

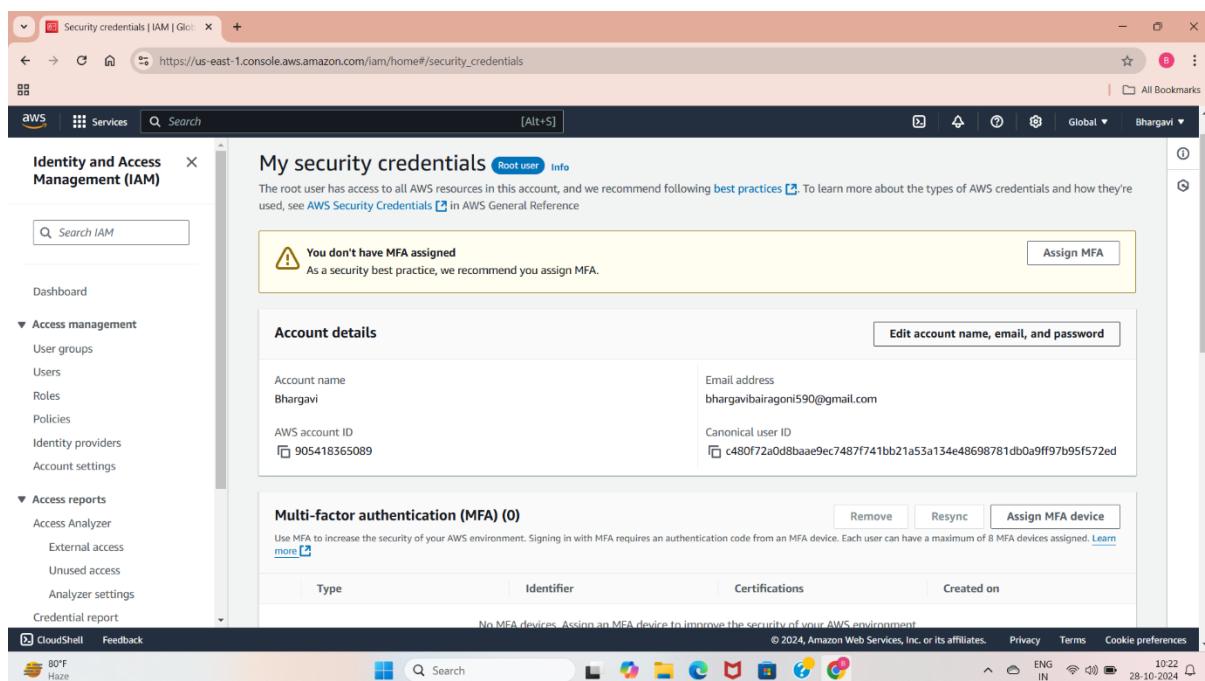
AWS MINI PROJECT – I

LAB-1: I AM HANDS ON

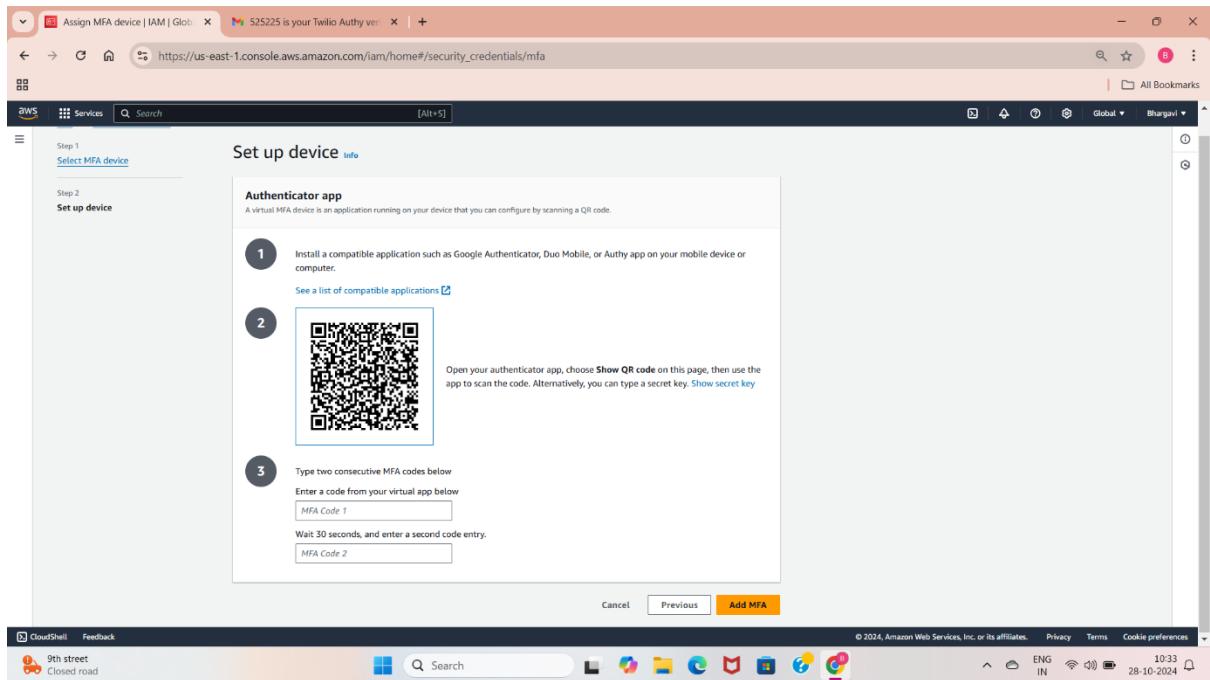
To secure our aws account we need to setup “Multi Factor Authentication (MFA)”.

Follow the below steps to create a MFA

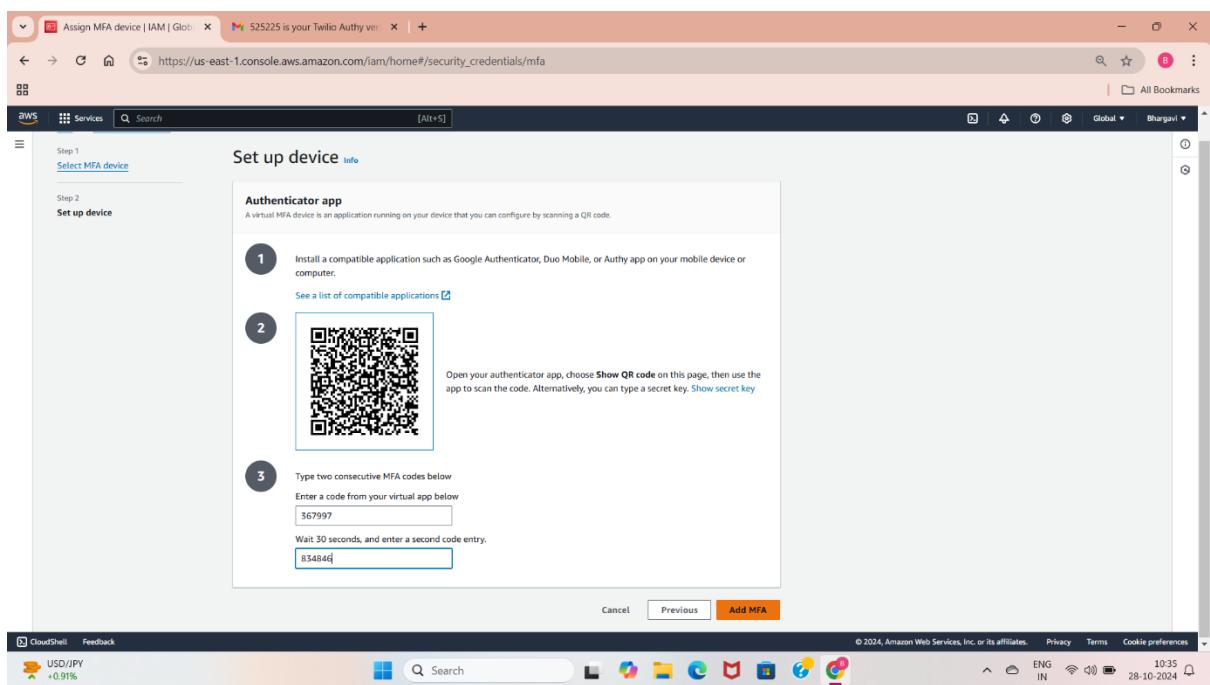
- Sign in to the AWS Console using root user sign in
- Go to security credentials(Using any service like IAM, EC2)
- We can see the Multi-Factor authentication (MFA) add MFA using this option.
- We can find three types of MFA Devices
 1. Passkey or Security key
 2. Authenticator App
 3. Hardware TOTP token
- Select any one of the three options and set MFA for AWS Console



This is the dashboard of identity and access management and there is no MFA for this account.



Am setting a MFA using Authenticator app. First we need to download the twilio authy application in mobile and then create account in that app, scan the above QR code.



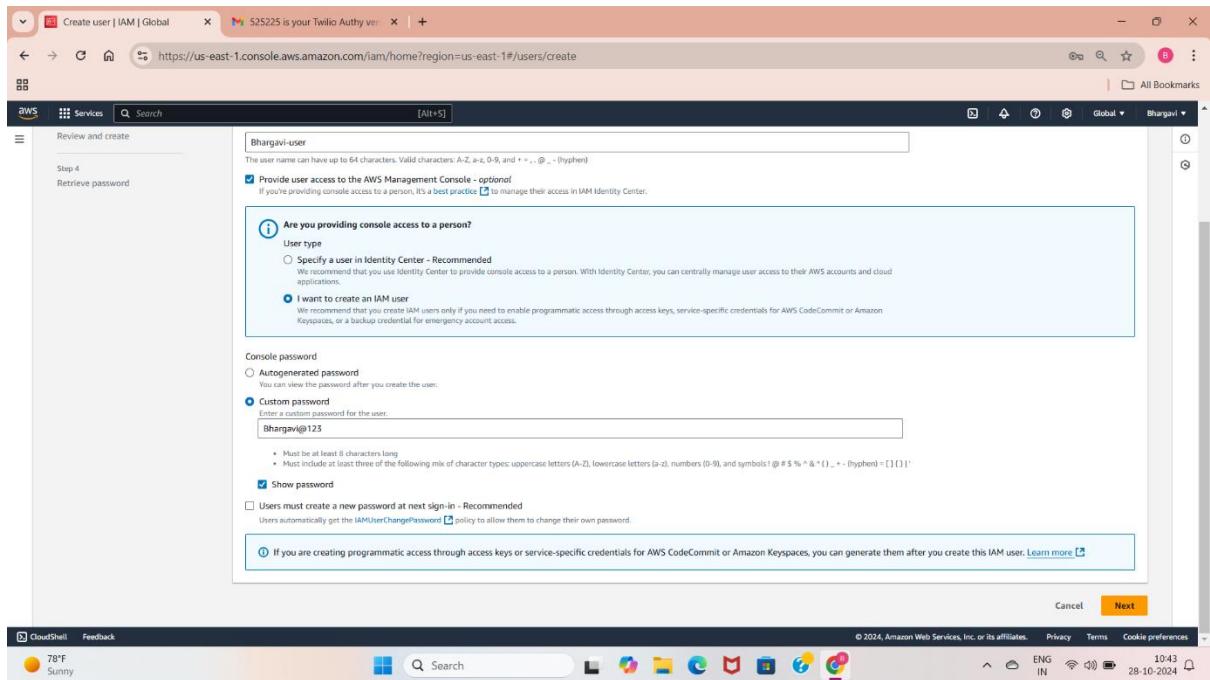
By scanning the QR code we can get a OTP to our registered mobile number enter the two consecutive OTPs. Now your account is secured using Authenticator app.

The screenshot shows the AWS IAM Security Credentials page. On the left, there's a sidebar with navigation links like Dashboard, Access management, Access reports, and Related consoles. The main content area has a header "MFA device assigned" with a note about registering up to 8 MFA devices. Below this, the "My security credentials" section shows account details (Account name: Bhargavi, Email address: bhargavibairagi590@gmail.com, AWS account ID: 905418365089) and a table for Multi-factor authentication (MFA) devices. One device is listed: Type: Virtual, Identifier: arn:aws:iam:905418365089:mfa/Bhargavi, Certifications: Not Applicable, Created on: Mon Oct 28 2024. At the bottom, there's a section for Access keys (0) with a "Create access key" button.

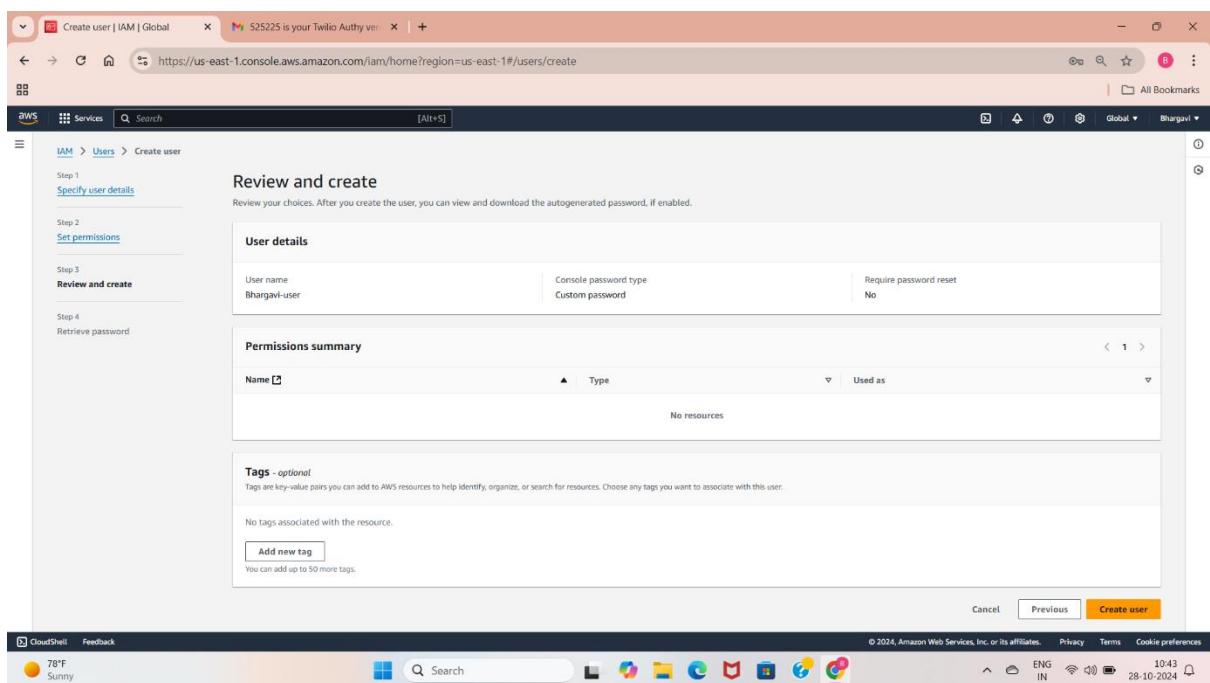
We can see my Security Credentials like type of my MFA , created date and identifier. Now sign out your account and try to login once again.

The screenshot shows the AWS Sign-in page. A message at the top says "Try the new sign in UI" with a link to "Enable new sign in". The main form for "Multi-factor authentication" asks for an "Email address" (bhargavibairagi590@gmail.com) and an "MFA code" (input field). Below the form are links for "Troubleshoot MFA" and "Cancel". To the right, there's a promotional banner for "AWS Cloud Institute" with the text "Become a cloud developer in as little as 9 months" and a "Enroll today >>" button. The banner features the AWS logo and a 3D cube icon. The browser status bar at the bottom shows the date as 28-10-2024.

When I am trying to login am getting a verification message by asking multi-factor authentication, open the twilio authy app and enter the security code.



Creating a user with the name of Bhargavi-user with only user access to aws management console.



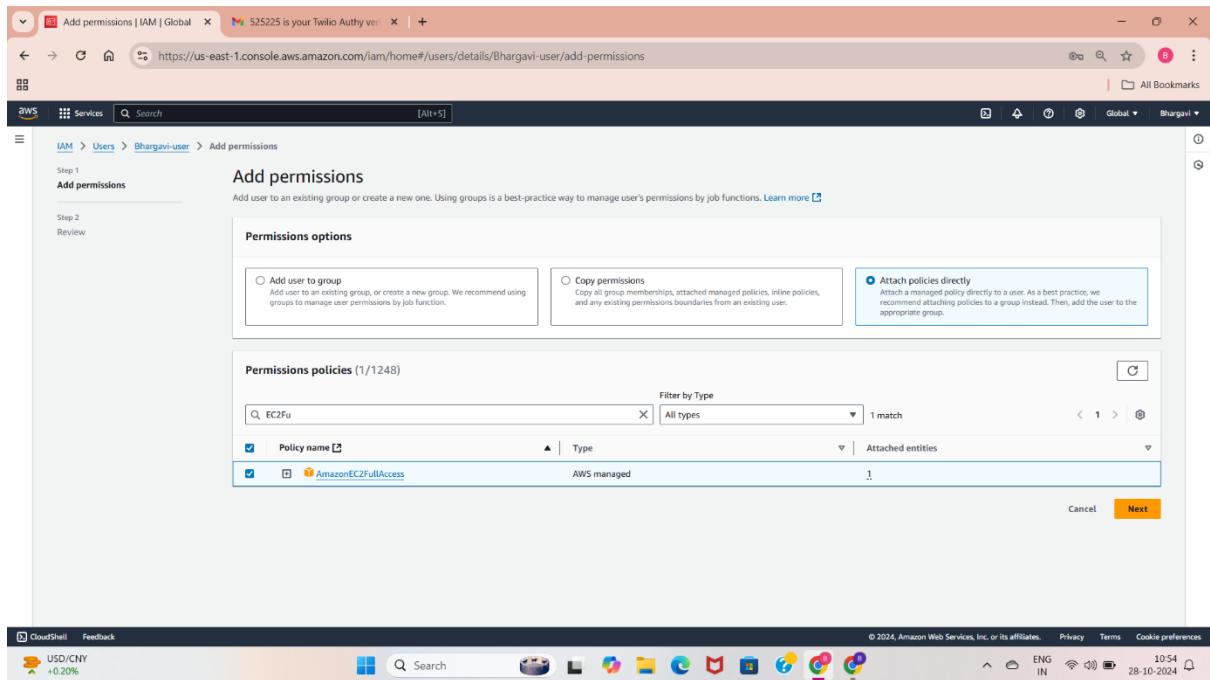
While creating user I have checked with the permissions of user, by default I Have taken AWS Management Console so I don't get any permissions.

The screenshot shows the AWS Management Console EC2 dashboard. On the left, the navigation pane includes 'Instances' (with 'Instances', 'Instance Types', 'Launch Templates', etc.), 'Images' (with 'AMIs', 'AMI Catalog'), and 'Elastic Block Store' (with 'Volumes', 'Snapshots', 'Lifecycle Manager'). The main 'Resources' section displays various EC2 metrics with red 'API Error' icons. Below it, the 'Launch instance' section shows a 'Launch instance' button and a 'Migrate a server' option. The 'Service health' section indicates an 'An error occurred' with a red 'X' icon. The 'Zones' section shows a single zone. The right side features a sidebar for 'EC2 Free Tier' (info, offers), a 'View Global EC2 resources' link, and an 'Account attributes' section. The bottom status bar shows the date (October 28, 2024), time (10:52), and user (Bhargavi-user).

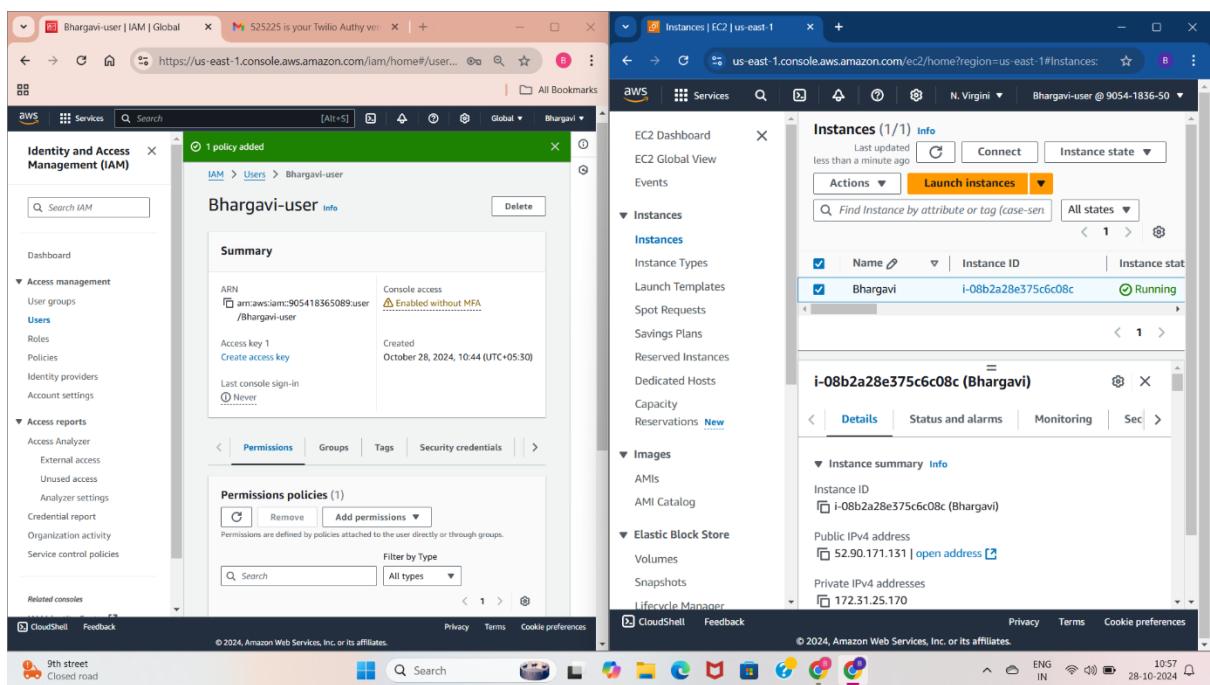
With assigning only aws management console access we cannot access any other services, we get error by saying you do not have any permissions to access this service.

This screenshot shows two open browser tabs. The left tab is the 'Identity and Access Management (IAM)' dashboard, showing the 'Create access key' section and a 'Permissions policies (0)' table. The right tab is the 'Instances | EC2 | us-east-1' dashboard, showing the 'Instances' section with a 'Launch instances' button and a message: 'You are not authorized to perform this operation. User: arn:aws:iam::905418365089:user/Bhargavi-user is not authorized to perform: ec2:DescribeInstances because no identity-based policy allows the ec2:DescribeInstances action'. Both tabs show the same user (Bhargavi-user) and date (October 28, 2024). The bottom status bar also shows the date and time.

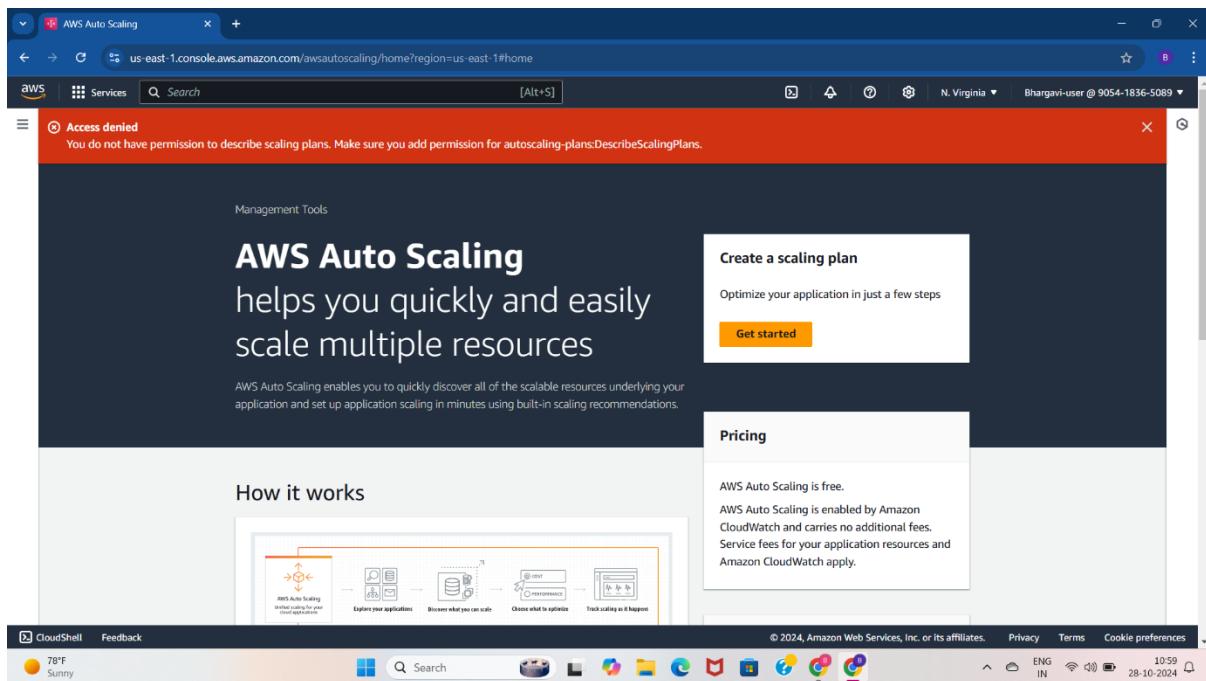
We cannot access or launch ec2 instance without giving ec2 access permission.



