling groups' section. A table lists one item:

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max
rao-venkat-auto	venkat-trmp Version Default	2	-	2	2	5

Below the table, it says '0 Auto Scaling groups selected'. At the bottom right, there are links for 'CloudShell', 'Feedback', and copyright information.

208) name= potti-nirasam-ag

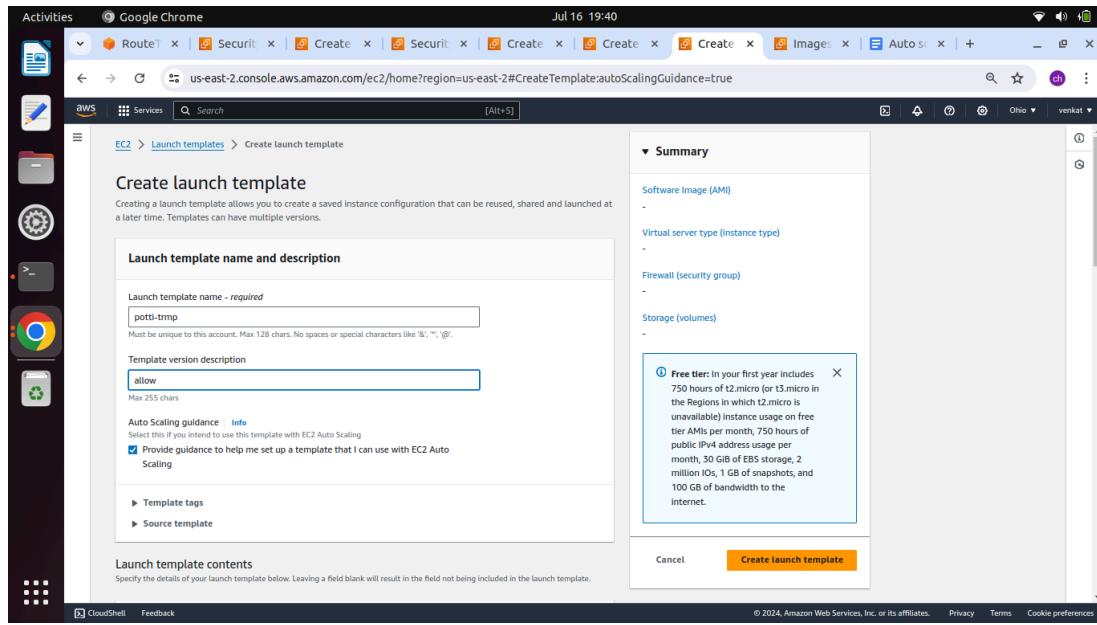
209) Click create launch template

The screenshot shows the 'Create Auto Scaling group' wizard, Step 1: Choose launch template. The left sidebar shows steps 1 through 7. Step 1 is active. The main content area has two sections:

- Name**: A text input field containing 'potti-nirasam-ag'.
- Launch template**: A dropdown menu with the placeholder 'Select a launch template' and a 'Create a launch template' link.

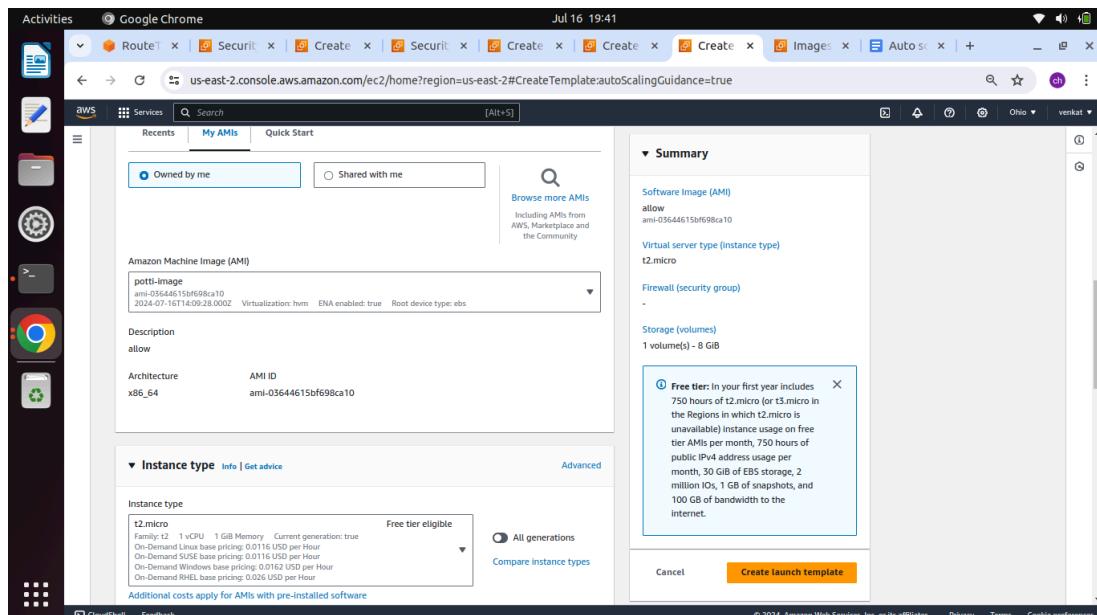
A note in the 'Launch template' section states: "For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023." At the bottom right, there are 'Cancel' and 'Next' buttons.

210) Launch template name= potti-temp



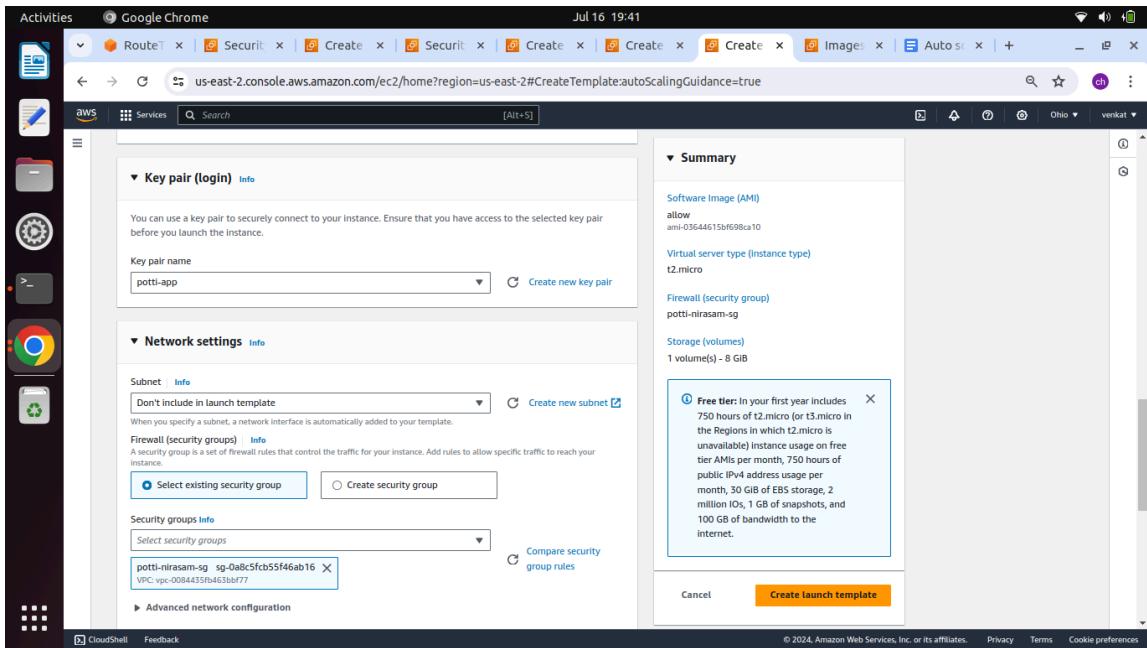
211)

