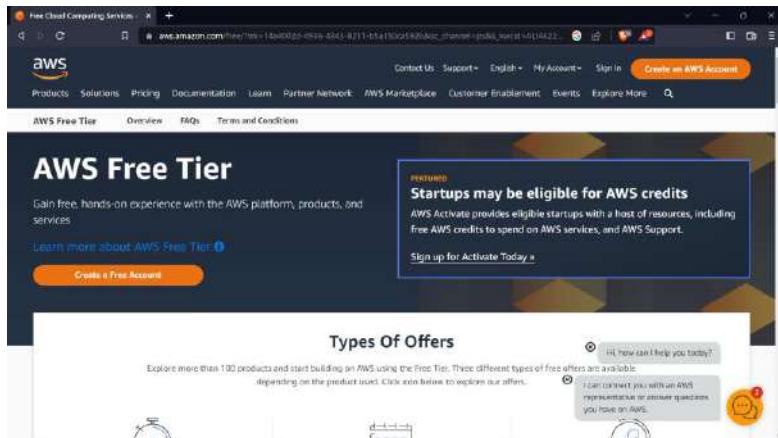


# ASSIGNMENT 1

**Problem Statement:** Create an account in AWS and configure a budget.

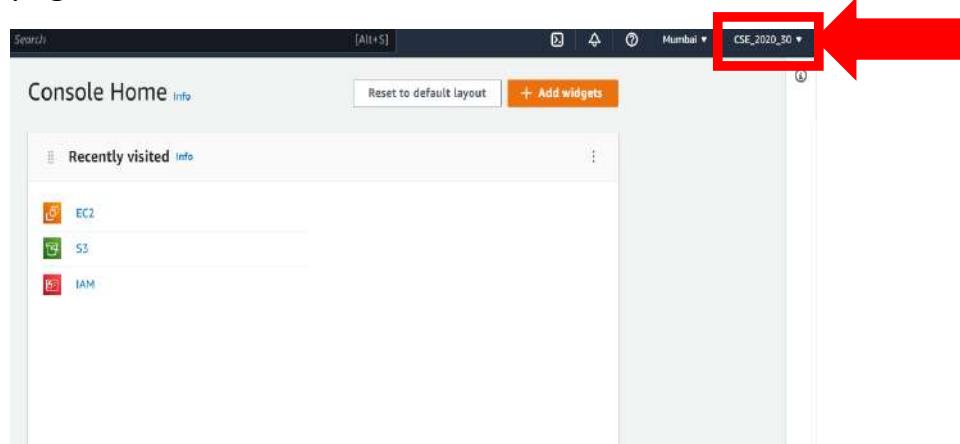
**Procedure: (a) Create an account in AWS**



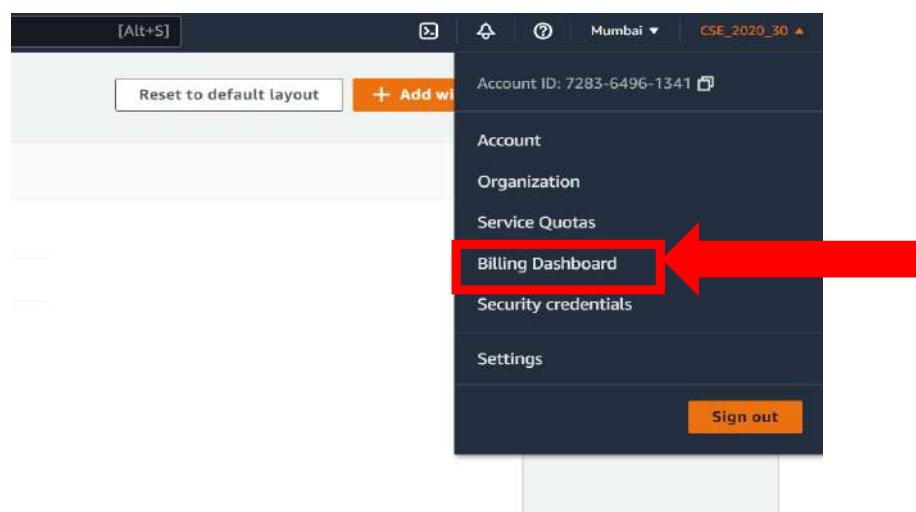
1. Open the [Amazon Web Services home page](#).
2. Choose **Create an AWS account**. Make sure to create the account as a **root user** to unlock full functionality of the newly created AWS account.
3. Enter your account information, and then choose **Continue**. Be sure that you enter your account information correctly, especially your email address.
4. Choose **personal** account.
5. Enter your personal information.
6. Read and accept the **AWS Customer Agreement**.
7. Choose Create Account and Continue.
8. On the Payment Information page, enter the information about your **payment method**, and then choose **Verify and Add**. You can't proceed with the sign-up process until you add a valid payment method.
9. Next, you must **verify your phone number**. Choose your country or region code from the list, and enter a phone number where you can be reached in the next few minutes.
10. Enter the code displayed in the CAPTCHA, and then submit.
11. When the automated system contacts you, enter the PIN you receive and then choose Continue.
12. On the **Select a Support Plan** page, choose **the free tier**.
13. Finally, wait for your new account to be **activated**. This usually takes a few minutes but can take up to **24 hours**.
14. When your account is **fully activated**, you receive a **confirmation email message**. Check your email and spam folder for the confirmation message. **After you receive this email message, you have full access to all AWS services.**