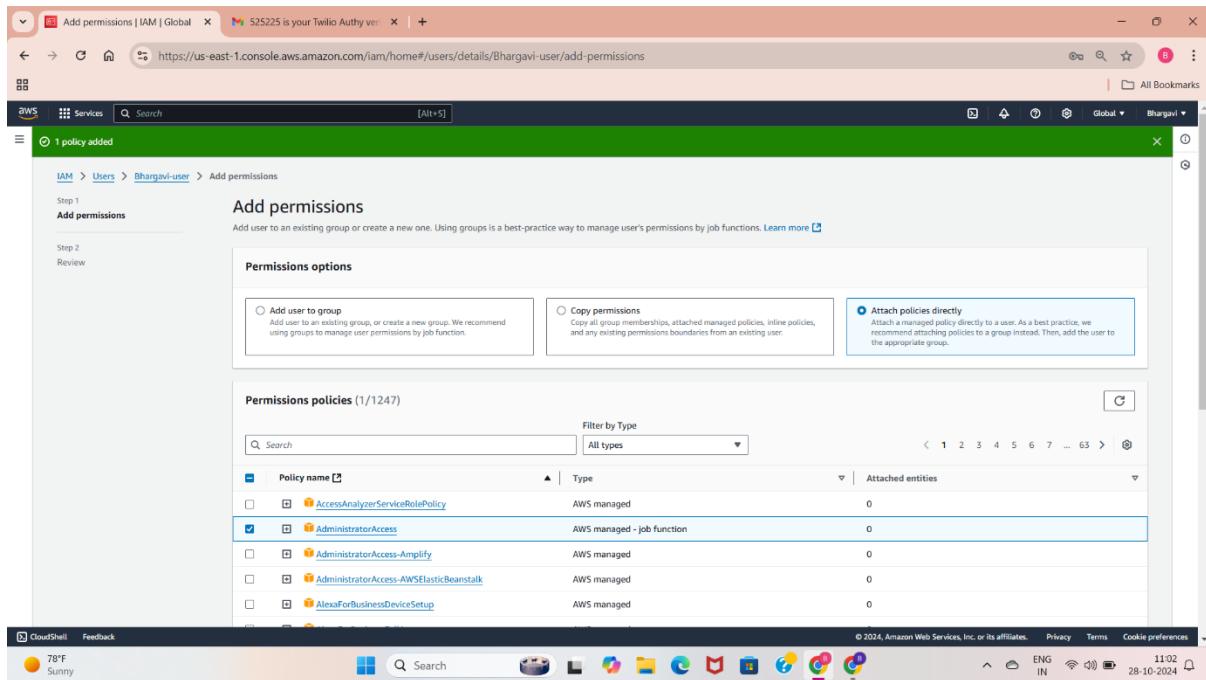
I am adding permissions to user using attach policy directly. Attaching a policy with the name "AmazonEc2fullAccess".



Now am able to access the ec2 service by creating ec2 instance.



By assigning only EC2full permissions am able to access other services, so we need to give permission to these services if we want to use.



Here I attached a policy "AdministrativeAccess" to the user named Bhargavi-user. Now we will check with the services.

Bhargavi-user | IAM | Global

https://us-east-1.console.aws.amazon.com/iam/home#/users/details/Bhargavi-user?section=permissions

1 policy added

Identity and Access Management (IAM)

Bhargavi-user Info

Summary

ARN: arn:aws:iam::905418365089:user/Bhargavi-user

Created: October 28, 2024, 10:44 (UTC+05:30)

Console access: Enabled without MFA

Last console sign-in: Never

Access key 1: Create access key

Permissions | Groups | Tags | Security credentials | Last Accessed

Permissions policies (2)

Policy name	Type	Attached via
AdministratorAccess	AWS managed - job function	Directly
AmazonEC2FullAccess	AWS managed	Directly

CloudShell Feedback

78°F Sunny

Search

CloudShell Feedback

CloudShell Feedback

CloudShell Feedback

AWS Auto Scaling

How would you rate your experience with this service console? ★ ★ ★ ★

aws Services Search

AWS Auto Scaling helps you quickly and easily scale multiple resources

Create a scaling plan

Get started

Pricing

AWS Auto Scaling is free. AWS Auto Scaling is enabled by Amazon CloudWatch and carries no additional fees. Service fees for your application resources and Amazon CloudWatch apply.

Documentation

User guide API reference FAQs Support forums

CloudShell Feedback

CloudShell Feedback

CloudShell Feedback

After assigning Administrative access to the user I can able to access other services like Auto scaling group,S3, Ec2ImageBuilder and EC2.

Final conclusion about this lab is we can able to create user and access all services using IAM user using credentials of created user in IAM service, but we get the permissions to our IAM user from root user(policies). Root user is having the authority to attach policy and remove policy.

LAB-2: BILLING AND COST MANAGEMENT:

We get bills whenever we use the aws services so to limit the budget we use billing alarms.

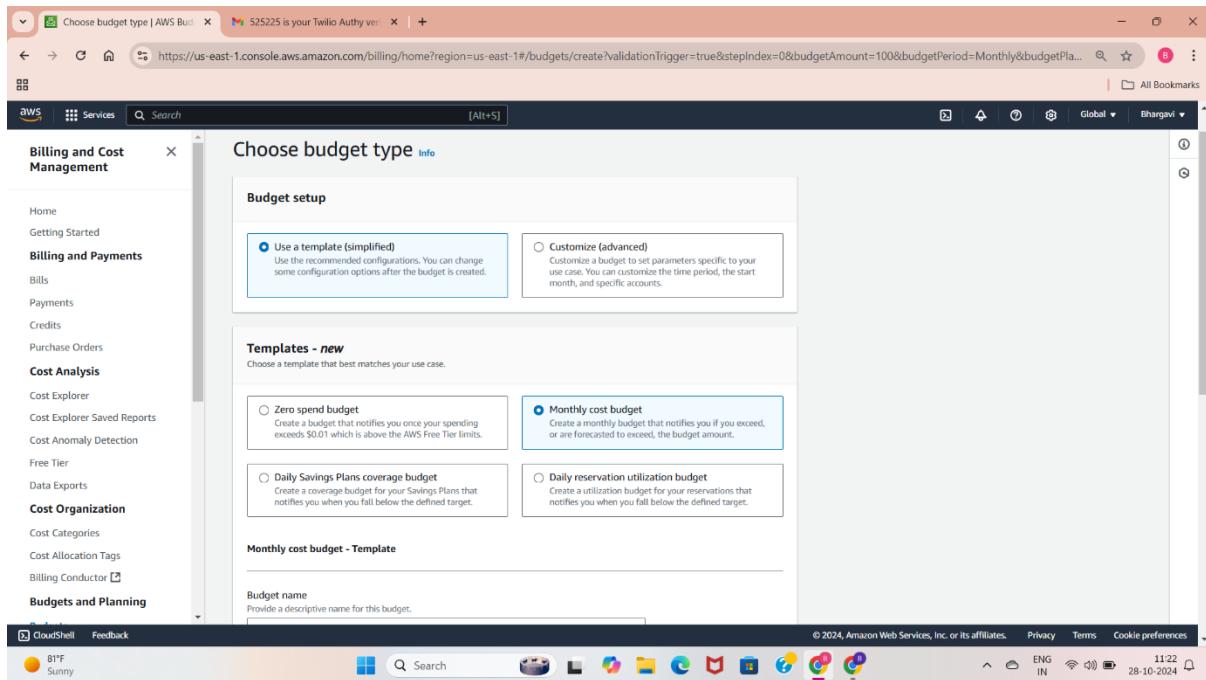
Billing alarms triggers whenever we cross the limit that is set by us in budgets. Here we use Billing and Cost Management to set the billing alarm.

The screenshot shows the AWS Cost Management Console home page. On the left, a sidebar menu includes options like Home, Getting Started, Billing and Payments, Cost Analysis, Cost Organization, and Budgets and Planning. The main content area displays a 'Billing and Cost Management home' summary. It shows 'Month-to-date cost' at \$1.02, 'Last month's cost for same time period' at \$0.06, and 'Total forecasted cost for current month' at \$1.07. Below this is a 'Cost breakdown' section where costs are grouped by service. To the right, there are 'Cost monitor' and 'Recommended actions' sections, which currently show '1 over budget' and '4 detected' anomalies respectively. At the bottom, there are notifications for 'Free Tier usage' and 'Budget status'. The browser address bar shows the URL: https://us-east-1.console.aws.amazon.com/costmanagement/home?region=us-east-1#/home.

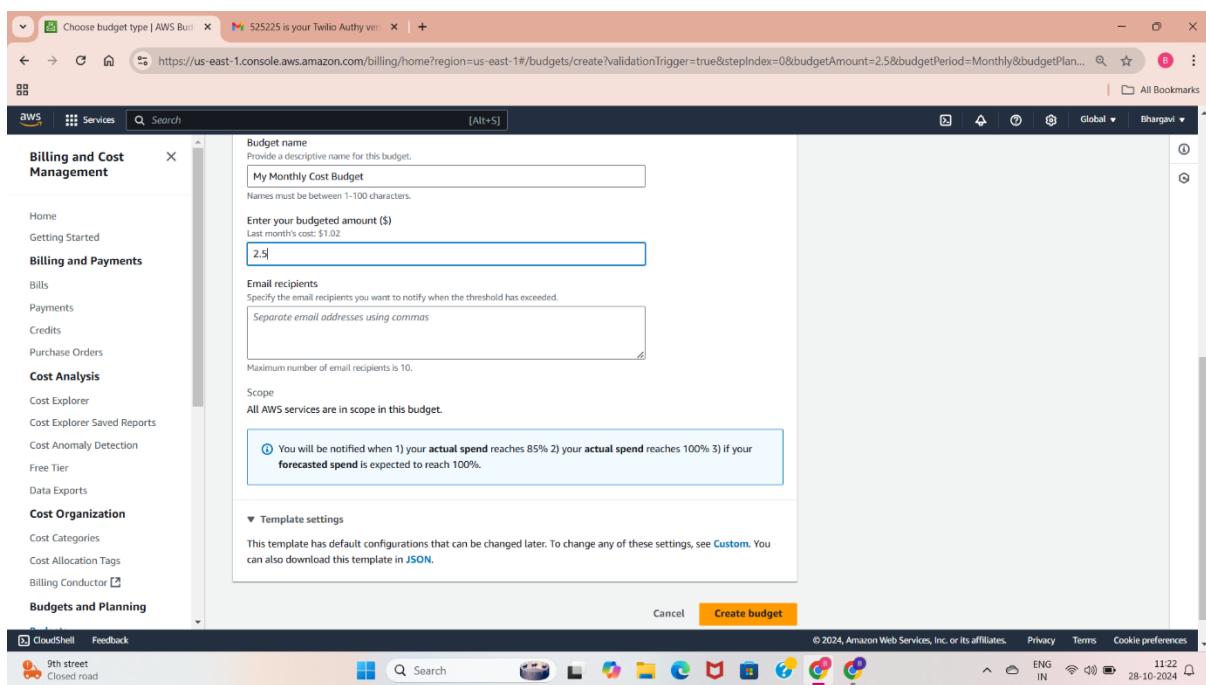
This is the home page of billing and cost management, go to budgets to set billing alarm.

The screenshot shows the 'Create budget' page within the AWS Billing and Cost Management section. The sidebar menu is identical to the previous screen. The main content is titled 'Choose budget type'. It features a 'Budget setup' section with two options: 'Use a template (simplified)' (selected) and 'Customize (advanced)'. Below this is a 'Templates - new' section with four options: 'Zero spend budget' (selected), 'Monthly cost budget', 'Daily Savings Plans coverage budget', and 'Daily reservation utilization budget'. A 'Zero spend budget - Template' section is also present. The browser address bar shows the URL: https://us-east-1.console.aws.amazon.com/billing/home?region=us-east-1#budgets/create?validationTrigger=true&stepIndex=0&budgetAmount=1&budgetPeriod=Monthly&budgetPlann...

Here am creating template by selecting particular configurations. In this step am using Zero spend budget that notifies us after we cross \$0.01 which is above aws free tier.



Creating other budget alarm using template. Here we are setting billing alarm for monthly usage using Monthly cost budget.



A monthly cost budget with \$2.5 and the recipient email. Here when we get notification whenever we cross half or more than half of the monthly cost budget.

The screenshot shows the AWS Budgets console with a green banner at the top stating "Your budget My Monthly Cost Budget has been created successfully." The main area displays a table of budgets:

Name	Thresholds	Budget	Amount used	Forecasted amount	Current vs. budgeted	Forecasted vs. budgeted
My Monthly Cost Budget	OK	\$2.50	\$1.02	\$1.07	40.84%	
My Zero-Spend Budget	Exceeded (1)	\$1.00	\$1.02	\$1.07	102.10%	

The sidebar on the left includes sections for Home, Getting Started, Billing and Payments, Bills, Payments, Credits, Purchase Orders, Cost Analysis, Cost Explorer, Cost Explorer Saved Reports, Cost Anomaly Detection, Free Tier, Data Exports, Cost Organization, Cost Categories, Cost Allocation Tags, Billing Conductor, and Budgets and Planning.

This is the setup with monthly Cost Budget and My Zero Spent Budget. Here it is showing "ok" for monthly cost budget because we didn't cross the amount \$2.5, we cross the zero spent budget so it is showing "Exceeded".

The screenshot shows the AWS Billing Management Console with a summary of the estimated bill for October 2024:

Account ID	Billing period	Bill status
905418365089	October 1 - October 31, 2024	Pending

The total estimated grand total is USD 1.02. Below this, there is a section for payment information and a table for the highest estimated cost by service provider.

Now to check where I have got bill, I need to select the bills option.

The screenshot shows the AWS Billing Management Console interface. On the left, there's a navigation sidebar with various service links like Home, Getting Started, Billing and Payments, Bills, Cost Analysis, Cost Organization, Budgets and Planning, and Savings and. The main content area displays a bill for the service 'Elastic Compute Cloud' in the region 'US East (N. Virginia)'. The 'Charges by service' tab is selected. It lists 8 active services, with a total pre-tax service charge of USD 0.87 and a total tax of USD 0.15, totaling USD 0.87. The table includes columns for Description, Usage Quantity, and Amount in USD.

Description	Usage Quantity	Amount in USD
Elastic Compute Cloud		USD 0.80
Virtual Private Cloud		USD 0.07
CloudWatch		USD 0.00
Data Transfer		USD 0.00
Elastic Load Balancing		USD 0.00
Key Management Service		USD 0.00
Simple Queue Service		USD 0.00
Simple Storage Service		USD 0.00
Total tax		USD 0.15
Total pre-tax service charges in USD		USD 0.87

Inside the bills we can see the particular service with the amount we spent.

This screenshot shows a detailed breakdown of EC2 usage within the AWS Billing Management Console. The table lists specific instance types and their usage details:

Description	Usage Quantity	Amount in USD
Elastic Compute Cloud		USD 0.80
US East (N. Virginia)		USD 0.80
Amazon Elastic Compute Cloud running Linux/UNIX		USD 0.79
\$0.00 per Linux t2.micro instance-hour (or partial hour) under monthly free tier	49.827 Hrs	USD 0.00
\$0.0464 per On Demand Linux t2.medium Instance Hour	10.693 Hrs	USD 0.50
\$0.0928 per On Demand Linux t2.large Instance Hour	3.083 Hrs	USD 0.29
EBS		USD 0.01
\$0.00 per GB-month of General Purpose (gp3) provisioned storage under monthly free tier	2.474 GB-Mo	USD 0.00
\$0.00 per GB-month of General Purpose (SSD) provisioned storage under monthly free tier	1.17 GB-Mo	USD 0.00
\$0.00 per GB-Month of snapshot data stored under monthly free tier	1 GB-Mo	USD 0.00
\$0.05 per GB-Month of snapshot data stored - US East (Northern Virginia)	0.297 GB-Mo	USD 0.01
Virtual Private Cloud		USD 0.07
CloudWatch		USD 0.00
Data Transfer		USD 0.00
Elastic Load Balancing		USD 0.00
Key Management Service		USD 0.00
Simple Queue Service		USD 0.00

I have got the bills in EC2 with \$0.80 because of using t2.medium and t2.large, For VPC with \$0.07 and total amount of \$1.02.

LAB-3:S3 BUCKET:

S3 is a Simple Storage Service which is used to store the large amount of data using buckets and objects.

Bucket:

We create buckets to store the data and the name of the bucket should be globally unique, because to identify name of bucket by aws whenever we ask for data.

In buckets we store data using objects(files)

We store data in the form of files and folder. Here we can upload the files/folders directly from local system or we create folders.

Step1: Creating S3 bucket

The screenshot shows the 'Create bucket' wizard on the AWS Management Console. In the 'General configuration' step, the user has selected the 'General purpose' bucket type and entered the bucket name 'bhargavi-b-balarami'. The 'Object Ownership' section is also visible, with the 'ACLs disabled (recommended)' option selected. In the second step, 'Block Public Access settings for this bucket', the 'Block all public access' checkbox is checked, and several other options like 'Block public access to buckets and objects granted through new access control lists (ACLS)' are listed but not selected. The status bar at the bottom indicates the session is in English (ENG) and the date is 28-10-2024.

While creating S3 bucket disable ACLs and public access.

The image consists of three vertically stacked screenshots of the AWS S3 console.

Screenshot 1: Create S3 bucket

This screenshot shows the "Create S3 bucket" wizard. The "Default encryption" section is open, showing options for SSE-S3, SSE-KMS, and Dual-layer server-side encryption with AWS KMS keys (DSS-E-KMS). The "Enable" option for SSE-KMS is selected. Other settings like Bucket Key and Advanced settings are also visible. A note at the bottom says: "After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings." A yellow "Create bucket" button is at the bottom right.

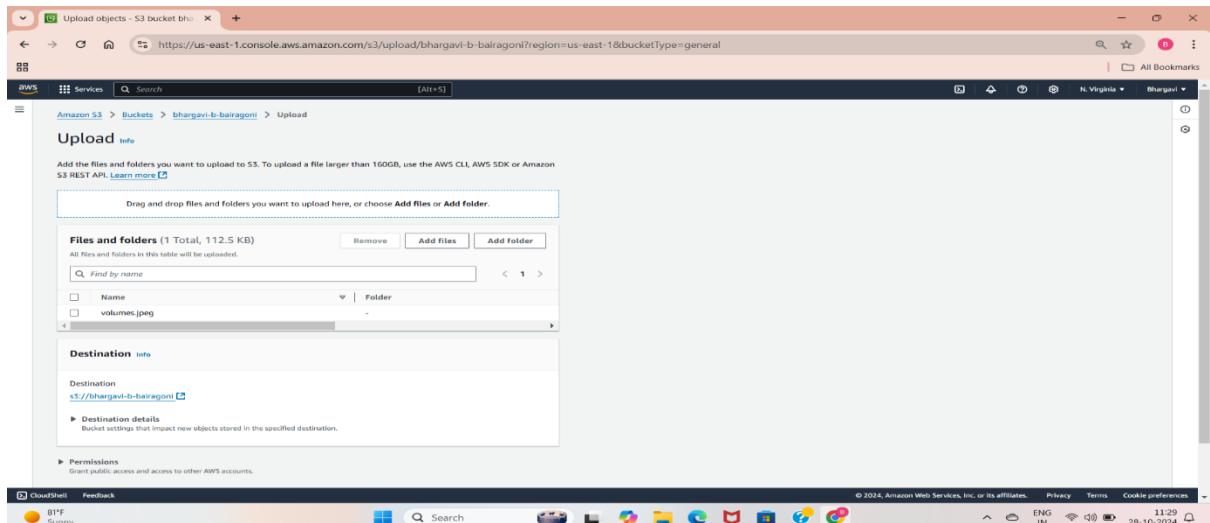
Screenshot 2: S3 buckets

This screenshot shows the "S3 buckets" page. It displays a green success message: "Successfully created bucket 'bhargavi-b-bhargavi'". Below this, it lists two buckets: "bhargavi-b-bhargavi" (created on Oct 28, 2024) and "elasticbeanstalk-us-east-1-905418365089" (created on July 30, 2024). The "General purpose buckets" tab is selected. A yellow "Create bucket" button is located at the top right of the list table.

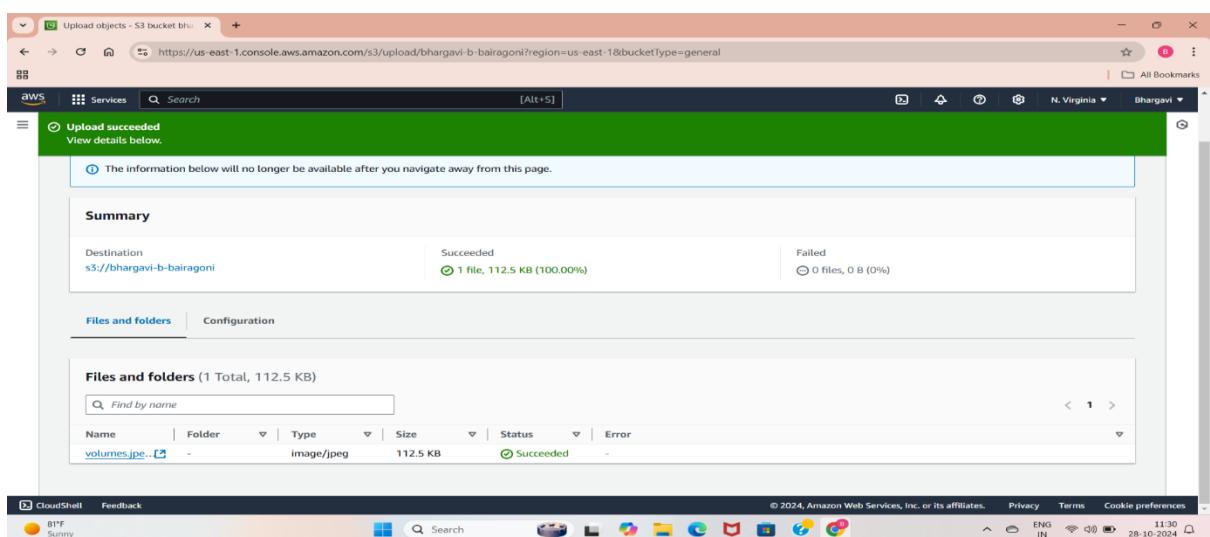
Screenshot 3: CloudShell Feedback

This screenshot shows the CloudShell interface with the feedback "81°F Sunny". At the bottom, there's a toolbar with various icons and the system status: ENG IN, 11:28, 28-10-2024.

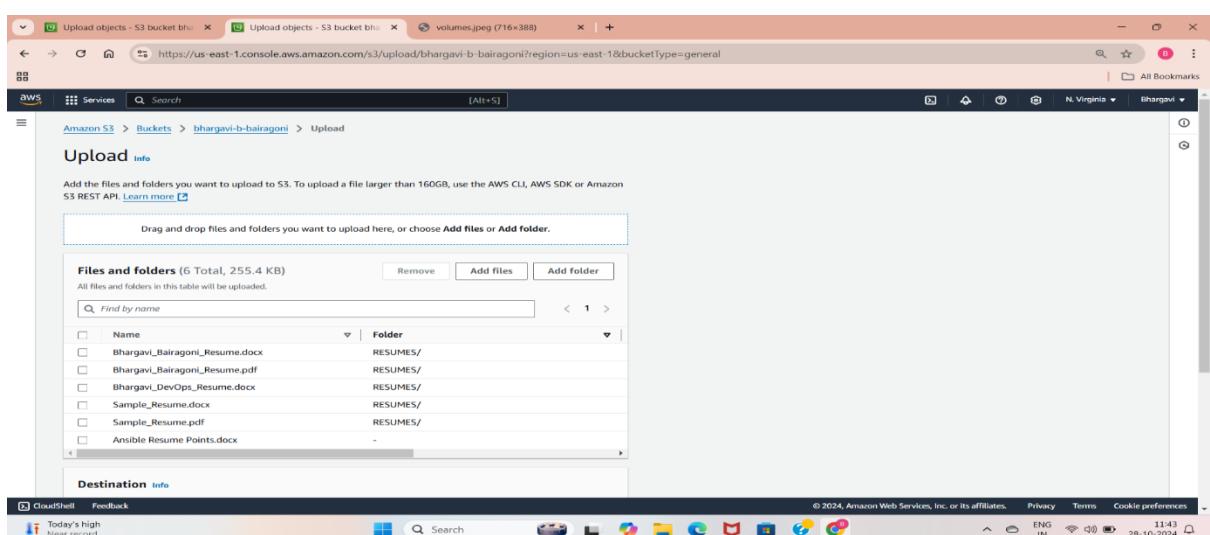
Created a bucket with the name of bhargavi-b-bhargavi in S3 using Create bucket button. Go inside bucket and check whether there are any objects or not.



Uploading the files into the s3 bucket using upload option.



Successfully uploaded files into the s3 bucket.



Uploading folder into the s3 bucket.

The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with various AWS services like CloudWatch Metrics, Lambda, and CloudWatch Logs. The main area displays the 'Object overview' for an object named 'volumes.jpeg'. Key details include:

- Owner:** bhargavibairagoni90
- AWS Region:** US East (N. Virginia) us-east-1
- Last modified:** October 26, 2024, 11:30:24 (UTC+05:30)
- Size:** 112.5 KB
- Type:** jpeg
- Key:** volumes.jpeg

Below the object overview, there's a section for 'Object management overview' which includes 'Bucket properties' and 'Management configurations'.

To access the objects and folders in s3 bucket we have object url paste the url in browser.

The screenshot shows a web browser window with the URL <https://bhargavi-b-bairagoni.s3.us-east-1.amazonaws.com/volumes.jpeg>. The page displays an XML error message:

```
<Error>
<Code>AccessDenied</Code>
<Message>Access Denied</Message>
<RequestId>22V4Q3AXP1C81YBT</RequestId>
<HostId>nB0x6+bChgR/FdZndt7V7nuMgPt4ZGvILxXGqgsKhQoR1u1z/iJWjldLdErBuzAJ3Dl+80s9o/E=</HostId>
</Error>
```

We are getting error while trying to access the object.

The screenshot shows the 'Permissions' tab for the 'bhargavi-b-bairagoni' bucket. Under the 'Permissions overview' section, it indicates that 'Block public access (bucket settings)' is turned on. A note states: "Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or ARNs. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on block all public access. These settings apply only to this bucket and its contents. If you have other buckets in your account, you must apply any of these settings, ensuring that your applications will work correctly without public access. If you require more level of public access to your buckets or objects, then you can implement this individual setting before to suit your specific storage use cases." Below this, there's a 'Bucket policy' section with a note: "The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts." At the bottom, a message says: "Public access is blocked because Block Public Access settings are turned on for this bucket. To determine which settings are turned on, check your Block Public Access settings for this bucket. Learn more about using Amazon S3 Block Public Access."

We can't access the object because we don't have access permissions because public access is blocked.

Enabling the Access Control lists.

Giving access to public by removing block public access.

