

# BABU BANARSI DAS

# UNIVERSITY



DEPARTMENT OF COMPUTER SCIENCE  
ENGINEERING

SUMMER TRAINING & INTERNSHIP  
PROJECT

AN ORGANIZATIONAL SETUP

SESSION:2024-25

SUBMITTED TO:

MR. MUDIT

MATHUR

TRAINER AT IKIGAI

SUBMITTED BY:

SHIWANSH

SRIVASTAVA

B.TECH CSE 3<sup>rd</sup> year

1210432306 CS32

# **Acknowledgement**

First and foremost, I would like to express my deepest gratitude to my project supervisor, Mr. Mudit Sir, for their invaluable guidance, constant support, and encouragement throughout the course of this project. Their insightful feedback and expertise were instrumental in shaping this work.

I am also grateful to him and the faculty members of the Computer Science & Engineering Department at BBD University for their assistance and for providing a conducive environment for research and learning.

Finally, I would like to extend my appreciation to all those who, in one way or another, contributed to the successful completion of this project.

Thank you.

SHIWANSH SRIVASTAVA

## AN ORGANIZATIONAL SETUP

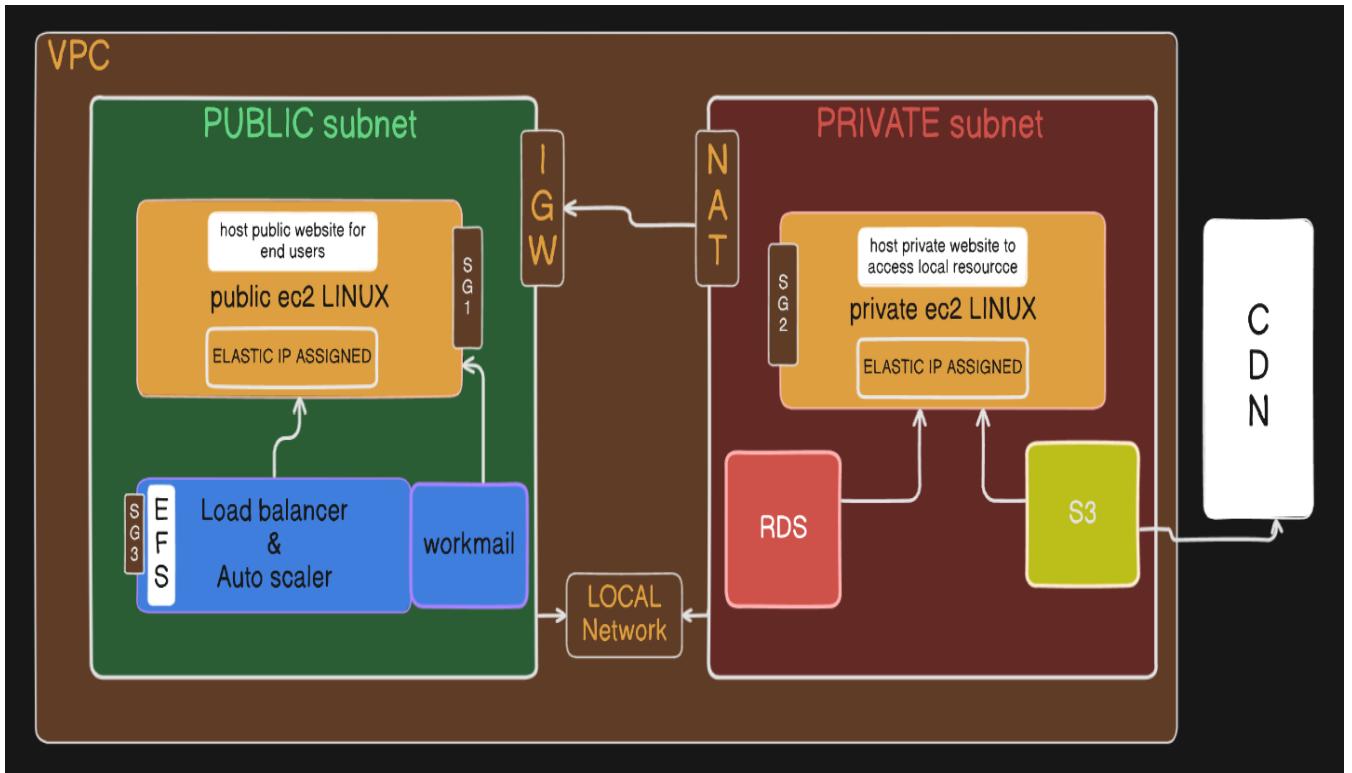
In this project, we are going to build an organisation which works on two networks: Public Network and Private Network .

The services available on public network can be accessed by the general audience but the services available on the private network are not available for general audiences and it can only be available for organization admins.

The platform that we are using is Amazon aws. We will be using some of the aws services like EC2, S3, RDS, AutoScalers etc.

As AWS is a paid platform, So as long as it's keep on functioning, it will induce charges for it.

# FRAMEWORK & COMPONENT



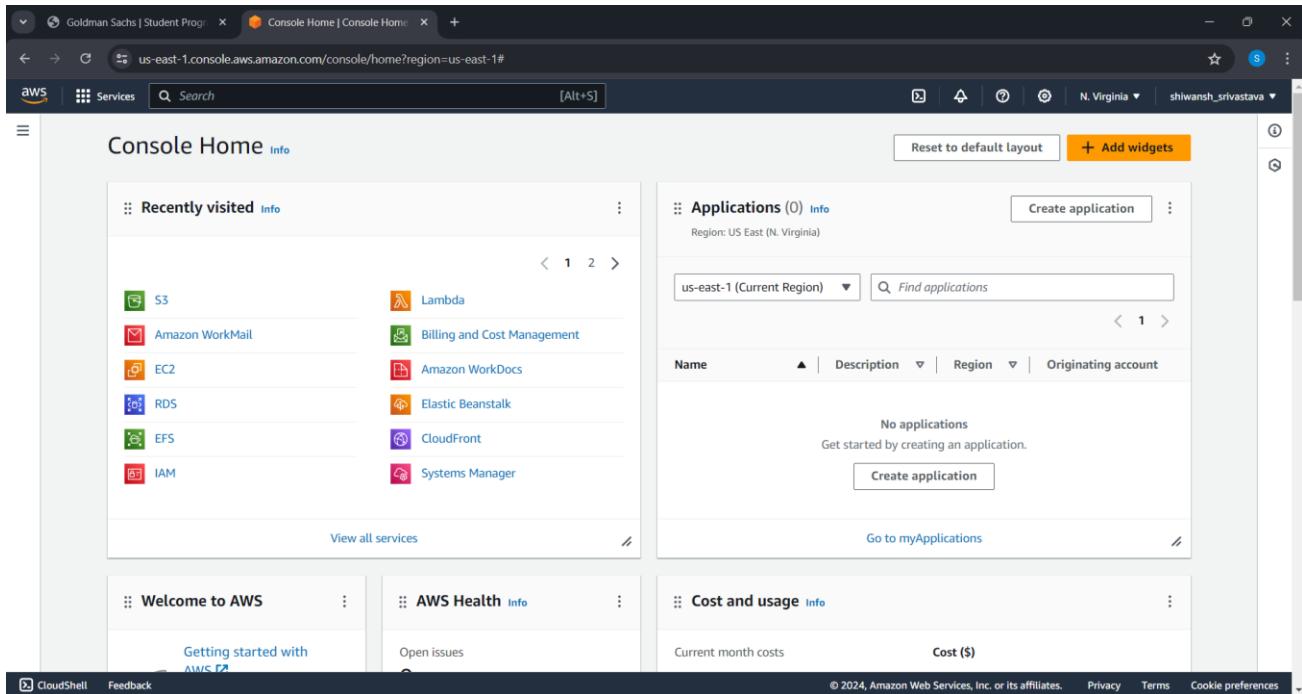
We are going to create this project on the basis of model shown above. It will contain the following components:-

- VPC
- EC2
- RDS
- S3
- CDN

# Designing VPC

To design VPC follow the given commands:

- Open console and search VPC.



- Click on create VPC.
- Now choose VPC and more.
- Choose the IP:10.0.0.0/16
- Choose at least 2 availability zones.
- Now select 2 public and private subnets
- Since we are making an organisational network, we are using larger subnets.

- Choose NAT gateway as none.
- Choose S3 gateway as VPC endpoints.
- Create VPC.

