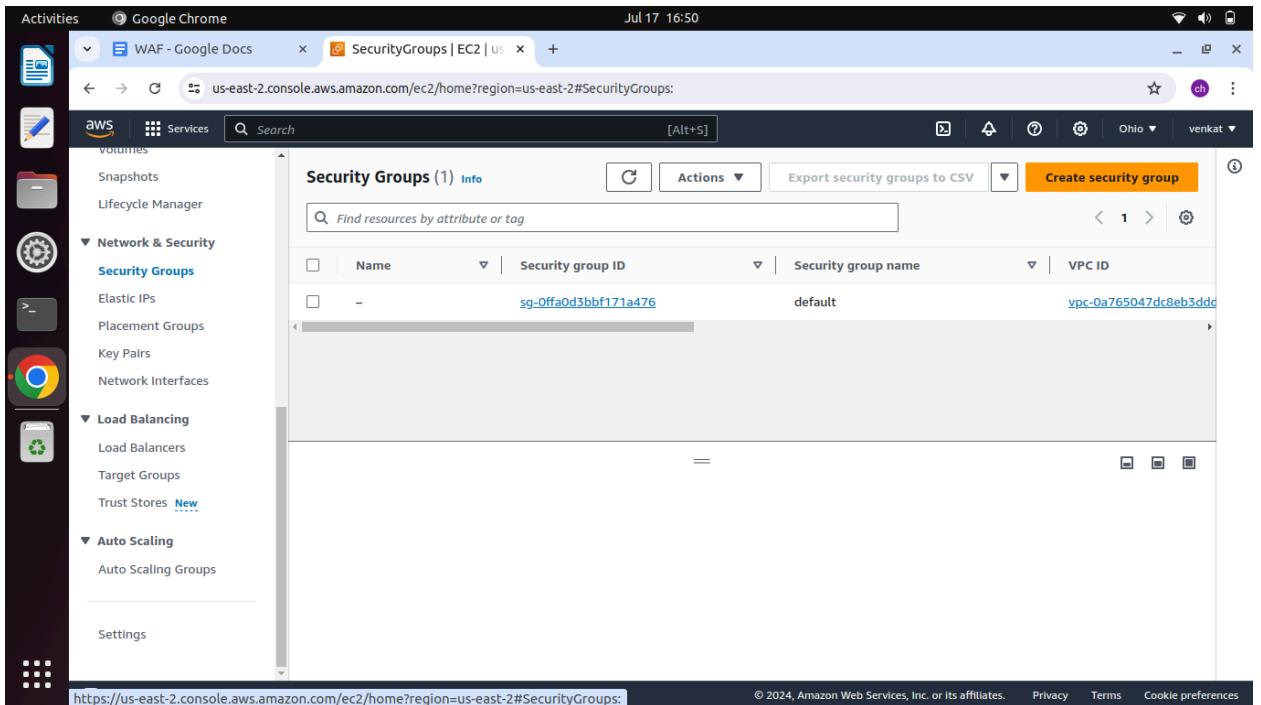


# WAF

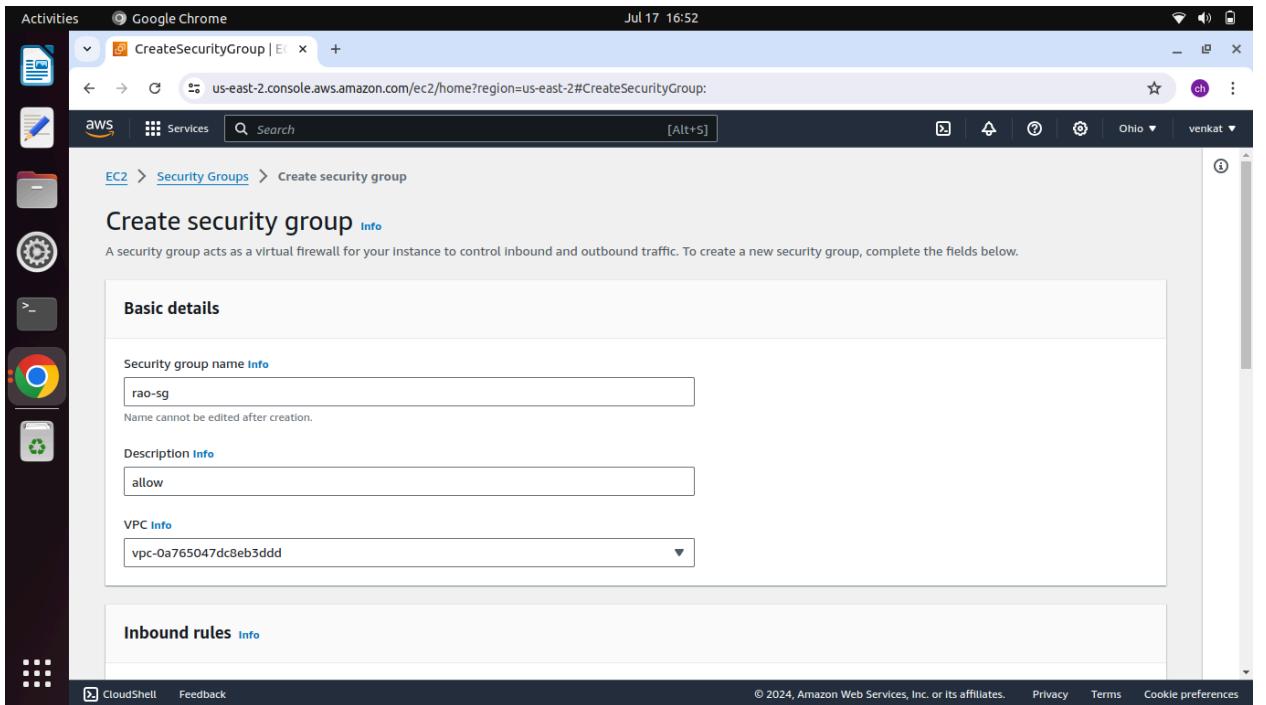
⇒ region single region use all → sg → ec2 server 2 → target group → load balancer → waf → ip set → rule group → web acl → load balancer dns → google

- 1) Region = ohio
- 2) Go to security group click create security group

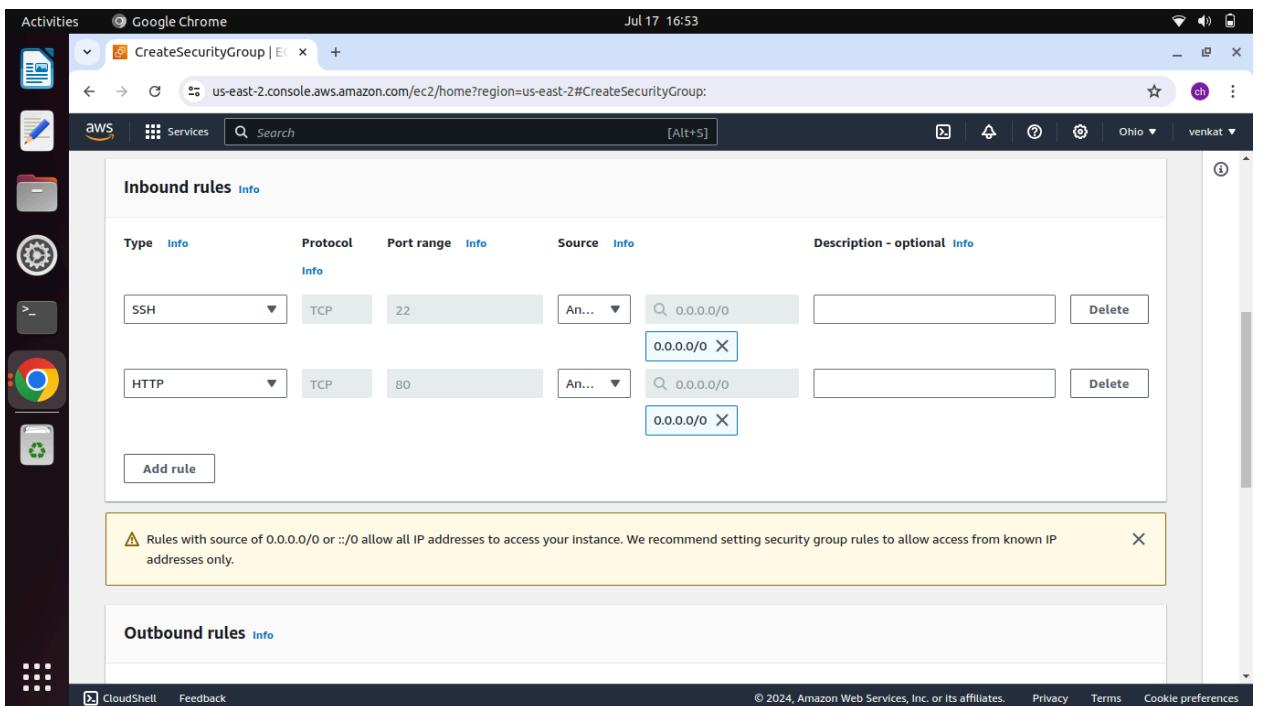


The screenshot shows the AWS Management Console interface for the EC2 service. The left sidebar has 'Activities' at the top, followed by icons for Google Docs, a file, a pen, a folder, and a browser. The main navigation bar includes 'Services' and a search bar. Below the search bar, the 'AWS' logo is followed by 'Services'. The main content area is titled 'Security Groups (1) Info'. It displays a table with one row, showing a security group named 'sg-Offa0d3bbf171a476' with a 'default' security group name and a VPC ID of 'vpc-0a765047dc8eb3ddc'. The table has columns for Name, Security group ID, Security group name, and VPC ID. There are 'Actions' and 'Create security group' buttons at the top right of the table. The bottom of the screen shows the URL 'https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#SecurityGroups:' and standard footer links for privacy, terms, and cookie preferences.