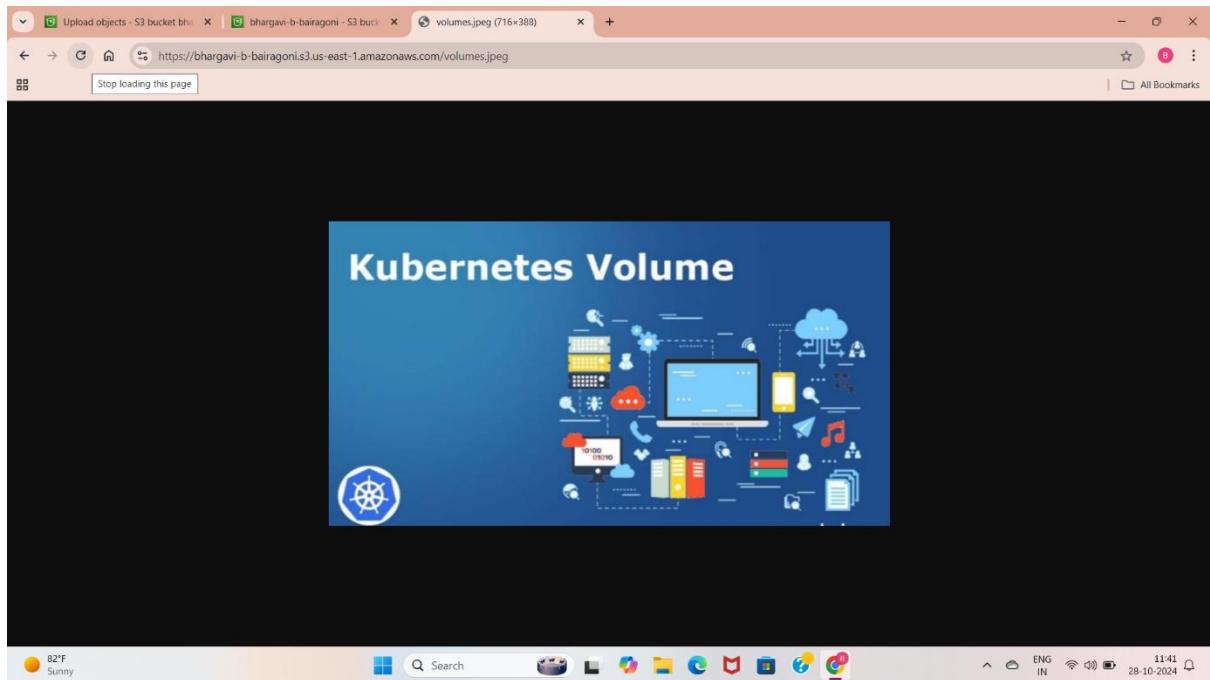
The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with options like Buckets, Access Grants, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Block Public Access settings for this account, Storage Lens, Dashboards, Storage Lens groups, AWS Organizations settings, Feature spotlight, and AWS Marketplace for S3. The main area displays a bucket named 'bhargavi-b-bairagoni'. Below the bucket name is a 'Objects' tab. A single object, 'volumes.jpeg', is listed in a table with columns for Name, Type, Last modified, and Size. The 'Last modified' column shows 'October 28, 2024, 11:30:24 (UTC+05:30)'. To the right of the table is a context menu with various actions such as Download, Copy, Move, and Make public using ACL.

We can access the objects by making them public. First we need to give public access and the enable access control list(ACL), select the object which you want to make public.

Here we can also make the object public by generating policy using JSON statement.

The screenshot shows the 'Make public' dialog box overlaid on the S3 console. At the top, it says 'Amazon S3 > Buckets > bhargavi-b-bairagoni > Make public'. Below that is a section titled 'When public read access is enabled and not blocked by Block Public Access settings, anyone in the world can access the specified objects.' A table titled 'Specified objects' lists the file 'volumes.jpeg'. At the bottom of the dialog is a large orange 'Make public' button.

Click on “make public” to that object to access that object.



After making object public using ACL am able to access the object.

A screenshot of the AWS S3 console. The left sidebar shows "Amazon S3 > Buckets > bhargavi-b-bairagoni". The main content area is titled "Properties" for the bucket "bhargavi-b-bairagoni". It includes sections for "Bucket overview", "Bucket Versioning" (disabled), and "Tags (0)". The "Bucket overview" section shows details like AWS Region (US East (N. Virginia) us-east-1), ARN (arn:aws:s3:::bhargavi-b-bairagoni), and Creation date (October 28, 2024, 11:28:10 (UTC+05:30)).

If the data is deleted wantedly or by mistakenly the data is lost if versioning is disabled.

Versioning is used as backup of the stored s3 bucket. After enabling versioning if any data is deleted we can restore them

If we made any change in the files and to store them in s3 we need to upload the files/folders again it will not updated automatically.

The screenshot shows the 'Edit Bucket Versioning' page for the 'bhargavi-b-bairagoni' bucket. The 'Bucket Versioning' section is open, showing the 'Enable' option selected. A note below states: 'After enabling Bucket Versioning, you might need to update your lifecycle rules to manage previous versions of objects.' The 'Multi-factor authentication (MFA) delete' section is also visible. At the bottom, there are 'Cancel' and 'Save changes' buttons.

Enabling the versioning we can able to save data from unintended data.

The screenshot shows the 'Objects' page for the 'bhargavi-b-bairagoni' bucket. The 'Objects (3)' section is displayed, listing three items: 'Ansible Resume Points.docx' (Type: docx, Last modified: October 28, 2024, Size: 28.5 KB, Storage class: Standard), a folder named 'RESUMES/' (Type: Folder), and 'volumes.jpeg' (Type: jpeg, Last modified: October 28, 2024, Size: 112.5 KB, Storage class: Standard). The 'Show versions' button is visible above the table. The bottom of the screen shows a taskbar with various icons and system status.

We can see the new option with the show versions which means versioning is enabled.

The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with navigation links like 'Buckets', 'Storage Lens', and 'Feature spotlight'. The main area displays the 'file1.txt' object within the 'bhargavi-b-bairagoni' bucket. The 'Properties' tab is selected, showing details such as the owner ('bhargavibairagoni590'), AWS Region ('US East (N. Virginia) us-east-1'), last modified ('October 28, 2024, 11:50:37 (UTC+05:30)'), size ('81.0 B'), type ('txt'), and key ('file1.txt'). To the right, there are buttons for 'Copy S3 URI', 'Download', 'Open', and 'Object actions'. The browser status bar at the bottom indicates the URL as <https://us-east-1.console.aws.amazon.com/s3/object/bhargavi-b-bairagoni?region=us-east-1&bucketType=general&prefix=file1.txt>.

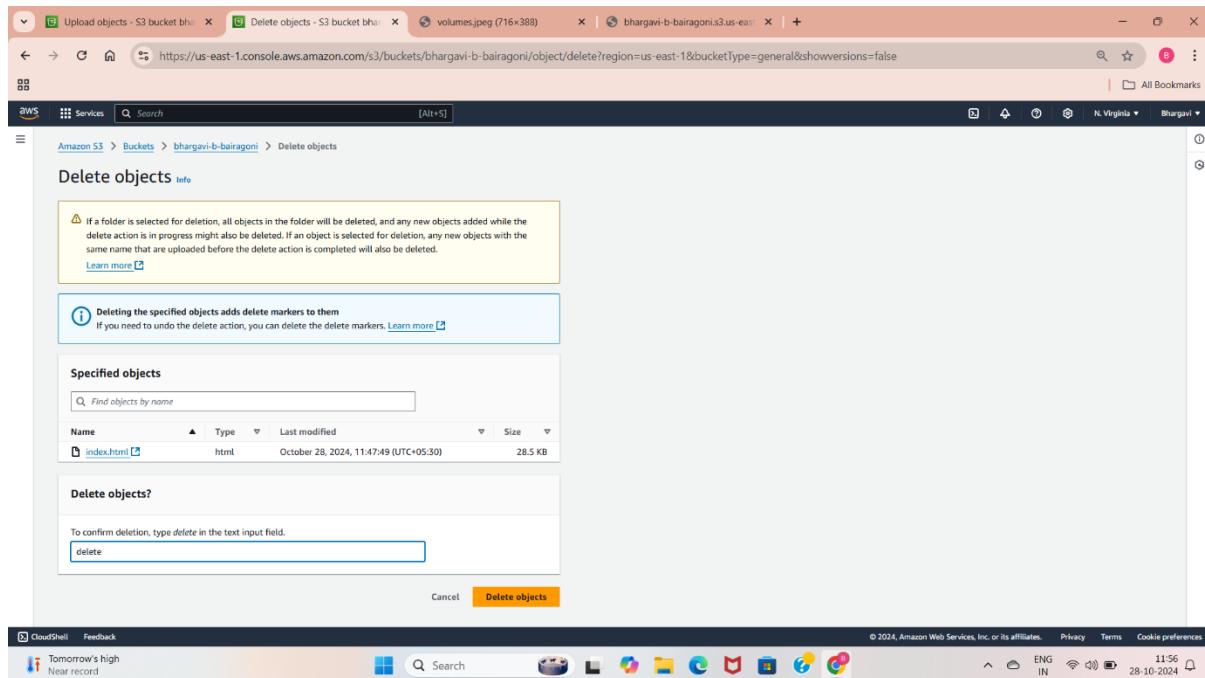
Created a text file and uploaded in s3 bucket with some content in the text file.

The screenshot shows a web browser window displaying the contents of the 'file1.txt' file. The text reads: 'Hi, This is Bhargavi. Today am starting my project which is given by Rabbani Sir'. The browser status bar at the bottom indicates the URL as <https://bhargavi-b-bairagoni.s3.us-east-1.amazonaws.com/file1.txt>.

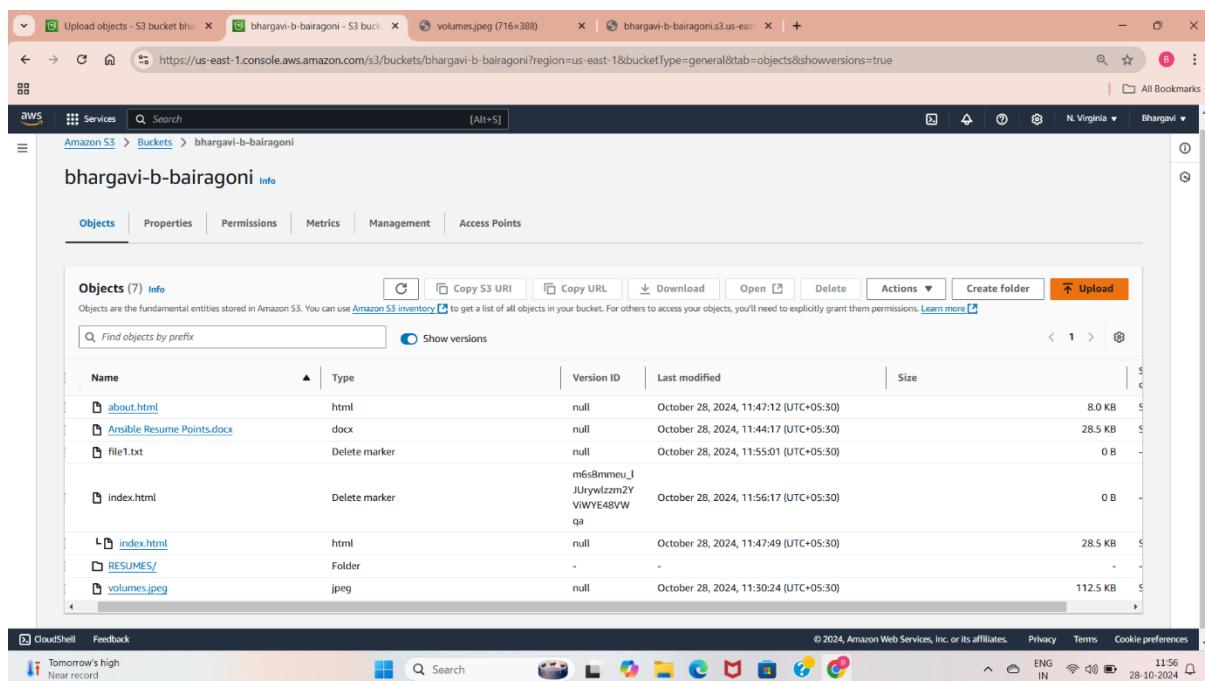
The screenshot shows a web browser window displaying the updated contents of the 'file1.txt' file. The text now includes an additional line: 'I have completed two tasks and currently am doing 3rd task which is s3'. The browser status bar at the bottom indicates the URL as <https://bhargavi-b-bairagoni.s3.us-east-1.amazonaws.com/file1.txt>.

Updated and uploaded the text file and we can see the changes by observing the above and this image.

Step3:



We can see a option of delete instead of delete maker.



Here it shows that versioning is enabled by seeing "show version" button. Here I have some objects. Before selecting show version option I have deleted joel.txt file but it is not permanently deleted we can identify this by seeing version objects. We can observe that the file is present but if we delete the joel.txt(type-txt) it will be permanently deleted.

If we delete the file with "Delete marker" we can able to restore the objects.

LAB4:EC2 INSTANCE

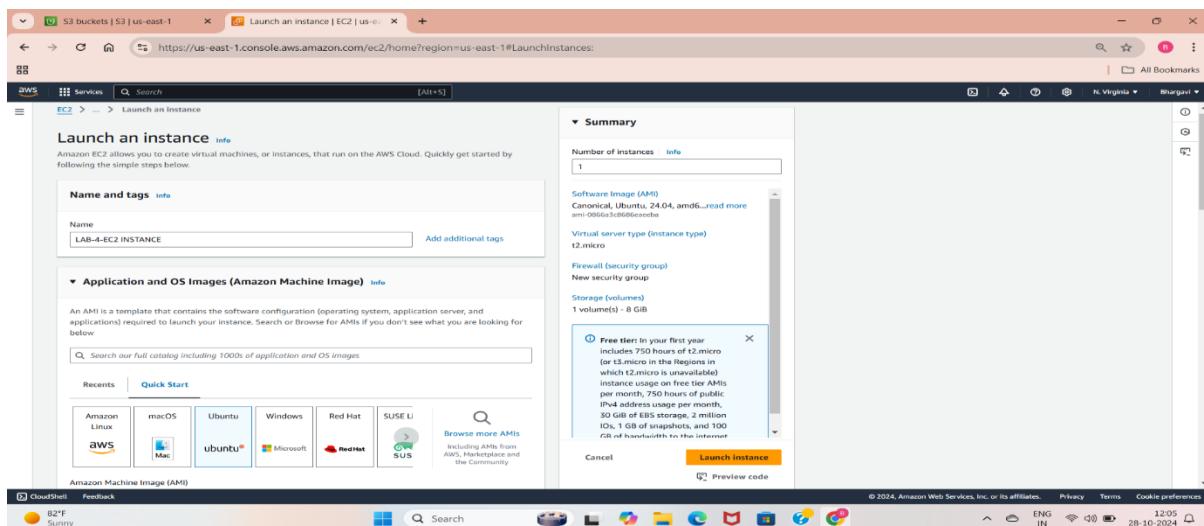
Elastic Compute Cloud is a service which provides computational power(speed to access any applications).

It will provide control over the virtual server, storage, security configurations and networking settings.

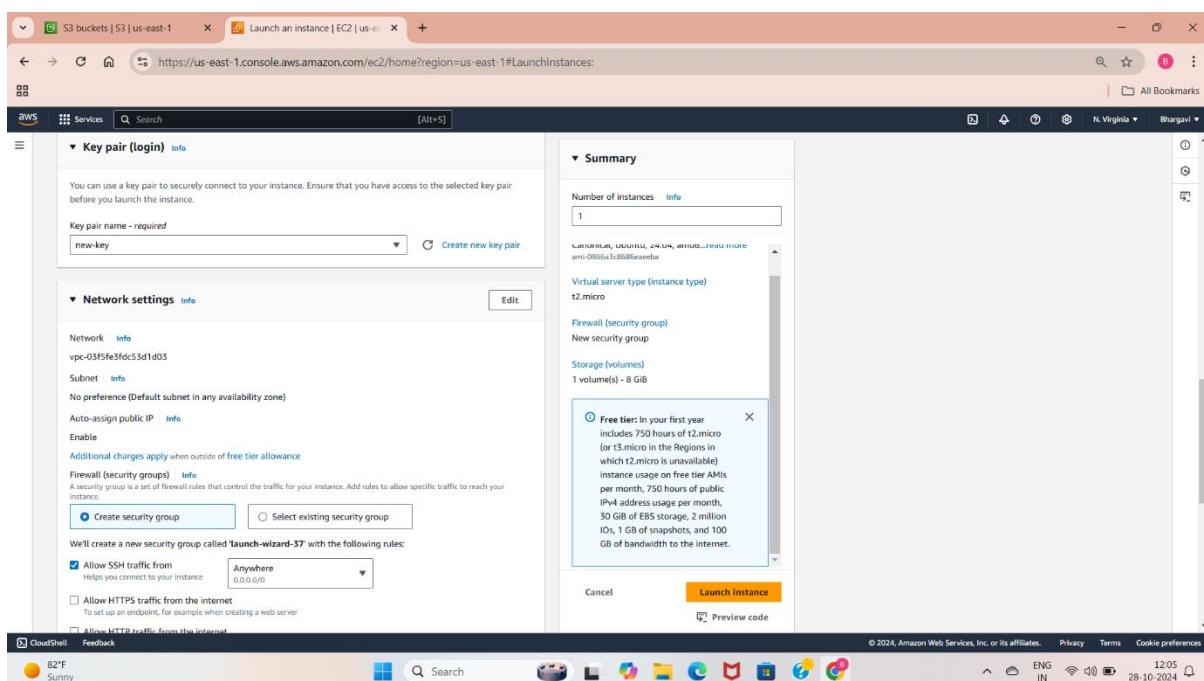
To access an ec2 instance we need to provide security with inbound and outbound rules.

Inbound: Allowing the traffic coming into your EC2 instance from external sources.

Outbound: Traffic going from your EC2 instance to the outside source.



Creating a ec2 instance with ubuntu, select t2.micro as instance type for ec2 instance.



Allowing ssh(22) to connect with putty and with default storage of 8gb.

The screenshot shows the AWS EC2 Instances page. A green banner at the top indicates "Successfully initiated termination (deletion) of i-08b2a28e375c6c08c". The main table lists two instances: "LAB-4-EC2 IN..." (instance ID i-0d45f24f524b65d3c) which is "Running" (t2.micro), and "Bhargavi" (instance ID i-08b2a28e375c6c08c) which is "Terminated" (t2.micro). The "Details" tab for the running instance is selected, showing its configuration: Instance ID i-0d45f24f524b65d3c, Public IPv4 address 18.234.167.216, Private IP DNS name (IPv4 only) ip-172-31-93-198.ec2.internal, Instance type t2.micro, VPC ID vpc-03f5fe3fd5d1d03, and Auto-assigned IP address 18.234.167.216 [Public IP].

We can see the above created instance is in running state.

The screenshot shows the AWS EC2 Instances page with a Putty configuration window overlaid. The Putty window is titled "PuTTY Configuration" and is set to connect to the instance's public IP address 18.234.167.216 on port 22 using SSH. The "Session" category is selected. Other tabs like "Appearance", "Translation", "Selection", "Colors", "Connection", "Data", "Proxy", "SSH", "Serial", "Telnet", "Rlogin", and "SUPDUP" are visible but not selected. The "Saved Sessions" section shows a single session named "Default Settings". The "VPC ID" field contains the value vpc-03f5fe3fd5d1d03. The status bar at the bottom right shows the date and time as 28-10-2024.

Connecting putty using public DNS and selecting the credentials of private key pair.

```

root@ip-172-31-93-198: ~
login as: ubuntu
Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Mon Oct 28 06:50:15 UTC 2024

System load: 0.0          Processes:           103
Usage of /: 22.9% of 6.71GB   Users logged in: 0
Memory usage: 20%          IPv4 address for enX0: 172.31.93.198
Swap usage: 0%             Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-93-198:~$ sudo su -
root@ip-172-31-93-198:~# time

real    0m0.000s
user    0m0.000s
sys     0m0.000s
root@ip-172-31-93-198:~# date
Mon Oct 28 06:50:39 UTC 2024
root@ip-172-31-93-198:~# exit

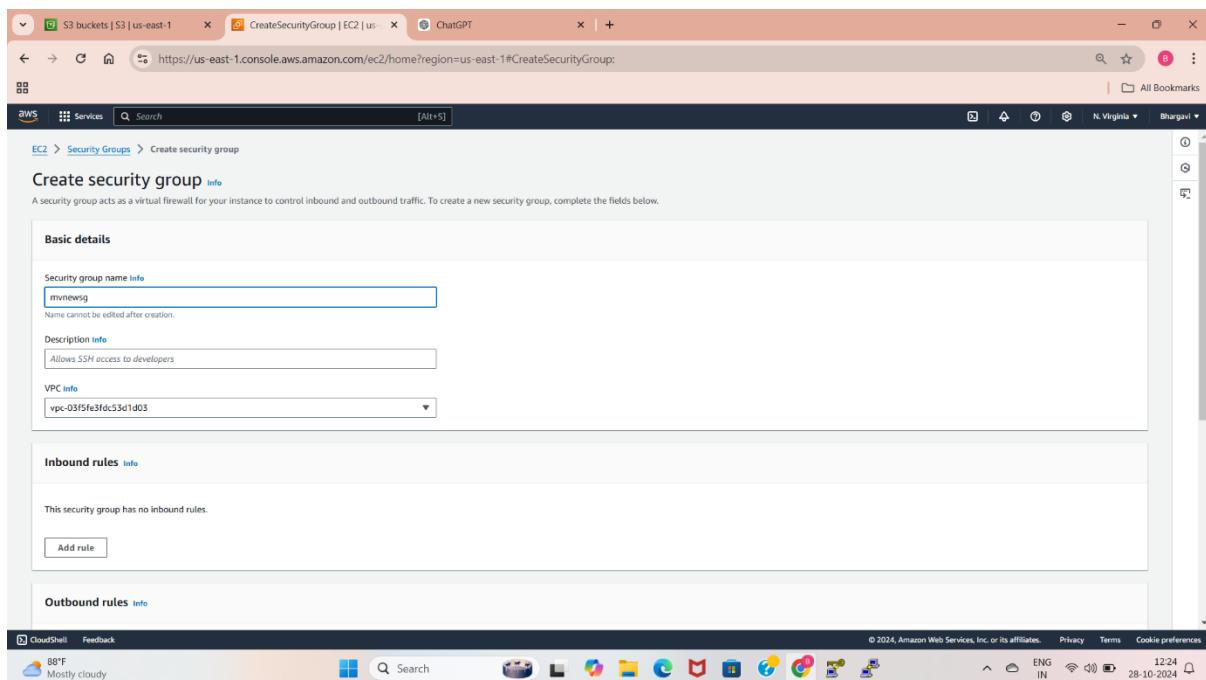
```

Accessing the local machine using putty using public ip, and private key pair and changed to root user using sudo su – command.

LAB-5: SECURITY GROUP:

A security group is a virtual firewall in cloud environments (like AWS) that controls inbound and outbound traffic to resources like EC2 instances.

To access any website, database we are having separate port numbers, once we allow the port number we can access.



Creating a security group with the name of “mvnewsg”.

The screenshot shows the AWS Management Console with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#SecurityGroup:groupId=sg-0b9706ebda4050365>. A green banner at the top indicates that the security group was created successfully. The main pane displays the details of the security group 'sg-0b9706ebda4050365 - mvnewsg'. It shows the security group name, ID, owner, and VPC ID. Under the 'Inbound rules' tab, there are two entries: one for port 80 (HTTP) and another for port 22 (SSH). The status bar at the bottom right shows the date as 28-10-2024.

Created a security group with the name “mvnewsg” and allowed port numbers 22(Secure shell) and 80(HTTP).

When I tried to change the cidr with 0.0.0.0/28 getting network error because it is invalid.

/28 tells how many addresses are in block and we get 16 ip addresses.

The screenshot shows the AWS Management Console with the URL <https://us-east-1.console.aws.amazon.com/ec2/instance-connect/ssh?addressFamily=ipv4&connType=standard&instanceId=i-026342341322a075080sUser=ubuntu®ion=us-east-1&sshPort=22>. A red banner at the top states: "Failed to connect to your instance. EC2 Instance Connect is unable to connect to your instance. Ensure your instance network settings are configured correctly for EC2 Instance Connect. For more information, see EC2 Instance Connect Prerequisites at <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-connect-prerequisites.html>". The status bar at the bottom right shows the date as 29-10-2024.

The screenshot shows the AWS Management Console with the URL [\\$case:tags:true%5Cclient:false;\\$regex:tags:false%5Cclient:false](https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:vv=3). A Putty Fatal Error dialog box is displayed, stating "Network error Connection timed out". The main pane shows the instance details for "ec2-18-207-200-70.compute-1.amazonaws.com - PuTTY". The status is "Running". The private IP address is listed as "ip-172-31-54-132.ec2.internal". The status bar at the bottom right shows the date as 29-10-2024.

Unable to connect the instance in putty and command prompt due to the cider with 0.0.0.0/28

LAB-6: VOLUMES AND SNAPSHOTS:

Volumes are used to store the data. By default when we launch a instance we get a storage volume as 8Gb but it is deleted when we delete the instance so data inside the volume is deleted. We create EBS volume to store data/persist data even after the termination of the instance. This can't be deleted until we delete it. It is used as backup.

Volumes can be regional specific and we need attach to the volume with the same availability zone. We can detach the volume from one instance and can be immediately attached to other instance with the same availability zone.

The screenshot shows the AWS Cloud9 IDE interface. The browser window displays the AWS Management Console with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Volumes>. The left sidebar shows navigation links for EC2, Volumes, and other services. The main content area shows a table titled "Volumes (1/2) Info" with two rows. The first row is for "volume5gb" (Volume ID: vol-0bd294469ff4529d3), which is 5 GiB gp3 type, created on 2024/10/29, and is available in us-east-1d. The second row is for a new volume (Volume ID: vol-0fb27e447dfa3...), which is 8 GiB gp3 type, created on 2024/10/29, and is in-use in us-east-1d. Below the table, a detailed view for "Volume ID: vol-0bd294469ff4529d3 (volume5gb)" is shown, including details like Size (5 GiB), Type (gp3), Volume status (Okay), and Created (Tue Oct 29 2024 10:40:20 GMT+0530 (India Standard Time)).

The screenshot shows the AWS Cloud9 IDE interface. The browser window displays the AWS Management Console with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#AttachVolume:volumeId=vol-0bd294469ff4529d3>. The left sidebar shows navigation links for EC2, Volumes, and other services. The main content area shows a form titled "Attach volume" under "Basic details". The "Volume ID" field is set to "vol-0bd294469ff4529d3 (volume5gb)". The "Availability Zone" is set to "us-east-1d". The "Instance" dropdown is set to "i-026345341522a0760". The "Device name" dropdown is set to "/dev/sdc". A note at the bottom states: "Never Linux kernels may rename your devices to /dev/vxvd through /dev/xvdy internally, even when the device name entered here (and shown in the details) is /dev/sdf through /dev/sdp." At the bottom right of the form are "Cancel" and "Attach volume" buttons.

After creating a EBS volume, we need to attach this to our instance so that the data can be stored in this volume.

The screenshot shows the AWS EC2 Instances page. A single instance, 'LAB-4-EC2 INSTANCE' (i-026343341322a0760), is listed as 'Running'. Two volumes are attached to it: 'vol-0fb27e47dfa3575b' (8GB) and 'vol-0bd294469ff4529d5' (5GB). Both volumes are labeled as 'Attached' and have an 'EBS' root device type. The interface includes tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. The Storage tab is currently selected. The bottom of the screen shows the Windows taskbar with various icons and system status.

Here we can see the two volumes with 8GB which is a default volume created with the instance and 5GB which is created manually to store data even after the instance termination.