212) Select my ami select potti-image and select instance t2.micro

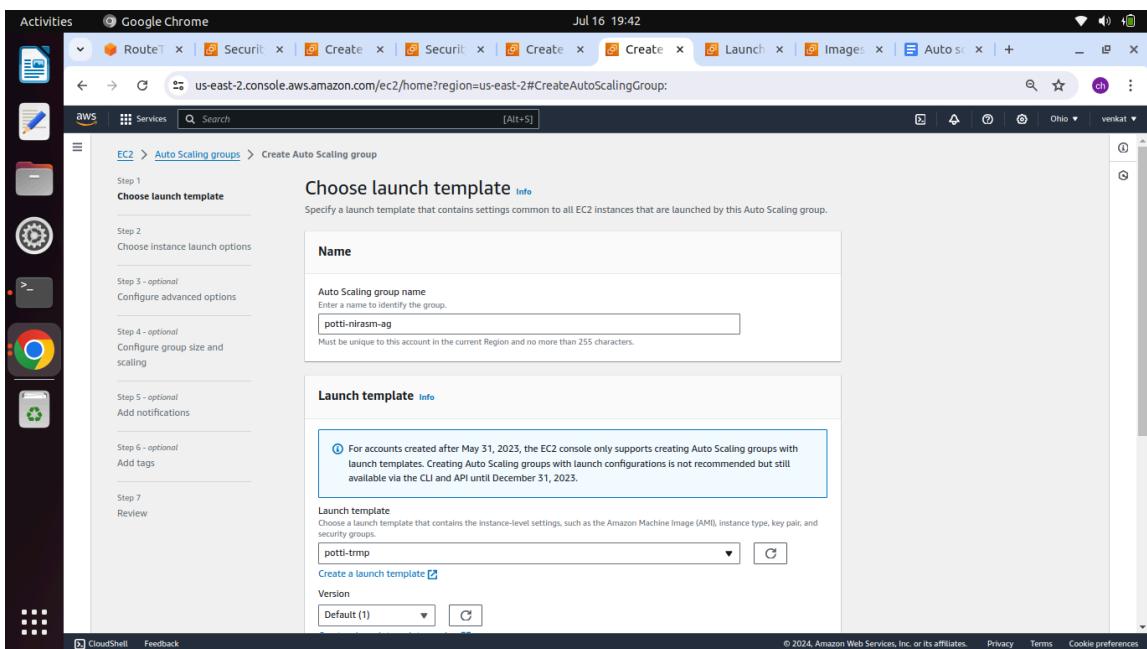


213)

214) Key pair= potti-app and don't select subnet it takes automatically select exiting sg=potti-nirasam-sg and click create launch template