- 3) name= rao-sg and vpc=default vpc



#### 4) Inbound rule add rule =ssh and http



#### 5) Click Create security group

Activities Google Chrome

Jul 17 16:54

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

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

Security group (sg-04ee2f334e2f44d50 | rao-sg) was created successfully

Details

Security Groups (2) Info Actions Export security groups to CSV Create security group

Find resources by attribute or tag

Name	Security group ID	Security group name	VPC ID
-	sg-Offa0d3bbf171a476	default	vpc-0a765047dc8eb5dd
rao-sg	sg-04ee2f334e2f44d50	rao-sg	vpc-0a765047dc8eb5dd

sg-04ee2f334e2f44d50 - rao-sg

Details Inbound rules Outbound rules Tags

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

- 6) Go to ec2 select Instances click launch instance
- 7) name= rao
- 8) Keypair =rao
- 9) Select vpc and subnet click assign ip= enable and sg= rao-sg
- 10) Click launch instance

Activities Google Chrome

Jul 17 17:01

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

Launch an instance | EC2 RouteTableDetails | VPC New Tab

Instance type

Key pair (login)

Network settings

Summary

Number of instances: 1

Software image (AMI): Amazon Linux 2023 AMI 2023.5.2... read more

Virtual server type (Instance type): t2.micro

Firewall (security group): rao-sg

Storage (volumes): 1 volume(s) - 8 GB

Free tier: Free tier in your first year includes 750 hours of t2.micro on t3.micro (unavailable) instance usage on free tier AMIs per month, 750 hours of t3.micro instance usage on pay-as-you-go AMIs per month, 30 GB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Launch instance Review commands

VPC - Region: vpc-0a765047dc8eb5dd (default)

Subnet: subnet-03400803687d72c7 sub1

Auto-assign public IP: Enabled

Firewall (security group): rao-sg

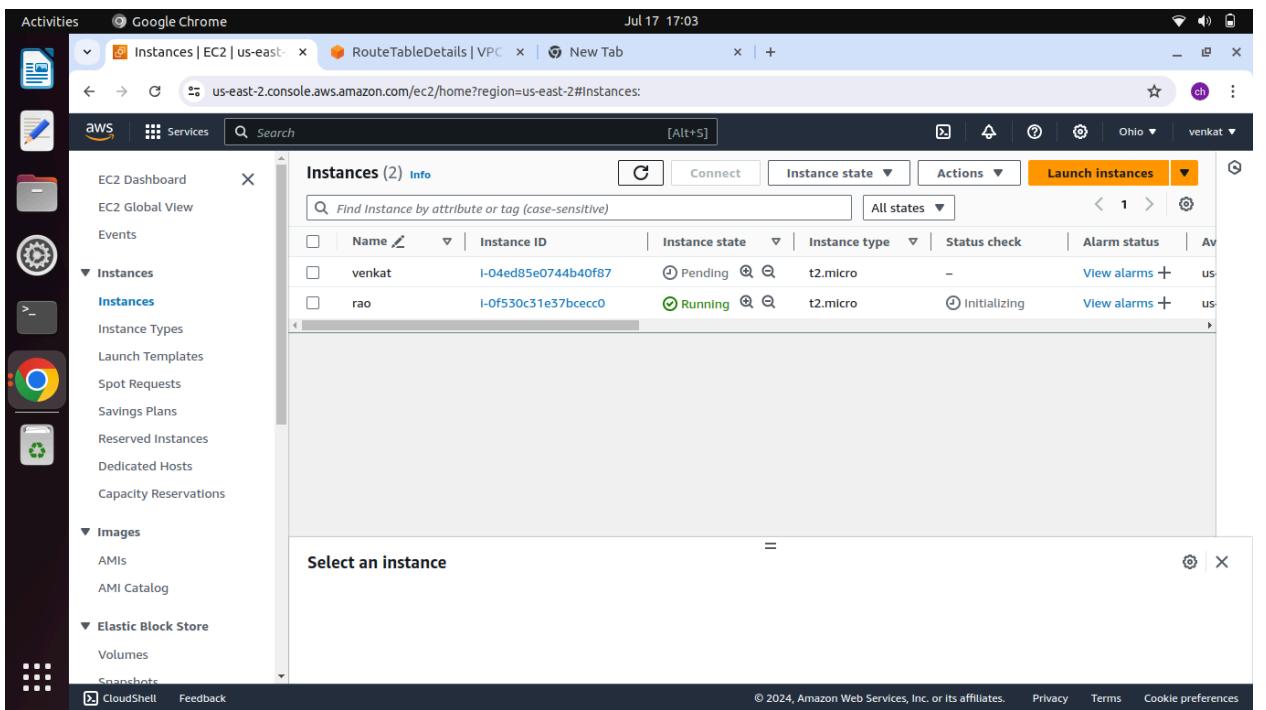
Common security groups info

Select security group: rao-sg (sg-04ee2f334e2f44d50)

Compare security group rules

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

## 11) Same as create venkat server



## 12) Connect rao server and install nginx

13) \$ sudo -i

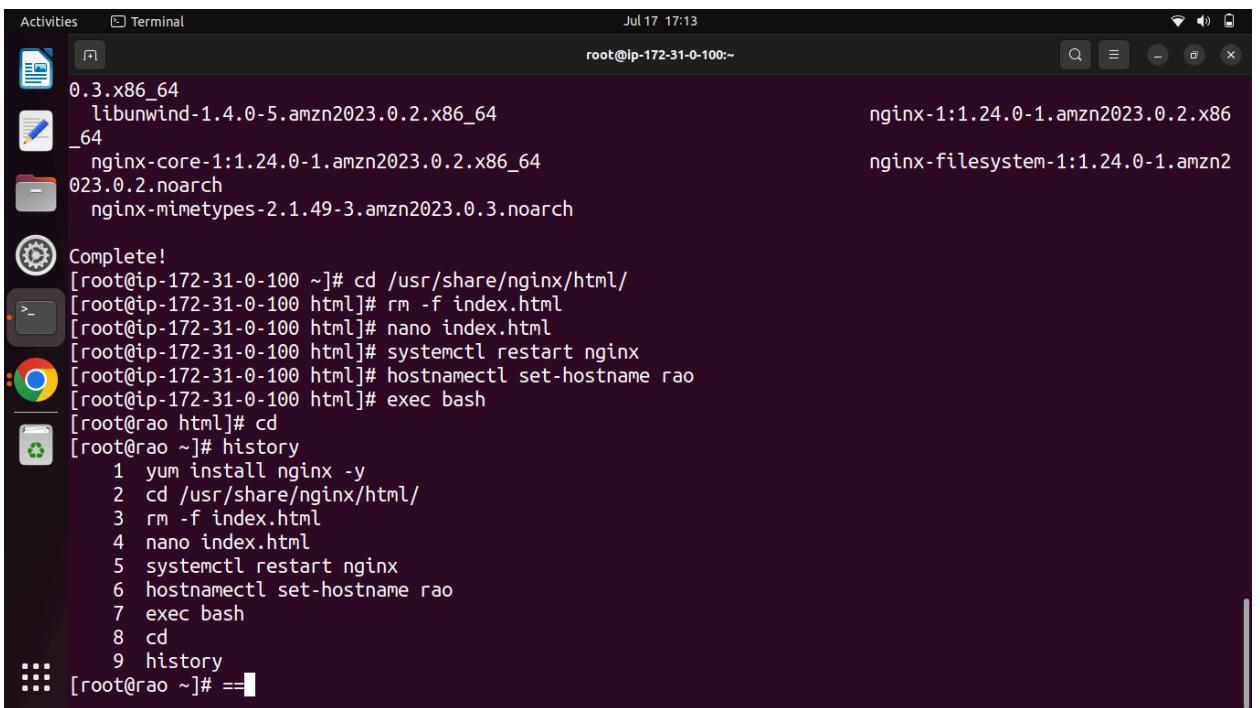
14) # yum install nginx -y

```
Activities Terminal Jul 17 17:05
root@ip-172-31-0-100:~#
challa@challa-HP-Laptop-15-da0xxx:~$ cd Downloads/
challa@challa-HP-Laptop-15-da0xxx:~/Downloads$ ssh -i "rao.pem" ec2-user@ec2-18-117-162-81.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-18-117-162-81.us-east-2.compute.amazonaws.com (18.117.162.81)' can't be established.
ED25519 key fingerprint is SHA256:FEJErINLXa/H5+Ed5F5DPDWJZ4DaCNLxQDD5ZmmBkps.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'ec2-18-117-162-81.us-east-2.compute.amazonaws.com' (ED25519) to the list of known hosts.

#_ _###_ Amazon Linux 2023
~~ \###\_
~~ \|##|
~~ \#/ V~`-->
~~ /`/ /
~~ ._. /`/
~~ /`/ /
~~ /`/ /
[ec2-user@ip-172-31-0-100 ~]$ sudo -i
[root@ip-172-31-0-100 ~]# yum install nginx -y
```

A screenshot of a terminal window. The title bar says 'Activities Terminal Jul 17 17:05'. The user is connected to the 'rao' instance via SSH. The terminal shows the command to change directory to 'Downloads', the SSH connection, and the successful installation of 'nginx' using 'yum'. The user is running as root ('root@ip-172-31-0-100:~#').