## (b)Configure a Budget

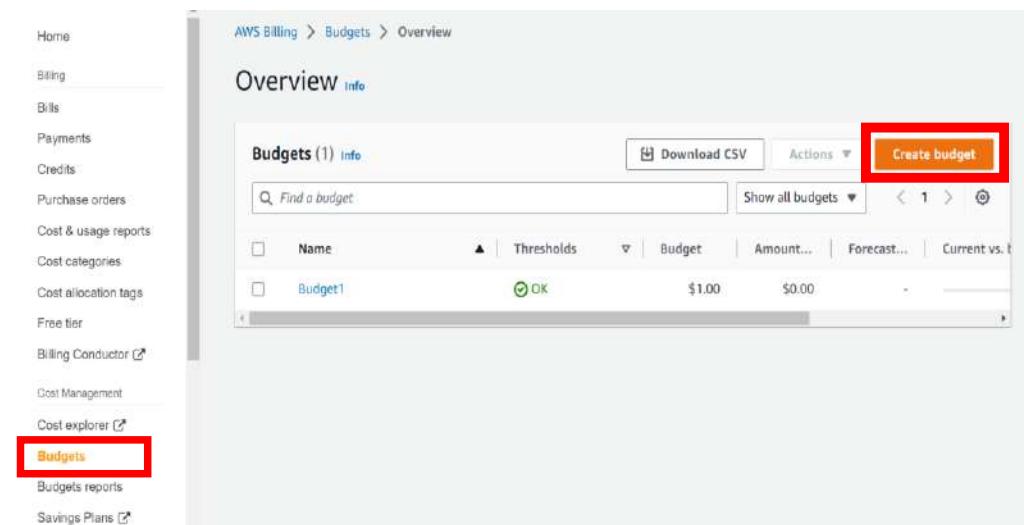
1. Sign in to the console using your credentials for your newly created account.
2. Now after successfully signing in we will arrive at the home page. Then click on down arrow beside your account name on the top right corner of the home page.



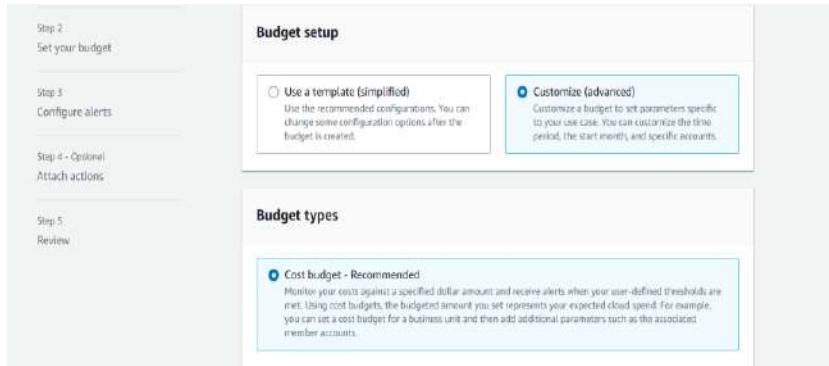
3. A drop down will appear and select **Billing Dashboard** from the list.



4. After arriving in Billing Dashboard, go to the **Budgets** section on the left side panel under cost management. Click on the **Create Budget** button in the **overview page** to start creating a new budget for your AWS account.



5. Select **Customize** in Budget Setup section and **Cost Budget** in the Budget Types section. Then click on **Next**.



6. Enter **Budget Name**. Then set your **Budget Period**, preferably **Monthly** and set **Renewal Type** to **Recurring**.