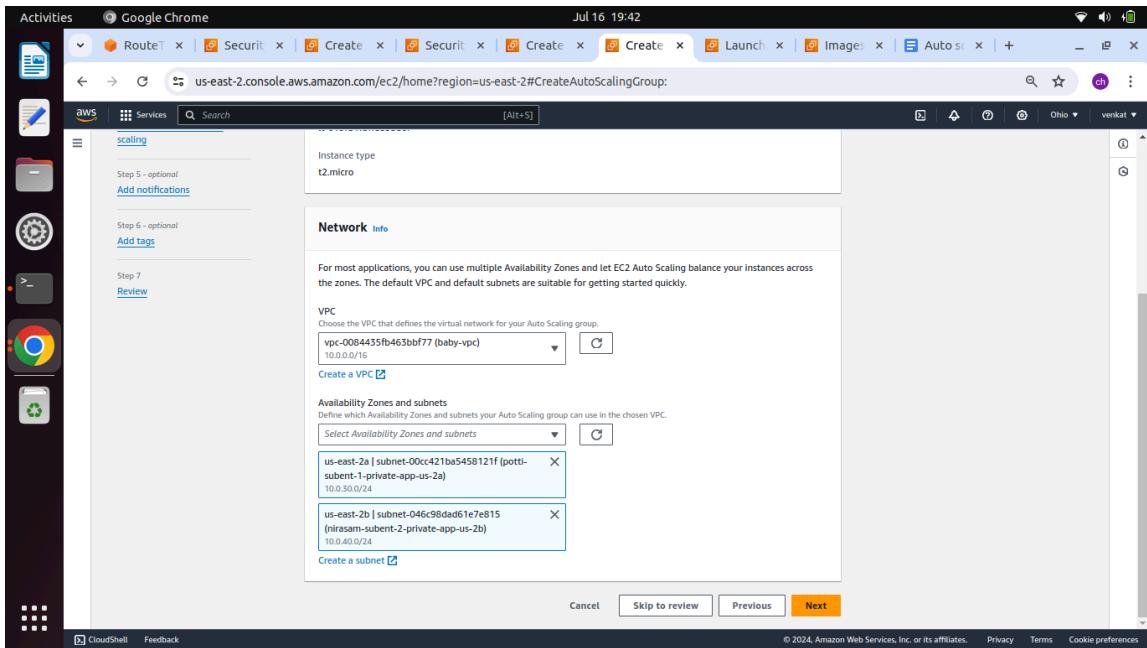


215) Now refresh launch template = potti-temp

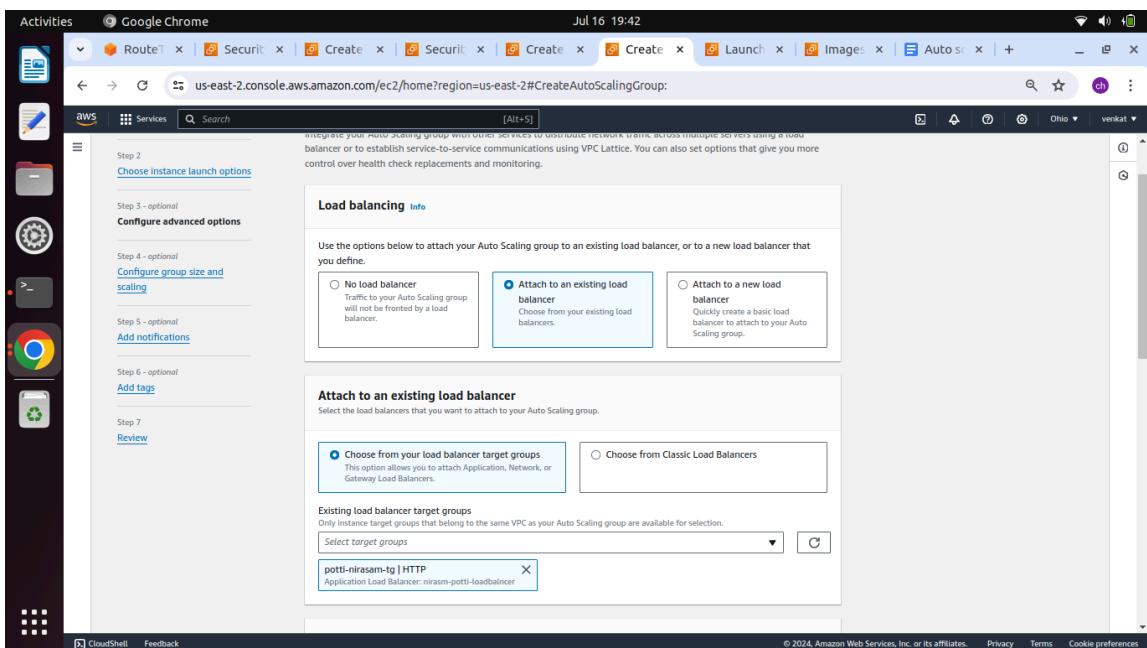


216) vpc=baby-vpc

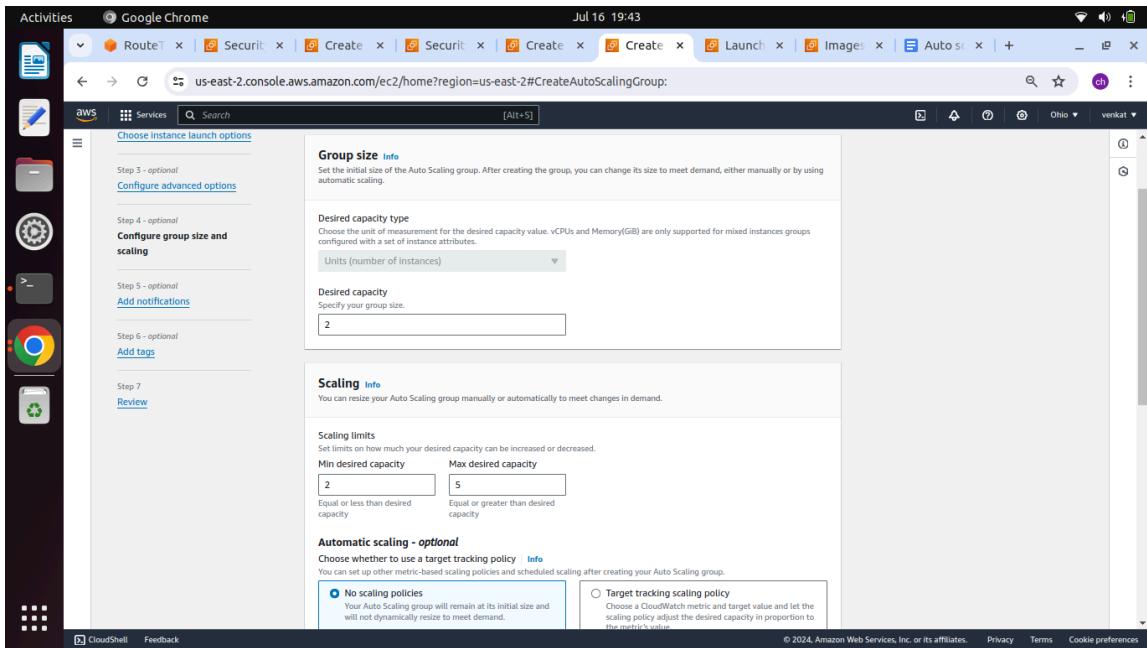
217) Availability zones= potti and nirasm subnets select and click next



218) Load balancing= attach to an existing load balancer click choose from your load balancer target groups and select potti-nirasam-tg