- 15) Go to /usr/share/nginx/html/ path and delete index.html file create new index.html file using vi or nano write file this is rao server save and exit and restart the nginx server using cmd # systemctl restart nginx
- 16) # cd /usr/share/nginx/html/
- 17) # rm -f index.html
- 18) # nano index.html
- 19) # systemctl restart nginx
- 20) # hostnamectl set-hostname rao
- 21) # exec bash
- 22) # cd



The screenshot shows a terminal window with a dark theme. The title bar says "Activities Terminal" and the status bar shows "Jul 17 17:13" and "root@ip-172-31-0-100:~". The terminal window displays the following command history:

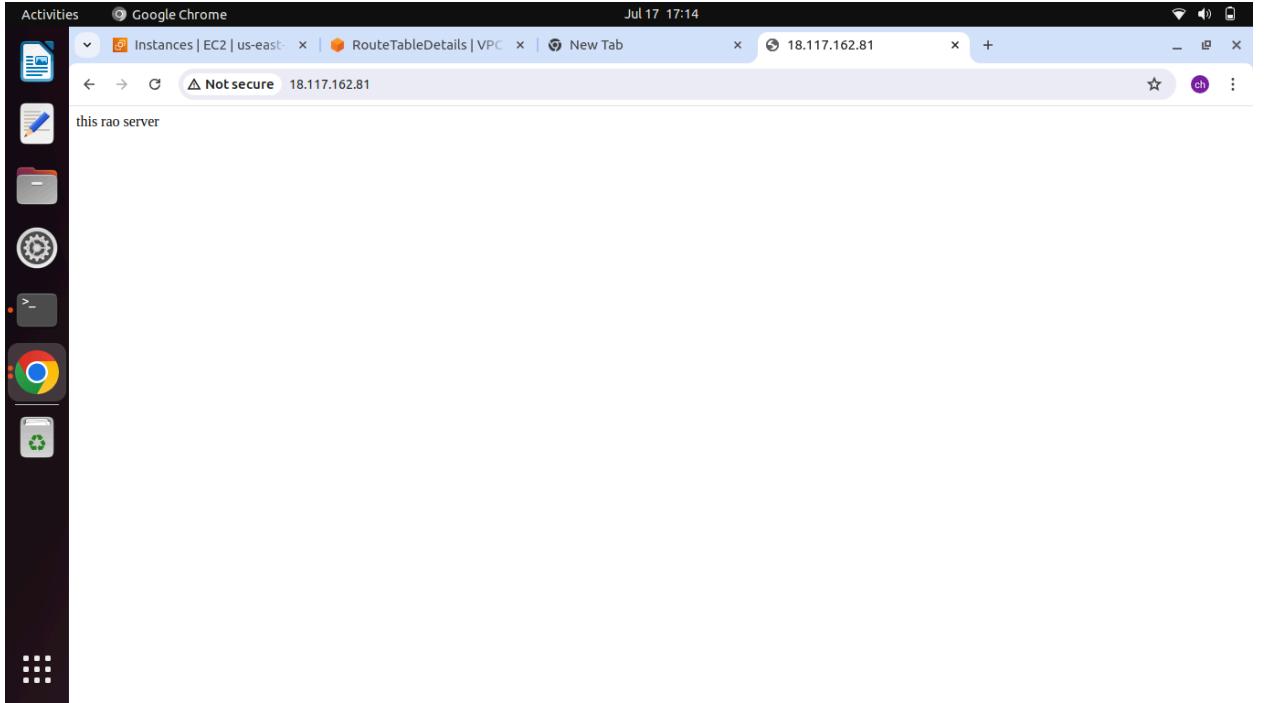
```

0.3.x86_64
libunwind-1.4.0-5.amzn2023.0.2.x86_64
_64
nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64
023.0.2.noarch
nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

Complete!
[root@ip-172-31-0-100 ~]# cd /usr/share/nginx/html/
[root@ip-172-31-0-100 html]# rm -f index.html
[root@ip-172-31-0-100 html]# nano index.html
[root@ip-172-31-0-100 html]# systemctl restart nginx
[root@ip-172-31-0-100 html]# hostnamectl set-hostname rao
[root@ip-172-31-0-100 html]# exec bash
[root@rao html]# cd
[root@rao ~]# history
      1  yum install nginx -y
      2  cd /usr/share/nginx/html/
      3  rm -f index.html
      4  nano index.html
      5  systemctl restart nginx
      6  hostnamectl set-hostname rao
      7  exec bash
      8  cd
      9  history
[root@rao ~]# ==■

```

- 23) Take rao sever public ip and paste in google you get this is rao sever



- 24) Now connect venkat server install nginx and set html file this is venkat server take venkat sever public ip paste in google you get this is venkat server

Activities Terminal Jul 17 17:18 root@ip-172-31-3-151:~

```
challa@challa-HP-Laptop-15-daxxxx:~/Downloads$ ssh -i "venkat.pem" ec2-user@ec2-18-191-172-123.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-18-191-172-123.us-east-2.compute.amazonaws.com (18.191.172.123)' can't be established.
ED25519 key fingerprint is SHA256:ST4mZZ9hKnaFQcM4cmOEM/LW1QjrtSeZWKnsvHog9kk.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-191-172-123.us-east-2.compute.amazonaws.com' (ED25519) to the list of known hosts.

[ec2-user@ip-172-31-3-151 ~]$ sudo -i
[ec2-user@ip-172-31-3-151 ~]# hostnamectl set-hostname venkat
Unknown command verb sey-hostname.
[ec2-user@ip-172-31-3-151 ~]# hostnamectl set-hostname venkat
[ec2-user@ip-172-31-3-151 ~]# exec bash
[ec2-user@ip-172-31-3-151 ~]# yum install nginx
Last metadata expiration check: 0:14:29 ago on Wed Jul 17 11:33:55 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-mimetypes   noarch        2.1.49-3.amzn2023.0.3  amazonlinux   21 k
```

25)

The screenshot shows a Linux desktop environment with a terminal window and a web browser window.

**Terminal Window:**

```

Activities Terminal Jul 17 17:20
root@ip-172-31-3-151:/usr/share/nginx/html%
root@ip-172-31-0-100:~% cd /usr/share/nginx/html/
root@venkat html% rm -f index.html
root@venkat html% nano index.html
root@venkat html% systemctl restart nginx
root@venkat html% systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
  Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; preset: disabled)
  Active: active (running) since Wed 2024-07-17 11:50:18 UTC; 11s ago
    Process: 25927 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
    Process: 25928 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
    Process: 25929 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
      Main PID: 25930 (nginx)
        Tasks: 2 (limit: 1114)
       Memory: 2.2M
          CPU: 56ms
        CGroup: /system.slice/nginx.service
            └─25930 "nginx: master process /usr/sbin/nginx"
              ├─25931 "nginx: worker process"

Jul 17 11:50:18 venkat systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Jul 17 11:50:18 venkat nginx[25928]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Jul 17 11:50:18 venkat nginx[25928]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Jul 17 11:50:18 venkat systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[root@venkat html]# history
 1 hostnamectl set-hostname venkat
 2 hostnamectl set-hostname venkat
 3 exec bash
 4 yum install nginx
 5 clear
 6 cd /usr/share/nginx/html/
 7 rm -f index.html
 8 nano index.html
 9 systemctl restart nginx
10 systemctl status nginx
11 history
[root@venkat html]#

```

**Web Browser Window:**

The browser window shows the URL `18.191.172.123`. The page content is:

```
this is venkat server
```

- 26) Go to target group click create target group select instances
- 27) name=rao-tg click next select rao and venkat server click include as pending below click create target group

The screenshot shows the AWS EC2 Target Groups page. In the top right, there is a search bar with the placeholder "[Alt+S]". Below it, a table titled "Available instances (2/2)" lists two instances: "venkat" (Instance ID: i-04ed85e0744b40f87) and "rao" (Instance ID: i-0f530c31e37bcecc0). Both instances are marked as "Running" and belong to the security group "rao-sg". A message at the bottom indicates "2 selected". Below the table, a section for "Ports for the selected Instances" shows a dropdown menu set to "80". A note says "1-65535 (separate multiple ports with commas)". A button labeled "Include as pending below" is present. At the bottom, a message says "2 selections are now pending below. Include more or register targets when ready." A "Review targets" button is located at the bottom left.

The screenshot shows the AWS EC2 Target groups page. The left sidebar has a tree view with "Instances" expanded, showing "Instances", "Instance Types", "Launch Templates", "Spot Requests", "Savings Plans", "Reserved Instances", "Dedicated Hosts", and "Capacity Reservations". The main area shows a table titled "Target groups (1/1) Info" with one entry: "rao-tg". The table columns are "Name", "ARN", "Port", "Protocol", and "Target type". The "Details" tab is selected in the "Target group: rao-tg" modal, which displays the ARN "arn:aws:elasticloadbalancing:us-east-2:058264331590:targetgroup/rao-tg/56d9f58391234ace" and the configuration: "Target type: Instance", "Protocol: Port HTTP: 80", "Protocol version: HTTP1", and "VPC: vpc-0a765047dc8eb3ddd".

- 28) Now go to load balancer click create load balancer select application load balancer click create set name= rao-lb click internet facing and ipv4 in mapping select both zones and sg= rao-sg
- 29) listeners and routing= rao-tg click create load balancer

Activities Google Chrome Jul 17 17:27

Create application load balancer RouteTables | VPC Cons... 18.191.172.123

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

AWS Services Search [Alt+S]

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

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.

**Network mapping** Info

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

Activities Google Chrome Jul 17 17:28

Create application load balancer RouteTables | VPC Cons... 18.191.172.123

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

AWS Services Search [Alt+S]

Network mapping Info

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

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](#).

vpc-0a765047dc8eb3dd  
IPv4 VPC CIDR: 172.31.0.0/16

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  
subnet-03460885b87d72ce7 sub1

IPv4 address  
Assigned by AWS

us-east-2b (use2-az2)  
Subnet  
subnet-035fc222e3b18a9b5 sub2

IPv4 address  
Assigned by AWS

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

Activities Google Chrome Jul 17 17:28

Create application load balancer RouteTables | VPC Cons... 18.191.172.123

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

aws Services Search [Alt+S]

Security groups Info

A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can create a new security group.