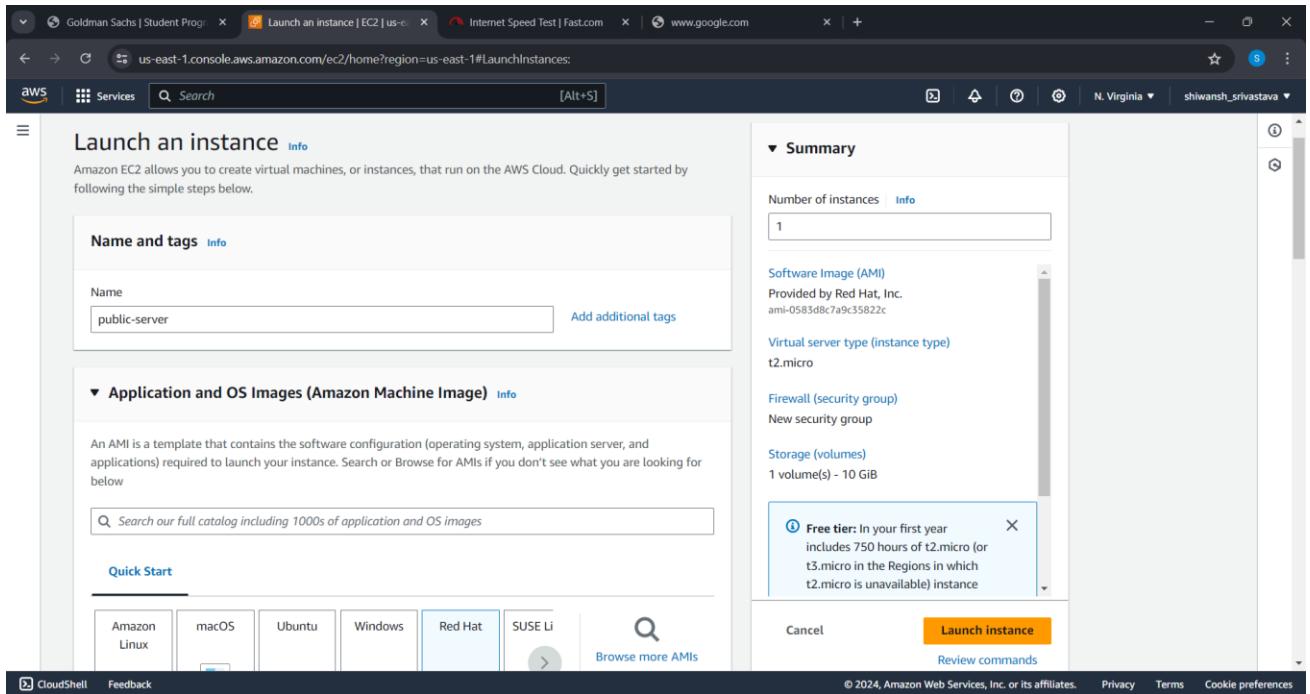
**Success**

**Details**

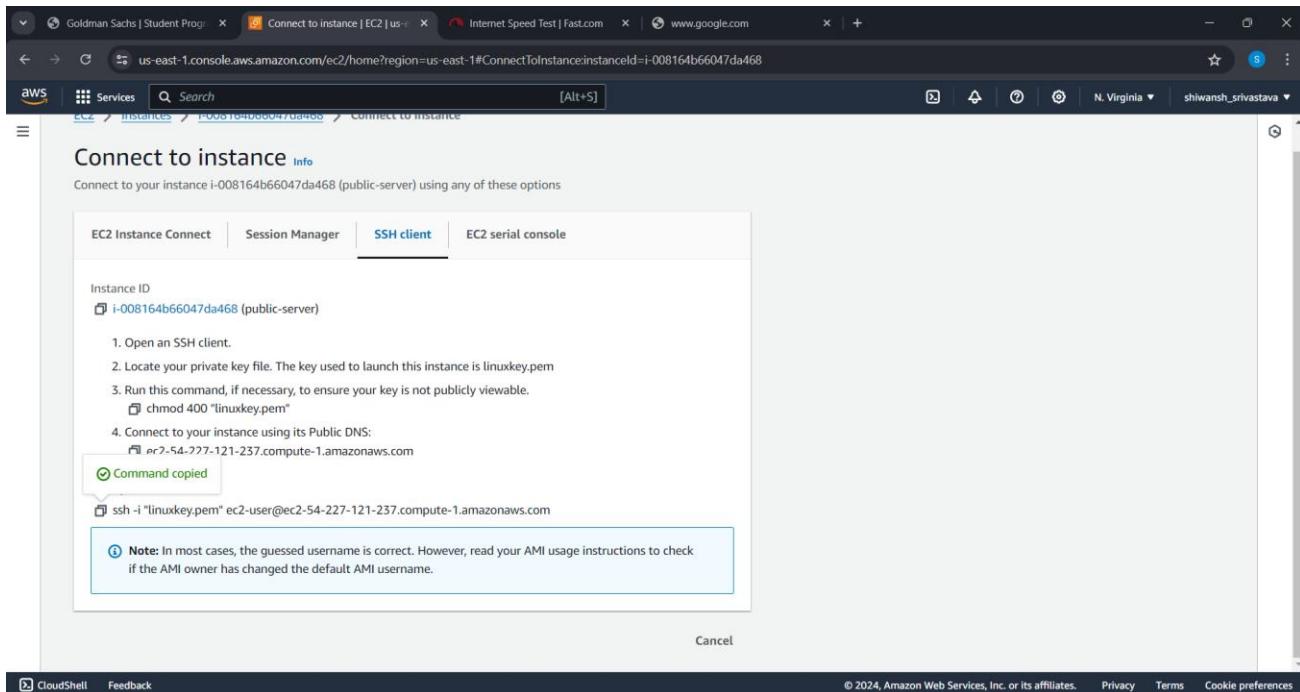
- Create VPC: vpc-02922df8acf7feac1
- Enable DNS hostnames
- Enable DNS resolution
- Verifying VPC creation: vpc-02922df8acf7feac1
- Create S3 endpoint: vpce-04d580e670b23ec67
- Create subnet: subnet-0dca37830292678d1
- Create subnet: subnet-04169a0dbccb7874f
- Create subnet: subnet-0b6baee48dc19c39f
- Create subnet: subnet-07b45b57c9f6720d0
- Create internet gateway: igw-088df45c930648f6d
- Attach internet gateway to the VPC
- Create route table: rtb-00d6d9c35b0dc0e04
- Create route
- Associate route table
- Associate route table
- Create route table: rtb-0c5125ef8aa4c4474
- Associate route table
- Create route table: rtb-0000dd00c71256301
- Associate route table
- Verifying route table creation
- Associate S3 endpoint with private subnet route tables: vpce-04d580e670b23ec67

# Creating a Public Instance

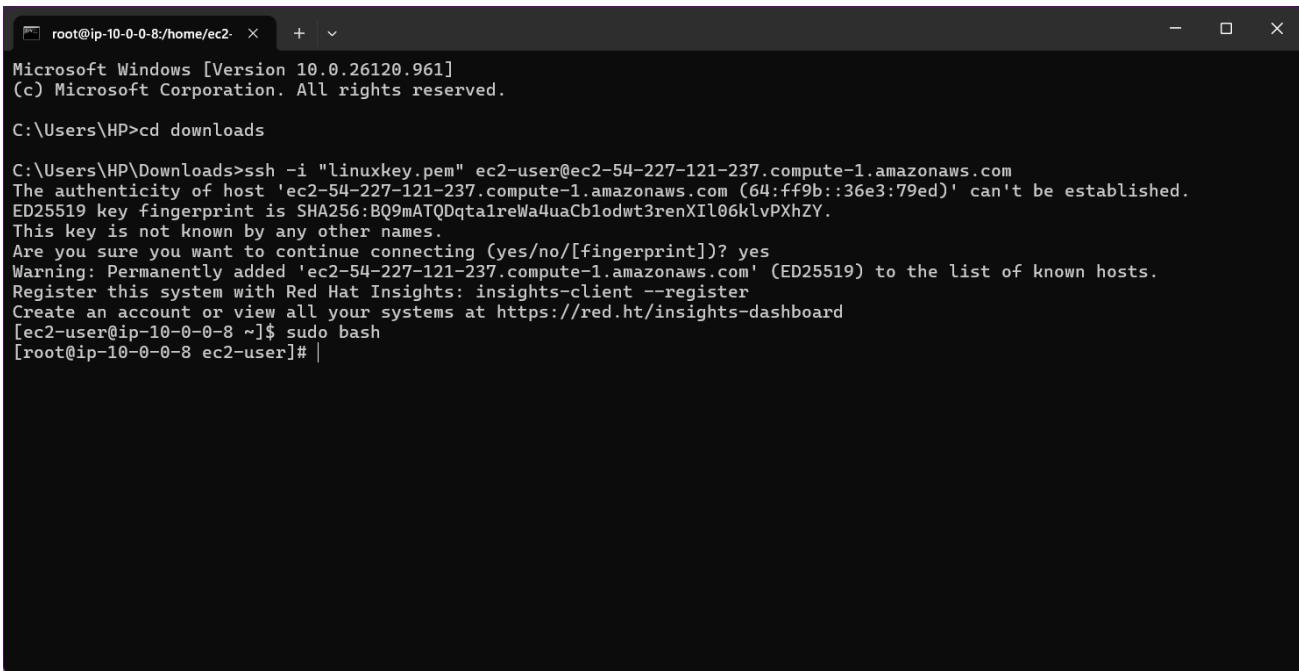
- Search EC2.
- Click on launch instance.



- Choose Linux OS.
- Select a key pair.
- In network configuration choose your project VPC and public subnet.
- Enable auto assign IP.
- And launch instance.
- Now click on connect instance.
- And choose SSH client.



- Copy the last command shown in the above image.
- Now open your systems terminal and write the following commands:
  1. cd downloads
  2. paste the copied command
  3. yes
  4. sudo bash
- Now you are in your instance.