We can see that the default volume and created volume is in the same availability zone(us-east-1a).

```

root@ip-172-31-34-132:~# mkdir /mnt/bhargavi_volume
root@ip-172-31-34-132:~# mount /mnt/bhargavi_volume /dev/xvdy
mount: /dev/xvdy: /mnt/bhargavi_volume is not a block device.
       dmesg(1) may have more information after failed mount system call.
root@ip-172-31-34-132:~# c
root@ip-172-31-34-132:~# mount /dev/xvdy /mnt/bhargavi_volume
root@ip-172-31-34-132:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root      6.8G  1.6G  5.2G  24% /
tmpfs          479M   479M  0     1% /dev/shm
tmpfs          192M  888K  191M  1% /run
tmpfs           5.0M   5.0M  0     0% /run/lock
/dev/xvda16    881M  76M  744M 10% /boot
/dev/xvda15    105M  6.1M  99M  6% /boot/efi
tmpfs          96M  12K  96M  1% /run/user/1000
/dev/xvdy      4.9G  8.0K  4.6G  1% /mnt/bhargavi_volume
root@ip-172-31-34-132:~# umount /mnt/bhargavi_volume
root@ip-172-31-34-132:~# mkdir /bhargavi_volume
root@ip-172-31-34-132:~# mount /dev/xvdy /bhargavi_volume
root@ip-172-31-34-132:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root      6.8G  1.6G  5.2G  24% /
tmpfs          479M   479M  0     1% /dev/shm
tmpfs          192M  888K  191M  1% /run
tmpfs           5.0M   5.0M  0     0% /run/lock
/dev/xvda16    881M  76M  744M 10% /boot
/dev/xvda15    105M  6.1M  99M  6% /boot/efi
tmpfs          96M  12K  96M  1% /run/user/1000
/dev/xvdy      4.9G  8.0K  4.6G  1% /bhargavi_volume
root@ip-172-31-34-132:~# lsblk
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0  7:0    0  25.2M  1 loop /snap/amazon-ssm-agent/7993
loop1  7:1    0  55.7M  1 loop /snap/core18/2829
loop2  7:2    0 38.8M  1 loop /snap/snapd/21759
xvda  202:0   0   8G  0 disk 
-xvda1 202:1   0   7G  0 part /
-xvda14 202:14  0   4M  0 part 
-xvda15 202:15  0 106M 0 part /boot/efi
-xvda16 259:0   0 913M 0 part /boot
xvdy  202:6144  0   5G  0 disk /bhargavi_volume
root@ip-172-31-34-132:~# cd /bhargavi_volume
root@ip-172-31-34-132:/bhargavi_volume# vim bhar
root@ip-172-31-34-132:/bhargavi_volume# ls -l
total 4
-rw-r--r-- 1 root root 59 Oct 29 05:59 bhar
root@ip-172-31-34-132:/bhargavi_volume# pwd
/bhargavi_volume
root@ip-172-31-34-132:/bhargavi_volume#

```

Check the volumes to see the EBS volume using list block devices and disk free

<df -h> using this we can see all the mounted file systems with human readable sizes.

<lblk> with this we can see all the mounted and unmounted volumes and shows the device path and the size.

we can find the volume of 5Gb and the path of volume is xvdi with the mount path.

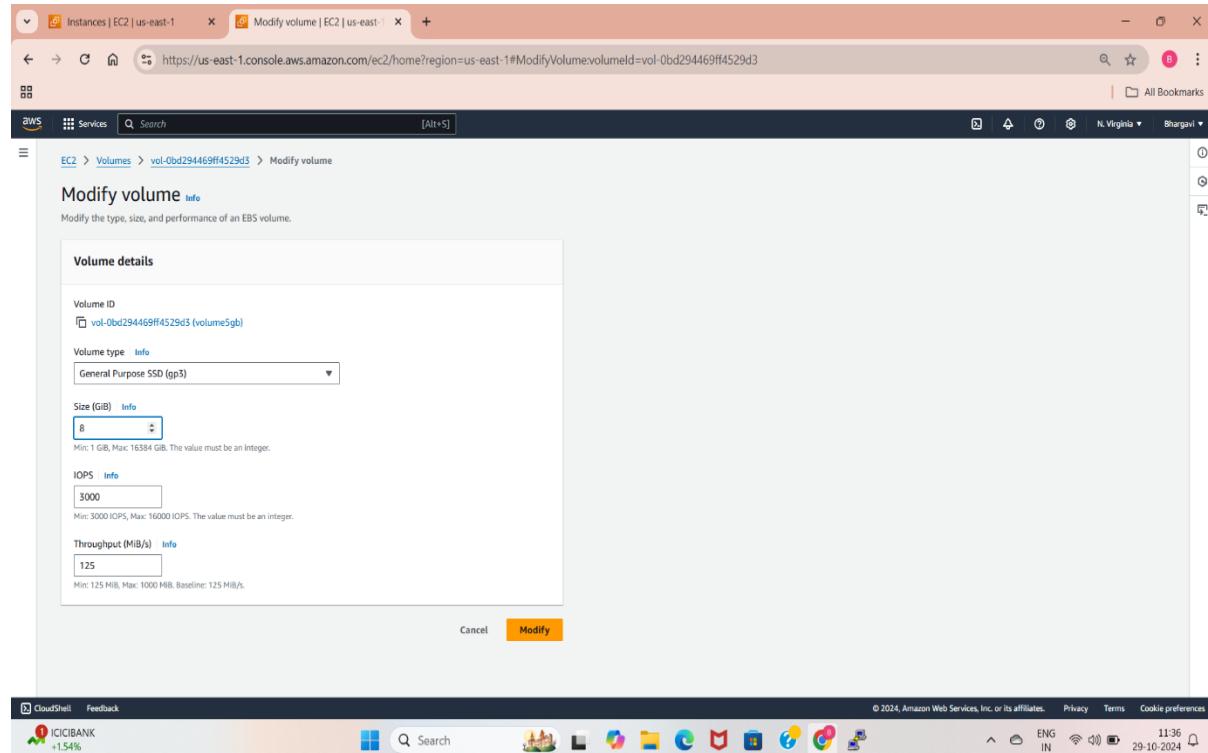
<mkfs -t ext4 /dev/xvdi> is used to create a new filesystem on a specified disk partition or volume.

Make filesystem using the type of file system with extended filesystem version 4 and the files created on the specific path(/dev/xvdi).

Create a folder to mount/attach the volume into that folder. Use the below command to attach the volume to created folder so it can be used as regular folder to store the files and the data.

<mount /dev/xvdi(path) /volume(foder)>

Go inside the created folder and create any files that will be stored in EBS volume.



Changing the size of EBS volume to 8GB.

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with 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, API Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, and Key Pairs. The main content area shows a table of instances. One instance is selected: i-026343341322a0760 (LAB-4-EC2 INSTANCE). The instance details show it's a t2.micro instance with a Public IPv4 of 18.207.200.70. Below the instance table, there's a tabbed view for Details, Status and alarms, Monitoring, Security, Networking, Storage (selected), and Tags. Under Storage, it shows Root device details (Root device name: /dev/sda1, Root device type: EBS) and Block devices (Volume ID: vol-0fb27e47dfa3575b, Device name: /dev/sda1, Volume size (GiB): 8, Attachment status: Attached, Attachment time: 2024/10/29 10:18 GMT+5:30, Encrypted: No, KMS key ID: -, Delete on termination: Yes). There's also a Volume monitoring section with a table showing one entry.

We can also increase the size of volume by modifying the size of the EBS volume. Here am modifying the 5gb volume to 8gb volume. When I modify the volume it will be automatically updated at the instance.

```

root@ip-172-31-34-132: ~
root@ip-172-31-34-132: ~# login as: ubuntu
root@ip-172-31-34-132: ~# Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1016-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Tue Oct 29 06:11:18 UTC 2024

System load: 0.0 Processes: 120
Usage of /: 23.2% of 6.71GB Users logged in: 1
Memory usage: 22% IPv4 address for enX0: 172.31.34.132
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

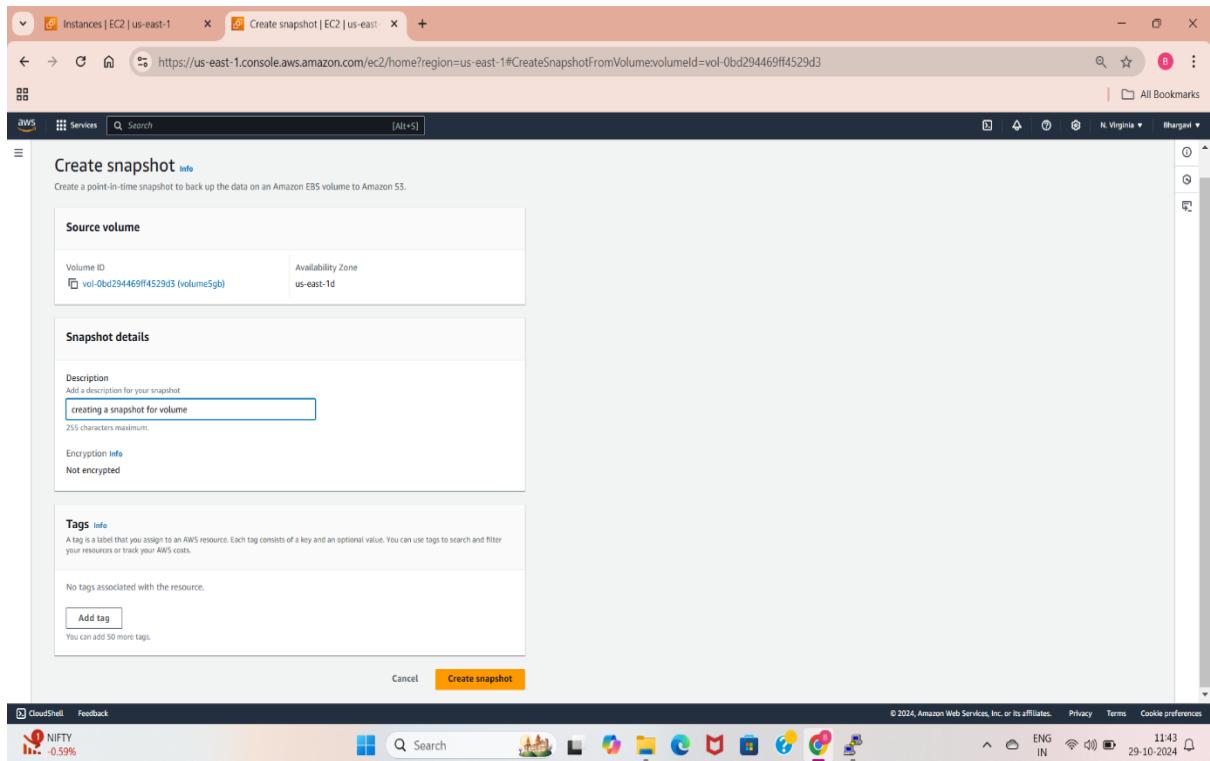
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

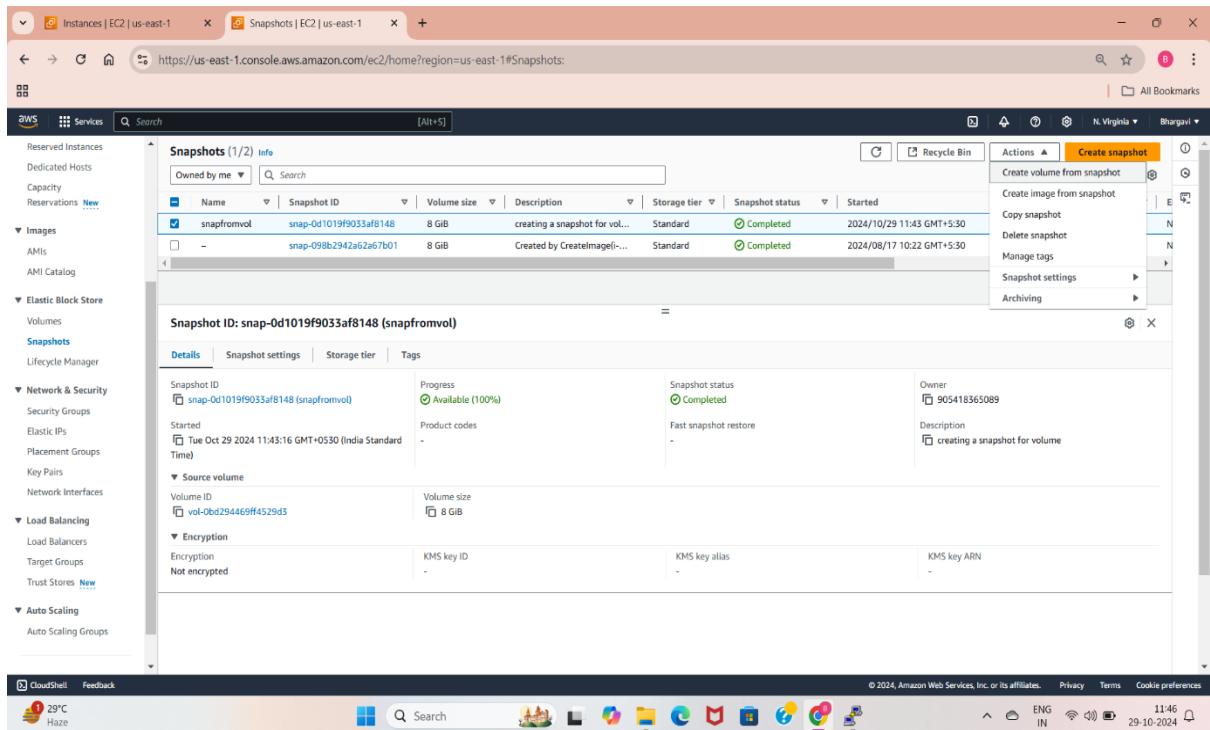
Last login: Tue Oct 29 05:49:56 2024 from 49.206.36.166
ubuntu@ip-172-31-34-132: ~$ sudo su -
root@ip-172-31-34-132: ~# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0    7:0    0  25.2M  1 loop /snap/amazon-ssm-agent/7993
loop1    7:1    0  55.7M  1 loop /snap/core18/2829
loop2    7:2    0  38.8M  1 loop /snap/snappy/21759
xvda   202:0    0   8G  0 disk
-xvda1  202:1    0   7G  0 part /
-xvda14 202:14   0   4M  0 part
-xvda15 202:15   0 106M  0 part /boot/efi
-xvda16 259:0    0  913M  0 part /boot
xvdy   202:6144   0   8G  0 disk /bhargavi_volume
                           /root/bhargavi_volume
root@ip-172-31-34-132: ~#

```

We can check the modification in the size of volume using list block devices and disk free.



Snapshot is a copy/backup of EBS volume. We can restore a snapshot to create a new EBS volume in the same or another region, which is useful for disaster recovery, data migration, or duplicating environments.



Create a snapshot from the previously created volume so that we can attach the volume in the same or different availability zone.

The screenshot shows the AWS Cloud Console interface. The left sidebar is collapsed. The main area displays a table titled 'Volumes (1/2) [info]'. The table has columns for Name, Volume ID, T., S..., I..., Thro..., Snap..., Created, Available..., Volume state, Alarm status, and Attached resources. Two rows are present: one for a volume named 'vol-0fb27e447dla3...' and another for a volume named 'vol'. The 'vol' row is highlighted with a blue selection bar. Below the table, a detailed view for 'Volume ID: vol-0cbfd4af04372446a (vol)' is shown. The 'Details' tab is selected, displaying information such as Volume ID (vol-0cbfd4af04372446a), Size (8 GiB), Type (gp3), Volume state (In-use), Availability Zone (us-east-1d), Created (Tue Oct 29 2024 11:48:42 GMT+0530 (India Standard Time)), and other details. The bottom of the screen shows the Windows taskbar with various pinned icons.

Creating a new volume from snapshot with 8gb storage.

The screenshot shows the 'Create volume' wizard in the AWS Cloud Console. The first step, 'Create volume | EC2 | us-east-1', is selected. The form includes fields for 'Volume type' (gp3), 'Volume size' (8 GiB), 'Availability zone' (set to 'us-east-1a'), and 'Encryption' (unchecked). Below these, there's a 'Tags - optional' section with a note about tags being optional labels for resources. A 'Snapshot summary' section shows a note to click refresh to view backup information. At the bottom, there are 'Cancel' and 'Create volume' buttons. The bottom of the screen shows the Windows taskbar with various pinned icons.

We can create a new volume from the snapshot so that we can change the availability zone of the volume and attach to the instance of different availability zone.

The screenshot shows the AWS EC2 Instances page. A single instance, "LAB-4-EC2 IN...", is listed as "Running". It has an "t2.micro" instance type, "us-east-1d" availability zone, and a public IP of "18.207.200.70". The instance is attached to an EBS volume, "/dev/sda1", which is listed in the "Block devices" section. The volume monitoring section shows no widget on the dashboard.

We can see the volume which is created from snapshot is attached to our instance.

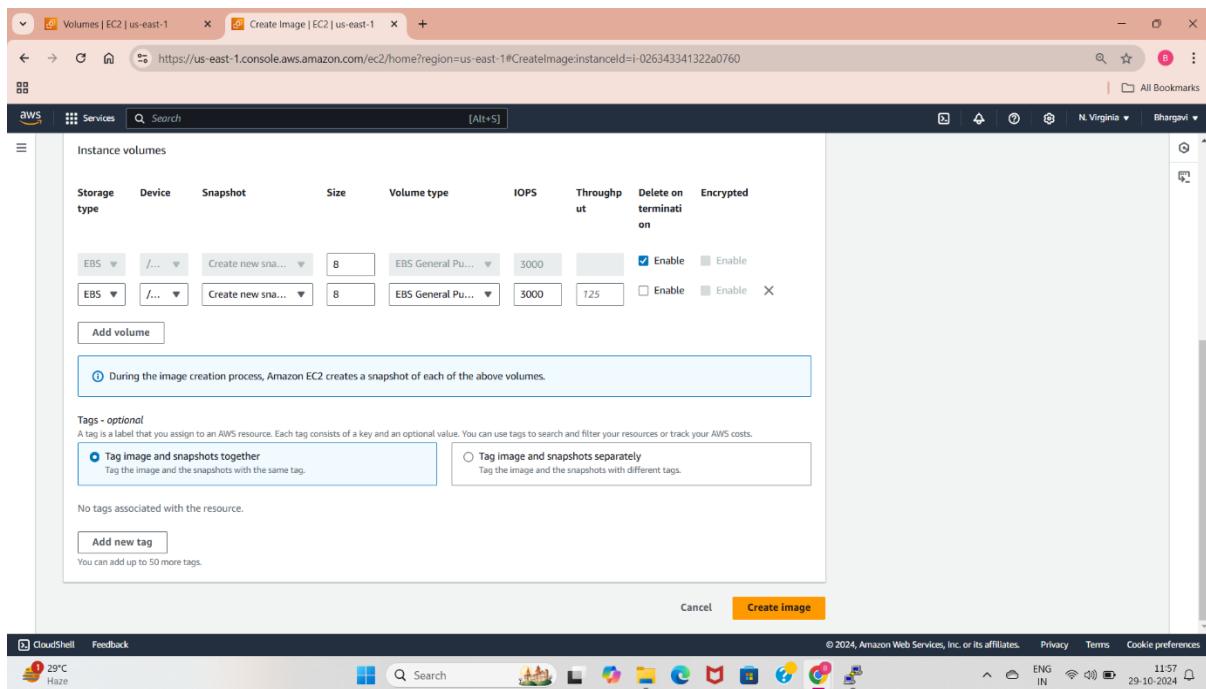
LAB:7:AMI

Amazon Machine Image is a pre-configured virtual image contains all the necessary components to launch an ec2 instance. For example if we want to launch 10 ec2 instances with same configuration we can create an AMI from first instance and launch other instances.

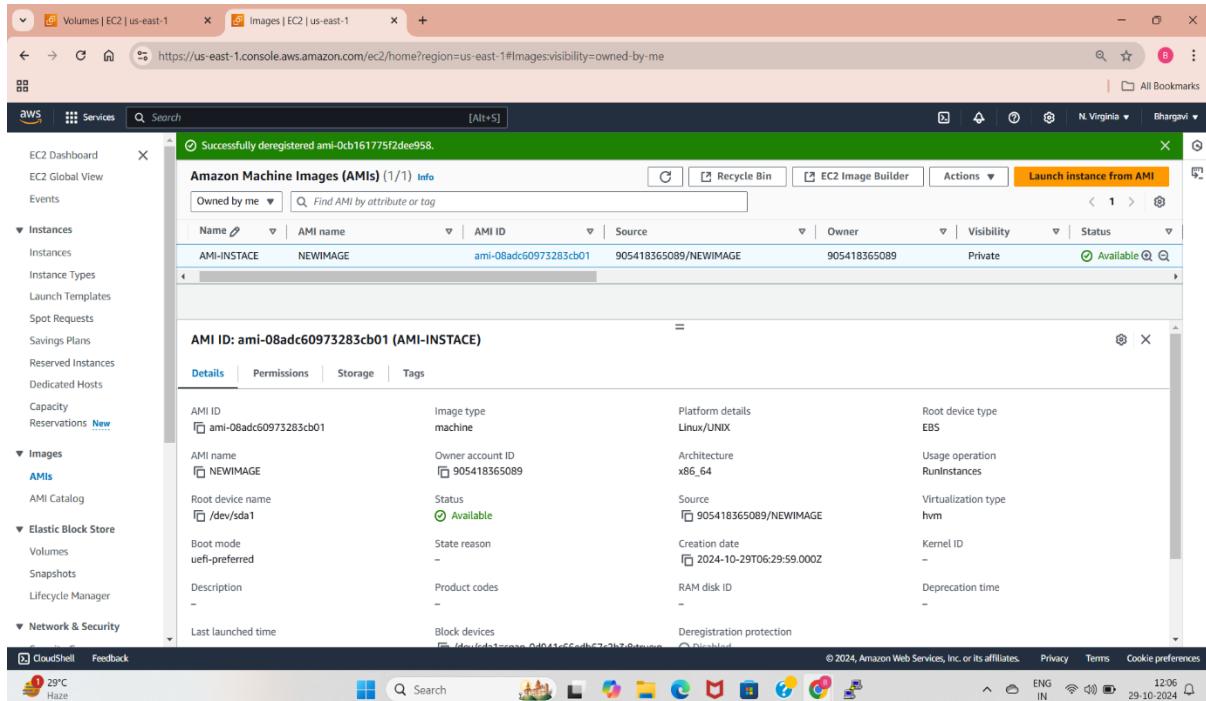
The screenshot shows the "Create image" page for the instance "i-026343341322a0760". The "Image name" field is set to "AMI". The "Reboot instance" checkbox is checked. In the "Instance volumes" section, the "Storage type" is "EBS", and the volume "8" is selected. The "Throughput" dropdown is set to "3000". The "Delete on termination" and "Encrypted" checkboxes are checked. The "Image description" field is empty.

Creating a AMI from already launched ec2 instance.

First we need to launch a ec2 instance with specific configurations.

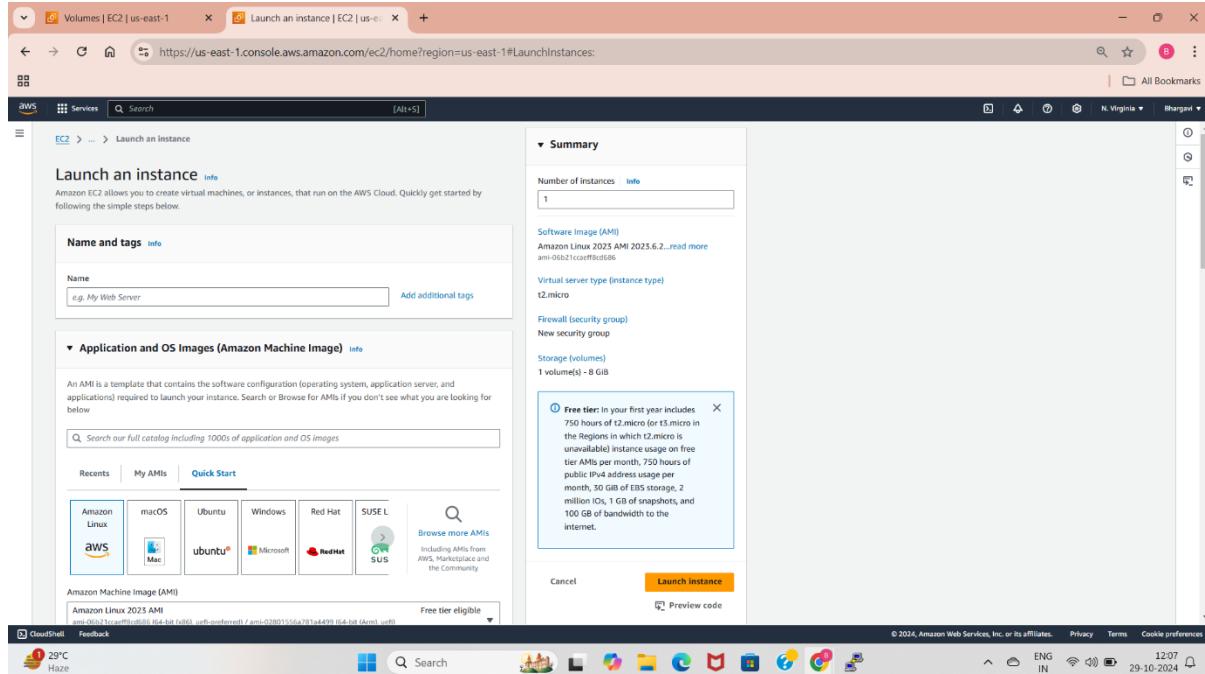


Stop the ec2 instance and create a Amazon Machine Image from the ec2 instance. Stopping the instance is necessary for the data integrity. When an EC2 instance is running, data may be in a volatile(it can change) state. Applications may be writing to disk, or processes may be using in-memory data. If you create an AMI while the instance is running, you risk capturing incomplete or inconsistent data.



Now the AMI consists all the configurations(operating system, storage, instance type) and settings of that instance.