Security groups

Select up to 5 security groups

rao-sg sg-04ee2f334e2f44d50 VPC: vpc-0a765047dc8eb3ddd

Listeners and routing Info

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

▼ Listener HTTP:80

Protocol Port Default action Info

HTTP : 80 Forward to rao-tg Target type: Instance, IPv4

HTTP 1-65535

Create target group

Listener tags - optional

Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

Add listener tag

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

The screenshot shows the AWS Lambda console with the following details:

- Function name:** HelloWorld
- Description:** A simple Lambda function that prints "Hello World" to the CloudWatch logs.
- Runtime:** Python 3.9
- Memory:** 128 MB
- Timeout:** 3 seconds
- Code:** A sample Python script is shown in the code editor.
- Test:** A test event is defined with the payload "Hello World".
- Logs:** CloudWatch Logs Insights link is provided.

Activities Google Chrome Jul 17 17:29

Load balancers | EC2 | us-east-2 RouteTables | VPC Cons... 18.191.172.123

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

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 Lifecycle Manager

Networking & Security

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

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 Zones	Type
rao-lb	rao-lb-314375763.us-east...	Provisioning	vpc-0a765047dc8eb3...	2 Availability Zones	application

Load balancer: rao-lb

Scheme Internet-facing	Hosted zone Z3AADJGX6KTTL2	Availability Zones subnet-03460885b87d72ce7 us-east-2a (use2-az1) subnet-035fc22e3b18a9b5 us-east-2b (use2-az2)	Date created July 17, 2024, 17:29 (UTC+05:30)
Load balancer ARN arn:aws:elasticloadbalancing:us-east-2:058264331590:loadbalancer/app/rao-lb/d25ab29e071d8e5	DNS name Info rao-lb-314375763.us-east-2.elb.amazonaws.com (A Record)		

30) Now go to waf

### 31) Select ip set click create ip set

The screenshot shows the AWS WAF & Shield IP sets page. The left sidebar has sections for AWS WAF (Getting started, Web ACLs, Bot control dashboard, Application integration, IP sets, Regex pattern sets, Rule groups, AWS Marketplace managed rules, Switch to AWS WAF Classic) and AWS Shield (Getting started). The main area is titled 'IP sets' and shows a table with one row: 'No IP sets found'. A note says 'You don't have any IP sets in the US East (N. Virginia) Region created with this latest version of AWS WAF.' Below it, a message says 'Resources created under AWS WAF Classic aren't compatible with the new AWS WAF.' A link 'If you are looking for web ACLs created in the past, please check the AWS WAF Classic console. Please click here' is provided. At the bottom right is a 'Create IP set' button.

32) name= rao-ipset

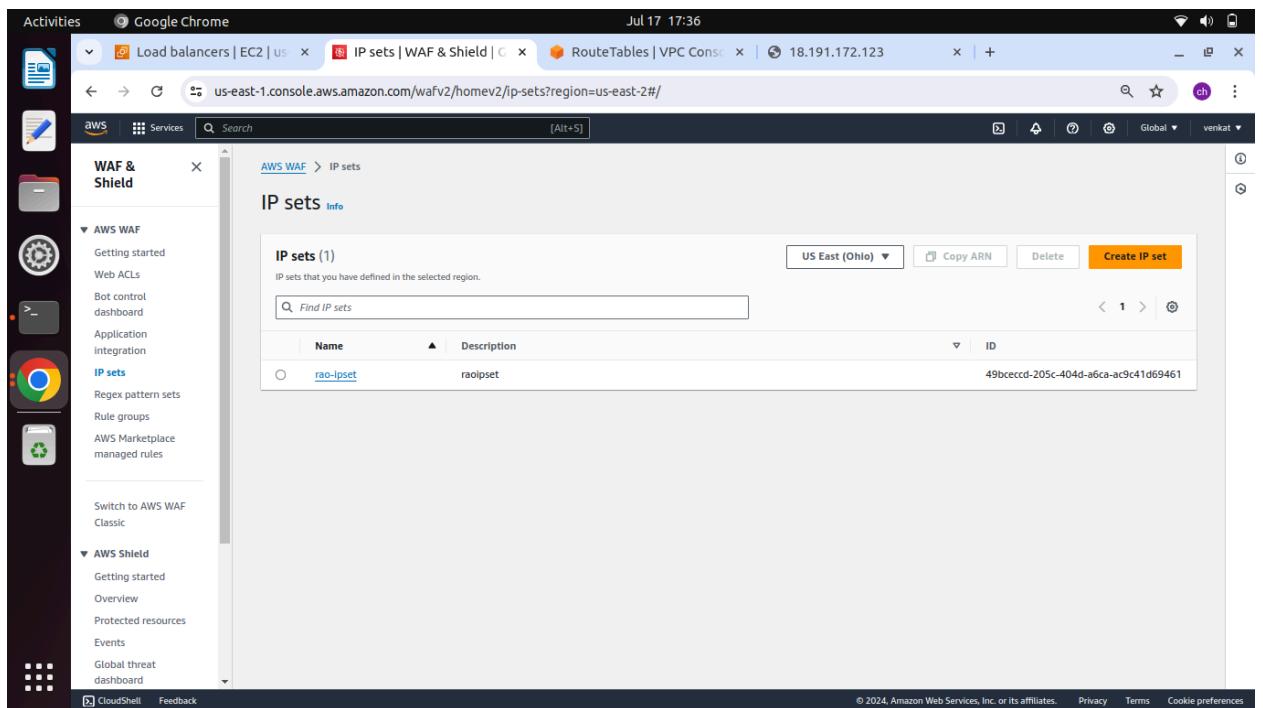
33) region= ohio

34) Ip version=1pv4

35) Give ip address you blocked ip=192.168.1.10/32

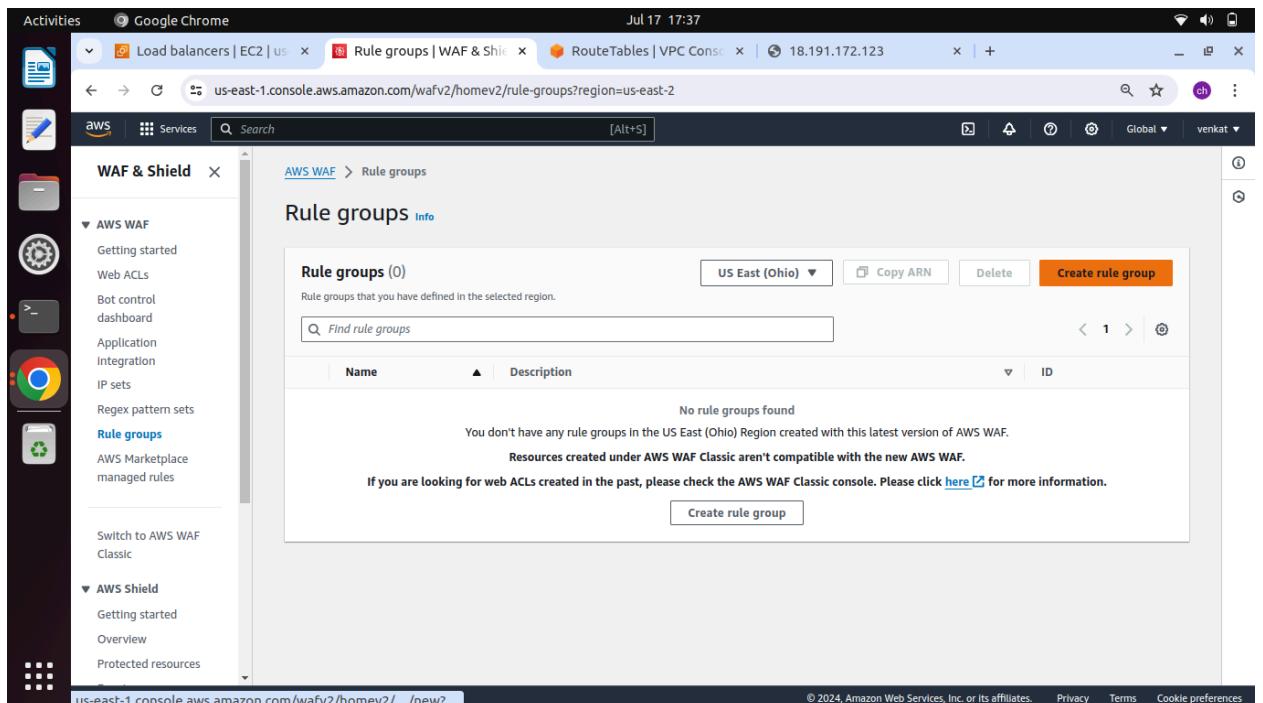
The screenshot shows the 'Create IP set' dialog box. It has fields for 'IP set name' (rao-ipset), 'Description - optional' (raoipset), 'Region' (US East (Ohio)), 'IP version' (IPv4 selected), and 'IP addresses' (192.168.1.10/32). At the bottom are 'Cancel' and 'Create IP set' buttons.

### 36) Click create ip set



The screenshot shows the AWS WAF IP sets page. The left sidebar has 'IP sets' selected under 'AWS WAF'. The main area shows 'IP sets (1)'. A table lists one item: Name 'rao-ipset' and Description 'raoipset'. A 'Create IP set' button is visible at the top right.