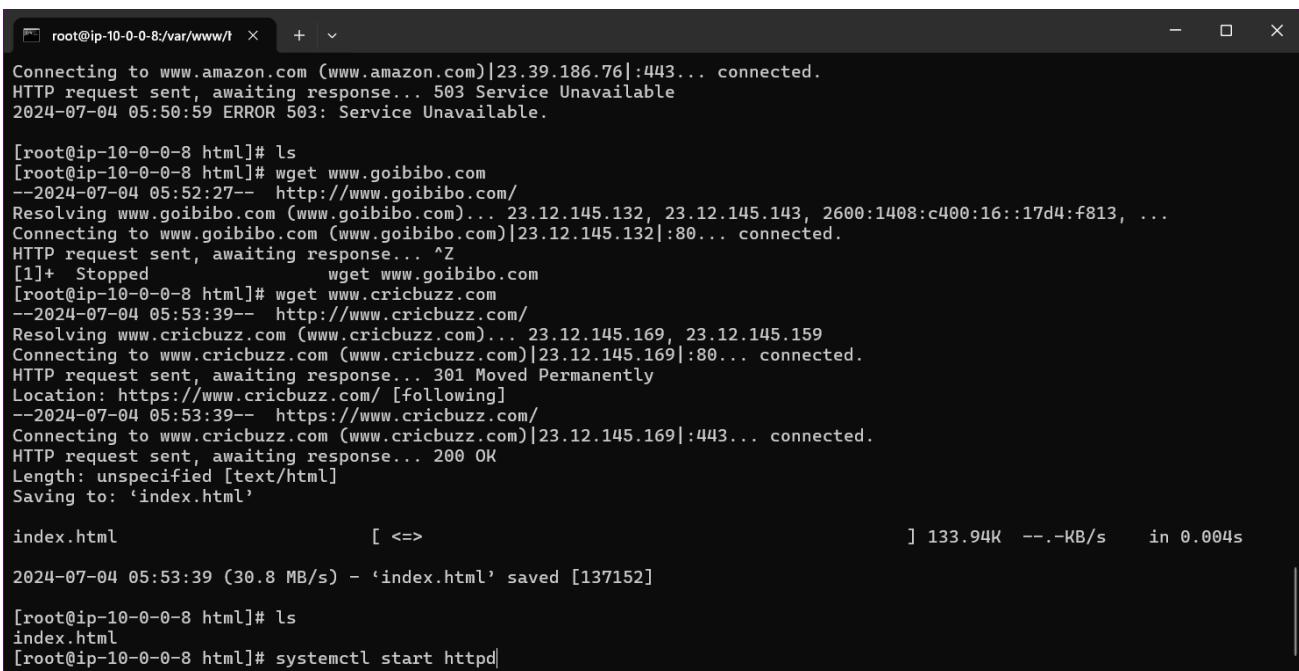


```
root@ip-10-0-0-8:/home/ec2-  × + ▾
Microsoft Windows [Version 10.0.26120.961]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>cd downloads

C:\Users\HP\Downloads>ssh -i "linuxkey.pem" ec2-user@ec2-54-227-121-237.compute-1.amazonaws.com
The authenticity of host 'ec2-54-227-121-237.compute-1.amazonaws.com (64:ff9b::36e3:79ed)' can't be established.
ED25519 key fingerprint is SHA256:BQ9mATQDqtaIreWa4uaCbldwt3renXIl06kLvPXhZY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-227-121-237.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
[ec2-user@ip-10-0-0-8 ~]$ sudo bash
[root@ip-10-0-0-8 ec2-user]#
```

- Now enter the following commands for instance configuration: yum install httpd -y; yum install wget\* -y; cd /var/www/html; wget [www.cricbuzz.com](http://www.cricbuzz.com);systemctl start httpd



```
root@ip-10-0-0-8:/var/www/html  × + ▾
Connecting to www.amazon.com (www.amazon.com)|23.39.186.76|:443... connected.
HTTP request sent, awaiting response... 503 Service Unavailable
2024-07-04 05:50:59 ERROR 503: Service Unavailable.

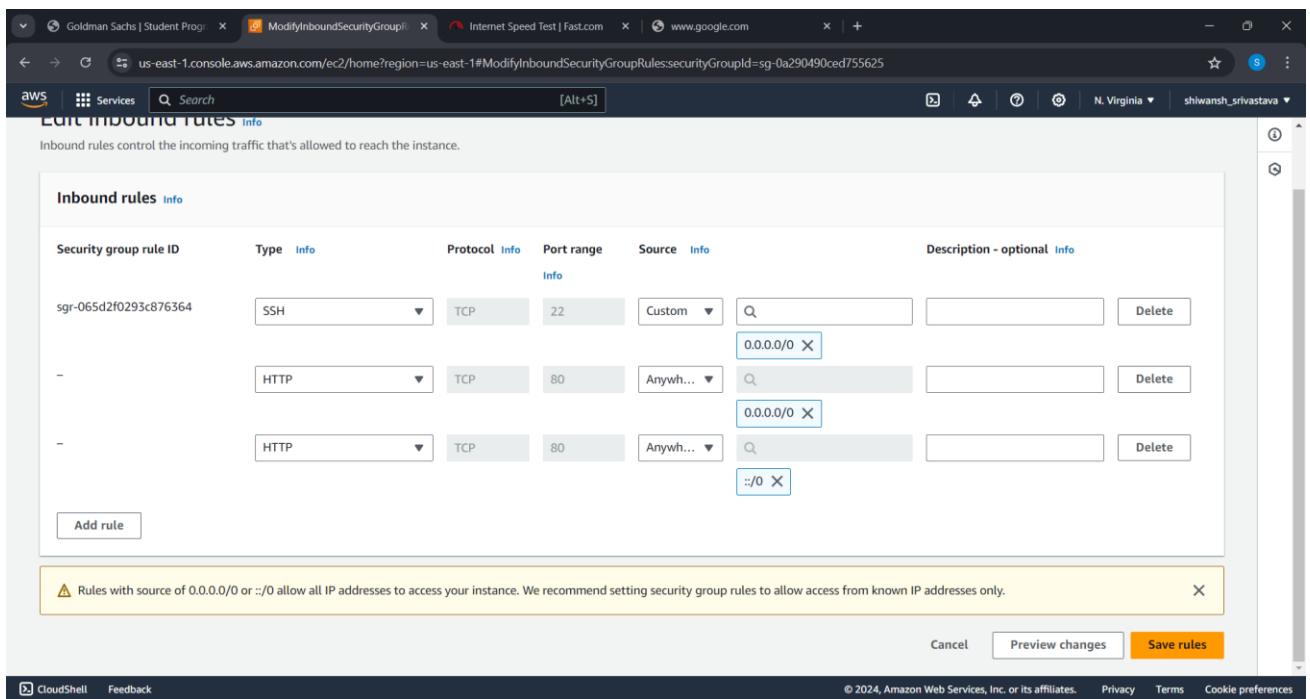
[root@ip-10-0-0-8 html]# ls
[root@ip-10-0-0-8 html]# wget www.goibibo.com
--2024-07-04 05:52:27-- http://www.goibibo.com/
Resolving www.goibibo.com (www.goibibo.com)... 23.12.145.132, 23.12.145.143, 2600:1408:c400:16::17d4:f813, ...
Connecting to www.goibibo.com (www.goibibo.com)|23.12.145.132|:80... connected.
HTTP request sent, awaiting response... ^Z
[1]+  Stopped                  wget www.goibibo.com
[root@ip-10-0-0-8 html]# wget www.cricbuzz.com
--2024-07-04 05:53:39-- http://www.cricbuzz.com/
Resolving www.cricbuzz.com (www.cricbuzz.com)... 23.12.145.169, 23.12.145.159
Connecting to www.cricbuzz.com (www.cricbuzz.com)|23.12.145.169|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://www.cricbuzz.com/ [following]
--2024-07-04 05:53:39-- https://www.cricbuzz.com/
Connecting to www.cricbuzz.com (www.cricbuzz.com)|23.12.145.169|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html'

index.html [ => ] 133.94K --.-KB/s   in 0.004s

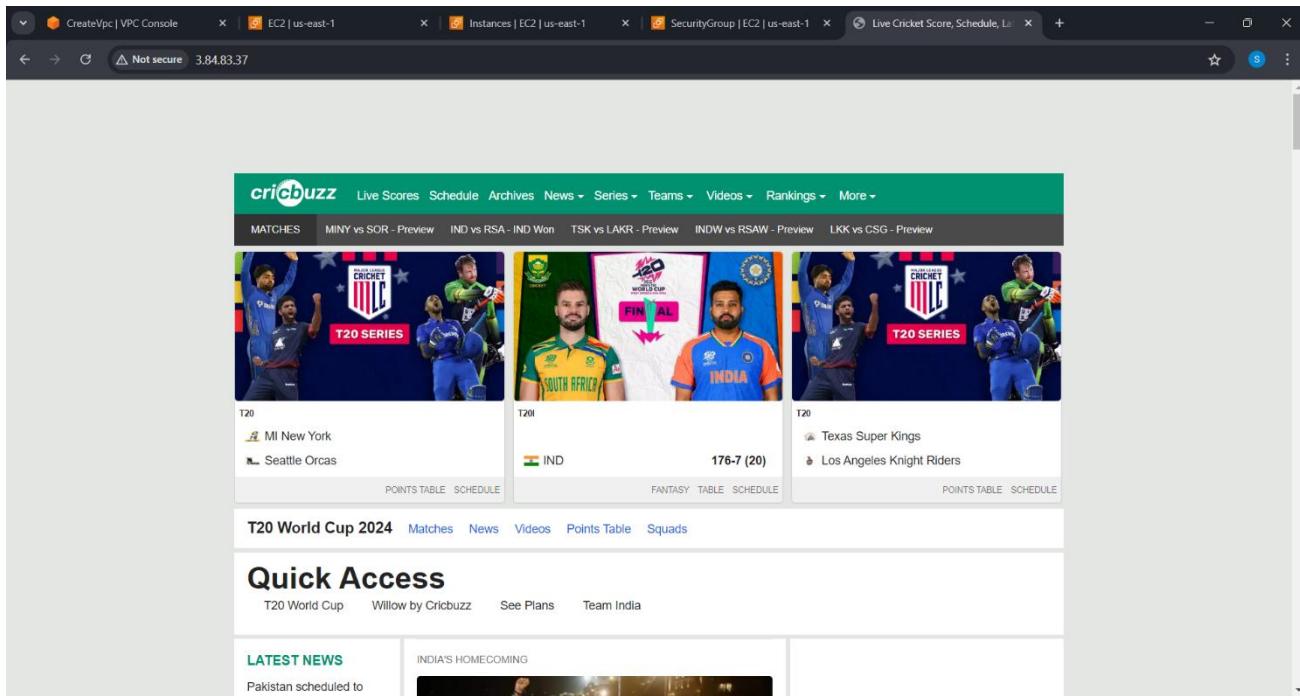
2024-07-04 05:53:39 (30.8 MB/s) - 'index.html' saved [137152]

[root@ip-10-0-0-8 html]# ls
index.html
[root@ip-10-0-0-8 html]# systemctl start httpd
```