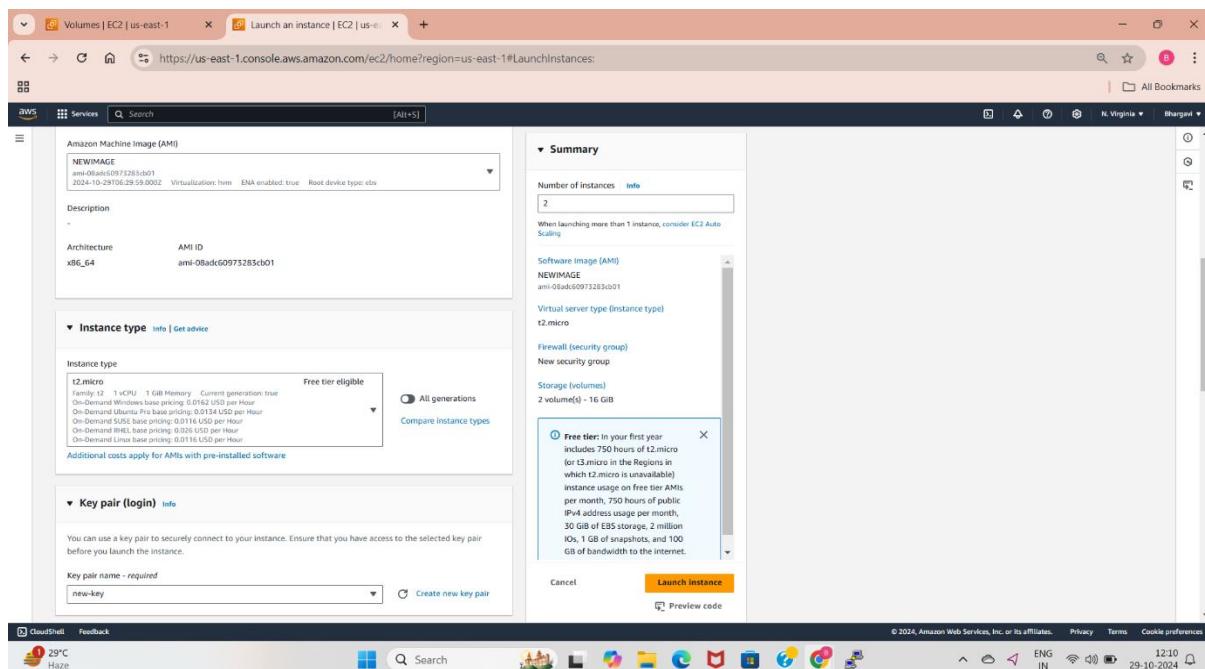
Here we can see the created AMI, by selecting the previously created image we can launch any number of instances which requires the same configurations and reduce the time to create instances.

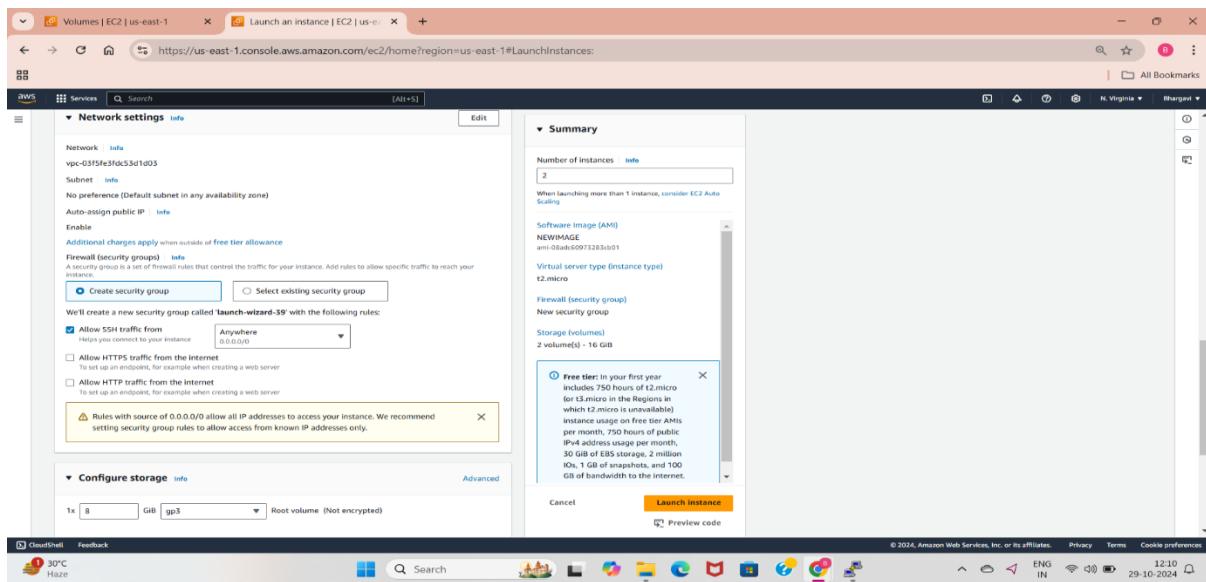


LAB:8-LOAD BALANCER:

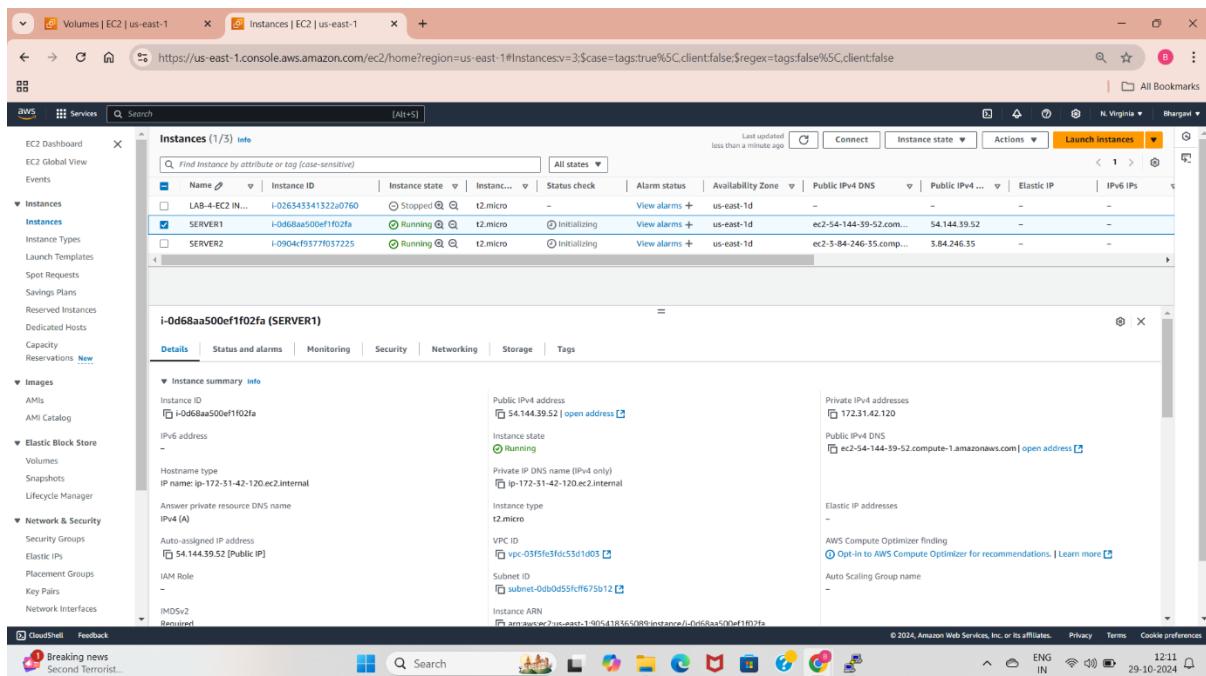
Load balancer is used to distributes the incoming traffic equally across the multiple servers. To reduce the load on the server we use load balancer so that all the servers will be in available state and preventing the failing of the server.

Create two ec2 instances and install nginx server on one machine and apache2 on server 2.





Creating two instances by allowing SSH and HTTP.



Install Httpd in one server and nginx in other server, start the two services.

Next copy the private ipaddresses of these instances and paste in the browser to access these applications.

Install the httpd – `apt install apache2 -y`

Start the httpd by – `systemctl start apache2`

Install the nginx by using

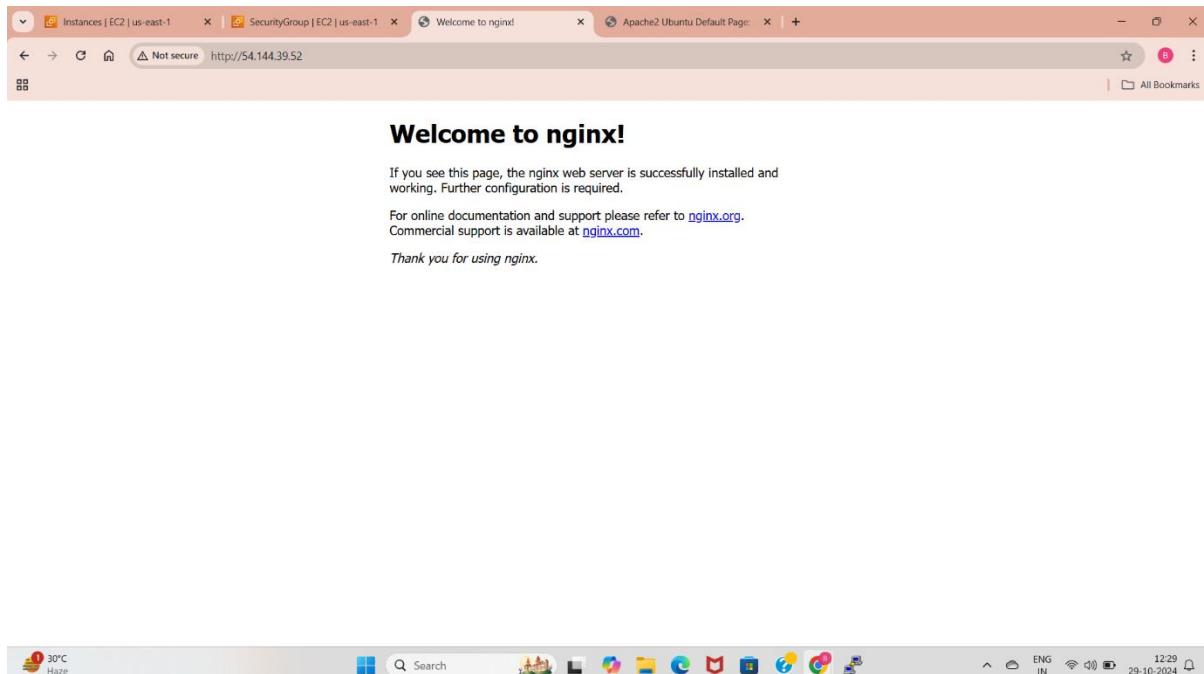
`apt install nginx -y`

`systemctl start nginx`

Access both the server over browser and check if their web page is visible.



Here we can see the httpd home page by copying the public ip address. Without allowing http port number we cannot access this webpages.



Here we can see the nginx home page by copying the public ip of other instance.

The screenshot shows the AWS EC2 Target groups page. A success message at the top states "Successfully created the target group LoadTarget. Anomaly detection is automatically applied to all registered targets. Results can be viewed in the Targets tab." The main section displays the "LoadTarget" configuration with the following details:

Target type	Protocol / Port	Protocol version	VPC
Instance	HTTP: 80	HTTP1	vpc-03f5fc3fd053d103
IP address type	IPv4		
0 Total targets	0 Healthy	0 Unhealthy	0 Unused
	0 Anomalous		0 Initial
			0 Draining

The "Targets" tab is selected. Below it, the "Registered targets (0) Info" section indicates "No registered targets" and "You have not registered targets to this group yet." There is a "Register targets" button.

To create a load balancer first we need to register the targets. Here we need to include the above created instances in target groups.

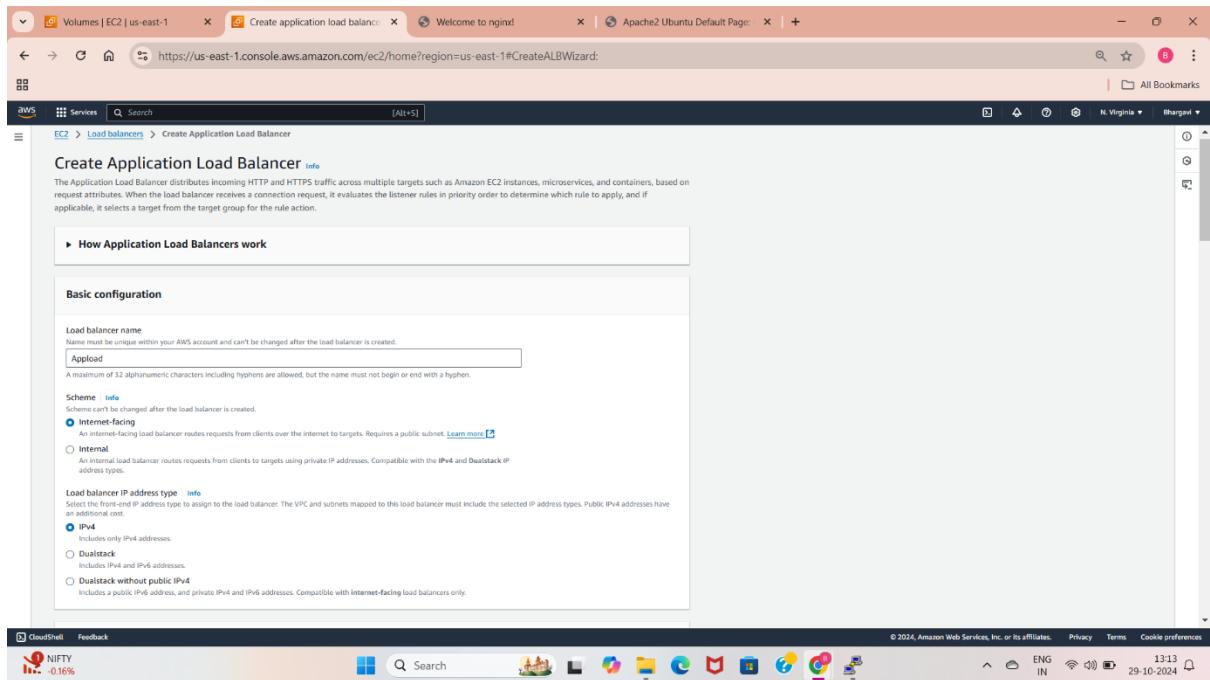
The screenshot shows the AWS EC2 Target groups page. A success message at the top states "One target registered successfully to LoadTarget." The main section displays the "Target group: LoadTarget" configuration with the following details:

Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
LoadTarget	arn:aws:elasticloadbalanc...	80	HTTP	Instance	None associated	vpc-03f5fc3fd053d103

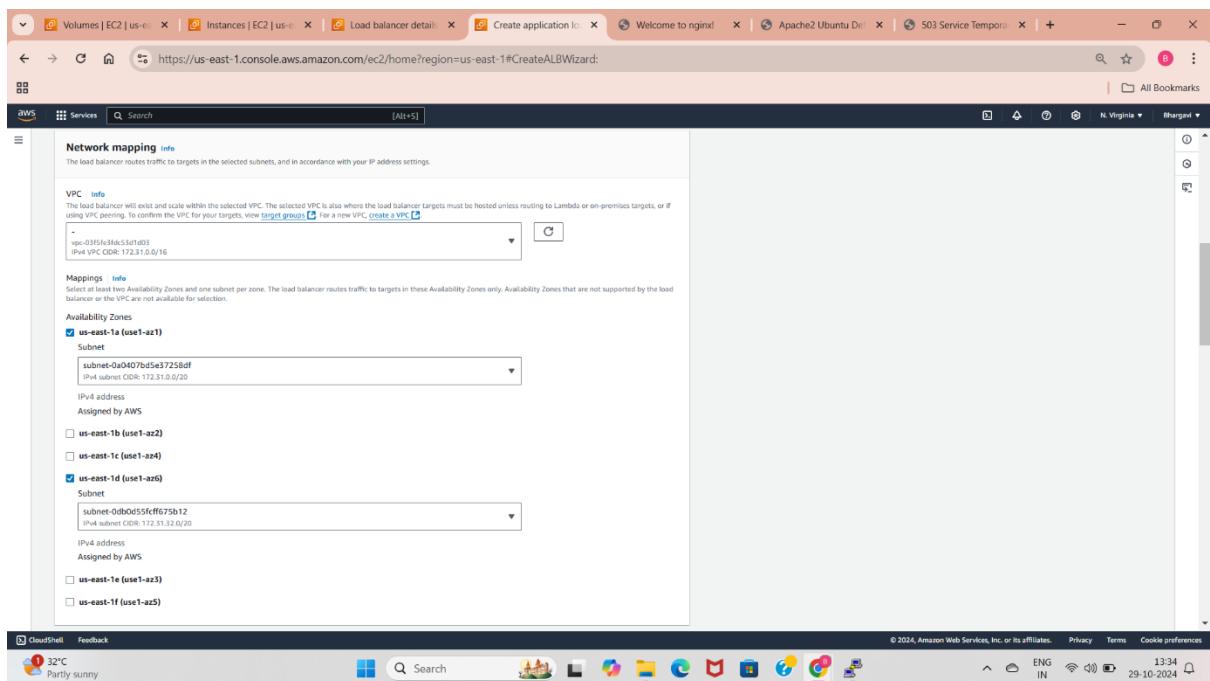
The "Targets" tab is selected. Below it, the "Registered targets (2) Info" section lists two instances:

Instance ID	Name	Port	Zone	Health ...	Health status de...	Launch...	Anomaly detection result
i-0904c98377f05...	SERVER2	80	us-east-1...	Unused	Target group is n...	October 2...	Normal
i-0d5aa500f11...	SERVER1	80	us-east-1...	Unused	Target group is n...	October 2...	Normal

Here we can see the two instances in the registered targets as a target group.



Go to Load balancers tab in the left side and create a load balancer with the name of AppLoad.



We need to select the available zones according to instance available zone. Select atleast two available zones.

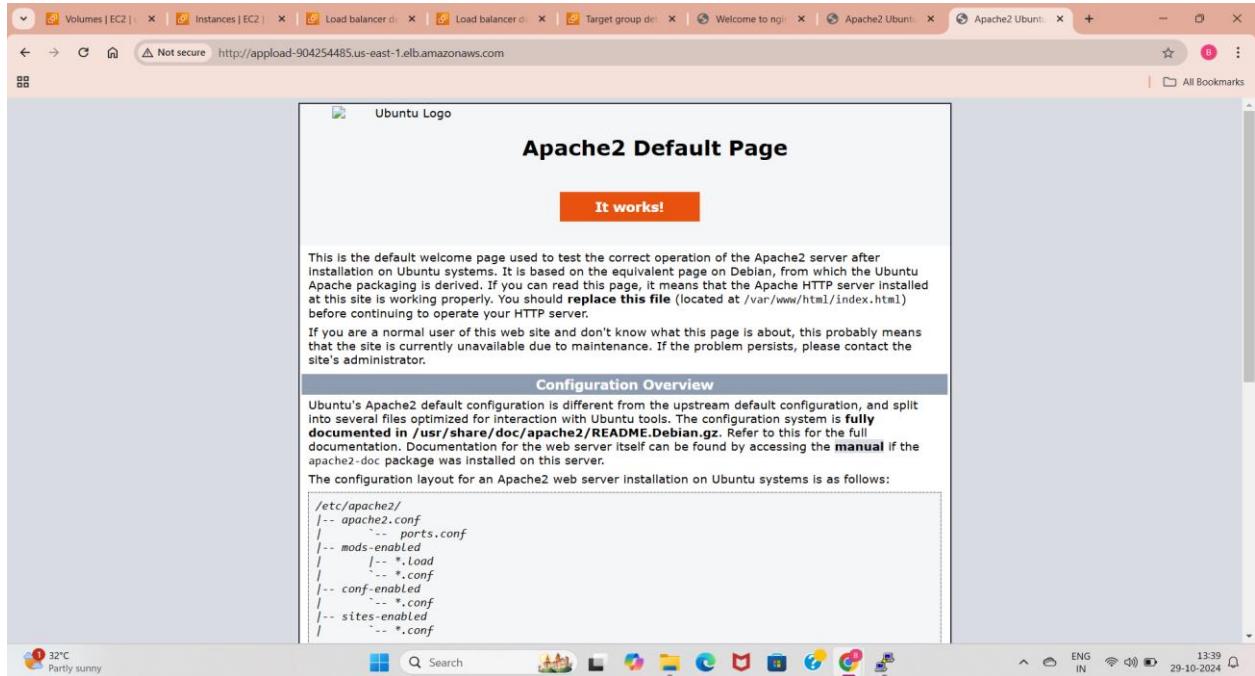
The screenshot shows the AWS CloudFront console with the URL <https://us-east-1.console.aws.amazon.com/cloudfront/home?region=us-east-1#CreateALBWizard>. The 'Listeners and routing' section is displayed, showing a listener for port 80 forwarding traffic to a target group named 'LoadTarget'. The 'LoadTarget' target group is defined as 'Target type: Instance, IPv4'.

While creating load balancer we need to specify the instances to which our load should be distributed. Am selecting the above created load balancer that is “LoadTarget”.

The screenshot shows the AWS CloudFront console with the URL <https://us-east-1.console.aws.amazon.com/cloudfront/home?region=us-east-1#CreateALBWizard>. The 'Listeners and rules' tab is selected, showing a single rule for port 80 forwarding to the 'LoadTarget' target group. The 'Forward to target group' dropdown is set to 'LoadTarget - 1 (100%)'.

We can see the created load balance with the load balancer url and target group. To access or check whether the load is distributing between the servers we need to use the url and increase the load by refreshing the website.

Access the load balancer link over the browser and hit it a couple of times. Check if both the webpages nginx and apache2 visible.



Here we can see the two web pages using the load balancer url by refreshing the web page

```

root@ip-172-31-42-120:~/project/sperring-html
inflating: sperring-html/images/experience-img.jpg
inflating: sperring-html/images/f1.png
inflating: sperring-html/images/f2.png
inflating: sperring-html/images/f3.png
inflating: sperring-html/images/f4.png
inflating: sperring-html/images/fb.png
inflating: sperring-html/images/freelance-img.jpg
inflating: sperring-html/images/icon-camera.png
inflating: sperring-html/images/icon-linkedin.png
inflating: sperring-html/images/icon-location.png
inflating: sperring-html/images/icon-mail.png
inflating: sperring-html/images/icon-menu.png
inflating: sperring-html/images/icon-next-angle.png
inflating: sperring-html/images/icon-prev-angle.png
inflating: sperring-html/images/icon-quote.png
inflating: sperring-html/images/icon-search-icon.png
inflating: sperring-html/images/icon-slider-img.png
inflating: sperring-html/images/icon-twitter.png
inflating: sperring-html/images/icon-youtube.png
inflating: sperring-html/index.html
creating: sperring-html/.gitignore
inflating: sperring-html/js/bootstrap.js
inflating: sperring-html/js/custom.js
inflating: sperring-html/js/jquery-3.4.1.min.js
inflating: sperring-html/work.html
root@ip-172-31-42-120:~/project# ll
total 5068
drwxr-xr-x 3 root root 4096 Oct 29 09:39 .
drwxr-xr-x 6 root root 4096 Oct 29 09:38 ..
drwxr-xr-x 5 root root 4096 Sep 16 2020 sperring-html/
-rw-r--r-- 1 root root 557415 Oct 29 09:44 sperring.zip
root@ip-172-31-42-120:~/project# cd sperring-html/
root@ip-172-31-42-120:~/project/sperring-html# ll
total 80
drwxr-xr-x 5 root root 4096 Sep 16 2020 .
drwxr-xr-x 3 root root 4096 Oct 29 09:46 ..
-rw-r--r-- 1 root root 10108 Jul 28 2020 about.html
-rw-r--r-- 1 root root 9824 Jul 28 2020 category.html
drwxr-xr-x 2 root root 4096 Sep 16 2020 css/
drwxr-xr-x 2 root root 4096 Sep 16 2020 images/
-rw-r--r-- 1 root root 23212 Jul 28 2020 index.html
drwxr-xr-x 2 root root 4096 Sep 16 2020 js/
-rw-r--r-- 1 root root 11825 Jul 28 2020 work.html
root@ip-172-31-42-120:~/project/sperring-html# mv * /var/www/html
root@ip-172-31-42-120:~/project/sperring-html# 

```

```

root@ip-172-31-42-20:~/project/mediplus-lite
inflating: mediplus-lite/js/jquery-migrate-1.0.0.js
inflating: mediplus-lite/js/jquery-ui.min.js
inflating: mediplus-lite/js/jquery.counterup.min.js
inflating: mediplus-lite/js/jquery.magnific-popup.min.js
inflating: mediplus-lite/js/jquery.min.js
inflating: mediplus-lite/js/jquery.nav.js
inflating: mediplus-lite/js/jquery.scrolltop.min.js
inflating: mediplus-lite/js/jquery.tether.min.js
inflating: mediplus-lite/js/jquery-active.js
inflating: mediplus-lite/js/niceselect.js
inflating: mediplus-lite/js/owl-carousel.js
inflating: mediplus-lite/js/popper.min.js
inflating: mediplus-lite/js/slicknav.min.js
inflating: mediplus-lite/js/steller.js
inflating: mediplus-lite/js/wilt.jquery.min.js
inflating: mediplus-lite/js/wilt.min.js
creating: mediplus-lite/mail/
inflating: mediplus-lite/mail/mail.php
inflating: mediplus-lite/portfolio-details.html
inflating: mediplus-lite/style.css
root@ip-172-31-42-20:~/project# ll
total 5068
drwxr-xr-x 3 root root 4096 Oct 29 09:39 .
drwxr-xr-x 6 root root 4096 Oct 29 09:38 ..
drwxr-xr-x 7 root root 4096 Oct 29 09:39 mediplus-lite/
-rw-r--r-- 1 root root 5175999 Oct 29 09:37 mediplus-lite/
root@ip-172-31-42-20:~/project# cd mediplus-lite/
root@ip-172-31-42-20:~/project/mediplus-lite# ll
total 220
drwxr-xr-x 7 root root 4096 Oct 29 09:39 .
drwxr-xr-x 3 root root 4096 Oct 29 09:39 ..
drwxr-xr-x 1 root root 12159 Jun 12 2020 404.html
-rw-r--r-- 1 root root 24600 Jun 12 2020 blog-single.html
drwxr-xr-x 2 root root 4096 Jun 13 2020 css/
drwxr-xr-x 2 root root 4096 Jun 13 2020 fonts/
drwxr-xr-x 2 root root 4096 Jun 13 2020 images/
-rw-r--r-- 1 root root 38017 Jun 12 2020 index.html
drwxr-xr-x 2 root root 4096 Jun 13 2020 js/
drwxr-xr-x 2 root root 4096 Jun 13 2020 mail/
-rw-r--r-- 1 root root 14449 Jun 12 2020 portfolio-details.html
-rw-r--r-- 1 root root 81593 Jun 12 2020 style.css
root@ip-172-31-42-20:~/project/mediplus-lite# mv * /var/www/html
root@ip-172-31-42-20:~/project/mediplus-lite# ll
total 8
drwxr-xr-x 2 root root 4096 Oct 29 09:44 .
drwxr-xr-x 3 root root 4096 Oct 29 09:39 ..
root@ip-172-31-42-20:~/project/mediplus-lite# 

```

32°C Partly sunny ENG IN 15:17 29-10-2024

Launching two static websites using two servers.