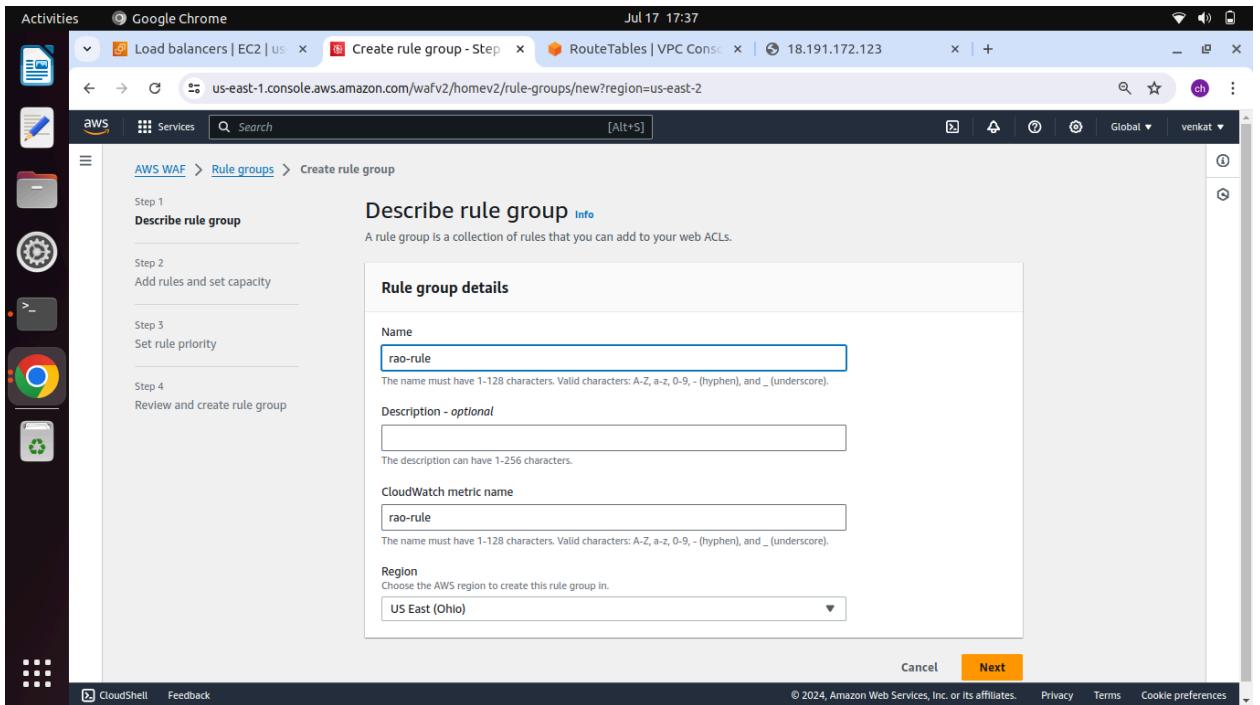
### 37) Click rule group click create rule group



The screenshot shows the AWS WAF Rule groups page. The left sidebar has 'Rule groups' selected under 'AWS WAF'. The main area shows 'Rule groups (0)'. A message states 'No rule groups found'. Below it, a note says 'Resources created under AWS WAF Classic aren't compatible with the new AWS WAF.' and 'If you are looking for web ACLs created in the past, please check the AWS WAF Classic console. Please click [here](#) for more information.' A 'Create rule group' button is at the bottom.

38) name=rao-rule

39) region= ohio



- 40) Next click add rule name=rao-rule
- 41) type=regular rule
- 42) inspect=originates from an ip address in
- 43) Ip set= rao-ipset
- 44) Select source ip address

45) action= block

The screenshot shows the AWS WAFV2 Rule Group creation interface. The current step is "Step 4: Review and create rule group". The rule name is "rao-rule", which is a Regular rule. The condition is "If a request matches the statement". The statement inspect "Originates from an IP address in IP set rao-ipset" and uses "Source IP address". The action is set to "Block". The "Add rule" button is visible at the bottom right.

The screenshot shows the AWS WAFV2 Rule Group creation interface. The current step is "Step 4: Review and create rule group". The rule name is "rao-rule", which is a Regular rule. The condition is "If a request matches the statement". The statement inspect "Originates from an IP address in IP set rao-ipset" and uses "Source IP address". The action is set to "Block". The "Add rule" button is visible at the bottom right.

46) Click add rule

47) Add rules and set capacity= select rao-rule click next

The screenshot shows the AWS WAF rule group creation process. The current step is "Add rules and set capacity". The "Rules" section contains one rule named "rao-rule" with a capacity of 1 and an action of "Block". The "Capacity" section shows a minimum required capacity of 1 and a capacity input field set to 1. Navigation buttons at the bottom include "Cancel", "Previous", and "Next".

48) Set rule priority= select rao-rule click next click rule group

The screenshot shows the AWS WAF Rule groups page. A success message indicates the rule group "rao-rule" was successfully created. The "Rule groups" table lists "rao-rule" with an ID of f9c41d81-825b-4552-9c61-14130fe39082. The left sidebar shows the "Rule groups" option under the AWS WAF section is selected.

49) Click web acl set region= ohio now click crete web acl

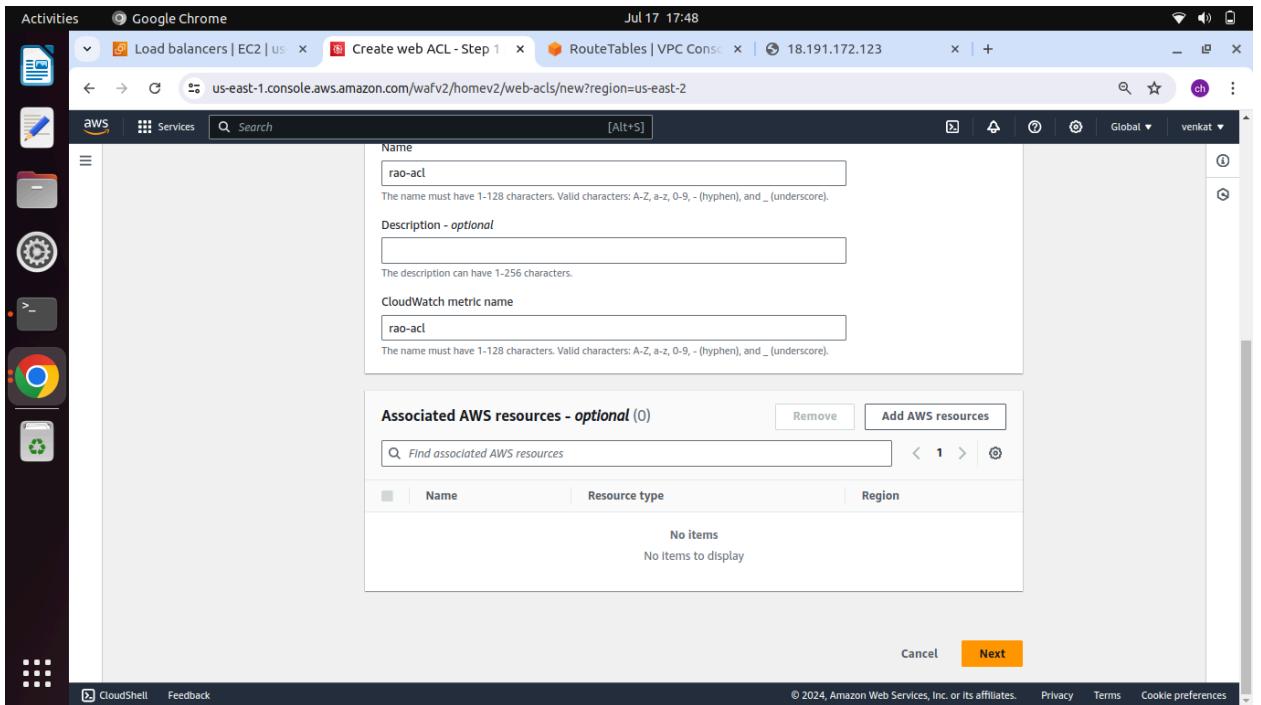
The screenshot shows the AWS WAF Web ACLs page. The left sidebar is titled 'AWS WAF & Shield' and includes sections for 'AWS WAF' (Getting started, Web ACLs, Bot control dashboard, Application integration, IP sets, Regex pattern sets, Rule groups, AWS Marketplace managed rules) and 'AWS Shield' (Switch to AWS WAF Classic, Getting started, Overview, Protected resources). The main content area is titled 'Web ACLs' and shows a message: 'New AWS WAF dashboards are now available. Check them out by selecting any of your web ACLs.' Below this is a table header for 'Web ACLs (0)' with columns for Name, Description, and ID. A message below the table states: 'No web ACLs found. You don't have any web ACLs in the US East (Ohio) Region created with this latest version of AWS WAF. Resources created under AWS WAF Classic aren't compatible with the new AWS WAF. If you are looking for web ACLs created in the past, please check the AWS WAF Classic console. Please click [here](#) for more information.' At the bottom right of the table is a 'Create web ACL' button.

50) Select =Regional resources (Application Load Balancers, Amazon API Gateway REST APIs, Amazon App Runner services, AWS AppSync GraphQL APIs, Amazon Cognito user pools and AWS Verified Access Instances)