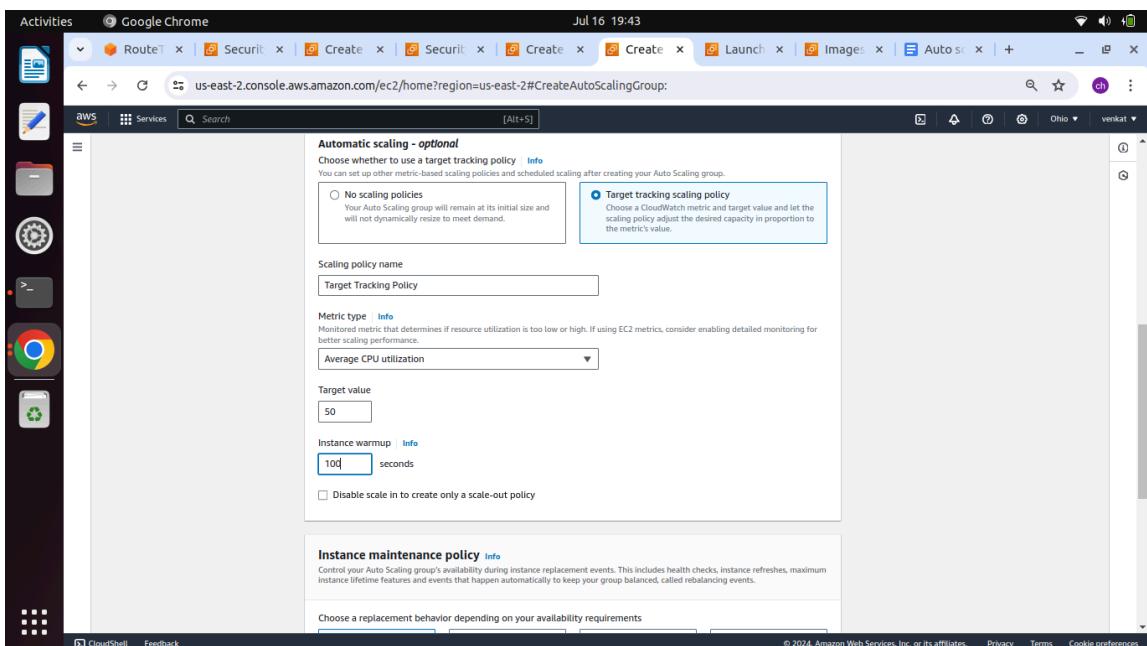


219) Desired capacity=2 , minimum capacity=2 maximum=5



220) Select target tracking scaling policy

221) Target value= 50 instance warmp=100



222) Click create auto scaling group

The screenshot shows the AWS Cloud9 IDE interface. The browser tab is titled "us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#AutoScalingGroups:". The main content area displays the "Auto Scaling groups (2) Info" page. It lists two Auto Scaling groups: "potti-nirasm-ag" and "rao-venkat-auto". The "potti-nirasm-ag" group has a status of "Updating capacity..." and a desired capacity of 2. The "rao-venkat-auto" group has a status of "Version Default" and a desired capacity of 2. The interface includes a search bar, filter buttons for Launch configurations, Launch templates, Actions, and Create Auto Scaling group, and a table header with columns for Name, Launch template/configuration, Instances, Status, Desired capacity, Min, Max, and Availability Zones.

223) Now go to ec2 instance it create automatically new instance

The screenshot shows the AWS Cloud9 IDE interface. The browser tab is titled "us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#Instances?v=3;\$.case=tags:true%5C;client:false;\$.regex=tags:false%5C;client:false". The main content area displays the "Instances (8) Info" page. It lists eight EC2 instances with the following details:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
i-00905c9a7ceeb8701	Pending	t2.nano	-	-	-	us-east-2b	-
nirasm-app	i-03fbfb1fb2ad84d2	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2b	-
venkat-web	i-0x64048a903aab6a0	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2b	-
	i-08e007962c7fcf128	Running	t2.micro	Initializing	View alarms +	us-east-2b	-
	i-09e1c0ab5f4a11e8	Running	t2.nano	2/2 checks passed	View alarms +	us-east-2a	-
potti-app	i-0b8f5606fb0d1614c	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2a	-
rao-web	i-00004c997d475d087	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2a	-
	i-0b4b7199180297e65	Running	t2.micro	Initializing	View alarms +	us-east-2a	-

224) Go to rds click subnet groups

225) name= devil-rakashi-sg and vpc=baby-vpc

The screenshots show the AWS RDS console interface for creating a DB subnet group.

Top Screenshot (Subnet Selection):

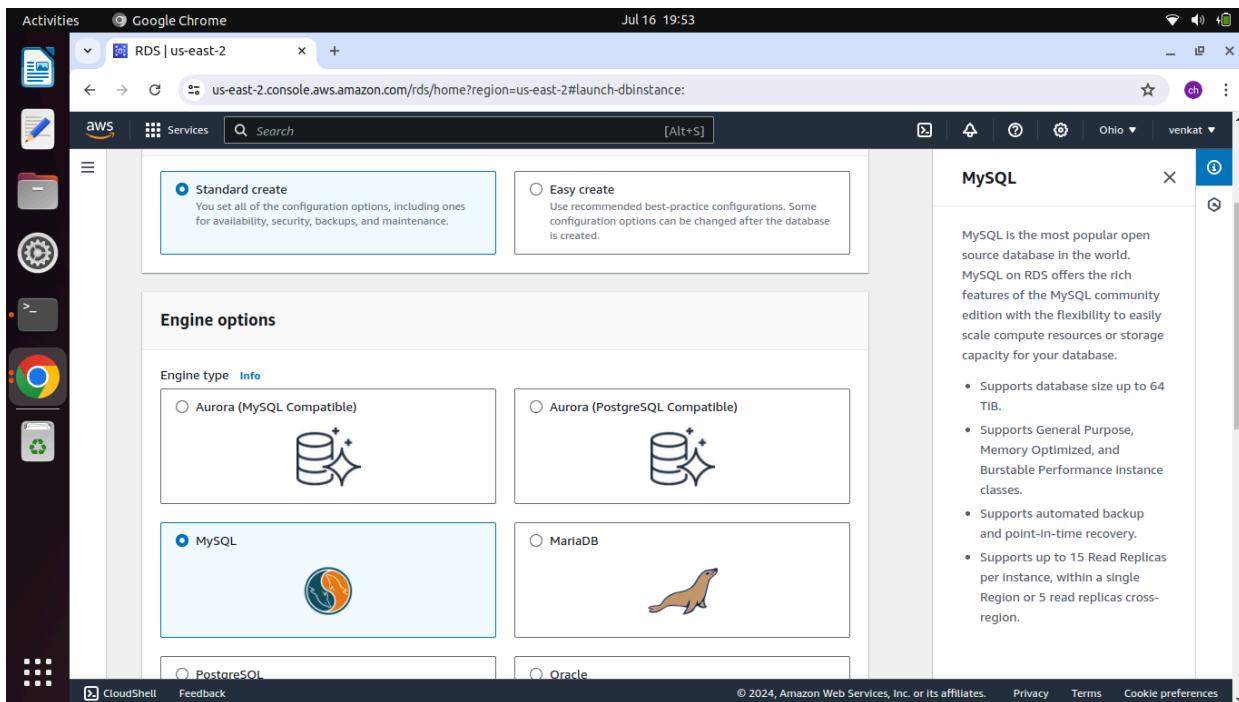
- Availability Zones:** Choose an availability zone. Options: us-east-2a, us-east-2b.
- Subnets:** Choose the subnets you want to add. Options: subnet-0a66662ac9b0a7987 (10.0.8.0/22), subnet-0966a3a1492cdced3 (10.0.48.0/20).
- Note:** For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones.