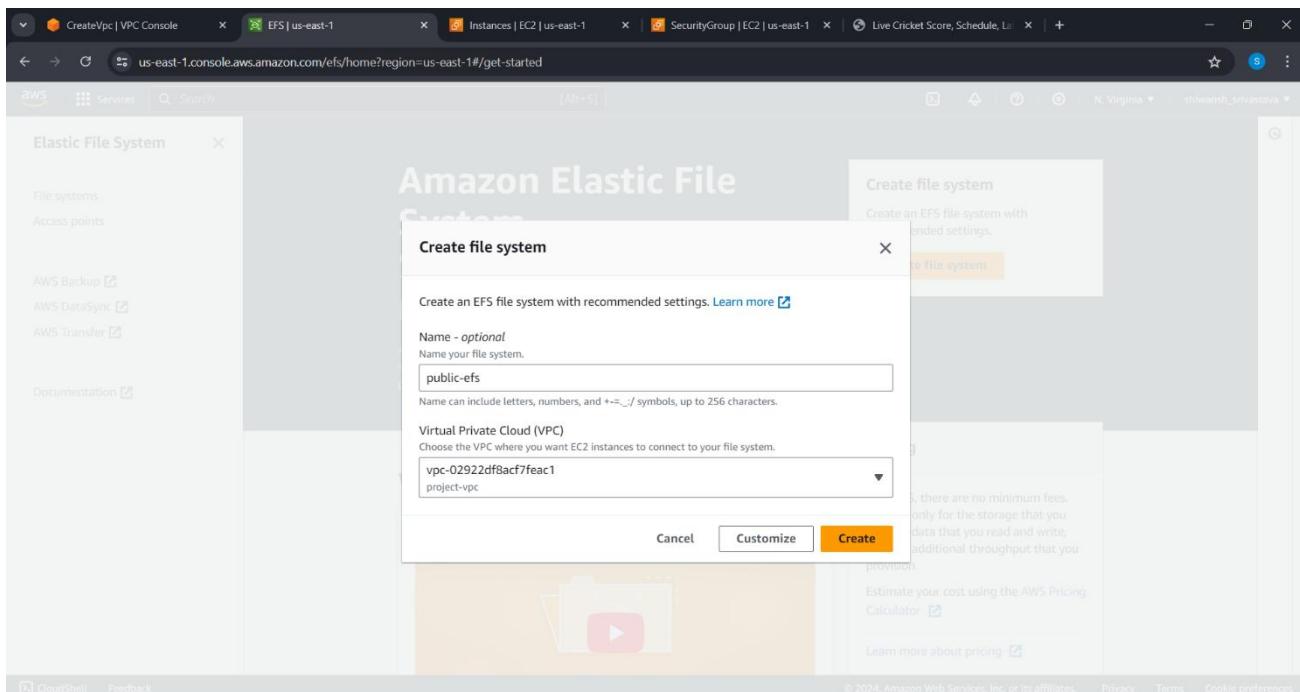
- Now go you AWS console and open your instance security group.
- There add new inbound rules for HTTP for all IPv4 and IPv6.



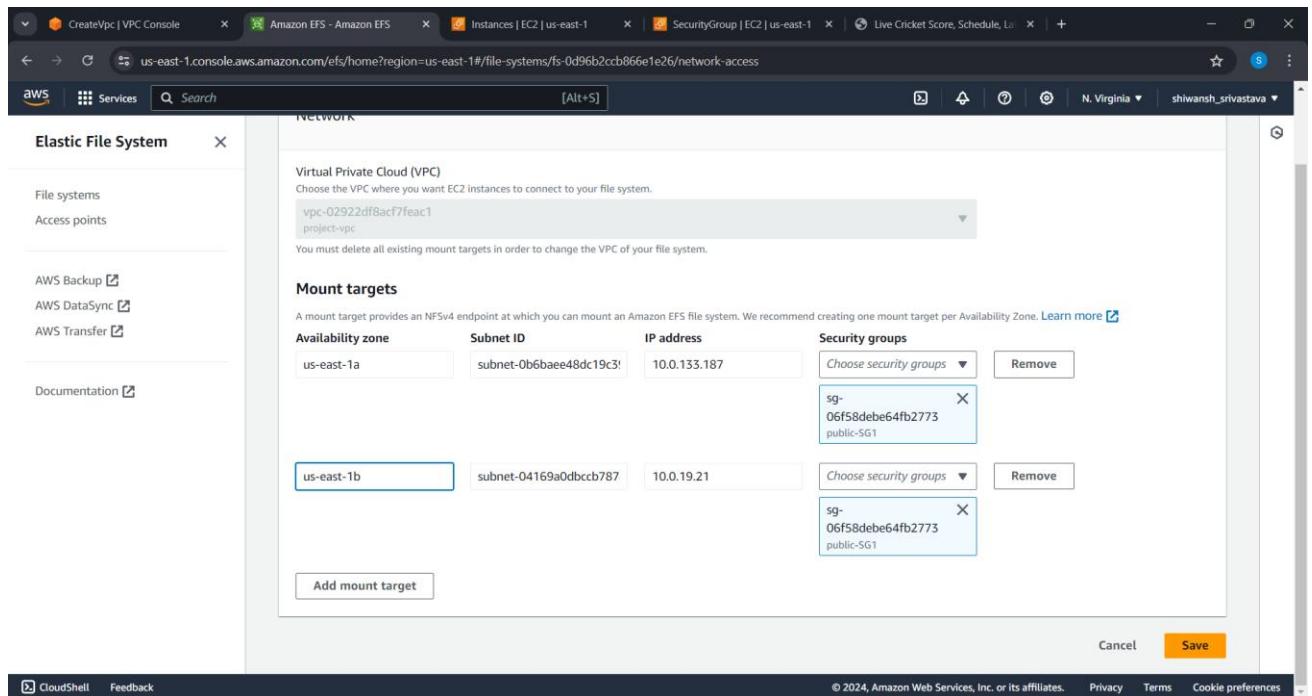
- Now copy the public address of you instance and check whether your website is working properly or not.



- Now you have to connect EFS to your instance.
- Search EFS and click on create



- Choose your project VPC while creating it.
- After creating efs in their network add your instance security group instead of default security group.



- Now add new inbound rule in your security group for port 2049 for all IPv4 and IPv6.

The screenshot shows the AWS EC2 Inbound Rules configuration page. It lists several security group rules:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-09e6a41f8f3b60b7c	HTTP	TCP	80	Custom	::/0
sgr-071911985dac07fc7	HTTP	TCP	80	Custom	0.0.0.0/0
sgr-065d2f0295c876364	SSH	TCP	22	Custom	0.0.0.0/0
-	Custom TCP	TCP	2049	Anyw...	0.0.0.0/0
-	Custom TCP	TCP	2049	Anyw...	::/0

At the bottom left is a "Add rule" button.

- Now open your efs and click on attach and copy the NFS client command.

The screenshot shows the AWS EFS Attach dialog box. It displays a success message: "Submitted all mount target changes successfully for file system (fs-0d96b2ccb866e1e26)". Below this, there are two options: "Mount via DNS" (selected) and "Mount via IP".

Using the EFS mount helper:

```
sudo mount -t efs -o tls fs-0d96b2ccb866e1e26:/ efs
```

Using the NFS client:

```
sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-0d96b2ccb866e1e26.efs.us-east-1.amazonaws.com:/ efs
```

See our user guide for more information. [Learn more](#)

Close

- Now go to your instance terminal and write the commands: yum install nfs\* -y;service nfs- utils start; paste the command and remove sudo from start and in end instead of efs write /mnt
- To auto install all the services in boot situation write: systemctl httpd on

```
root@ip-10-0-15-242:~# yum install nfs* -y
Running scriptlet: nfs4-acl-tools-0.4.2-3.el9.x86_64                                         14/14
Verifying   : nfs-utils-coreos-1:2.5.4-25.el9.x86_64                                         1/14
Verifying   : nfsv4-client-utils-1:2.5.4-25.el9.x86_64                                     2/14
Verifying   : libev-4.33-5.el9.x86_64                                                 3/14
Verifying   : libverto-libev-0.3.2-3.el9.x86_64                                         4/14
Verifying   : quota-1:4.06-6.el9.x86_64                                              5/14
Verifying   : quota-nls-1:4.06-6.el9.noarch                                         6/14
Verifying   : keyutils-1.6.3-1.el9.x86_64                                         7/14
Verifying   : gssproxy-0.8.4-6.el9.x86_64                                         8/14
Verifying   : libnfsidmap-1:2.5.4-25.el9.x86_64                                         9/14
Verifying   : libtirpc-1.3.3-8.el9_4.x86_64                                         10/14
Verifying   : nfs-utils-1:2.5.4-25.el9.x86_64                                         11/14
Verifying   : nfs4-acl-tools-0.4.2-3.el9.x86_64                                         12/14
Verifying   : rpcbind-1.2.6-7.el9.x86_64                                              13/14
Verifying   : sssd-nfs-idmap-2.9.4-6.el9_4.x86_64                                         14/14
Installed products updated.

Installed:
  gssproxy-0.8.4-6.el9.x86_64          keyutils-1.6.3-1.el9.x86_64          libev-4.33-5.el9.x86_64
  libnfsidmap-1:2.5.4-25.el9.x86_64    libtirpc-1.3.3-8.el9_4.x86_64     libverto-libev-0.3.2-3.el9.x86_64
  nfs-utils-1:2.5.4-25.el9.x86_64      nfs-utils-coreos-1:2.5.4-25.el9.x86_64  nfs4-acl-tools-0.4.2-3.el9.x86_64
  nfsv4-client-utils-1:2.5.4-25.el9.x86_64  quota-1:4.06-6.el9.x86_64           quota-nls-1:4.06-6.el9.noarch
  rpcbind-1.2.6-7.el9.x86_64          sssd-nfs-idmap-2.9.4-6.el9_4.x86_64

Complete!
[root@ip-10-0-15-242 ~]# service nfs-utils start
Redirecting to /bin/systemctl start nfs-utils.service
[root@ip-10-0-15-242 ~]# mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport
fs-0d96b2ccb866e1e26.efs.us-east-1.amazonaws.com:/ /mnt
[root@ip-10-0-15-242 ~]#
```

- Now to create and auto scaler for our public instance create an image of the instance first.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like EC2 Dashboard, EC2 Global View, Events, Console-to-Code Preview, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, and AMI Catalog. The main area displays 'Instances (1/1)'. A table lists one instance: 'public-server' (i-02e32c77032265a80). The instance is 'Running' (t2.micro) and has '2/2 checks passed'. A context menu is open over this instance, with 'Launch instances' highlighted in orange. Other options in the menu include Connect, View details, Manage instance state, Instance settings, Networking, Security, Image and templates, and Monitor and troubleshoot.