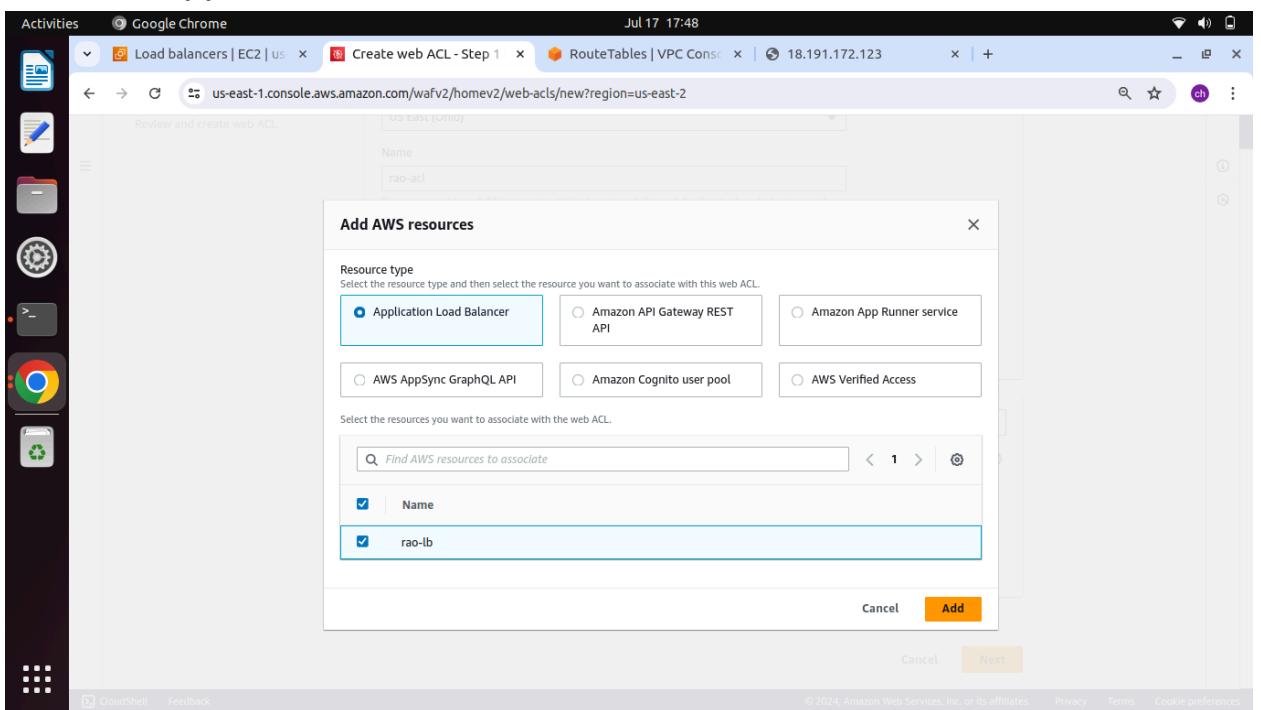
51) region= ohio

The screenshot shows the 'Create web ACL - Step 1' page. On the left, a sidebar lists steps: Step 1 (Describe web ACL and associate it to AWS resources), Step 2 (Add rules and rule groups), Step 3 (Set rule priority), Step 4 (Configure metrics), and Step 5 (Review and create web ACL). The main content area is titled 'Describe web ACL and associate it to AWS resources'. It has a 'Web ACL details' section with fields for 'Resource type' (set to 'Regional resources (Application Load Balancers, Amazon API Gateway REST APIs, Amazon App Runner services, AWS AppSync GraphQL APIs, Amazon Cognito user pools and AWS Verified Access Instances)'), 'Region' (set to 'US East (Ohio)'), 'Name' ('rao-acl'), 'Description - optional' (empty), and 'CloudWatch metric name' ('rao-acl').

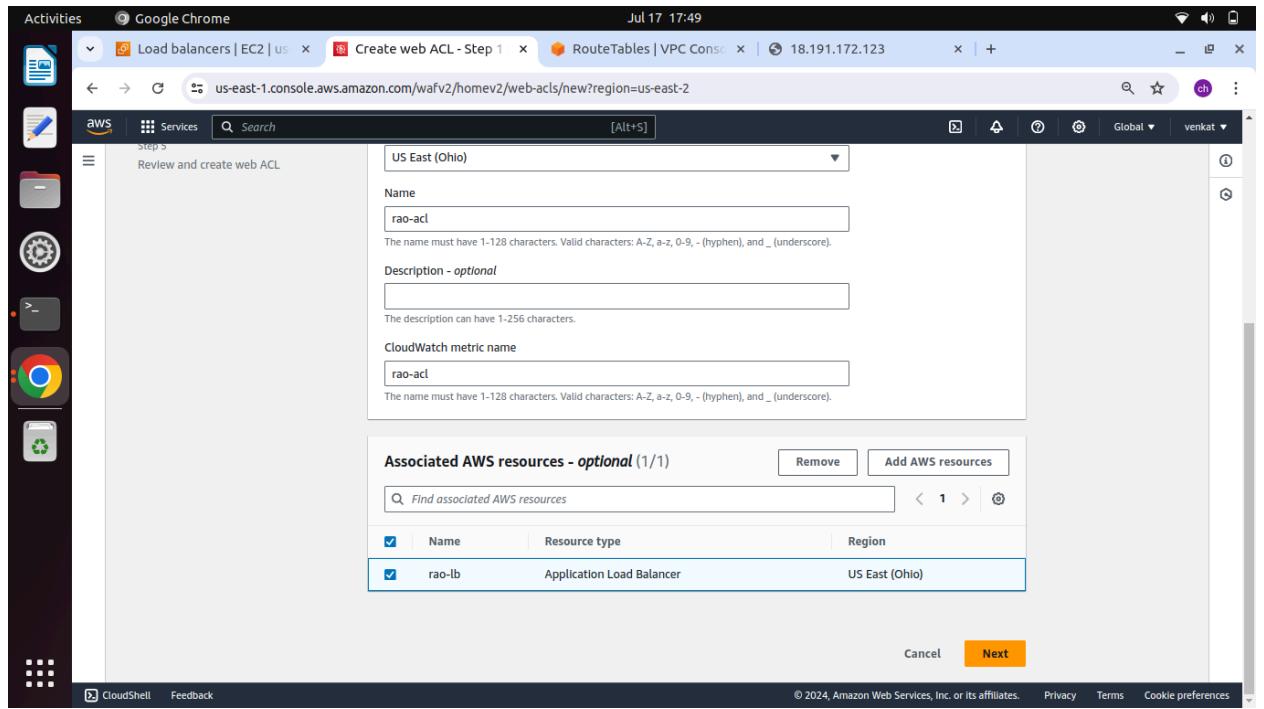
52) Click add aws resources



53) Select application load balancer and select rao-lb click add

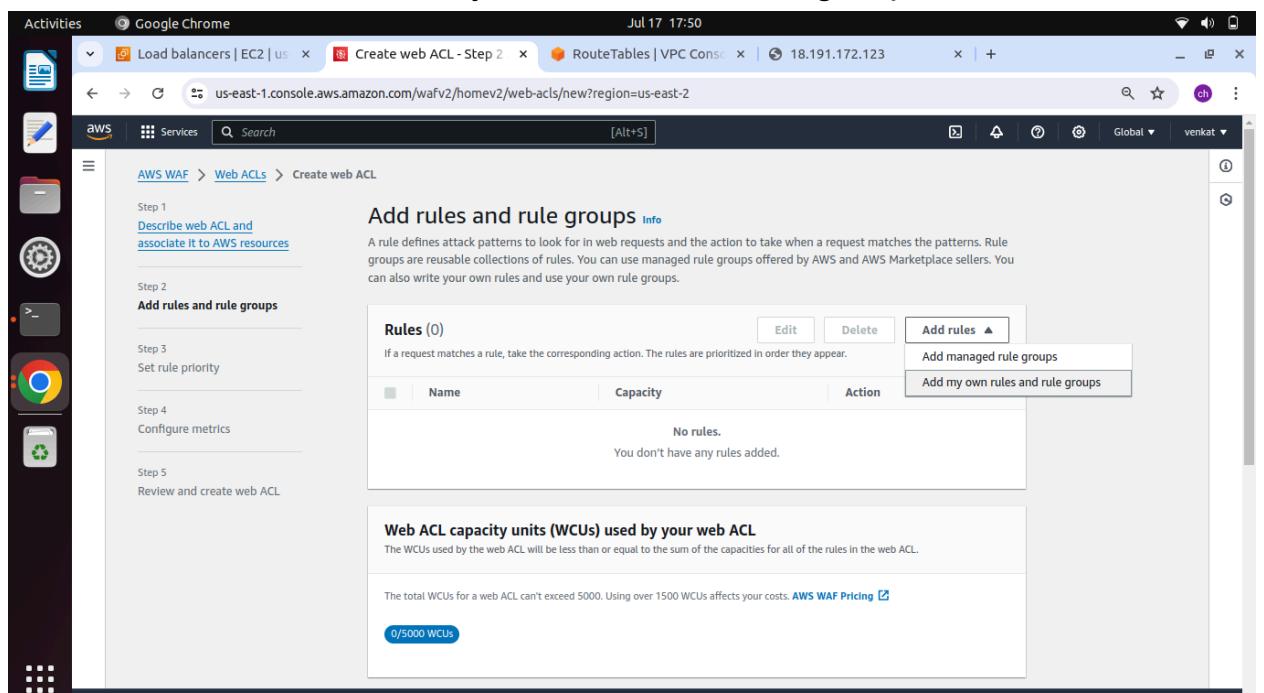


54) Now select rao-lb click next



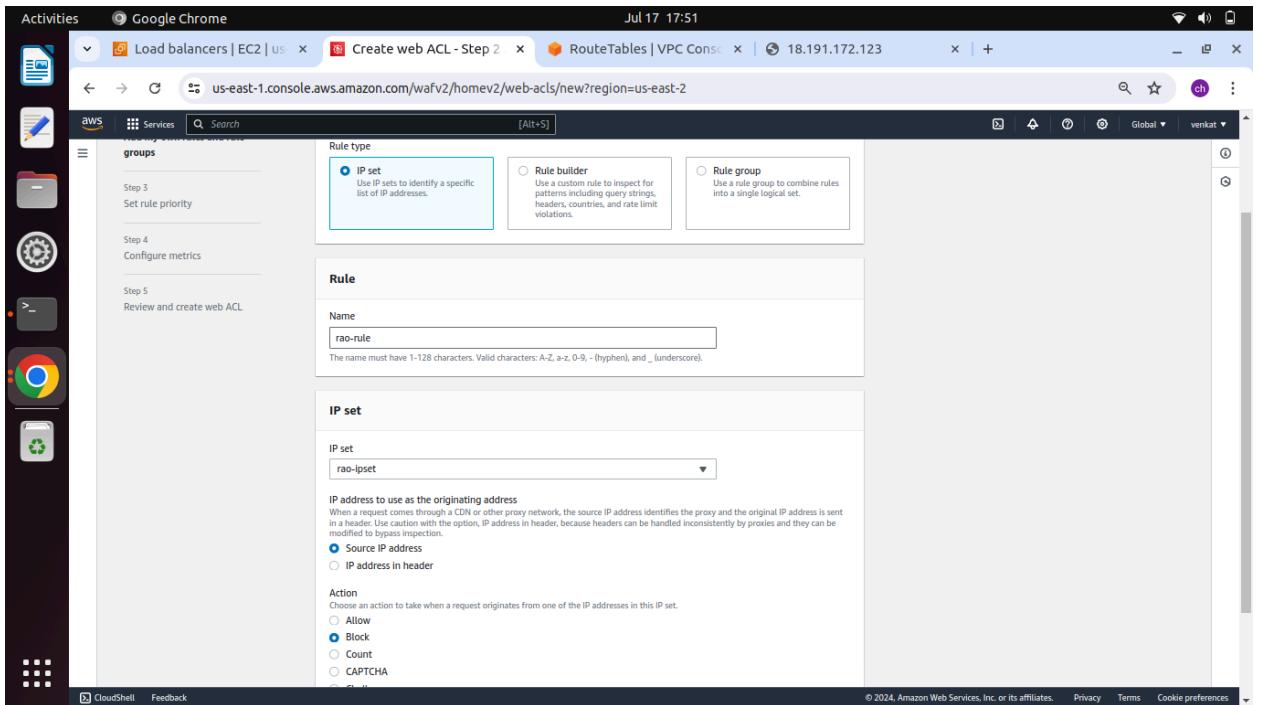
55)