Availability zone	Subnet ID	CIDR block
us-east-2b	subnet-0a66662ac9b0a7987	10.0.8.0/22
us-east-2a	subnet-0966a3a1492cdced3	10.0.48.0/20

Bottom Screenshot (Create DB subnet group):

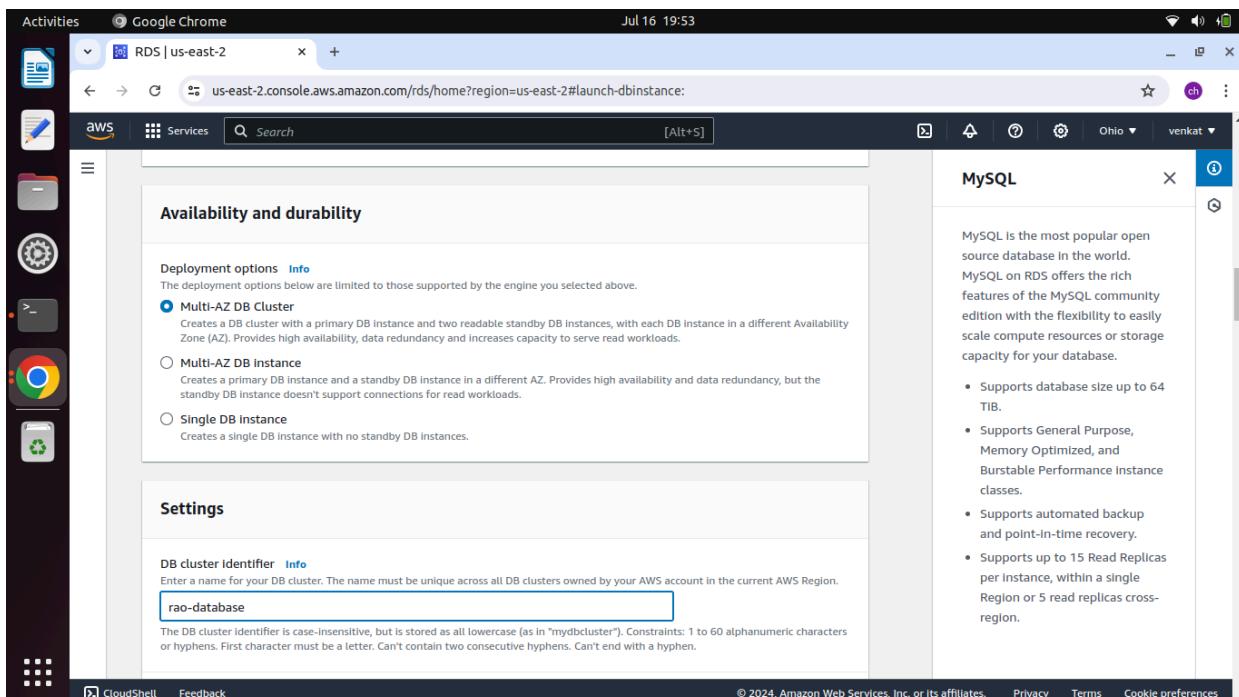
- Name:** devil-rakshi-sg
- Description:** allow
- VPC:** baby-vpc (vpc-008445fb463bbf77)

- 226) Availability zones= 2a and 2b
- 227) subnets= devil and rakshi subnets select
- 228) Click create subnet group
- 229) Now go to databases
- 230) Select standard create select MY SQL