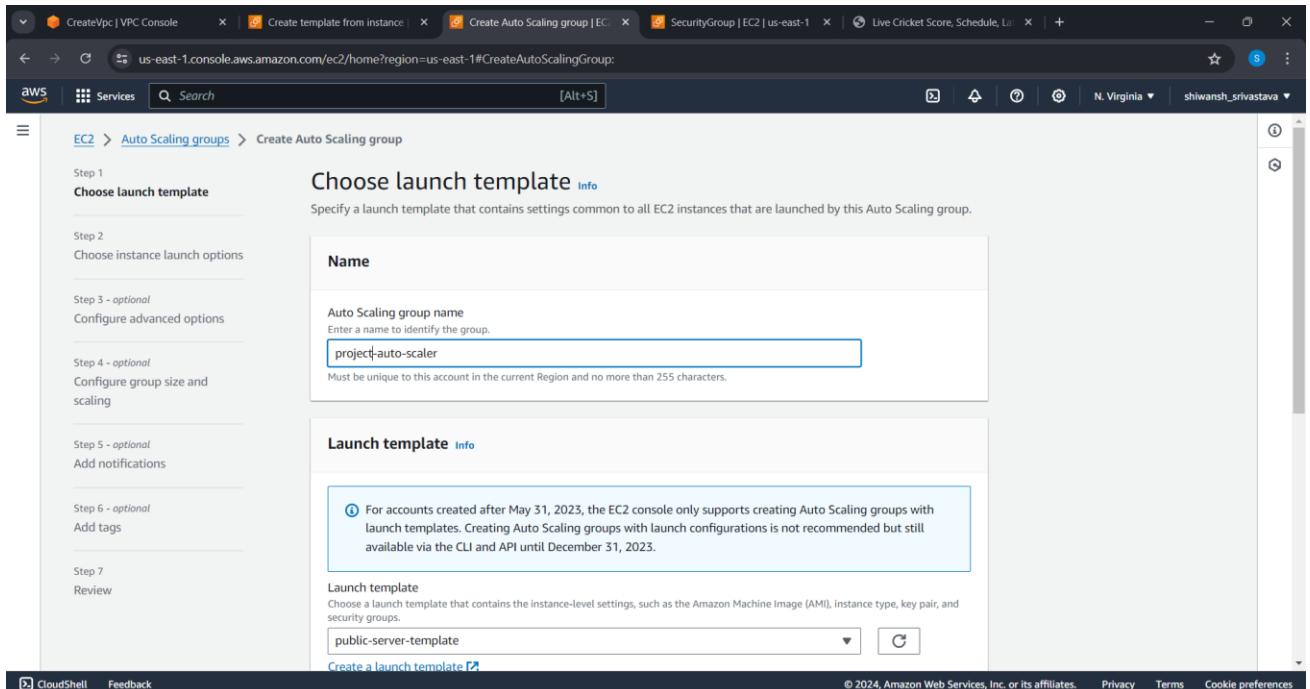
## ● Now create launch template.

The screenshot shows the 'Create template from instance' wizard. Step 1: 'Create launch template'. It shows the 'Source instance' as i-02e32c77032265a80, 'Launch template name - required' as 'public-server-template', and 'Template version description' as 'A prod webserver for MyApp'. A tooltip for the 'Free tier' is visible, stating: 'In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 50 GiB of EBS storage, 2 million I/Os, 1 GB of temporary storage, and 1000 CPU credits per hour.'

Step 2: 'Summary'. It lists the configuration: Software Image (AMI) provided by Red Hat, Inc. (ami-0583d8c7a9c35822c), Virtual server type (instance type) t2.micro, Firewall (security group) public-SG1, and Storage (volumes) 1 volume(s) - 10 GiB.

Step 3: 'Create launch template' button is at the bottom right.

- Now search auto scaler and click on create.
- Give a name to your auto scaler.
- Choose your created launch template



- Now create a new load balancer.

The screenshot shows the AWS Auto Scaling group creation wizard at Step 3 - optional: Configure advanced options. The main section is titled "Configure advanced options - optional". It includes a brief description of how to integrate the Auto Scaling group with other services like VPC Lattice or set more control over health checks. Below this, there's a "Load balancing" section with three options:

- No load balancer: Traffic to your Auto Scaling group will not be fronted by a load balancer.
- Attach to an existing load balancer: Choose from your existing load balancers.
- Attach to a new load balancer: (selected) Quickly create a basic load balancer to attach to your Auto Scaling group.

Below the load balancing section, there's a "Attach to a new load balancer" section with a note about defining a new load balancer. It shows two radio button options for "Load balancer type": "Application Load Balancer" (selected) and "Network Load Balancer".

- Set the max capacity up to 10.
- Choose tracking scaling policies.
- Set average CPU utilization percentage to 50 and instance warmup time to 10.
- Add notifications to get an update whenever your instance is up or terminated.

The screenshot shows the 'Add notifications' step of the Auto Scaling group creation wizard. It displays two notification configurations:

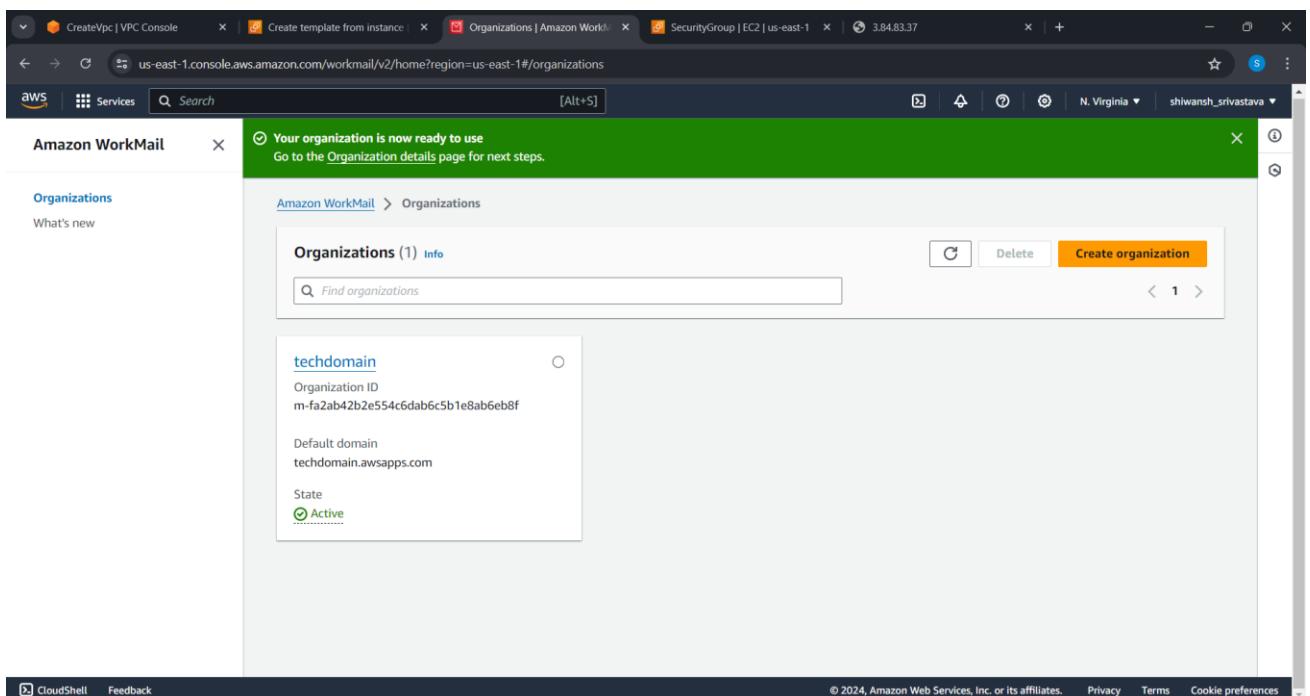
- Notification 1:** Send a notification to 'instance-is-up' with recipient 'shivanshrivastava345@gmail.com'. Event types: Launch (checked), Terminate, Fail to launch, Fail to terminate.
- Notification 2:** Send a notification to 'instance-is-down' with recipient 'shivanshrivastava345@gmail.com'. Event types: Launch, Terminate, Fail to launch, Fail to terminate.

- Now take a review of all configurations and create.

The screenshot shows the 'Review' step of the Auto Scaling group creation wizard. It summarizes the configuration:

- Step 1: Choose launch template**
  - Group details:
    - Auto Scaling group name: project-auto-scaler
  - Launch template:
    - Launch template: public-server-template
    - Version: Default
    - Description: lt-0c8af74916c8ca8b1
- Step 2: Choose instance launch options**
  - Network:
    - VPC: vpc-0202249a-Subnet-1

- Now we must attach the workmail for our organization.
- Search workmail and click on create
- Choose your organization domain and create.