56) Click add rules select add my own rules and rule groups

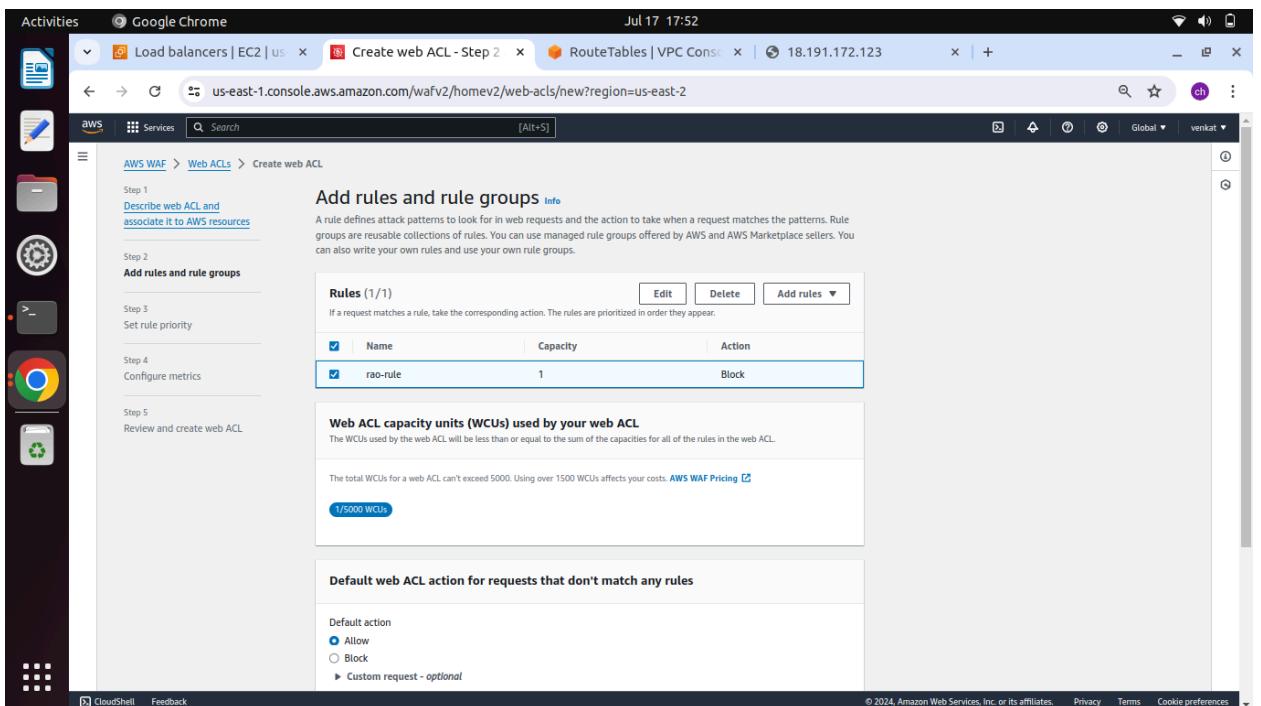


57)

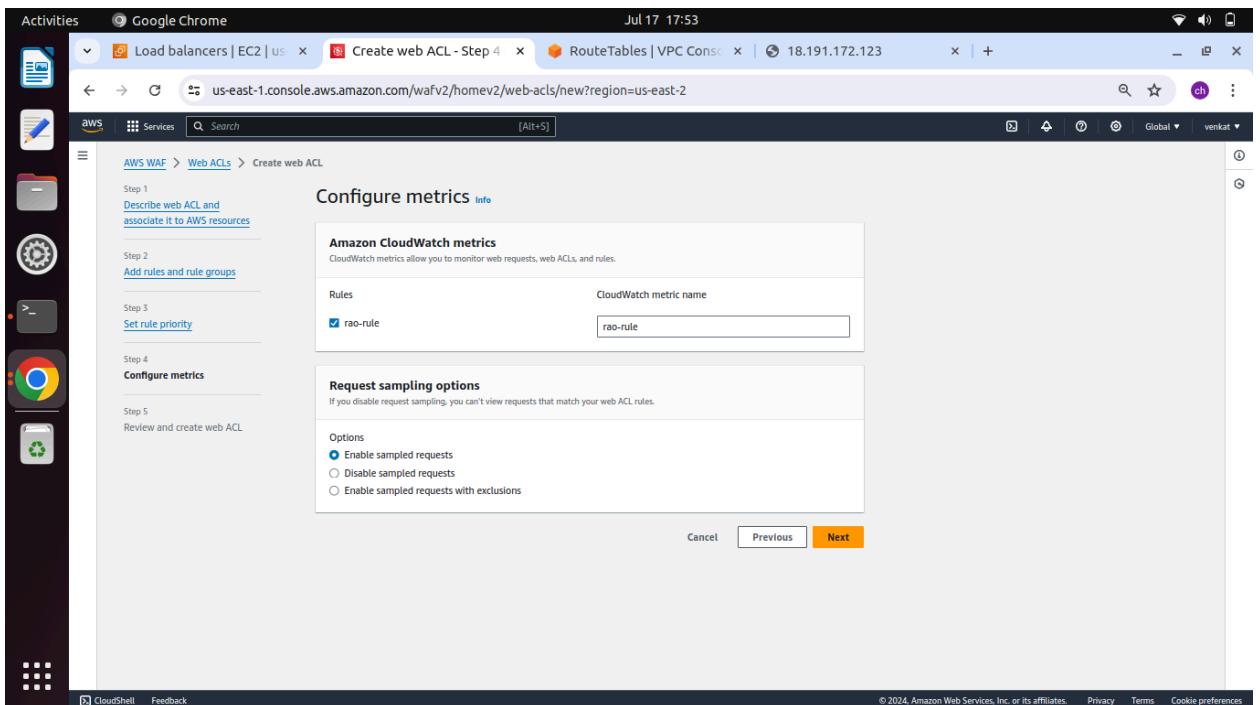
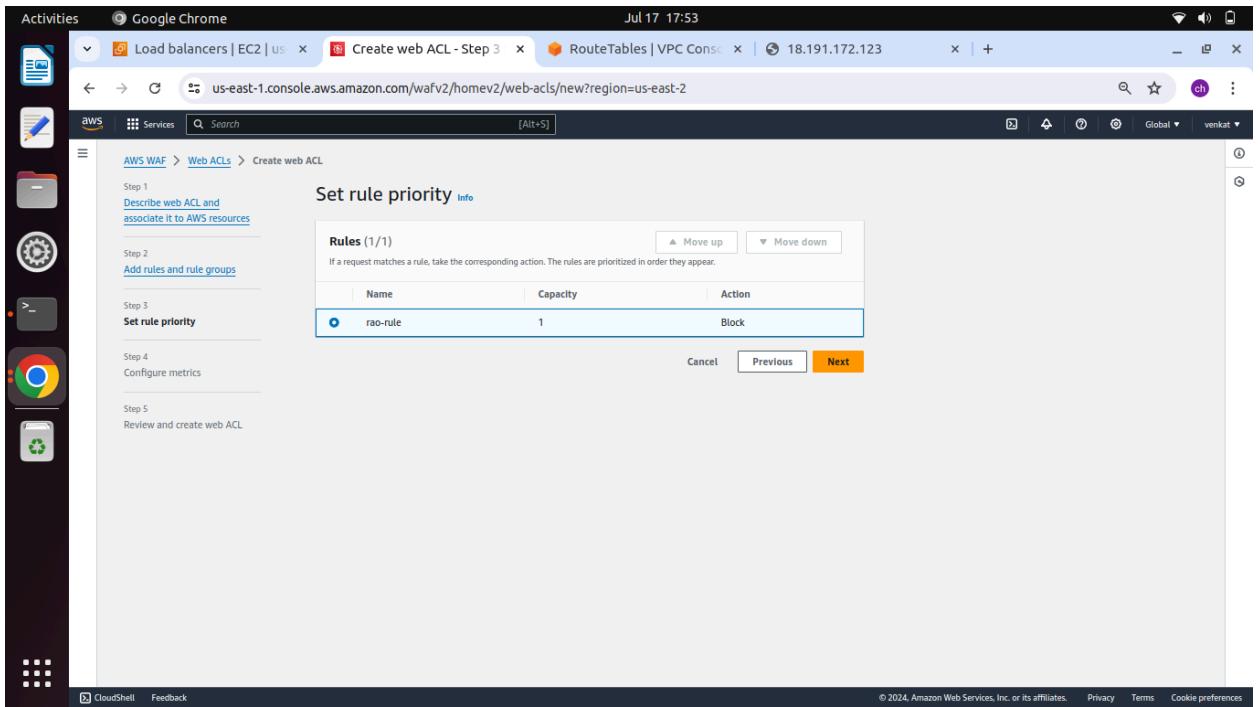
58) Select ip set and rule=rao-rule and ipset=rao-ip-set select source ip address and action= block click add rule



## 59) Select rule group =rao-rule



## 60) Set rule priority= rao-rule



61) Next and create web acl

62) Select rao acl and open it scroll down check

The screenshot shows two browser windows. The top window displays the 'Web ACLs' page under the 'AWS WAF' section. It lists one Web ACL named 'rao-acl' with the ARN 9a500f6d-be10-42ea-98ef-c7298a51dfb0. The bottom window shows a detailed view of the 'rao-acl' rule, specifically the 'Action totals for the specified time range - all traffic' section. This section includes four metrics: Total (0), Blocked (0), Allowed (0), and Captcha (0). Below this, there are sections for 'Challenge' (0) and 'Action totals' (Request counts for each selected terminating action, with a 'View in CloudWatch' link). To the right, there is a 'Top 10 rules' section (Requests counts for the ten rules that matched the most requests during the selected time range, with a 'View in CloudWatch' link).