231) Select multi-az-db cluster

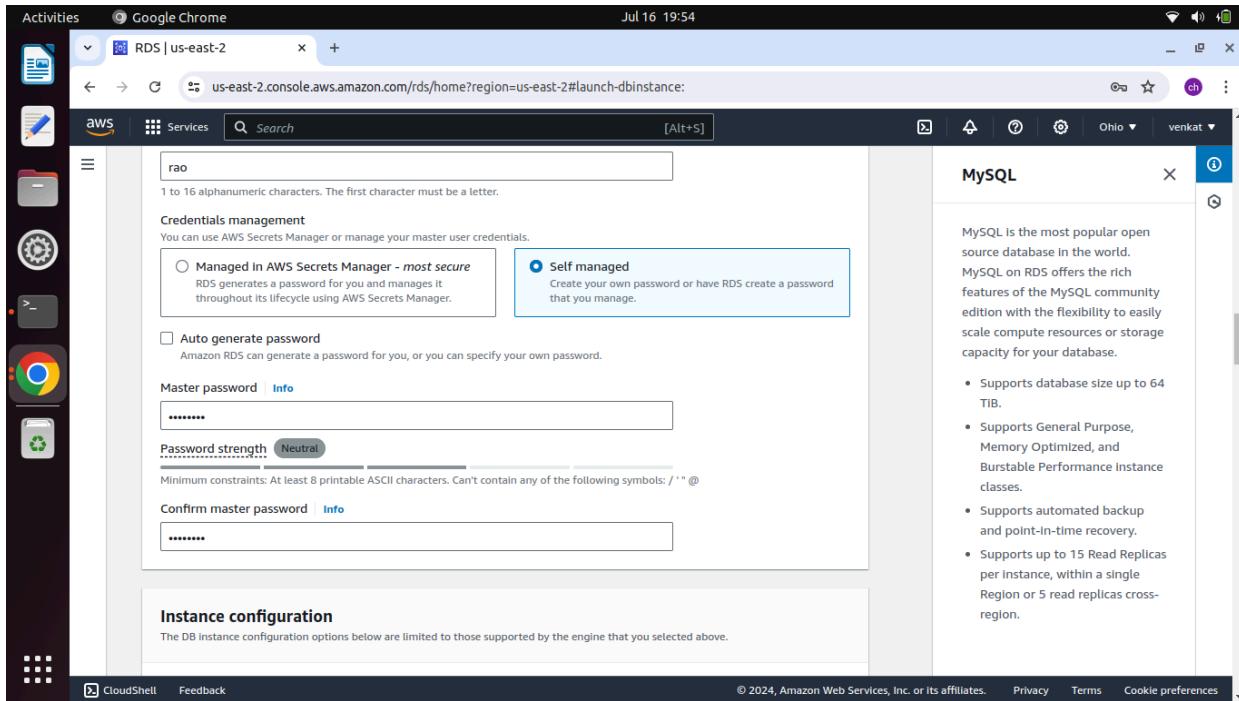
232) Db cluster identifier=rao-database



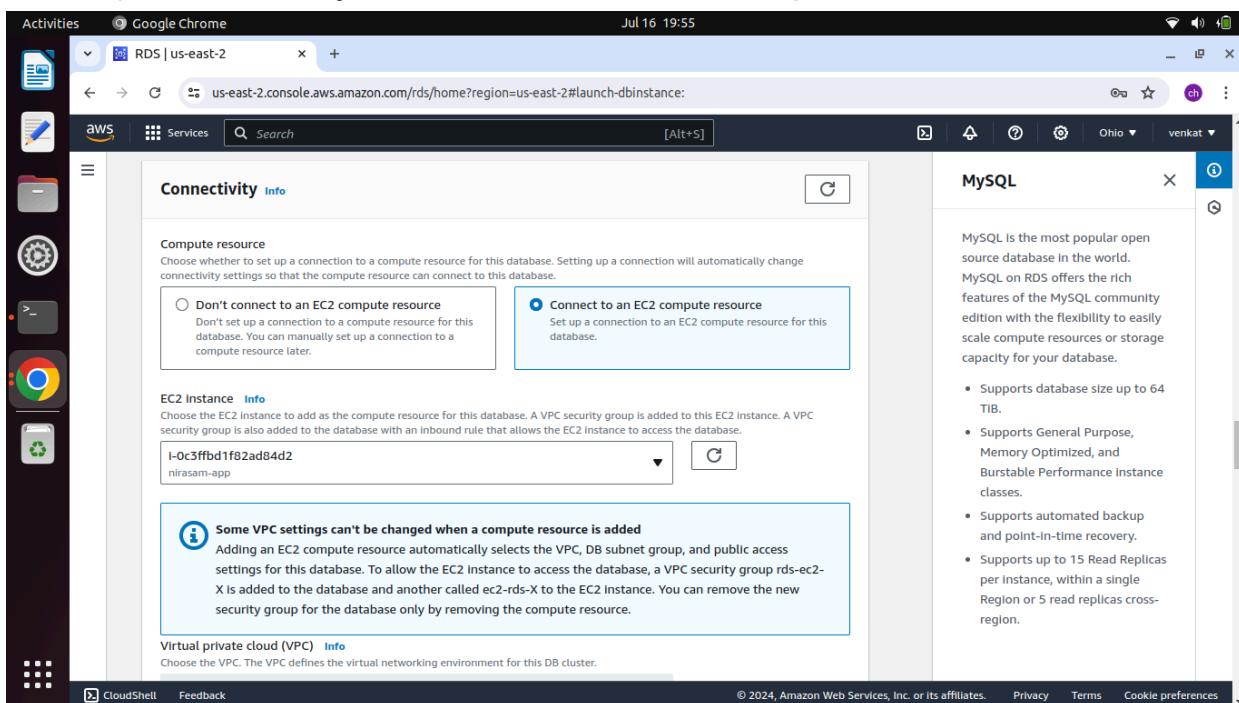
233) username= rao

234) Cluster management type= self mange

235) Master password= rao1234



236) connectivity= connect to an ec2 compute resource



237) Do create rds all availability zones each one one subnet need so i miss us-east-2c so click add new subnet create subnet in 2c

The screenshot shows the AWS RDS CreateSubnet page. On the left, there's a sidebar with various AWS services like Lambda, CloudWatch, and S3. The main content area has a heading "CreateSubnet | VPC Console". A message box says "Some VPC settings can't be changed when a compute resource is added". Below it, a dropdown shows "baby-vpc (vpc-0064435fb463bbf77)" with "6 Subnets, 2 Availability Zones". A note says "Only VPCs with a corresponding DB subnet group are listed." A warning box says "The VPC subnets must be in 3 Availability Zones (AZs) for the Multi-AZ DB cluster. The current subnets are in 2 AZs (us-east-2b, us-east-2a). Add a subnet in a different AZ than the current subnets." Another message box says "After a database is created, you can't change its VPC." At the bottom, there's a "DB subnet group" section with a note about choosing a DB subnet group. To the right, a sidebar for "MySQL" lists its features: supports database sizes up to 64 TiB, general purpose, memory optimized, and burstable performance instance classes, automated backup and point-in-time recovery, and up to 15 read replicas per instance. The URL in the address bar is <https://us-east-2.console.aws.amazon.com/vpc/home#CreateSubnet>.

238) az= us-east-2c

239) ipv4= 10.0.22.0/22 click crete subnet

The screenshot shows the AWS VPC CreateSubnet page. It has sections for "Availability Zone", "IPv4 VPC CIDR block", "IPv4 subnet CIDR block" (set to 10.0.22.0/22), and "Tags - optional" (with a tag named "db-sub-3"). At the bottom, there are "Cancel" and "Create subnet" buttons. The URL in the address bar is <https://us-east-2.console.aws.amazon.com/vpcconsole/home?region=us-east-2#CreateSubnet>.

240) Click automatic setup

241) Public access= no

242) Vpc sg=choose exciting

243) Select sg click create database

Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB cluster can use in the VPC that you selected.

Choose existing
Choose existing DB subnet group

Automatic setup
RDS creates a new subnet group for you or reuses an existing subnet group

DB subnet group name
rds-ec2-db-subnet-group-1

New DB subnet group created.

Public access [Info](#)

Yes
RDS assigns a public IP address to the cluster. Amazon EC2 instances and other resources outside of the VPC can connect to your cluster. Resources inside the VPC can also connect to the cluster. Choose one or more VPC security groups that specify which resources can connect to the cluster.

No
RDS doesn't assign a public IP address to the cluster. Only Amazon EC2 instances and other resources inside the VPC can connect to your cluster. Choose one or more VPC security groups that specify which resources can connect to the cluster.

VPC security group (firewall) [Info](#)

Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

Choose existing
Choose existing VPC security groups

Create new
Create new VPC security group

Additional VPC security group
Choose one or more options

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

Consider creating a Blue/Green Deployment to minimize downtime during upgrades
You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. [RDS User Guide](#) [Aurora User Guide](#)

Databases (4)

Group resources [C](#) [Modify](#) [Actions](#) [Restore from S3](#) [Create database](#)

Filter by databases

DB identifier	Status	Role	Engine	Region & AZ
rao-database	Creating	Multi-AZ DB cluster	MySQL Community	us-east-2
rao-database-instance-1	Creating	Reader Instance	MySQL Community	us-east-2b
rao-database-instance-2	Creating	Reader Instance	MySQL Community	us-east-2a
rao-database-instance-3	Creating	Reader Instance	MySQL Community	us-east-2c

244) Now go to rao server connect it

245) Connect rao server through nirasam server

```
Activities Terminal Jul 16 20:04
ec2-user@ip-10-0-40-100:~ ec2-user@ip-10-0-40-100:~ challa@challa-HP-Laptop-15-da0xxx:~/Downloads
challa@challa-HP-Laptop-15-da0xxx:~/Downloads$ cd Downloads/
challa@challa-HP-Laptop-15-da0xxx:~/Downloads$ ssh -i "rao.pem" ec2-user@18.225.55.140
Last login: Tue Jul 16 12:28:39 2024 from 60.243.169.79
[ec2-user@rao ~]$ sudo -i
[root@rao ~]# ll
total 0
[root@rao ~]# exit
logout
[ec2-user@rao ~]$ ll
total 4
-r----- 1 ec2-user ec2-user 1675 Jul 16 12:29 nirasam-app.pem
[ec2-user@rao ~]$ ssh -i "nirasam-app.pem" ec2-user@10.0.40.100
Last login: Tue Jul 16 12:31:13 2024 from 10.0.0.26
[ec2-user@ip-10-0-40-100 ~]$
```