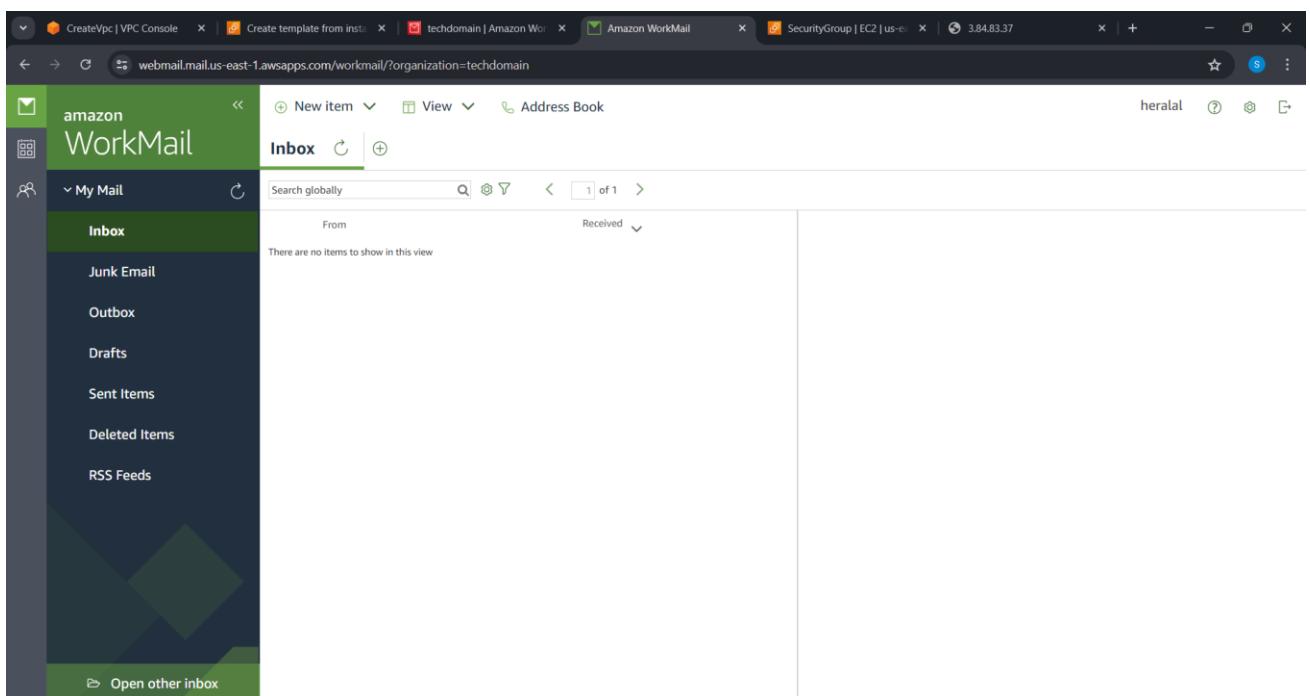
- Now add some users to your organization.

The screenshot shows the Amazon WorkMail console interface. On the left, there's a sidebar with navigation links for Organizations, Users (selected), Groups, Resources, Domains, Mobile policies, Organization settings, Tags, Access control rules, Retention policies, Impersonation roles, and Monitoring. The main content area is titled 'Amazon WorkMail > Organizations > techdomain > Users'. It shows a table with the following columns: Username, Display name, Primary email address, and State. A search bar at the top is set to 'Find users' and has a dropdown menu showing 'Username'. Below the table, a message says 'No users to display.'

- Now open your organization and click on amazon mail web application link.

The screenshot shows a web browser window with the URL 'techdomain.awsapps.com/auth/?client\_id=6b9615ec01be1c8d&redirect\_uri=https%3A%2F%2Fwebmail.mail.us-east-1.awsapps.com%2Fworkmail%2F'. The page itself is the Amazon WorkMail login screen. It features the Amazon logo and the text 'Please log in with your techdomain credentials'. There are input fields for 'Username' and 'Password', a 'Remember username' checkbox, and a large green 'Sign In' button. At the bottom, there's a small note about AWS Customer Agreement and Privacy Notice, along with a link to the Cookie Notice.

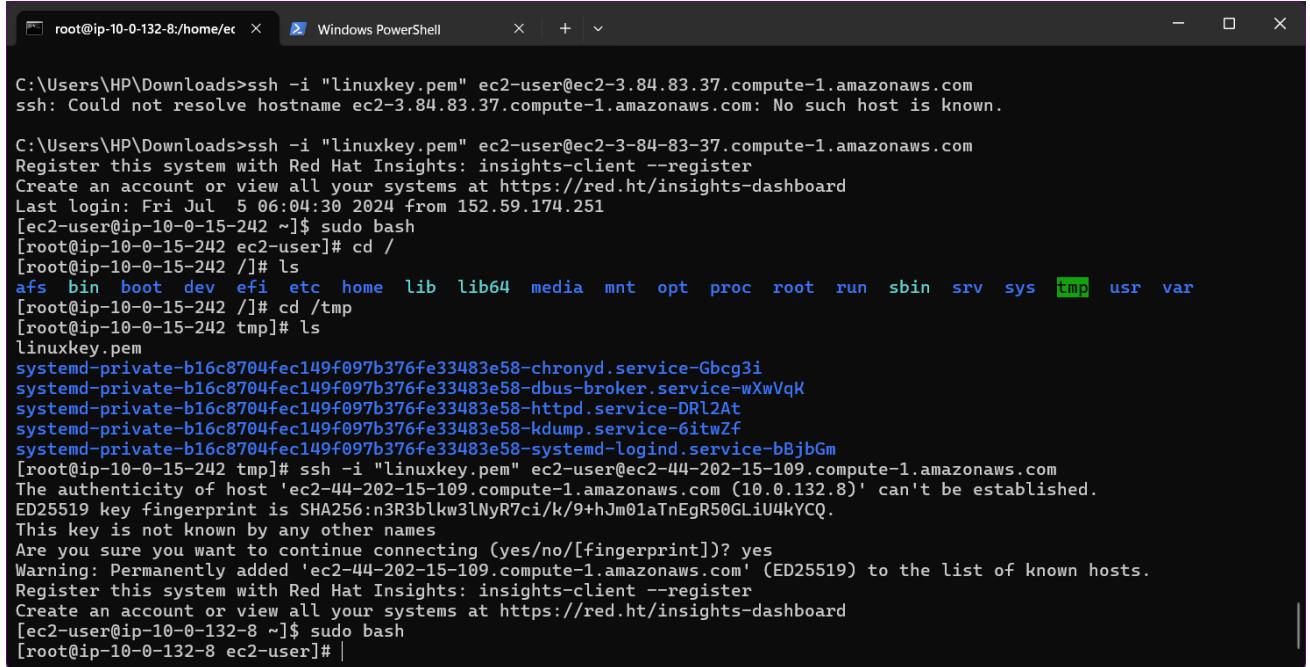
- You can check your mail services by login as a user.
- You can choose any user details to check your mail services.



# Creating Private Instance

- Now launch a private instance with same configuration of public instance just in subnets choose private subnet of same availability zone.
- Since this instance is on private network it can't be accessed directly through internet.
- You first need to upload your keypair to your public instance from your computer.
- Write following commands in your device's terminal: cd downloads; scp -i .\linuxkey.pem .\linuxkey.pem [ec2-user@3.84.83.37:/tmp](#)
- Now open your instance terminal and write following commands:

```
cd /tmp; ssh -i linuxkey.pem ec2-user@44.202.15.109; yes ; sudo bash
```



```
root@ip-10-0-132-8:/home/ec  ×  Windows PowerShell  ×  +  ▾

C:\Users\HP\Downloads>ssh -i "linuxkey.pem" ec2-user@ec2-3-84-83-37.compute-1.amazonaws.com
ssh: Could not resolve hostname ec2-3-84-83-37.compute-1.amazonaws.com: No such host is known.

C:\Users\HP\Downloads>ssh -i "linuxkey.pem" ec2-user@ec2-3-84-83-37.compute-1.amazonaws.com
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
Last login: Fri Jul  5 06:04:30 2024 from 152.59.174.251
[ec2-user@ip-10-0-15-242 ~]$ sudo bash
[root@ip-10-0-15-242 ec2-user]# cd /
[root@ip-10-0-15-242 /]# ls
afs  bin  boot  dev  efi  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
[root@ip-10-0-15-242 /]# cd /tmp
[root@ip-10-0-15-242 tmp]# ls
linuxkey.pem
systemd-private-b16c8704fec149f097b376fe33483e58-chrony.service-Gbcg3i
systemd-private-b16c8704fec149f097b376fe33483e58-dbus-broker.service-wXwVqK
systemd-private-b16c8704fec149f097b376fe33483e58-httpd.service-DRL2At
systemd-private-b16c8704fec149f097b376fe33483e58-kdump.service-6itwZf
systemd-private-b16c8704fec149f097b376fe33483e58-systemd-logind.service-bBjbGm
[root@ip-10-0-15-242 tmp]# ssh -i "linuxkey.pem" ec2-user@ec2-44-202-15-109.compute-1.amazonaws.com
The authenticity of host 'ec2-44-202-15-109.compute-1.amazonaws.com (10.0.132.8)' can't be established.
ED25519 key fingerprint is SHA256:n3R3bLkw3lNyR7ci/k/9+hJm01aTnEgR50GLiU4kYQ.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-202-15-109.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
[ec2-user@ip-10-0-132-8 ~]$ sudo bash
[root@ip-10-0-132-8 ec2-user]# |
```

- Now we must create an S3 bucket.
- Search S3 and click on create bucket.
- Don't enable ACLs while creating a bucket it can be enabled later.