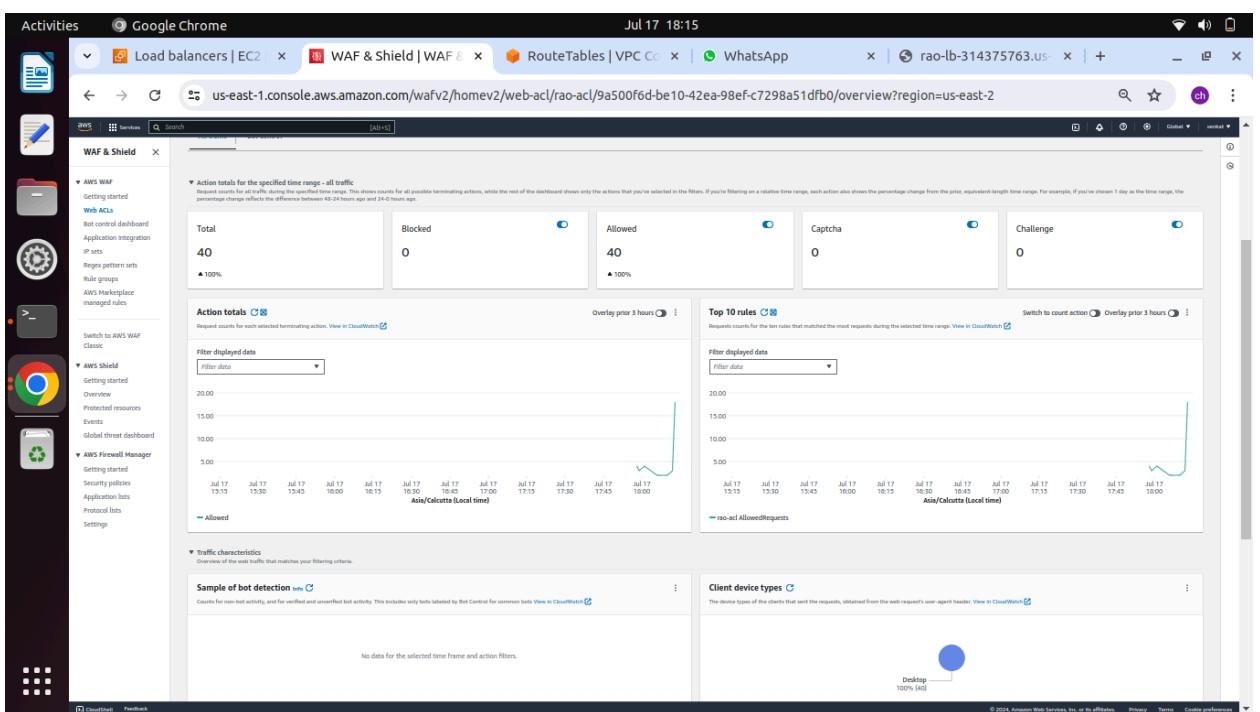
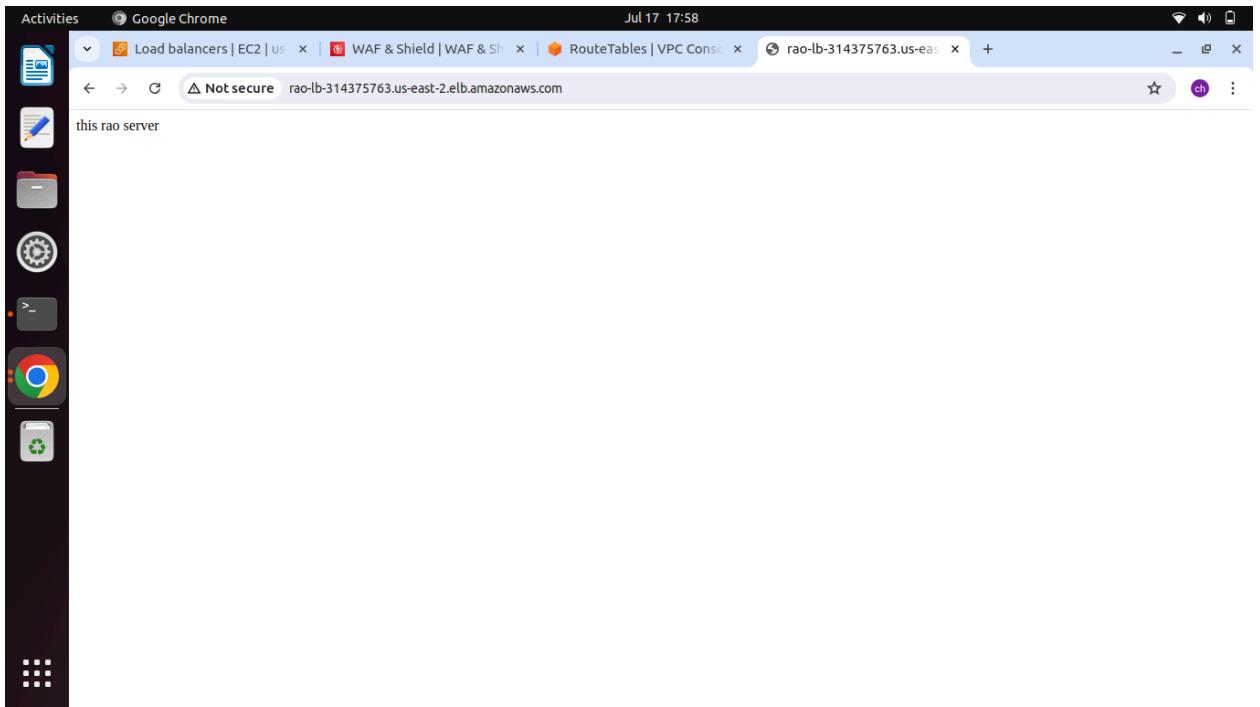
63) Go to load balancer copy rao-lb dns link paste in google and search and send friends check working or not after check ip user access or not

The screenshot shows the AWS EC2 Load Balancers console. On the left, there's a sidebar with various services like EC2 Dashboard, EC2 Global View, Instances, Images, Elastic Block Store, and Network & Security. The main area shows a table titled "Load balancers (1/1)" with one entry: "rao-lb". The table includes columns for Name, DNS name, State, VPC ID, Availability Zones, and Type. The "DNS name" column shows "rao-lb-314375763.us-east-2.elb.amazonaws.com (A Record)". A tooltip "DNS name copied" appears over this field. The "Availability Zones" column lists "subnet-03460085b87d72ce7 us-east-2a (use2-az1)" and "subnet-035fc22e3b18a9b5 us-east-2b (use2-az2)". The "Date created" field shows "July 17, 2024, 17:29 (UTC+05:30)".

The screenshot shows a web browser window with the URL "rao-lb-314375763.us-east-2.elb.amazonaws.com". The page content is "this is venkat server".

64)

65) Refresh once



Activities Google Chrome Jul 17 18:15

Load balancers | EC2 WAF & Shield | WAF RouteTables | VPC WhatsApp rao-lb-314375763.us

us-east-1.console.aws.amazon.com/wafv2/homev2/web-acl/rao-acl/9a500f6d-be10-42ea-98ef-c7298a51dfb0/overview?region=us-east-2

**WAF & Shield**

**AWS WAF**

- Getting started
- Web ACLs
- Not control dashboard
- Application integration
- IP sets
- Regex pattern sets
- Rule groups
- AWS Marketplace managed rules

Switch to AWS WAF Classic

**AWS Shield**

- Getting started
- Overview
- Protected resources
- Events
- Global threat dashboard

**AWS Firewall Manager**

- Getting started
- Security policies
- Application lists
- Protocol lists
- Settings

CloudWatch Feedback

Jul 17 18:15

2000

2000

Allowed

Jul 17 13:15 Jul 17 13:30 Jul 17 13:45 Jul 17 16:00 Jul 17 16:15 Jul 17 17:00 Jul 17 17:15 Jul 17 17:30 Jul 17 17:45 Jul 17 18:00 Asia/Calcutta (local time)

Jul 17 13:15 Jul 17 13:30 Jul 17 13:45 Jul 17 16:00 Jul 17 16:15 Jul 17 17:00 Jul 17 17:15 Jul 17 17:30 Jul 17 17:45 Jul 17 18:00 Asia/Calcutta (local time)

rao-acl AllowedRequests

Traffic characteristics

Sample of bot detection

No data for the selected time frame and action filters.

Client device types

Desktop 100% (48)

Attack types

No data for the selected time frame and action filters.

Top 10 countries

Filter displayed data

India 100% (48)

Netherlands 2%

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

The screenshot shows the AWS WAF & Shield console in Google Chrome. The left sidebar contains navigation links for AWS WAF, AWS Shield, and AWS Firewall Manager. The main content area displays various metrics and charts. The 'Client device types' section shows a single blue circle representing 'Desktop' at 100% (48). The 'Top 10 countries' section shows a bar chart with India at 100% (48) and Netherlands at 2%, both represented by blue bars.