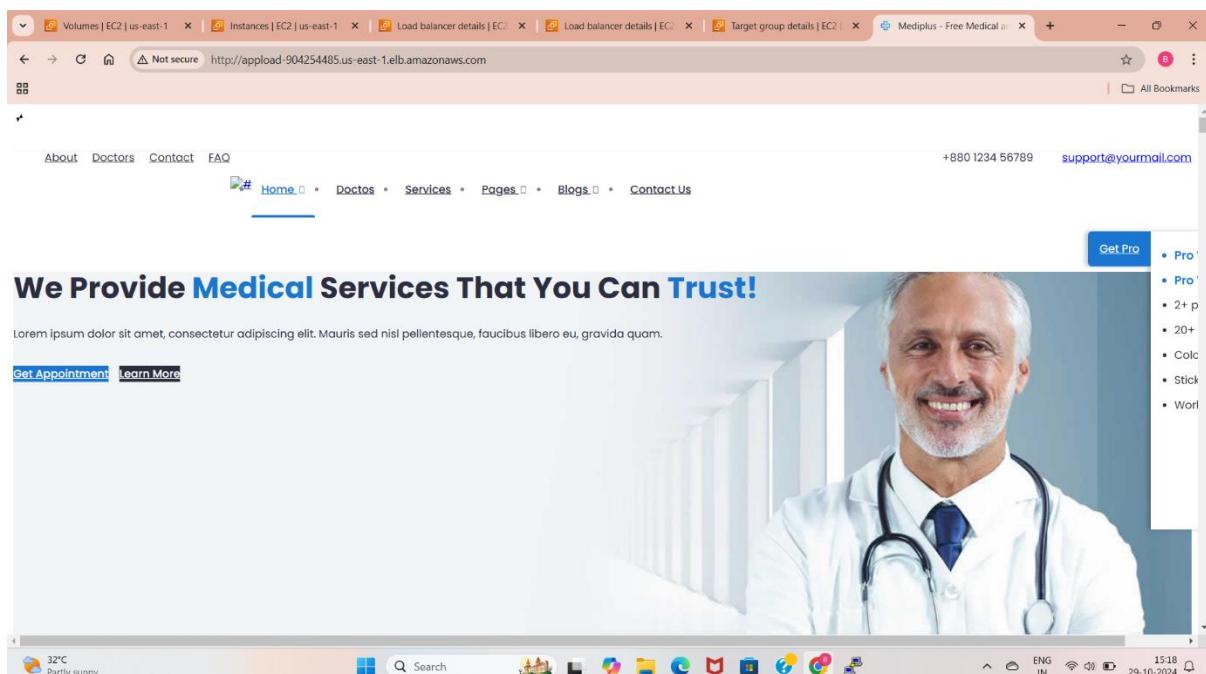
Take any static website to the server using

`wget url(your static website)`

we will get the zip or tar file, now extract those files using

`tar -xvf foldername`

Now go inside the extracted folder we can see the files with the code, move these files to `/var/www/html` so that we can access using the website.



[Read More](#)

Work Freelancer Has Done

\$250 Million
Paid to Freelancers

2 Million
Paid Invoices

700,000
Worldwide Freelancer

98%
Customer
Satisfaction Rate

Testimonial

When we copy the load balancer url and paste in browser we can see these two static websites by refreshing the page.

LAB-9: ASG AND LAUNCH TEMPLATE:

Launch template is nothing but setting all the configurations related to the instance like instance type, security group and storage options etc. in one template so that we can launch as many servers as we want.

Step 5 - optional
Add notifications

Step 6 - optional
Add tags

Step 7
Review

Launch template info

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.

Launch template

Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security group.

tempubuntu

Version

Create a launch template version

Description	Launch template	Instance type
A launch template	tempubuntu lt-0858f9e872e7bcca	t2.micro
AMI ID	ami-08adc60973283cb01	Security groups
Key pair name	new-key	Request Spot Instances
	Security group IDs	No
	sg-00dfccaa2f0a40ba0c	

Additional details

Storage (volumes)	Date created
/dev/sda1 /dev/sdy	Tue Oct 29 2024 15:24:53 GMT+0530 (India Standard Time)

[Cancel](#) [Next](#)

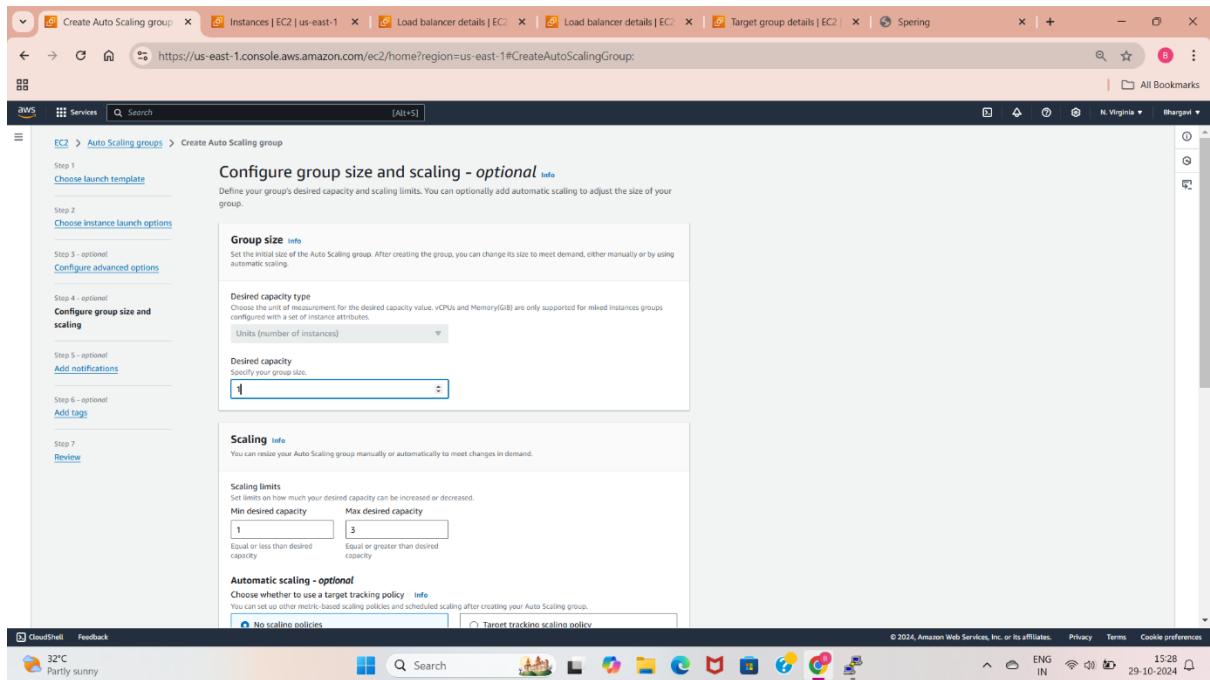
The screenshot shows the AWS EC2 Launch Templates page. On the left, there's a navigation sidebar with options like EC2 Dashboard, EC2 Global View, Instances, Launch Templates (selected), Images, Elastic Block Store, Network & Security, and more. The main content area displays a table titled 'Launch Templates (1/1)'. The table has columns for Launch Template ID, Launch Template Name, Default Version, Latest Version, Create Time, and Created By. One row is selected, showing 'lt-085bf9e872e7bbc' and 'tempubuntu'. Below the table, a section titled 'tempubuntu (lt-085bf9e872e7bbc)' shows 'Launch template details' and 'Launch template version details'. The version details show an AMI ID ('ami-08ade60973283cb01'), instance type ('t2.micro'), and availability zone ('us-east-1'). The date created is '2024-10-29T09:54:53.000Z' and the created by is 'arn:aws:iam::905418365089:root'.

Launched a launch template selecting the option from left side and including the configurations related to the ec2 instance and the template name is tempubuntu.

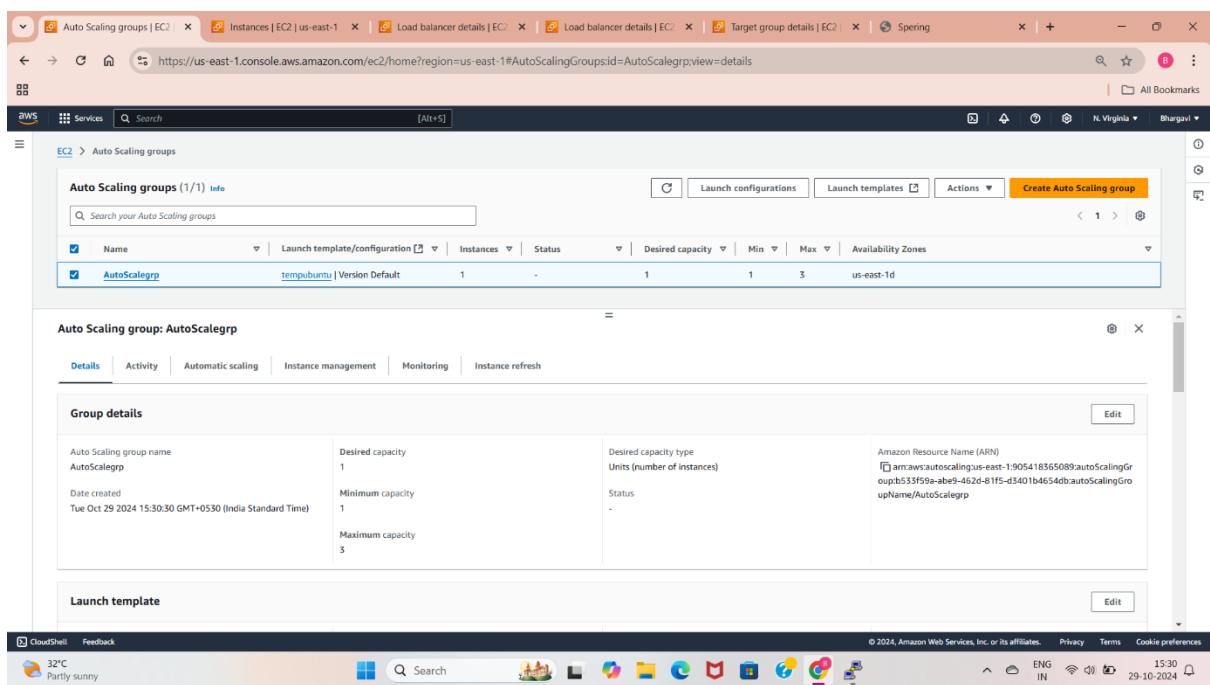
The screenshot shows the 'Create Auto Scaling group' wizard, Step 1: Choose launch template. It has seven steps: Step 1 (Choose launch template), Step 2 (Choose instance launch options), Step 3 (optional: Configure advanced options), Step 4 (optional: Configure group size and scaling), Step 5 (optional: Add notifications), Step 6 (optional: Add tags), and Step 7 (Review). The 'Choose launch template' step is active. It asks to specify a launch template that contains settings common to all EC2 instances. A 'Name' field is present, and a note says 'Auto Scaling group name' (Enter a name to identify the group). Below it, a 'Launch template' dropdown is set to 'tempubuntu'. Other fields include 'Version' (Default (1)), 'Description' (A launch template), and 'Launch template' (tempubuntu) and 'Instance type' (t2.micro).

An Auto Scaling group is a feature in cloud computing that automatically adjusts the number of running servers (instances) based on demand.

- If traffic increases, it automatically adds more servers to handle the load.
- If traffic decreases, it removes unnecessary servers to save costs.



Creating the auto scaling group by selecting Desired capacity with 1 and min desired capacity with 1 and max capacity with 3.



Created autoscaling group by selecting the above created launch template with 1 running instance(desired capacity) and we need 3 instances should be available everytime(max capacity) in the region of us-east-1a.

Now try to change the maximum capacity and see the new instance should get created.

The screenshot shows the AWS EC2 Instances page. A notification at the top states "Successfully initiated termination (deletion) of i-0806500cf293d142c". The main table lists four instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP
SERVER1	i-0806500cf293d142c	Terminated	t2.micro	-	View alarms +	us-east-1d	-	-	-
SERVER1	i-0d68aa500ef1f02fa	Terminated	t2.micro	-	View alarms +	us-east-1d	-	-	-
SERVER2	i-0904cf9377f037225	Terminated	t2.micro	-	View alarms +	us-east-1d	-	-	-
SERVER1	i-09be4d3853a8fe356	Running	t2.micro	Initializing	View alarms +	us-east-1d	ec2-54-91-33-214.com...	54.91.33.214	-

The instance **i-09be4d3853a8fe356 (SERVER1)** is detailed in the modal:

- Details**: Instance ID: i-09be4d3853a8fe356, Public IPv4 address: 54.91.33.214, Instance state: Running.
- Instance summary**: Hostname type: IP name: ip-172-31-37-117.ec2.internal, Private IP DNS name (IPv4 only): ip-172-31-37-117.ec2.internal, Instance type: t2.micro.
- Networking**: Public IPv4 addresses: 172.31.37.117, Public IP DNS: ec2-54-91-33-214.compute-1.amazonaws.com, Private IP addresses: -.

One ec2 instance is automatically launched by using auto scaling group. If we terminate this instance we can see other instance is launching by the autoscaling group. That means one server is always available even if there is heavy traffic or if there is no traffic.

The screenshot shows the AWS Auto Scaling Groups page. The left sidebar shows the navigation path: EC2 > Auto Scaling groups > AutoScalegrp. The main content area is titled "Edit AutoScalegrp" and contains the following sections:

- Group size**: Set the size of the Auto Scaling group by changing the desired capacity. It also allows specifying minimum and maximum scaling limits.
- Desired capacity**: Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes. The dropdown shows "Units (number of instances)" with a value of 2 selected.
- Scaling limits**: Set limits on how much your desired capacity can be increased or decreased. It includes fields for "Min desired capacity" (1) and "Max desired capacity" (4), and dropdowns for "Equal or less than desired capacity" and "Equal or greater than desired capacity".
- Launch template**: A note 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."

Here we changed the autoscaling group with min capacity to 2 instances, maximum capacity 4 and we can see the two running instances.

The screenshot shows two separate browser tabs open in a Windows 10 environment.

Top Tab (Auto Scaling Groups):

- URL: <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#AutoScalingGroupsId=AutoScalegrp;view=details>
- Content: Shows the "Auto Scaling groups (1/1) Info" page. An alert at the top says "Auto Scaling group updated successfully". A table lists one instance: "AutoScalegrp" (tempubuntu | Version Default), Desired capacity 2, Min 1, Max 4, Availability Zones us-east-1d. Status is "Updating capacity...".

Bottom Tab (Instances):

- URL: <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances>
- Content: Shows the "Instances (1/5) Info" page. An alert at the top says "Successfully initiated termination (deletion) of i-0806500cf293d142c". A table lists five instances, with the first one (SERVER1) selected and its status changed to "Running". Other instances are terminated or initializing.

We can see two instances are running after changing autoscaling desired capacity to 2 instances.

LAB:10: RDS

RDS (Relational Database Service) is a cloud service that makes it easy to set up, operate, and manage a database.

- It automatically takes care of backups, security, and updates.
- You don't have to worry about the server setup or maintenance.
- It supports popular databases like MySQL, PostgreSQL, Oracle, and SQL Server.

1. Provision an RDS instance

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like 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, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, and CloudWatch Metrics. The main content area displays a table titled 'Instances (1/1) Info'. The table has columns for Name, Instance ID, Instance state, Status check, Alarm status, Availability Zone, Public IPv4 DNS, Public IPv4 address, and Elastic IP. One row is selected, showing 'RDS-INSTANCE' with Instance ID 'i-082b3d74ef2d525b0', State 'Running', Status 'Initializing', Availability Zone 'us-east-1c', Public IPv4 DNS 'ec2-54-86-192-52.com...', and Public IPv4 address '54.86.192.52'. Below the table, a detailed view for 'i-082b3d74ef2d525b0 (RDS-INSTANCE)' is shown with tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. The 'Details' tab is active, displaying various configuration details such as Instance ID, Public IPv4 address, Instance state, Hostname type, IP name, Answer private resource DNS name, Auto-assigned IP address, IAM Role, Subnet ID, Instance ARN, VPC ID, and Auto Scaling Group name.

Launch a ec2 instance by allowing the mysql port number 3306 in security groups.

The screenshot shows the AWS RDS Databases page. On the left, there's a navigation sidebar with options like Dashboard, Databases (selected), Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL integrations, Events, Event subscriptions, Recommendations (0), and Certificate update. The main content area shows a progress bar for creating a database named 'rds'. A message says 'Your database might take a few minutes to launch. You can use settings from rds to simplify configuration of suggested database add-ons while we finish creating your DB for you.' Below the progress bar, a notification bar says 'Introducing Global Database writer endpoint'. It explains that each global cluster now has a writer endpoint that can be used to send application requests to the writer instance in the primary cluster of the Global Database. Aurora automatically updates the endpoint upon a cross-region failover or switch-over operation, ensuring that requests are routed to the writer instance in the new primary cluster without needing changes to application code or configuration. A link to 'Learn more' is provided. At the bottom, a 'Notifications' section shows 0 notifications. The 'Databases' section shows a table with one entry: 'rds' (Status: Creating, Instance Type: MySQL, Engine: MySQL Co..., Region: us-east-1c, Size: db.t3.micro). There are buttons for 'Group resources', 'Modify', 'Actions', 'Restore from S3', and 'Create database'.

Creating a database with db.t3.micro and allow the ec2 instance to connect with the RDS database.

Now the Database is in available state we can connect with our instance to access this database by using the endpoint of the database, endpoint is like the address for your database. By using this endpoint, your app can send queries to the database over the network.

```

root@ip-172-31-191-~:
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'database' at line 1
MySQL [(none)]> show databases;
+-----+
| Database |
+-----+
| RDS |
| data |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
6 rows in set (0.00 sec)

MySQL [(none)]> use data
Database changed
MySQL [data]> CREATE TABLE students (
    ->     student_id INT PRIMARY KEY AUTO_INCREMENT,
    ->     first_name VARCHAR(50),
    ->     last_name VARCHAR(50),
    ->     age INT,
    ->     major VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.03 sec)

MySQL [data]> show table students
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'students' at line 1
MySQL [data]> INSERT INTO students (first_name, last_name, age, major)
    -> VALUES ('John', 'Doe', 21, 'Computer Science');
Query OK, 1 row affected (0.01 sec)

MySQL [data]> show table students;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'students' at line 1
MySQL [data]> INSERT INTO students (first_name, last_name, age, major) VALUES ('John', 'Doe', 21, 'Computer Science');
Query OK, 1 row affected (0.00 sec)

MySQL [data]> show table students;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'students' at line 1
MySQL [data]> select * from students;
+-----+-----+-----+-----+
| student_id | first_name | last_name | age | major |
+-----+-----+-----+-----+
| 1 | John | Doe | 21 | Computer Science |
| 2 | John | Doe | 21 | Computer Science |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

MySQL [data]>

```

Here I have connected the database to ec2 instance using the following command

`mysql -h <endpoint of database> -u <username> -p`

-h specifies the host address

-u specifies the username

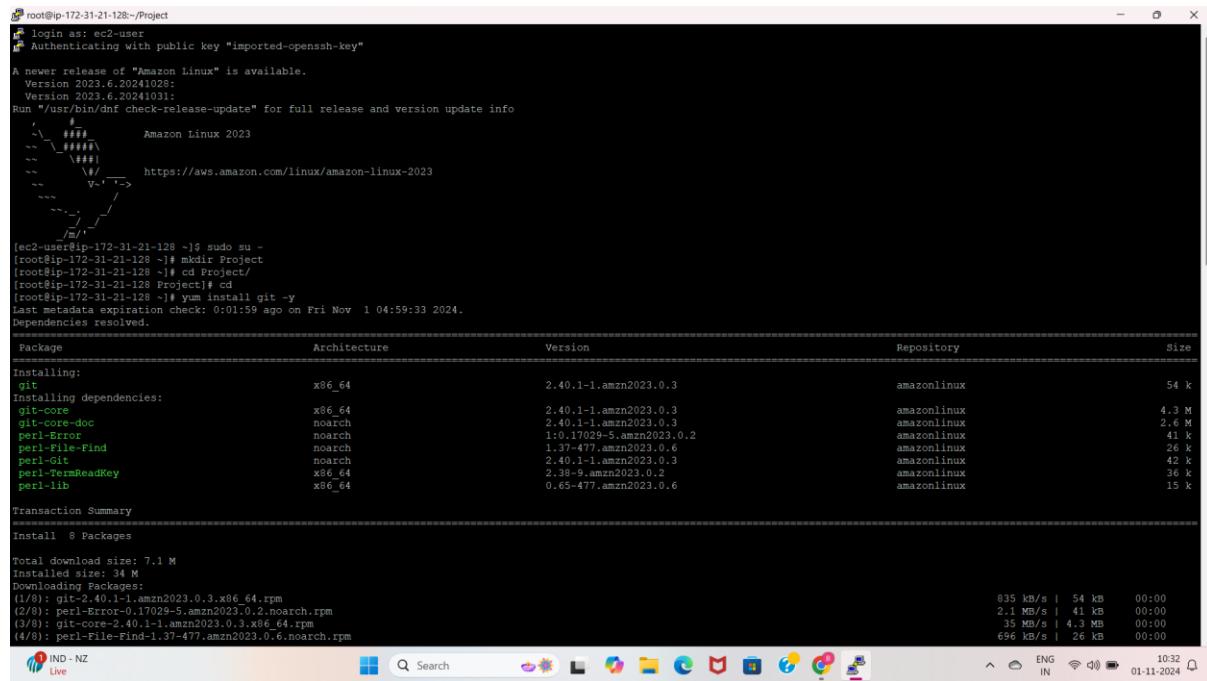
-p is the password

Next enter the password that you have created while creating database. If you enter the correct password you are successfully connected and can mysql prompt.

MINI PROJECT 2

LAB1: CREATING EC2 INSTANCE

1. Login to AWS Console
2. Create a server with Amazon Linux two/AMI/RHEL/Ubuntu
3. Connect the server only with putty/Gitbash



```
root@ip-172-31-21-128:~/Project
└── login as: ec2-user
    └── Authenticating with public key "imported-openssh-key"

A never release of "Amazon Linux" is available.
Version 2023.6.20241028;
Version 2023.6.20241031;
Run '/usr/bin/dnf check-release-update' for full release and version update info

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-172-31-21-128 ~]$ sudo su -
[root@ip-172-31-21-128 ~]# mkdir Project
[root@ip-172-31-21-128 ~]# cd Project/
[root@ip-172-31-21-128 Project]# cd
[root@ip-172-31-21-128 ~]# yum install git -y
Last metadata expiration check: 0:01:59 ago on Fri Nov  1 04:59:33 2024.
Dependencies resolved.

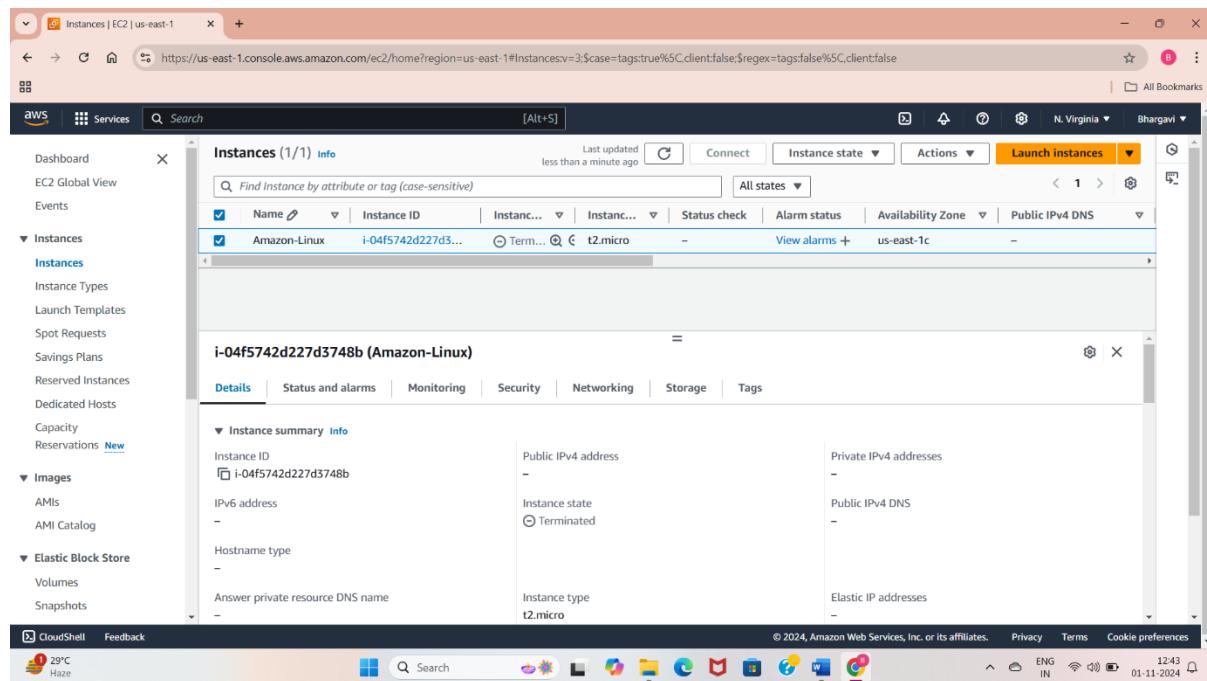
Transaction Summary
=====
Package          Architecture Version      Repository   Size
Installing:
  git             x86_64       2.40.1-1.amzn2023.0.3  amazonlinux  54 k
Installing dependencies:
  git-core        x86_64       2.40.1-1.amzn2023.0.3  amazonlinux  4.3 M
  git-core-doc    noarch      2.40.1-1.amzn2023.0.3  amazonlinux  2.6 M
  perl-Error      noarch      1:0.17029-5.amzn2023.0.2  amazonlinux  41 k
  perl-File-Find  noarch      1.37-477.amzn2023.0.6   amazonlinux  26 k
  perl-Git        noarch      2.40.1-1.amzn2023.0.3  amazonlinux  42 k
  perl-TermReadKey x86_64     2.38-9.amzn2023.0.2   amazonlinux  36 k
  perl-Lib        x86_64     0.65-477.amzn2023.0.6  amazonlinux  15 k

Transaction Summary
=====
Install  8 Packages

Total download size: 7.1 M
Installed size: 34 M
Downloads checked:
(1/8): git-2.40.1-1.amzn2023.0.3.x86_64.rpm  835 kB/s | 54 kB   00:00
(2/8): perl-Error-0.17029-5.amzn2023.0.2.noarch.rpm  2.1 MB/s | 41 kB   00:00
(3/8): git-core-2.40.1-1.amzn2023.0.3.x86_64.rpm  35 MB/s | 4.3 MB  00:00
(4/8): perl-File-Find-1.37-477.amzn2023.0.6.noarch.rpm 696 kB/s | 26 kB   00:00

[ec2-user@ip-172-31-21-128 ~]$ 10:32 01-11-2024
```

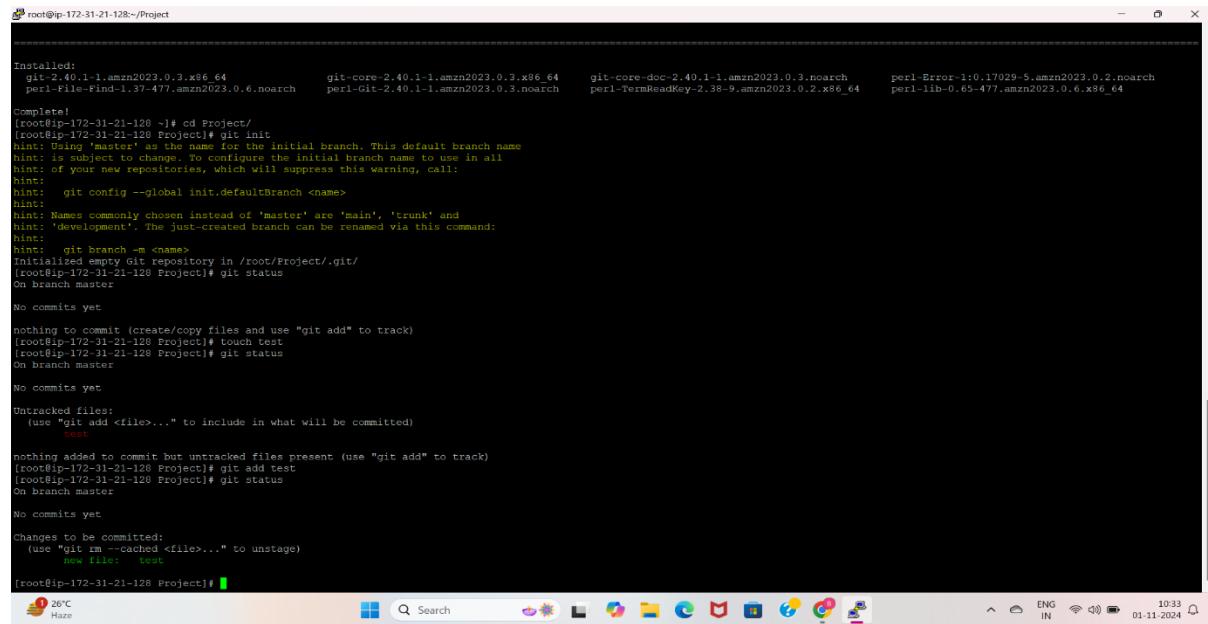
Created a ec2 instance with amazon linux and enabled ssh in the security group to connect with the putty. Connected the above created ec2 instance with the putty using public ip and private key pair, changed to root user using “super user do” command.