246) Nirasam-sg add http and mysql/aurora port numbers

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-09f379501c29a6512	SSH	TCP	22	Cus... 0.0.0.0/0	
-	HTTP	TCP	80	An... 0.0.0.0/0	
-	MYSQL/Aurora	TCP	3306	An... 0.0.0.0/0	

Warning: Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Preview changes Save rules

247) After connect nirasam server install my sql in nirasam server using bellow commands because i take server amazon linux

248) #sudo dnf update -y

249) #sudo dnf install mariadb105

Note
The `-y` option installs the updates without asking for confirmation. To examine updates before installing, omit this option.

```
sudo dnf update -y
```

4. To install the mysql command-line client from MariaDB on Amazon Linux 2023, run the following command:

```
sudo dnf install mariadb105
```

5. Connect to the MySQL DB instance. For example, enter the following command. This action lets you connect to the MySQL DB instance using the MySQL client.
Substitute the DB instance endpoint (DNS name) for `endpoint`, and substitute the master username that you used for `admin`. Provide the master password that you used when prompted for a

On this page

- Prerequisites
- Step 1: Create an EC2 instance
- Step 2: Create a MySQL DB instance
- (Optional) Create VPC, EC2
- Step 3: Connect to a MySQL DB instance
- Step 4: Delete the EC2 instance and

Introducing Amazon Q
Receive guidance, get troubleshooting tips, and learn about AWS services and capabilities.

250) Check install or not #mysql –version

251) Now go to rao-database click rao and copy writer end point

Endpoint	Type	Port
rao-database.cluster-cxs6cq6muyc7.us-east-2.rds.amazonaws.com	Writer	3306
rao-database.cluster-ro-cxs6cq6muyc7.us-east-2.rds.amazonaws.com	Reader	3306

252) Go to terminal

253) #mysql -h paste here endpoint -u username -p click enter and type password now you connect my sql

254) MYSQL> show databases;

```
Activities Terminal Jul 16 20:11 root@ip-10-0-40-100:~  
Verifying : mariadb105-3:10.5.23-1.amzn2023.0.1.x86_64 3/5  
Verifying : mariadb105-common-3:10.5.23-1.amzn2023.0.1.x86_64 4/5  
Verifying : perl-Sys-Hostname-1.23-477.amzn2023.0.6.x86_64 5/5  
  
Installed:  
mariadb-connector-c-3.1.13-1.amzn2023.0.3.x86_64 mariadb-connector-c-config-3.1.13-1.amzn2023.0.3.noarch  
mariadb105-3:10.5.23-1.amzn2023.0.1.x86_64 mariadb105-common-3:10.5.23-1.amzn2023.0.1.x86_64  
perl-Sys-Hostname-1.23-477.amzn2023.0.6.x86_64  
  
Complete!  
[root@ip-10-0-40-100 ~]# mysql --version  
mysql Ver 15.1 Distrib 10.5.23-MariaDB, for Linux (x86_64) using EditLine wrapper  
[root@ip-10-0-40-100 ~]# mysql -h rao-database.cluster-cxs6cq6muy7.us-east-2.rds.amazonaws.com -u rao -p  
Enter password:  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MySQL connection id is 38  
Server version: 8.0.35 Source distribution  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MySQL [(none)]> create database rao;  
Query OK, 1 row affected (0.003 sec)  
  
MySQL [(none)]> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| rao |  
| sys |  
+-----+  
5 rows in set (0.001 sec)  
  
MySQL [(none)]>
```

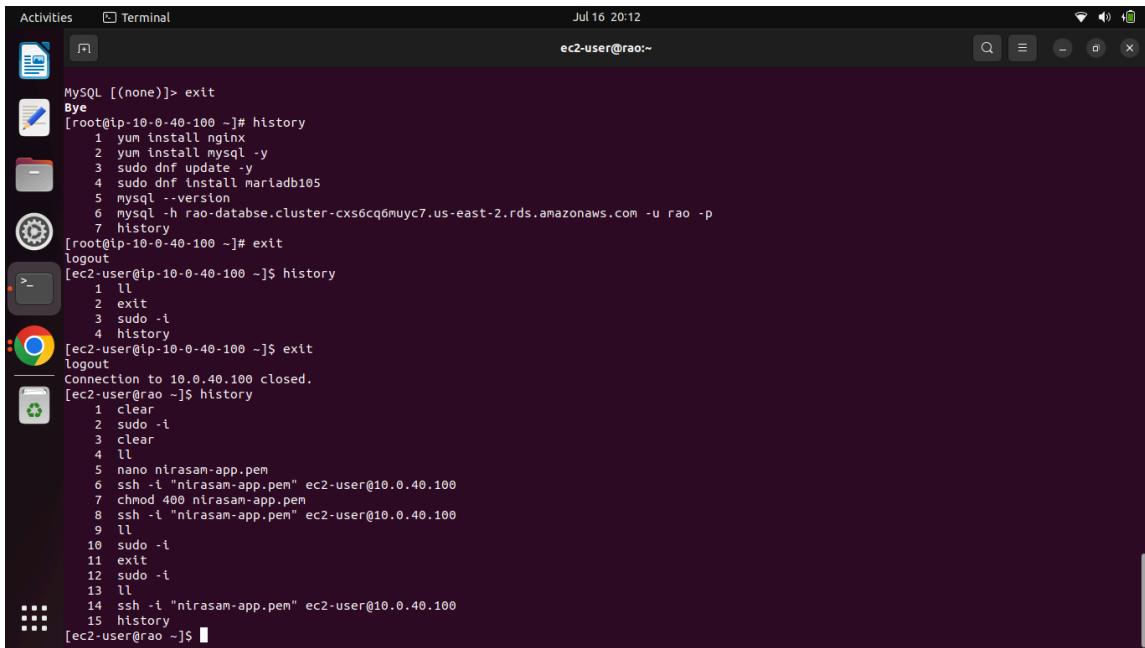
255) MySQL>create database rao;

256) MySQL> show databases;