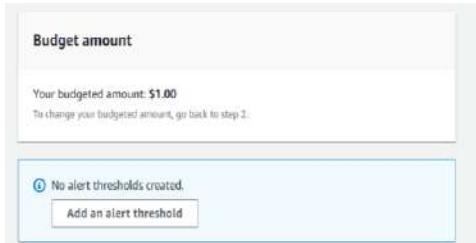
The screenshot shows the 'Set budget amount' step. It includes fields for 'Budget name' (with placeholder 'Provide a descriptive name for this budget.'), 'Period' (set to 'Monthly'), 'Budget renewal type' (radio button selected for 'Recurring budget' with description 'Recurring budgets renew on the first day of every monthly billing period.'), and 'Start month' (set to 'Feb 2023').

7. Scroll down and Enter your Budgeted amount. For best free usage enter 1 in the box. (All amounts are automatically considered in dollars '\$')

The screenshot shows the 'Budgeting method' step. It includes a dropdown 'Budgeting method' set to 'Fixed' (description 'Create a budget that tracks against a single monthly budgeted amount.') and an input field 'Enter your budgeted amount (\$)' containing the value '1'. Below it, a note says 'Last month's cost: \$0.00'.

8. No need to change any other options in the page. So, now we click on **Next**.

9. Next, we move on to Configure Alerts. This is required to let AWS know when to show us a warning when we reach a certain limit or threshold of our budgeted amount in our stipulated time period. Here, we select **Add an Alert Threshold** button.



10. Now set Alert #1. Give a suitable threshold value in percentage of the actual budgeted amount. If that threshold is reached an alert will be send by AWS. Also, mention the email address(/es) where AWS should send you the alerts.

**▼ Alert #1** Remove

**Set alert threshold**

**Threshold** When should this alert be triggered?  
70 % of budgeted amount Actual

**Trigger** How should this alert be triggered?

**Summary:** When your actual cost is greater than 70.00% (\$0.70) of your budgeted amount (\$1.00), the alert threshold will be exceeded.

**Notification preferences**  
Select one or more notification preferences to receive alerts.

**Email recipients**  
Specify the email recipients you want to notify when the threshold has exceeded.  
debrup202002@gmail.com

Maximum number of email recipients is 10.

11. Click on Next.
12. Again, click on Next after reviewing yours alert summary.
13. Now, we are finally at the review or summary page of the whole budget. Click on **Create Budget** button to finally create your Budget!
14. After this step you will be redirected to the **overview page** where all your budgets will be shown. You can see your newly created budget in the table format with various information related to it. Our Budget creation is complete!

AWS Billing > Budgets > Overview

**Overview** Info

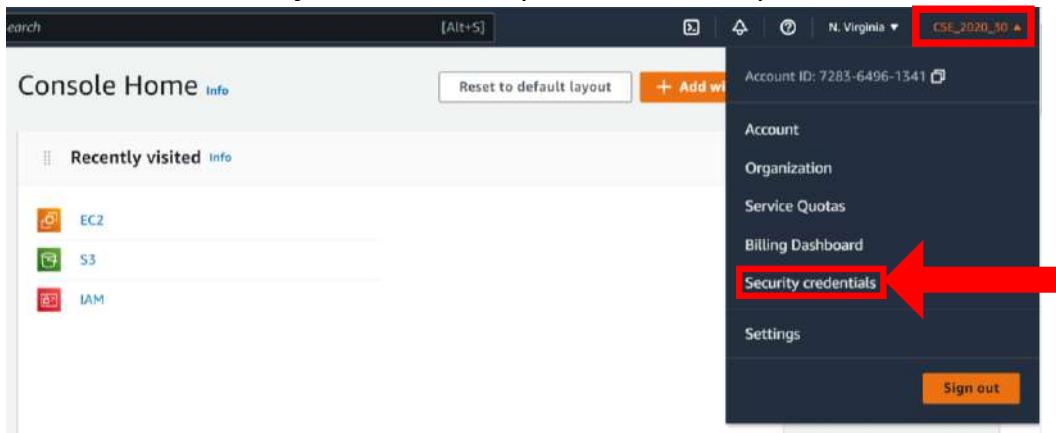
Budgets (2) <a href="#">Info</a>		<a href="#">Download CSV</a>	Actions	<a href="#">Create budget</a>	
<input type="text"/> Find a budget		Show all budgets	< 1 >		
Name	Thresholds	Budget	Amount...	Forecast...	Current vs. t
Budget1	OK	\$1.00	\$0.00	-	-
Budget2	OK	\$1.00	\$0.00	-	-