This shows the instance is created with the name Amazon-linux and it is terminated after creating and completing work.

LAB2: CREATE REPO IN LOCAL MACHINE:

Git is a Source Code Management System developed by Linus Torvalds. It is called as distributed version control system because it tracks(manages) every change in the code and saves the versions of code like updates with the new features and rollback to previous version if there are any errors.



```
root@ip-172-31-21-128:~/Project
Installed:
git-2.40.1-1.amzn2023.0.3.x86_64      git-core-2.40.1-1.amzn2023.0.3.x86_64      git-core-doc-2.40.1-1.amzn2023.0.3.noarch    perl-BioPerl-1.0.17029-5.amzn2023.0.2.noarch
perl-File-Find-1.37-477.amzn2023.0.6.noarch  perl-CIO-2.40.1-1.amzn2023.0.3.noarch      perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64   perl-lib-0.65-477.amzn2023.0.6.x86_64

Complete!
[root@ip-172-31-21-128 ~]# cd Project/
[root@ip-172-31-21-128 Project]# git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: may be subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:  git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:  git branch -m <name>
Initialized empty Git repository in /root/Project/.git/
[root@ip-172-31-21-128 Project]# git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
[root@ip-172-31-21-128 Project]# touch test
[root@ip-172-31-21-128 Project]# git status
On branch master

No commits yet

Untracked files:
(use "git add <file>..." to include in what will be committed)
        test

nothing added to commit but untracked files present (use "git add" to track)
[root@ip-172-31-21-128 Project]# git add test
[root@ip-172-31-21-128 Project]# git status
On branch master

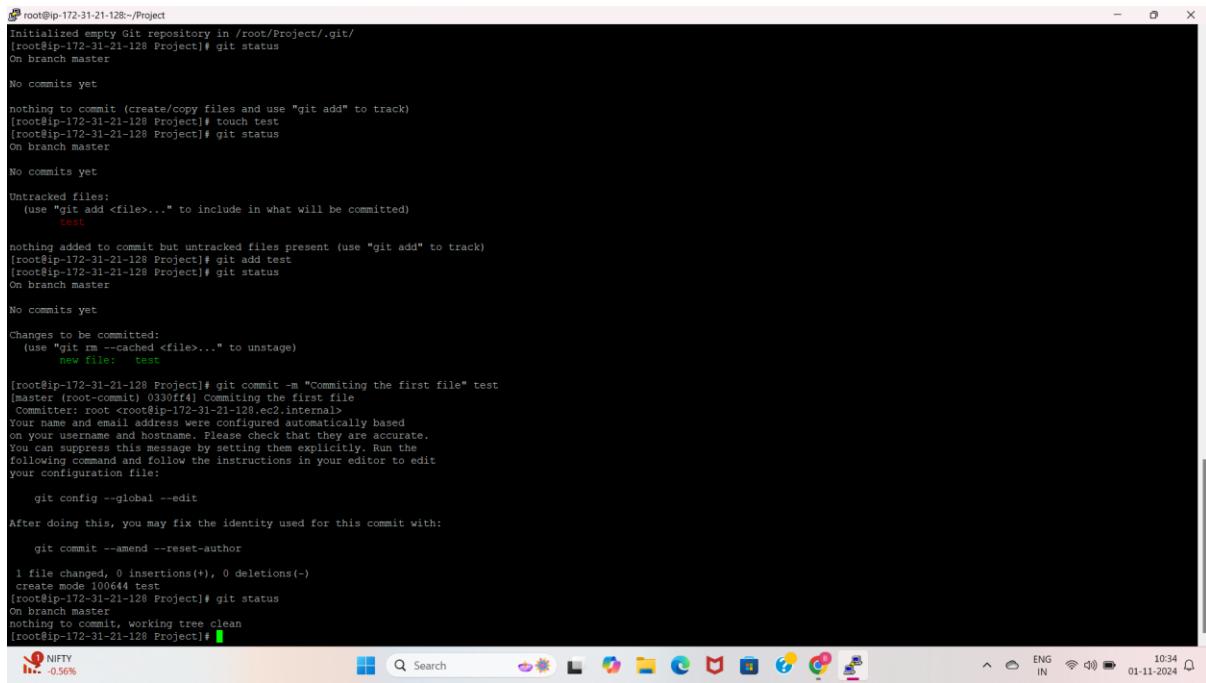
No commits yet

Changes to be committed:
(use "git rm --cached <file>..." to unstage)
  new file:  test

[root@ip-172-31-21-128 Project]#
```

Creating a folder in local machine with the name Project. After creating changed to that folder

And installed and initialized git inside the folder to work on the git. Created a file and added to staging area using `<git add filename>`. Staging area files and folders are tracked if there is any changes made.



```
root@ip-172-31-21-128:~/Project
Initialized empty Git repository in /root/Project/.git/
[root@ip-172-31-21-128 Project]# git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
[root@ip-172-31-21-128 Project]# touch test
[root@ip-172-31-21-128 Project]# git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    test

nothing added to commit but untracked files present (use "git add" to track)
[root@ip-172-31-21-128 Project]# git add test
[root@ip-172-31-21-128 Project]# git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file: test

[root@ip-172-31-21-128 Project]# git commit -m "Committing the first file" test
[master (root-commit 0330f44) Committing the first file]
Committer: root <root@ip-172-31-21-128.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:
  git config --global --edit

After doing this, you may fix the identity used for this commit with:
  git commit --amend --reset-author

1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 test
[root@ip-172-31-21-128 Project]# git status
On branch master
nothing to commit, working tree clean
[root@ip-172-31-21-128 Project]#
```

Before committing the files from working directory check the status we can't find any files on the master branch. We can only see the tracking files need to be commit.

The file which is added to staging area are committed to repository using git commit command with a message and file name.

<git commit -m “message” filename>

LAB3: CREATING A REPO IN REMOTE LOCATION-GITHUB

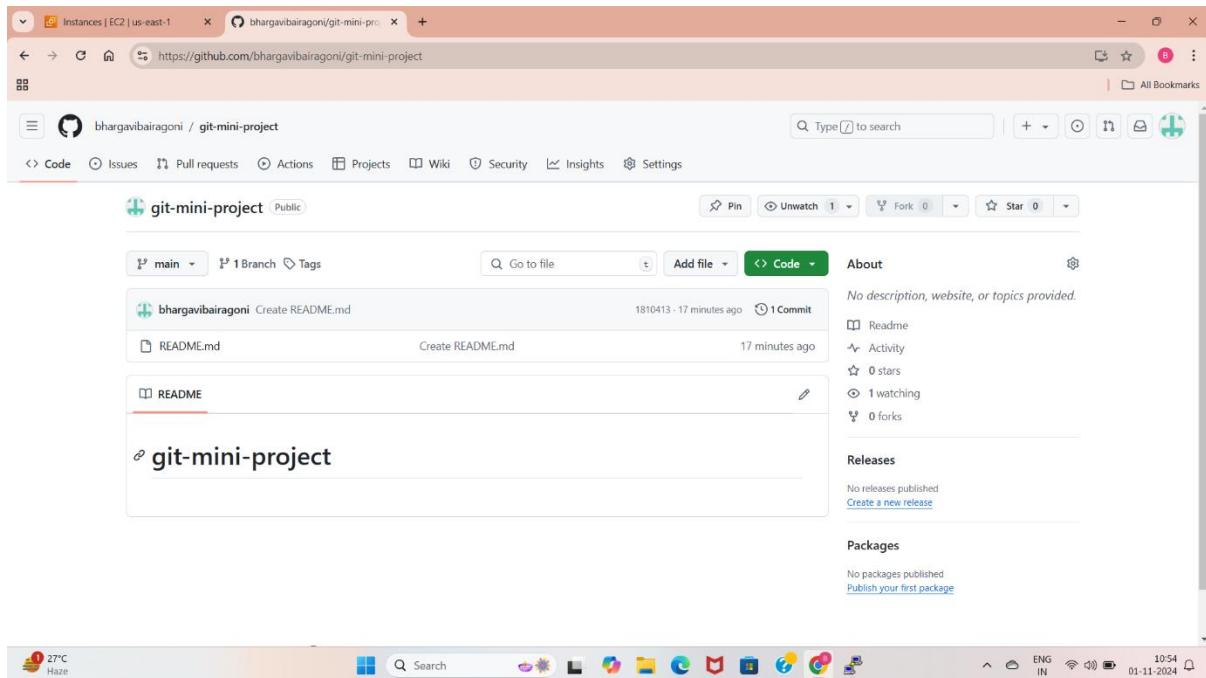
Github is a project management tool, it stores and manages our project such as issues, pull requests which helps to track our project. We are having two type of repositories in github .

Public repository:

Whatever code, files and folder we store in this repository can be seen and accessed by everyone over the internet and they can fork this and use this project code.

Private Repository:

Whatever we code, files and folders we save in the repository cannot be accessible by people until we share it with them.



Created a private repo with the name git-mini-project with readme.md file.

LAB4: WORKING WITH REMOTE REPO:

Take the remote repository to your local machine and make some changes by adding some files and folders.

- ❖ After making changes push the changes into remote repository
- ❖ From remote machine copy the URL of repository to clone in the local machine using <git clone url> we can see the URL at code button
- ❖ After cloning you will be asked username and password
- ❖ If you enter password and username you will get one error with authentication is removed
- ❖ To solve the error we need to generate a token and keep in the place of password. As we have created a private repository, to access private repository we need personal token.

The screenshot shows the GitHub developer settings page for personal access tokens. A specific token named 'token — repo' is selected. The token was last used on December 1, 2024, and will expire on December 1, 2024. It is described as being used for Git over HTTPS or for authenticating to the API over Basic Authentication.



- ❖ Go to the setting on top right corner of your repository and go to the developer settings and generate a personal classic token.
- ❖ Generate a token with no. of valid days and copy and save the token to access GitHub for valid no. of days
- ❖ Now clone the repository in your local machine and enter the GitHub username and the generated token in the place of password

```

root@ip-172-31-21-128:~#
Date: Fri Nov 1 05:04:29 2024 +0000
Committing the first file
[root@ip-172-31-21-128 Project]# git push -u origin master
Username for 'https://github.com': bhargavibairagoni
Password for 'https://bhargavibairagoni@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 221 bytes | 221.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/bhargavibairagoni/git-mini-project/pull/new/master
remote:
To https://github.com/bhargavibairagoni/git-mini-project.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
[root@ip-172-31-21-128 Project]# git clone https://github.com/bhargavibairagoni/git-mini-project.git
Cloning into 'git-mini-project'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (6/6), done.
[root@ip-172-31-21-128 Project]# 11
total 0
drwxr-xr-x. 3 root root 35 Nov 1 05:35 git-mini-project
-rw-r--r--. 1 root root 0 Nov 1 05:02 test
[root@ip-172-31-21-128 Project]# cd git-mini-project/
[root@ip-172-31-21-128 git-mini-project]# 11
total 4
-rw-r--r--. 1 root root 19 Nov 1 05:35 README.md
[root@ip-172-31-21-128 git-mini-project]# git clone https://github.com/bhargavibairagoni/git-mini-project.git
Cloning into 'git-mini-project'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (6/6), done.
[root@ip-172-31-21-128 git-mini-project]# 11
total 4
-rw-r--r--. 1 root root 19 Nov 1 05:35 README.md
[root@ip-172-31-21-128 git-mini-project]# cd
[root@ip-172-31-21-128 ~]# 

```

Repository which we created above is cloned into our local machine using <git clone project-url> with this we download the repository and get all the files from the remote repository to local machine.

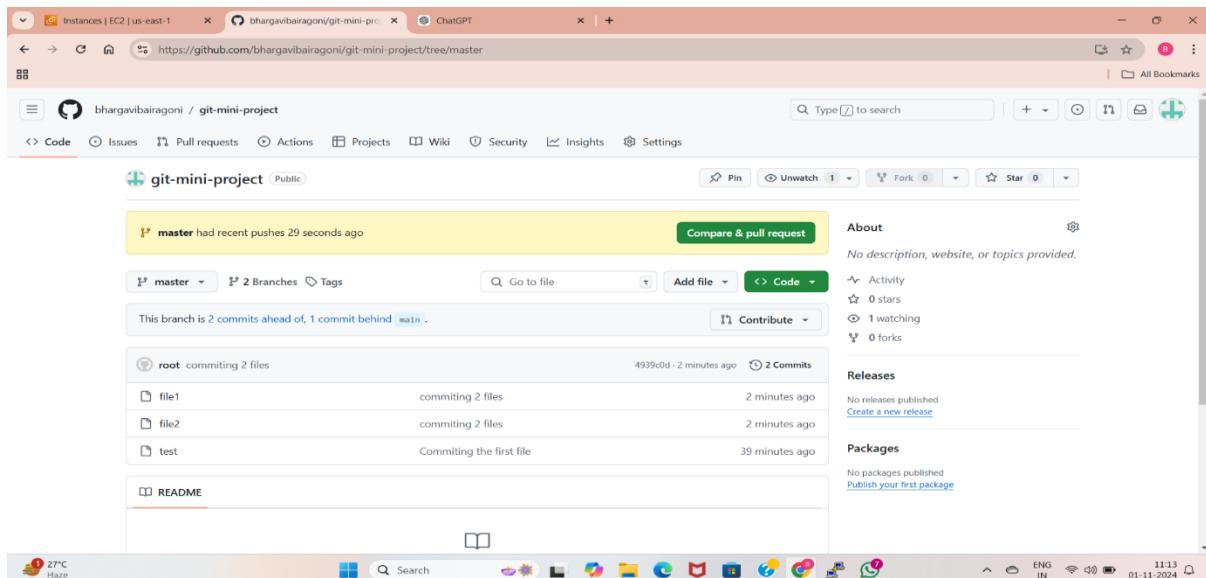
```

root@ip-172-31-21-128:~/Project
[root@ip-172-31-21-128 git-mini-project]# cd git-mini-project/
[root@ip-172-31-21-128 git-mini-project]# ll
total 4
-rw-r--r--, 1 root root 19 Nov 1 05:36 README.md
[root@ip-172-31-21-128 git-mini-project]# cd ..
[root@ip-172-31-21-128 Project]# touch file1 file2
[root@ip-172-31-21-128 Project]# git config --file test
[root@ip-172-31-21-128 Project]# git add file1 file2
[root@ip-172-31-21-128 Project]# git commit -m "committing 2 files" file1 file2
[master 4939c0d] committing 2 files
 1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file1
[root@ip-172-31-21-128 Project]# git push -u origin master
Username for 'https://github.com': bhargavibairagoni
Password for 'https://bhargavibairagoni@github.com':
remote: Counting objects: 1008 (3/3), done.
remote: Compressing objects: 1008 (2/2), done.
remote: Writing objects: 1008 (2/2), done.
remote: Total 1008 (delta 0), pack-reused 0
To https://github.com/bhargavibairagoni/git-mini-project.git
 0330ff4..4939c0d master -> master
Branch 'master' set up to track 'origin/master'.
[root@ip-172-31-21-128 Project]#

```

- ❖ Now go to repository add some files using touch command it will create empty files in working directory.
- ❖ Now add these files to staging area using < git add filename>
- ❖ We can track every change from staging area
- ❖ Now commit the changes from staging area to repository using <git commit -m "provide any message" filenames>
- ❖ After committing push the changes into remote repository using < git push origin main>
- ❖ It will ask for username and password enter username and token
- ❖ If the repository is empty the changes will be pushed if it is not empty we need to pull the remote repository.

Now go to your remote repository we can see the new changes like files.



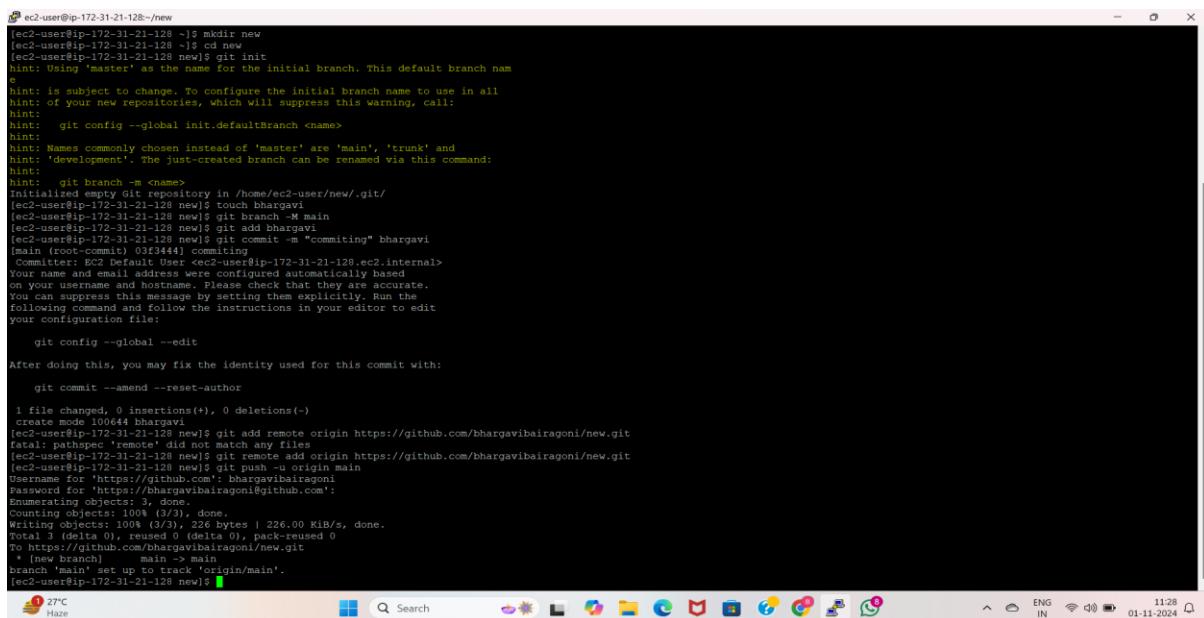
Here we can see the remote repository with the change that is the files from local repository is pushed to here.

LAB5: Pushing a locally created repo to Github:

Creating one remote repository with the same name as local repository in GitHub and do not initiate it

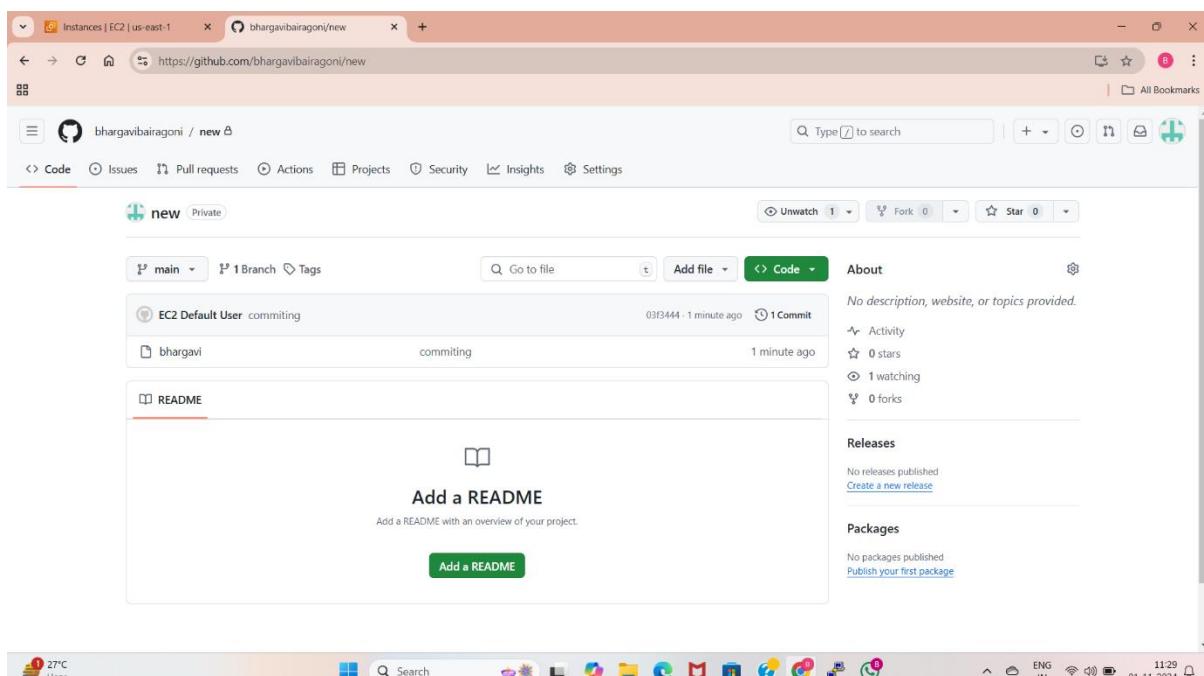
Now from inside the locally created repo execute the following commands

- <git branch -M main> change the name of master branch to main in locally created repo.
- <git remote add origin url> to add the remote repository in locally created repository
- <git push -u origin main> to push the changes from the local machine to remote repository



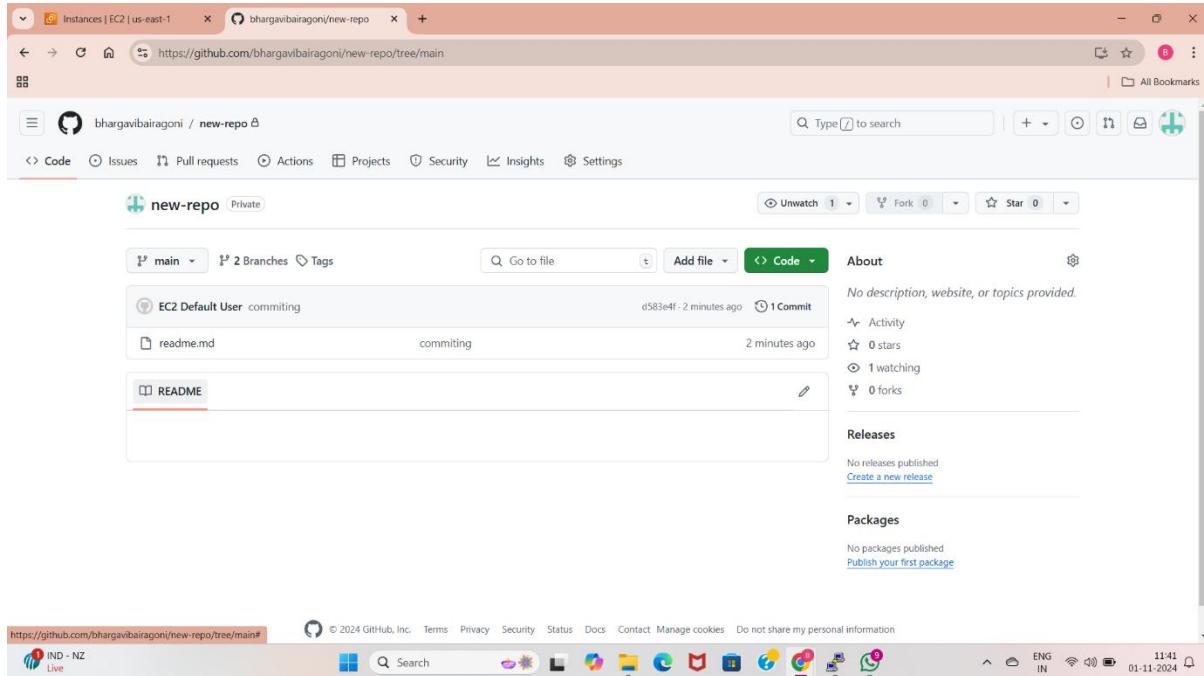
```
ec2-user@ip-172-31-21-128:~/new
[ec2-user@ip-172-31-21-128 ~]$ mkdir new
[ec2-user@ip-172-31-21-128 ~]$ cd new
[ec2-user@ip-172-31-21-128 new]$ git init
hint: Using 'master' as the name for the initial branch. This default branch nam
e is
hint: subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/ec2-user/new/.git/
[ec2-user@ip-172-31-21-128 new]$ touch bhangavi
[ec2-user@ip-172-31-21-128 new]$ git branch -M main
[ec2-user@ip-172-31-21-128 new]$ git add bhangavi
[ec2-user@ip-172-31-21-128 new]$ git commit -m "committing" bhangavi
[Email from user: 03B444]
Committer: EC2 Default User <ec2-user@ip-172-31-21-128.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:
git config --global --edit
After doing this, you may fix the identity used for this commit with:
git commit --amend --reset-author
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 bhangavi
[ec2-user@ip-172-31-21-128 new]$ git add remote origin https://github.com/bhargavibairagoni/new.git
fatal: pathspec 'remote' did not match any files
[ec2-user@ip-172-31-21-128 new]$ git remote add origin https://github.com/bhargavibairagoni/new.git
[ec2-user@ip-172-31-21-128 new]$ git push -u origin main
Username for 'https://github.com': bhargavibairagoni
Password for 'https://bhargavibairagoni@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 226 bytes | 226.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/bhargavibairagoni/new.git
 * [new branch]  main -> main
branch 'main' set up to track 'origin/main'.
[ec2-user@ip-172-31-21-128 new]$
```