257) Click exit

```
Activities Terminal Jul 16 20:11 root@ip-10-0-40-100:~  
Verifying : mariadb105-3:10.5.23-1.amzn2023.0.1.x86_64 3/5  
Verifying : mariadb105-common-3:10.5.23-1.amzn2023.0.1.x86_64 4/5  
Verifying : perl-Sys-Hostname-1.23-477.amzn2023.0.6.x86_64 5/5  
  
Installed:  
mariadb-connector-c-3.1.13-1.amzn2023.0.3.x86_64 mariadb-connector-c-config-3.1.13-1.amzn2023.0.3.noarch  
mariadb105-3:10.5.23-1.amzn2023.0.1.x86_64 mariadb105-common-3:10.5.23-1.amzn2023.0.1.x86_64  
perl-Sys-Hostname-1.23-477.amzn2023.0.6.x86_64  
  
Complete!  
[root@ip-10-0-40-100 ~]# mysql --version  
mysql Ver 15.1 Distrib 10.5.23-MariaDB, for Linux (x86_64) using EditLine wrapper  
[root@ip-10-0-40-100 ~]# mysql -h rao-database.cluster-cxs6cq6muy7.us-east-2.rds.amazonaws.com -u rao -p  
Enter password:  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MySQL connection id is 38  
Server version: 8.0.35 Source distribution  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MySQL [(none)]> create database rao;  
Query OK, 1 row affected (0.003 sec)  
  
MySQL [(none)]> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| rao |  
| sys |  
+-----+  
5 rows in set (0.001 sec)  
  
MySQL [(none)]> exit  
Bye  
[root@ip-10-0-40-100 ~]# history  
1 yum install nginx  
2 yum install mysql -y  
3 sudo dnf update -y  
4 sudo dnf install Mariadb105  
5 mysql --version  
6 mysql -h rao-database.cluster-cxs6cq6muy7.us-east-2.rds.amazonaws.com -u rao -p  
7 history  
[root@ip-10-0-40-100 ~]
```

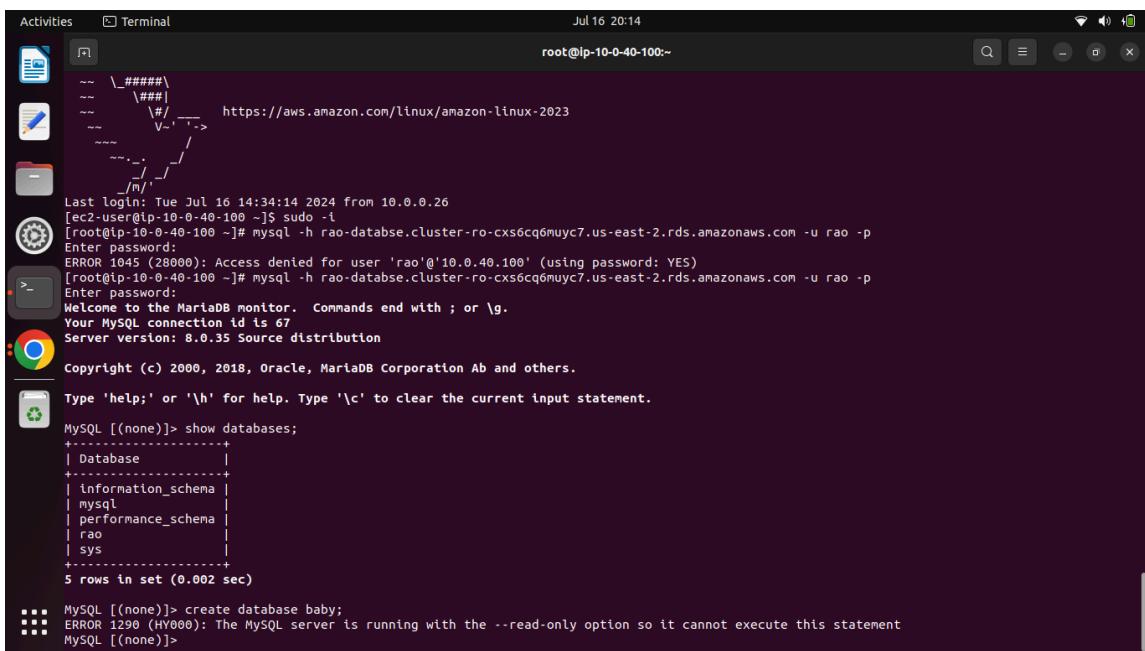
258)



```
Activities Terminal Jul 16 20:12
ec2-user@rao:~>

MySQL [(none)]> exit
Bye
[root@ip-10-0-40-100 ~]# history
 1 yum install nginx
 2 yum install mysql -y
 3 sudo dnf update -y
 4 sudo dnf install mariadb105
 5 mysql --version
 6 mysql -h rao-database.cluster-cxs6cq6muyc7.us-east-2.rds.amazonaws.com -u rao -p
 7 history
[root@ip-10-0-40-100 ~]# exit
logout
[ec2-user@ip-10-0-40-100 ~]$ history
 1 ll
 2 exit
 3 sudo -i
 4 history
[ec2-user@ip-10-0-40-100 ~]$ exit
Connection to 10.0.40.100 closed.
[ec2-user@rao ~]$ history
 1 clear
 2 sudo -i
 3 clear
 4 ll
 5 nano nirasam-app.pem
 6 ssh -i "nirasam-app.pem" ec2-user@10.0.40.100
 7 chmod 400 nirasam-app.pem
 8 ssh -i "nirasam-app.pem" ec2-user@10.0.40.100
 9 ll
10 sudo -i
11 exit
12 sudo -i
13 ll
14 ssh -i "nirasam-app.pem" ec2-user@10.0.40.100
15 history
[ec2-user@rao ~]$
```

259) Now connect reader try to show database and create database you can show here but you can't crete here



```
Activities Terminal Jul 16 20:14
root@ip-10-0-40-100:~

~~ \#####
~~ \|##|
~~ #/ _--> https://aws.amazon.com/linux/amazon-linux-2023
~~ `-'`_`_`_
~~ .-`_`_`_
~~ /`_`_`_
~~ /`_`_`_
~~ /`_`_`_
Last login: Tue Jul 16 14:34:14 2024 from 10.0.0.26
[ec2-user@ip-10-0-40-100 ~]$ sudo -l
[root@ip-10-0-40-100 ~]# mysql -h rao-database.cluster-ro-cxs6cq6muyc7.us-east-2.rds.amazonaws.com -u rao -p
Enter password:
ERROR 1045 (28000): Access denied for user 'rao'@'10.0.40.100' (using password: YES)
[ec2-user@ip-10-0-40-100 ~]# mysql -h rao-database.cluster-ro-cxs6cq6muyc7.us-east-2.rds.amazonaws.com -u rao -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 67
Server version: 8.0.35 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| rao |
| sys |
+-----+
5 rows in set (0.002 sec)

MySQL [(none)]> create database baby;
ERROR 1290 (HY000): The MySQL server is running with the --read-only option so it cannot execute this statement
MySQL [(none)]>
```

=====

=====