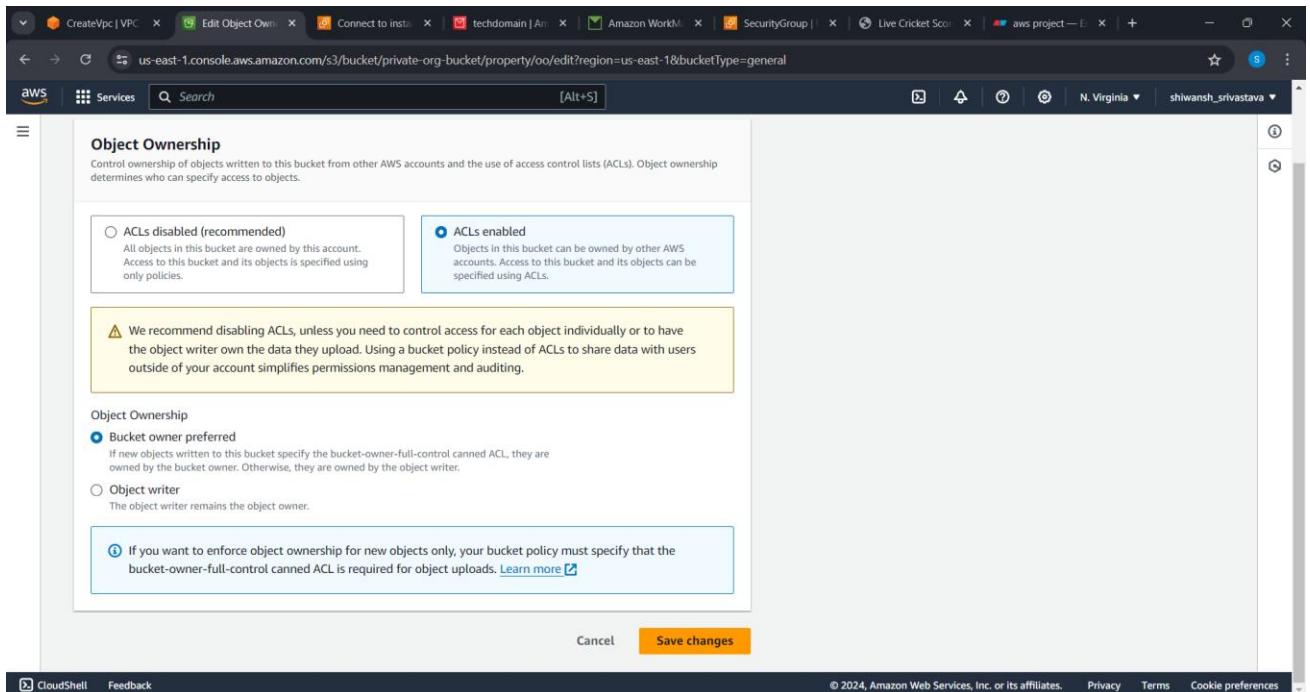
The screenshot shows the 'Create bucket' configuration page in the AWS Management Console. The 'General configuration' section is selected. Under 'Bucket type', 'General purpose' is chosen. The 'Bucket name' field contains 'project-private-bucket'. The 'Copy settings from existing bucket - optional' section is collapsed.

- Now upload an object in the bucket.

The screenshot shows the 'Objects' tab in the AWS S3 console for the 'private-org-bucket'. A single file, 'Screenshot 2024-07-05 120000.png', has been uploaded. The file is a PNG image, 147.1 KB in size, and was last modified on July 5, 2024, at 12:01:03 UTC+05:30.

Name	Type	Last modified	Size	Storage class
Screenshot 2024-07-05 120000.png	png	July 5, 2024, 12:01:03 (UTC+05:30)	147.1 KB	Standard

- Now inside bucket permissions enable ACLs.



- Edit access control list of both bucket and object.
- Now open the object and copy the objects public URL to check the accessibility of the object.

Screenshot 2024-07-05 120000.png

Successfully edited access control list for object "Screenshot 2024-07-05 120000.png".

Amazon S3 > Buckets > private-org-bucket > Screenshot 2024-07-05 120000.png

Screenshot 2024-07-05 120000.png [Info](#)

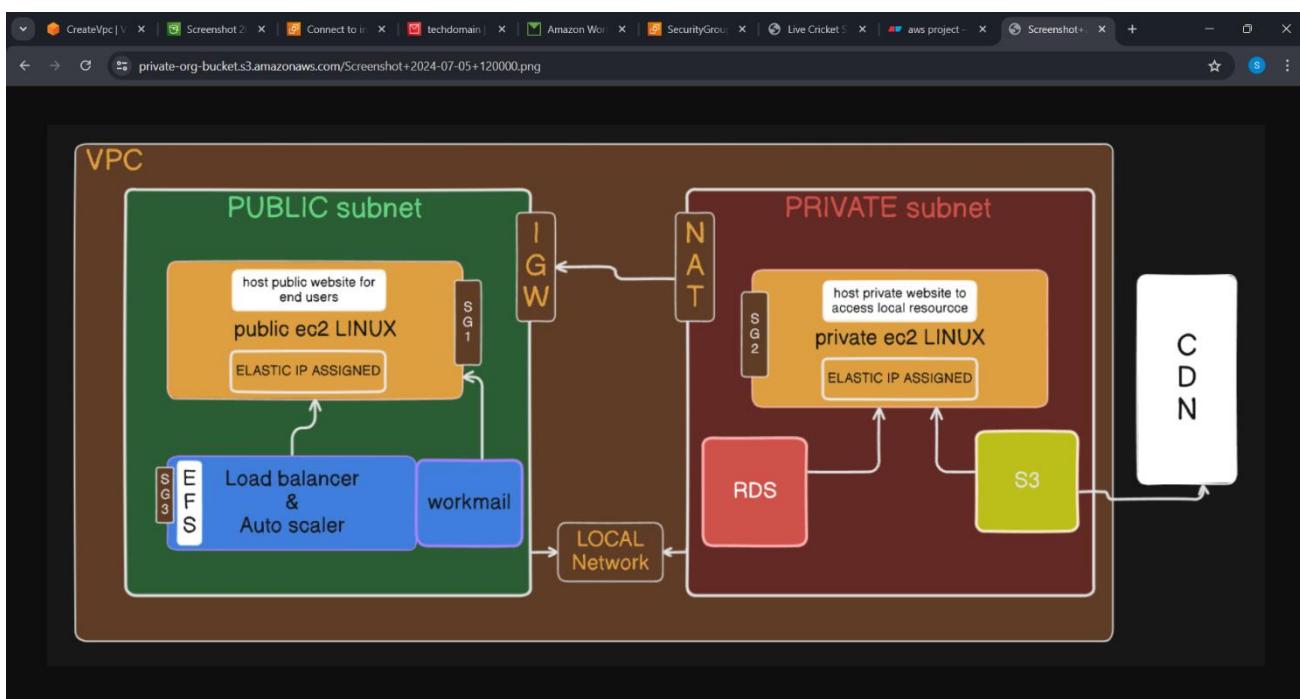
[Copy S3 URI](#) [Download](#) [Open](#) [Object actions](#)

[Properties](#) [Permissions](#) [Versions](#)

**Object overview**

Owner	sivanshsrivastava345	S3 URI	<a href="s3://private-org-bucket/Screenshot 2024-07-05 120000.png">s3://private-org-bucket/Screenshot 2024-07-05 120000.png</a>
AWS Region	US East (N. Virginia) us-east-1	Amazon Resource Name (ARN)	<a href="arn:aws:s3:::private-org-bucket/Screenshot 2024-07-05 120000.png">arn:aws:s3:::private-org-bucket/Screenshot 2024-07-05 120000.png</a>
Last modified	July 5, 2024, 12:01:03 (UTC+05:30)	Entity tag (Etag)	<a href="3e23c284085c6e1efdbf13a1f4f75842">3e23c284085c6e1efdbf13a1f4f75842</a>
Size	147.1 KB	Object URL	<a href="https://private-org-bucket.s3.amazonaws.com/Screenshot+2024-07-05+120000.png">https://private-org-bucket.s3.amazonaws.com/Screenshot+2024-07-05+120000.png</a>
Type	png		<a href="#">Copy Object URL</a>

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



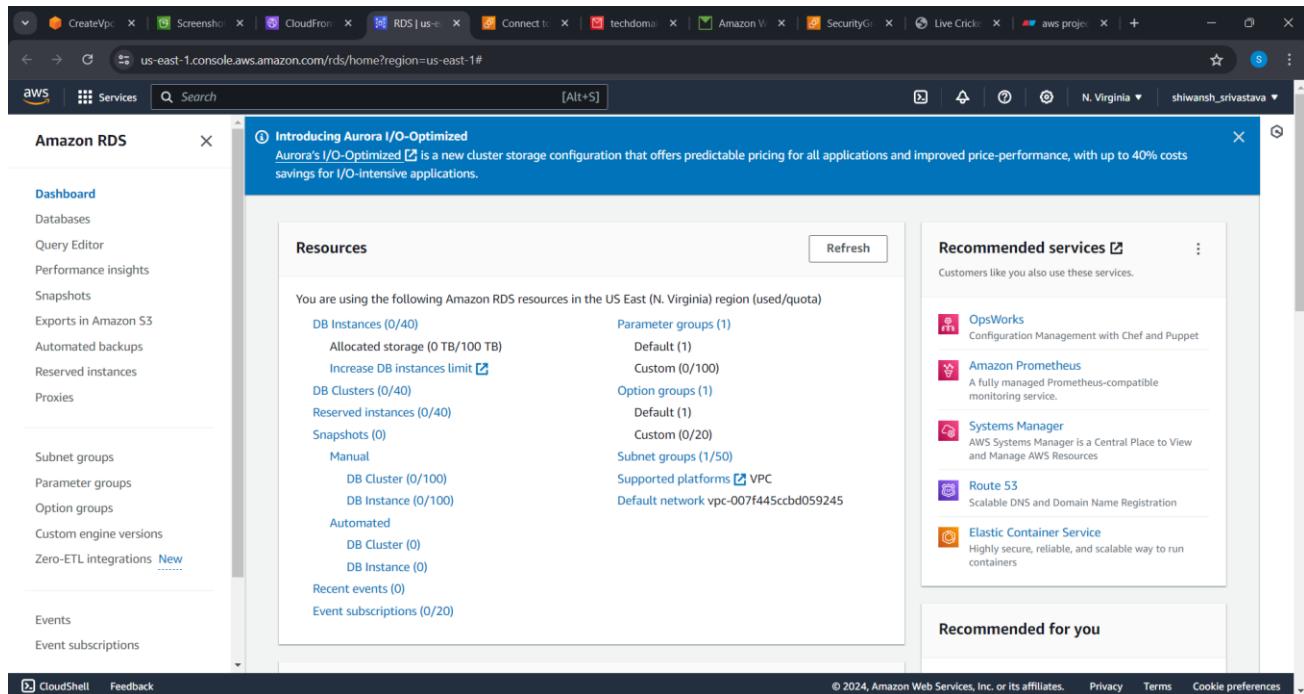
- Now create CloudFront distribution in all edge locations in world.
- Search CloudFront and click create.