Created a local repository with the

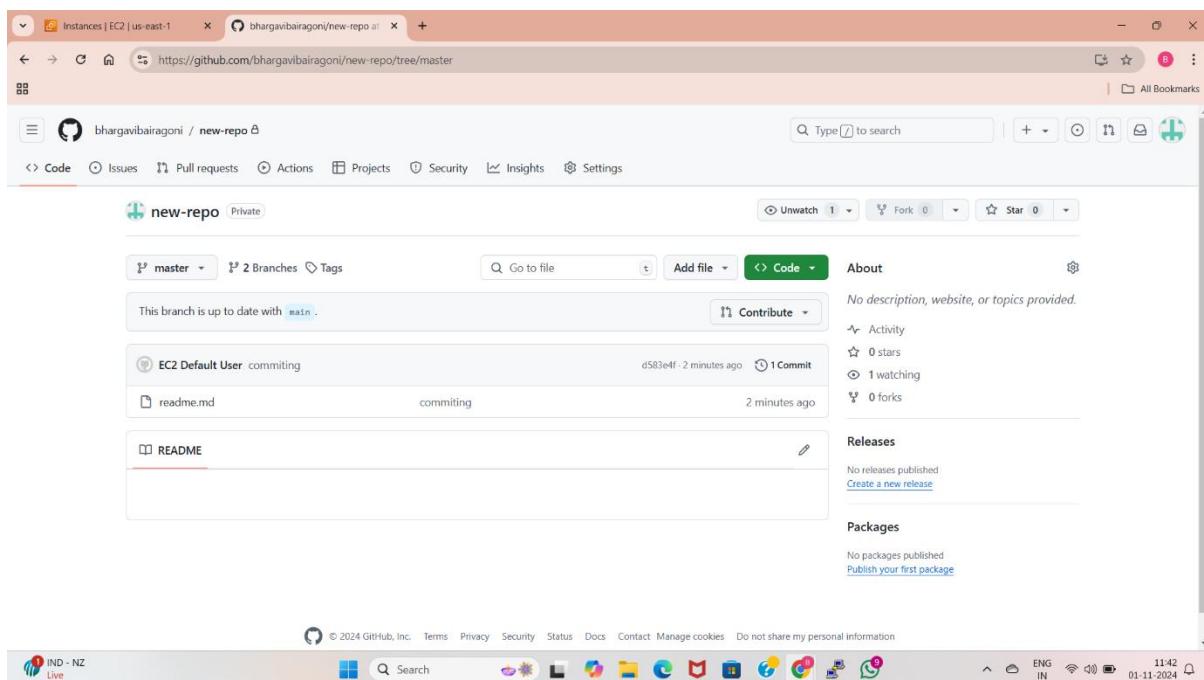


LAB:6: Creating a New branch from your main branch:

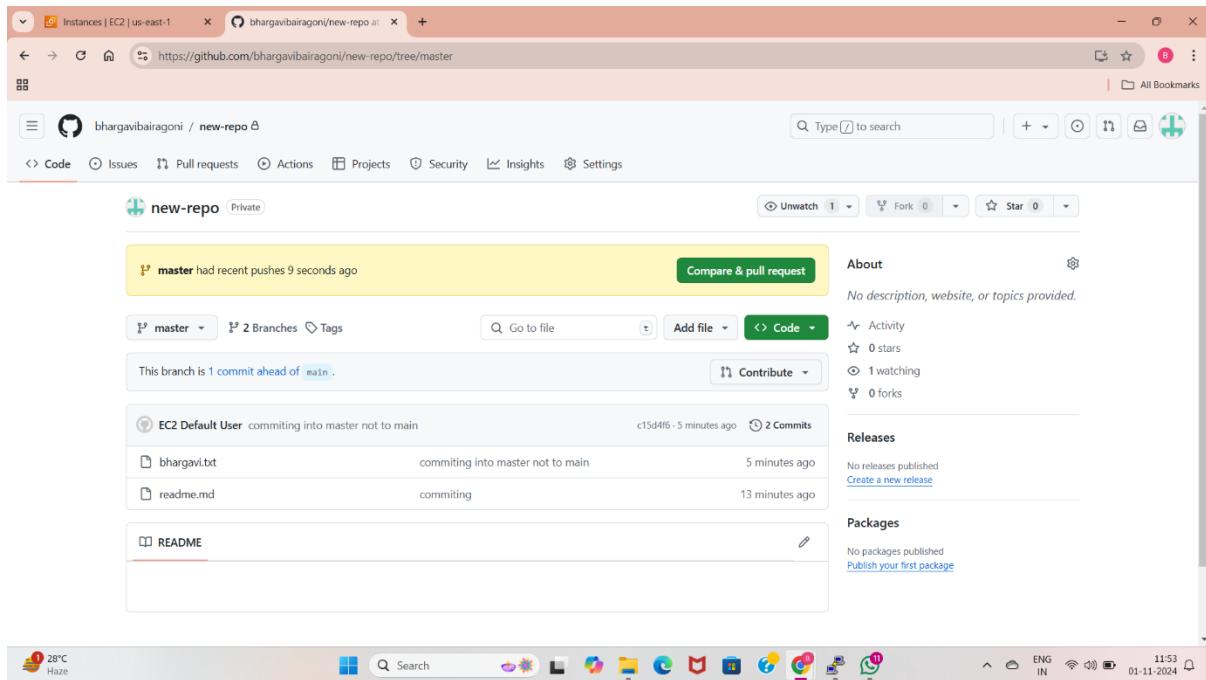
Now go to your remote repository and click on the branch. In remote repository we are having main as default branch.



Click on the main branch and type the name you want to create and you can see create new branch from main.



Now a new branch with the given name master is created.



Make some changes in new branch and commit changes. The changes made in new branch will not be applied in main branch.

LAB:7: Pull all the branches in your local machine:

Go to the local repository inside the local machine where are having remote repository as origin .

```
ec2-user@ip-172-31-21-128:~/new-repo
[ec2-user@ip-172-31-21-128 ~]$ rm -rf new
[ec2-user@ip-172-31-21-128 ~]$ 11
total 0
drwxr-xr-x. 2 ec2-user ec2-user 6 Nov  1 06:07 new-repo
[ec2-user@ip-172-31-21-128 new-repo]$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/ec2-user/new-repo/.git/
[ec2-user@ip-172-31-21-128 new-repo]$ touch readme.md
[ec2-user@ip-172-31-21-128 new-repo]$ git add readme.md
[ec2-user@ip-172-31-21-128 new-repo]$ git commit -m "committing"
[ec2-user@ip-172-31-21-128 new-repo]$ git config user.name "EC2 Default User"
[ec2-user@ip-172-31-21-128 new-repo]$ git config user.email "ec2.internal"
[ec2-user@ip-172-31-21-128 new-repo]$ git remote add origin https://github.com/bhargavibairagoni/new-repo.git
[ec2-user@ip-172-31-21-128 new-repo]$ git push -u origin main
Username for 'https://github.com': bhargavibairagoni
Password for 'https://bhargavibairagoni@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 228 bytes | 228.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/bhargavibairagoni/new-repo.git
 * [new branch]      main    -> main
Branch 'main' set up to track 'origin/main'.
[ec2-user@ip-172-31-21-128 new-repo]$
```

```

[ec2-user@ip-172-31-21-128-/new-repo]
[ec2-user@ip-172-31-21-128 new-repo]$ git checkout main
Already on 'main'
Your branch is up to date with 'origin/main'.
[ec2-user@ip-172-31-21-128 new-repo]$ git branch
* main
[ec2-user@ip-172-31-21-128 new-repo]$ git pull origin main
Username for 'https://github.com': bhargavibairagi
Password for 'https://bhargavibairagi@github.com':
remote: Invalid username or password.
fatal: Authentication failed for 'https://github.com/bhargavibairagi/new-repo.git'
[ec2-user@ip-172-31-21-128 new-repo]$ git fetch origin
Username for 'https://github.com': bhargavibairagi
Password for 'https://bhargavibairagi@github.com':
From https://github.com/bhargavibairagi/new-repo
 * [new branch]      master    -> origin/master
[ec2-user@ip-172-31-21-128 new-repo]$ git branch
* main
[ec2-user@ip-172-31-21-128 new-repo]$ vim bhargavi.txt
[ec2-user@ip-172-31-21-128 new-repo]$ git add bhargavi.txt
[ec2-user@ip-172-31-21-128 new-repo]$ git commit -s "committing into master not to main" bhargavi.txt
[master c15d4f6] committing into master not to main
Committer: EC2 Default User <ec2-user@ip-172-31-21-128.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:
git config --global --edit
After doing this, you may fix the identity used for this commit with:
git commit --amend --reset-author
1 file changed, 2 insertions(+)
create mode 100644 bhargavi.txt
[ec2-user@ip-172-31-21-128 new-repo]$ git pull origin master
Username for 'https://github.com': bhargavibairagi
Password for 'https://bhargavibairagi@github.com':
From https://github.com/bhargavibairagi/new-repo
 * branch      master    -> FETCH_HEAD
Already up to date.
[ec2-user@ip-172-31-21-128 new-repo]$ git push -u origin master

```

Run the `<git pull>` command to bring the changes from remote repository to local repository.

```

root@ip-172-31-21-128-/new-repo
[ec2-user@ip-172-31-21-128 new-repo]#
[root@ip-172-31-21-128 new-repo]# git pull origin master
Username for 'https://github.com': bhargavibairagi
Password for 'https://bhargavibairagi@github.com':
From https://github.com/bhargavibairagi/new-repo
 * branch      master    -> FETCH_HEAD
[root@ip-172-31-21-128 new-repo]# git pull origin main
Username for 'https://github.com': bhargavibairagi
Password for 'https://bhargavibairagi@github.com':
From https://github.com/bhargavibairagi/new-repo
 * branch      main     -> FETCH_HEAD
Already up to date.
[root@ip-172-31-21-128 new-repo]# git branch -a
* master
  remotes/origin/main
  remotes/origin/master
[root@ip-172-31-21-128 new-repo]# git branch -D master
error: cannot delete branch 'master' checked out at '/root'
[root@ip-172-31-21-128 new-repo]# git branch -a
* main
  remotes/origin/main
  remotes/origin/master
[root@ip-172-31-21-128 new-repo]# git checkout main
branch 'main' set up to track 'origin/main'.
Switched to a new branch 'main'.
[root@ip-172-31-21-128 new-repo]# git branch -a
* main
  remotes/origin/main
  remotes/origin/master
[root@ip-172-31-21-128 new-repo]# git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    ./._bash_history
    ./._bash_logout
    ./._bash_profile
    ./._bashrc
    ./._cshrc
    ./._lesshst
    ./._ssh/
    ./._tcschrc
    ./._project/
nothing added to commit but untracked files present (use "git add" to track)
[root@ip-172-31-21-128 new-repo]#

```

To see the locally created branches and remotely created branches run this command

`< git branch -a>`

The names with red mark and `remote/origin/` branch are the branches created in remote repository and the names with green color , * are the locally created branches.

Now checkout to the new branch using `<git checkout branchname>`

Make sure you are in the new branch by checking `git status` or `git branch` command

```

root@ip-172-31-21-128:~/new-repo
.../Project/
nothing added to commit but untracked files present (use "git add" to track)
[root@ip-172-31-21-128 new-repo]# ll
total 0
[root@ip-172-31-21-128 new-repo]# touch file3 file4
[root@ip-172-31-21-128 new-repo]# git add *
[root@ip-172-31-21-128 new-repo]# git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   file3
    new file:   file4

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    ./bash_history
    ./bash_logout
    ./bash_profile
    ./bashrc
    ./cshrc
    ./lessht
    ./ssh/
    ./tcshrc
    ./viminfo
    ./Project/
[root@ip-172-31-21-128 new-repo]# git commit -m "committing file3 file4"
[master 870b2cd] committing file3 file4
Committer: root <root@ip-172-31-21-128.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:
  git config --global --edit

After doing this, you may fix the identity used for this commit with:
  git commit --amend --reset-author

2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 new-repo/file3
create mode 100644 new-repo/file4
[root@ip-172-31-21-128 new-repo]# git push -u origin master
Username for 'https://github.com': bhargavibairagoni
Password for 'https://bhargavibairagoni@github.com': [REDACTED]

```

Now make some changes in the newly created branch using touch file3 file4

And push to the remote repository

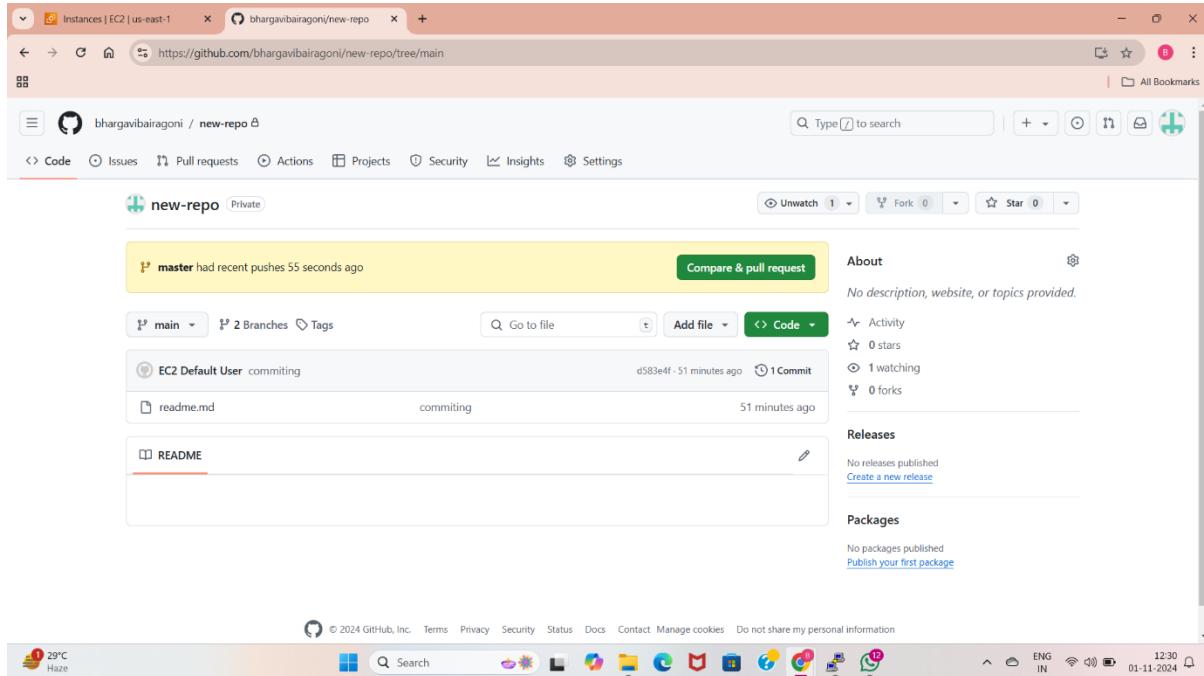
Name	Last commit message	Last commit date
..		
file3	committing file3 file4	4 days ago
file4	committing file3 file4	4 days ago

We can see the files from local repository are pushed into remote repository using

`<git push -u origin master>`

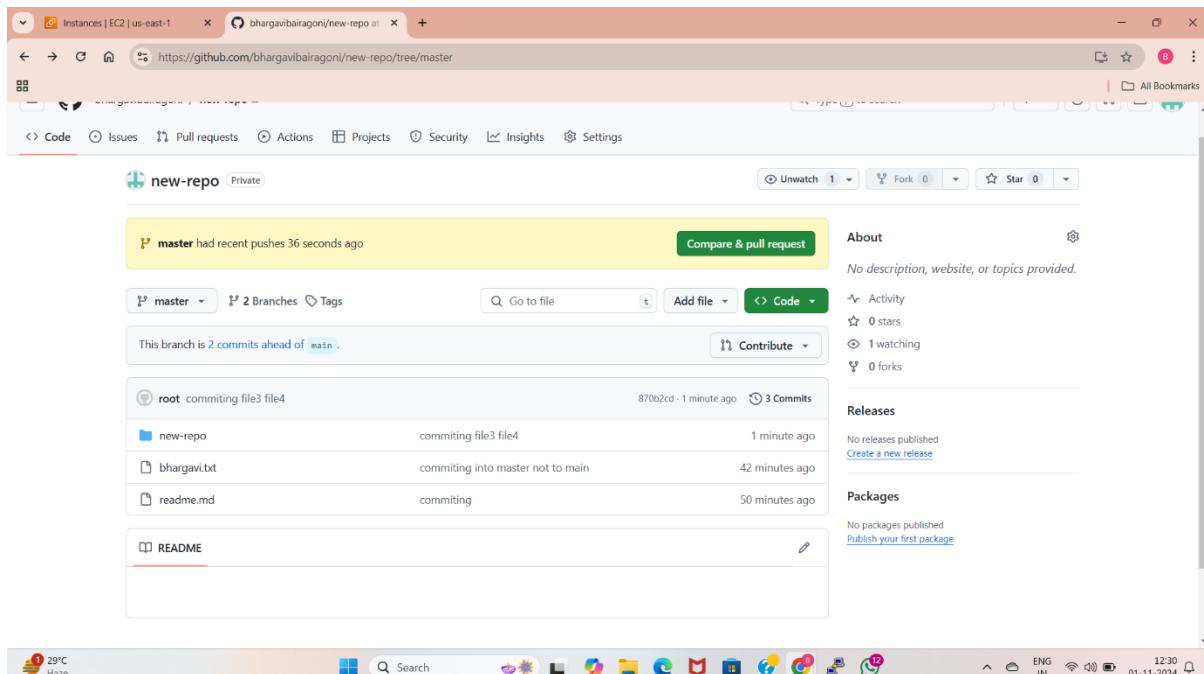
LAB:8: Merge our feature branch with main branch:

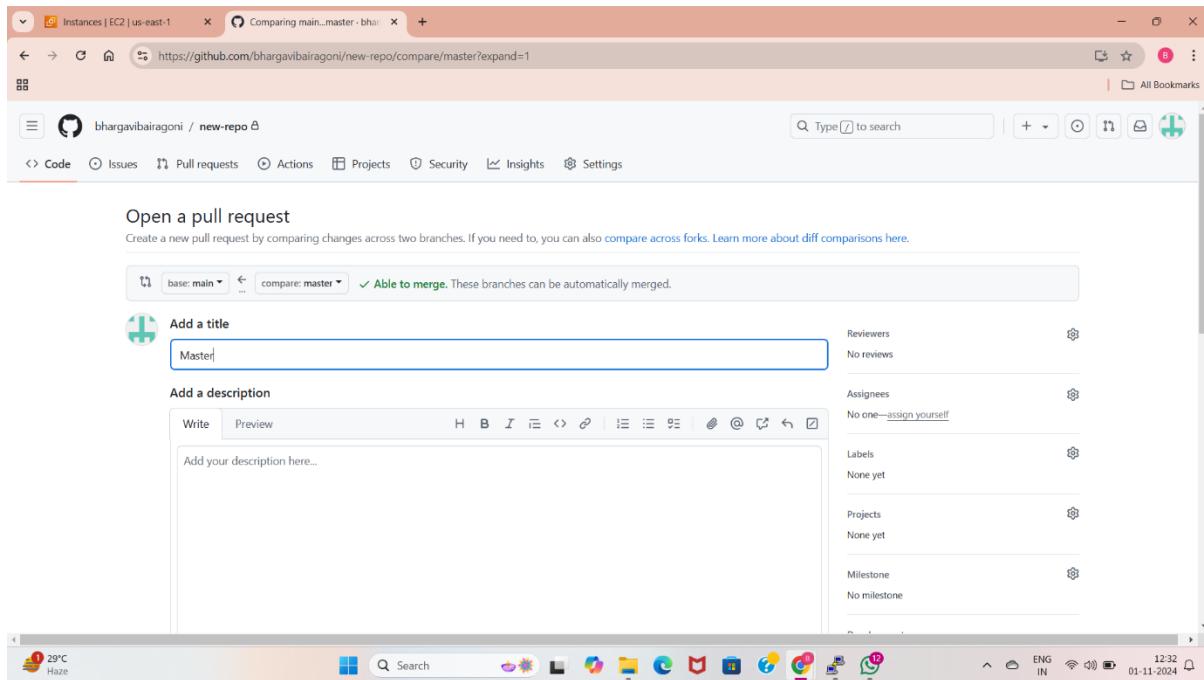
Go to your GitHub repository and you can see some changes are made in newly created branch



Go to "pull request" tab in the repository and create a pull request from master(new branch)to main

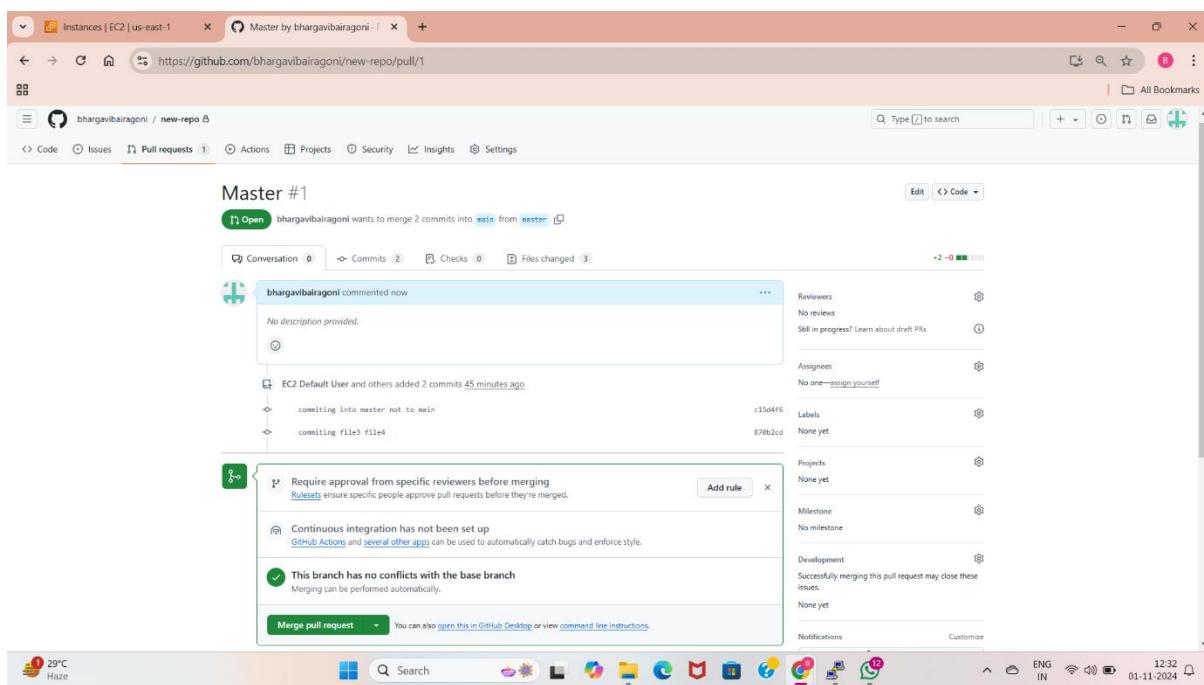
To take the changes from master to main we need to specify the destination as main and source as master





Now click on the pull request and it will ask for title of pull request just ignore that and click on pull request.

Again go to pull request tag and click on. Pull request there we can find the pull request just now created.



Now click on the created pull request, review the request and accept the request or merge the request.

Now the files from the new branch is merged to main branch we can see this changes by going to the repository page with the code button and there change the branch to main branch see all the files.

LAB:9: Go to local machine:

Go to the local machine having remote repository.

Check out to the main branch <git checkout branchname>and pull the changes into local repository.

```
root@ip-172-31-21-128:~/new-repo
...
Last login: Fri Nov  1 06:40:24 2024 from 49.206.56.90
[ec2-user@ip-172-31-21-128 ~]$ sudo su -
Last login: Fri Nov  1 06:40:28 UTC 2024 on pts/7
[root@ip-172-31-21-128 ~]$ ll
total 4
drwxr-xr-x  3 root root 44 Nov  1 05:40 Project
-rw-r--r--  1 root root 86 Nov  1 06:49 bhargavi.txt
drwxr-xr-x  2 root root  6 Nov  1 06:43 new
drwxr-xr-x  2 root root 32 Nov  1 06:58 new-repo
-rw-r--r--  1 root root  0 Nov  1 06:46 readme.md
[root@ip-172-31-21-128 ~]$ cd new-repo/
[root@ip-172-31-21-128 new-repo]$ git branch
main
* master
[root@ip-172-31-21-128 new-repo]$ git checkout main
Switched to branch 'main'.
Your branch is up to date with 'origin/main'.
[root@ip-172-31-21-128 new-repo]$ git branch
* main
  master
[root@ip-172-31-21-128 new-repo]$ git pull origin main
Username for 'https://github.com':bhargavibairagoni
Password for 'https://bhargavibairagoni@github.com':
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (1/1), 895 bytes | 895.00 KiB/s, done.
From https://github.com/bhargavibairagoni/new-repo
 * branch      main      -> FETCH HEAD
   d583ef..44fea3a main      -> origin/main
Updating d583ef..44fea3a
Fast-forward
  bhargavi.txt | 2 ++
  new-repo/file3 | 0
  new-repo/file4 | 0
  3 files changed, 2 insertions(+)
  create mode 100644 bhargavi.txt
  create mode 100644 new-repo/file3
  create mode 100644 new-repo/file4
[root@ip-172-31-21-128 new-repo]$ ll
total 0
-rw-r--r--  1 root root 0 Nov  1 07:05 file3
-rw-r--r--  1 root root 0 Nov  1 07:05 file4
[root@ip-172-31-21-128 new-repo]$
```

Now we can see the new changes in the main branch of local repository.