The screenshot shows the AWS CloudFront Distributions page. At the top, there is a search bar and several filter buttons: 'Enable', 'Disable', 'Delete', and a prominent orange 'Create distribution' button. Below this is a table header with columns: ID, Description, Type, Domain name, Alternate do..., Origins, Status, and Last modified. A message 'No distributions' and 'You don't have any distributions.' is displayed. At the bottom of the table area is a 'Create distribution' button.

- Choose a domain and name for your distribution.
- Set origin access to public.

The screenshot shows the 'Create distribution' wizard on the 'Origin' configuration step. It includes fields for 'Origin domain' (set to 'private-org-bucket.s3.us-east-1.amazonaws.com'), 'Origin path - optional' (empty), 'Name' (set to 'private-org-bucket.s3.us-east-1.amazonaws.com'), and 'Origin access' (set to 'Public'). The 'Public' option is selected, with a note that the bucket must allow public access. Other options include 'Origin access control settings (recommended)' and 'Legacy access identities'.

- Now we must create a database for our private networks or instance.
- Search RDS and click on create data base.



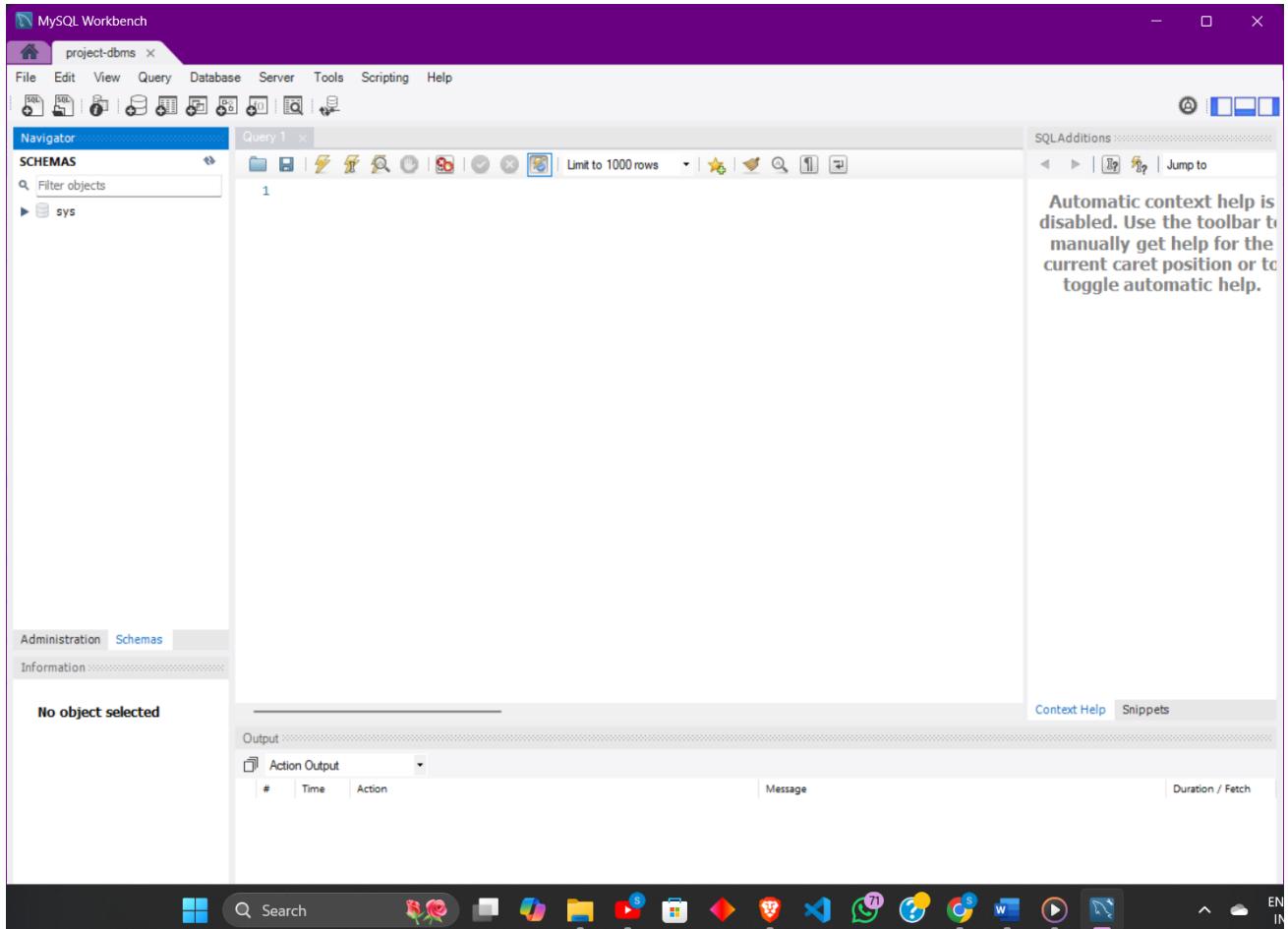
- Choose the standard settings.
- Select MySQL software and select free tier setup.
- Now create a password for your database.
- In networks section set publicly accessible.
- Create database.

The screenshot shows the Amazon RDS console with the 'Databases' tab selected. A prominent blue banner at the top left reads 'Introducing Aurora I/O-Optimized' with a subtext about predictable pricing and improved performance. Below the banner, a callout box suggests creating a Blue/Green Deployment to minimize downtime during upgrades. The main area displays a table titled 'Databases (1)' with one entry: 'private-database' (Status: Available, Role: Instance, Engine: MySQL Community, Region & AZ: us-east-1a, Size: db.t3.micro). The left sidebar includes links for Dashboard, Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, and Zero-ETL integrations (marked as New).

- Open MySQL in your device and setup new connection.

The screenshot shows the 'Setup New Connection' dialog box. The 'Parameters' tab is selected. The 'Connection Name' field contains 'project-dbms'. The 'Connection Method' dropdown is set to 'Standard (TCP/IP)'. The 'Hostname' field is filled with 'iyyk51m.ap-south-1.rds.amazonaws.com' and the 'Port' field is set to '3306'. The 'Username' field contains 'admin'. The 'Password' field has two buttons: 'Store in Vault ...' and 'Clear'. The 'Default Schema' field is empty. To the right of each input field, there is a descriptive tooltip. At the bottom of the dialog are buttons for 'Configure Server Management...', 'Test Connection', 'Cancel', and 'OK'.

- Now enter your password and open your created RDS.



- You can check the proper working of your database by creating schemas and tables.

MySQL Workbench

project-dbms

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

project

- Tables
- Views
- Stored Procedures
- Functions

sys

- Tables
- Views
- Stored Procedures
- Functions

Query 1

```
1 select *from employee;
```

Result Grid

roll_no	name	salary	employeecol
HULL	HULL	HULL	HULL

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Administration Schemas

Information

Schema: project

employee 1

Action Output

#	Time	Action	Message	Duration / Fetch
4	11:33:58	create table employee	Error Code: 1044. Access denied for user 'admin'@'%' to database 'sys'	0.063 sec
5	11:35:08	Apply changes to employee	Changes applied	
6	11:35:49	select *from employee LIMIT 0, 1000	Error Code: 1146. Table 'sys.employee' doesn't exist	0.141 sec
7	11:35:58	select *from employee LIMIT 0, 1000	Error Code: 1146. Table 'sys.employee' doesn't exist	0.078 sec

EN IN

The screenshot shows the MySQL Workbench interface. In the top-left, the 'project-dbms' connection is selected. The main area contains a 'Query 1' window with the SQL command 'select \*from employee;'. Below it is a 'Result Grid' showing four columns: roll\_no, name, salary, and employeecol, all containing 'HULL' values. To the right is a 'SQLAdditions' panel with a message about context help. At the bottom, the 'Output' tab displays the 'Action Output' section with a table of log entries. The log includes actions like creating a table, applying changes, and performing selects, along with their timestamps, messages, and duration/fetch times. The Windows taskbar at the bottom shows various open applications.

So, after following all these aforementioned steps our organisation will have two subnets of private networks with connected storage and database which will be accessible only to the administrator and the public network in which the websites are hosted that are publicly available and it will help us in this setting up organisational infrastructure at a low cost and helps us to make our setup more scalable as we need in future.

As we know, how cloud computing is gaining popularity in the current world scenario these skills will help us to make our organization more cost effective and scalable.