# **ASSIGNMENT 2**

**Problem Statement:** Create Multi Factor Authentication(MFA) for account authentication.

## **Procedure:**

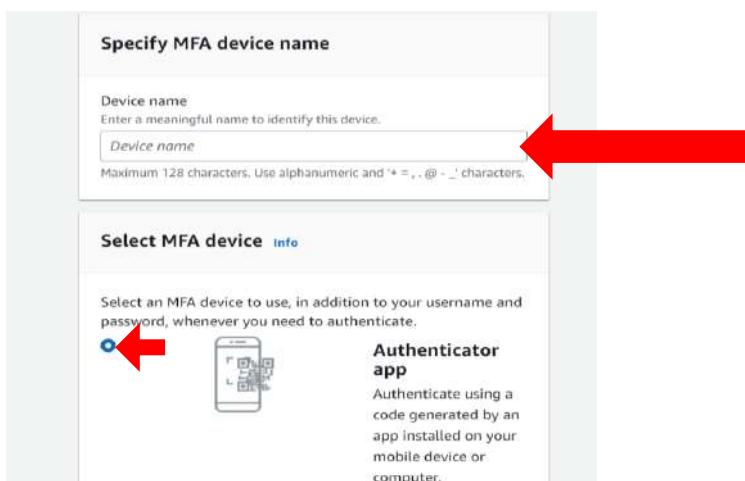
1. Sign-in to your AWS console. Then select the **down arrow** beside your **account name** in the top right side of the page.
2. Now select **Security Credentials** option in the drop-down menu.



3. Now after arriving in My Security Credentials page, now scroll down to **Multi-Factor Authentication (MFA)** section. Click on the **Assign MFA device** button.

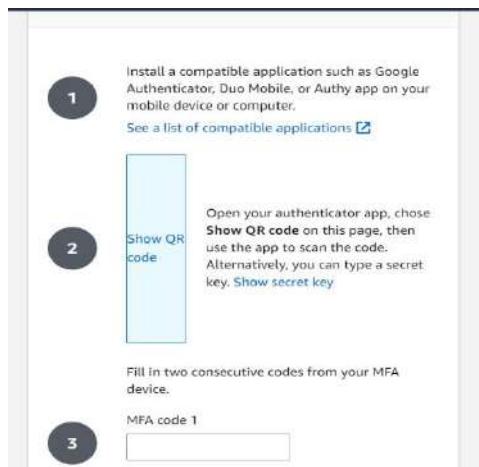


4. Next Assign a **unique device name** (very important and needs to be unique). This name is actually the name of the account that shows up in your authenticator app with the security codes and hence is essential for identification later. Select **Authenticator app** in **Select MFA device** section.



5. Click on **Next**.
6. After that you have to make sure you download an authenticator app from app store in your android/iOS device. Google/Microsoft authenticator is preferred. After installation, click on the **Show QR code box** here in the

website. Scan the QR code with your authenticator app. Your device name given will show up in your authenticator main page.



*Download some authenticator app from an App Store in your mobile phone (Android/iOS). Then continue with steps 2 and 3 of the section as seen in the snapshot.*



Google Authenticator

7. We will see a certain unique combination appearing against our given device name for our AWS account and stays only for 30-60 seconds. Enter 2 consecutive codes appearing in the given box in the website to authenticate your MFA.
8. After successfully verifying your MFA, click on the Add MFA button.
9. You will be redirected to the security credentials page and see your newly added MFA method with your given device name for your account.

Multi-factor authentication (MFA) (1)		
Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. <a href="#">Learn more</a>		
<a href="#">Remove</a>	<a href="#">Resync</a>	<a href="#">Assign MFA device</a>
<input checked="" type="radio"/> Virtual	arn:aws:iam::728364961341:mfa/AWS_College	5 days ago

10. Hence, we have successfully added an MFA device.
11. Now sign-out and try to re-login to the console.
12. Now after providing user email and password, from now on you have to enter the MFA code which is given by the authenticator app in your phone. Be mindful that the code changes every 30-60 seconds and we have to enter the current or existing one which has not expired (or in this case stopped showing).



#### Multi-factor authentication

Your account is secured using multi-factor authentication (MFA). To finish signing in, turn on or view your MFA device and type the authentication code below.

Email address: debrup202002@gmail.com

MFA code

Submit

Troubleshoot MFA

Cancel

## Amazon DocumentDB AllScale Clusters

Scale your document database to handle virtually any number of reads and writes

LEARN MORE

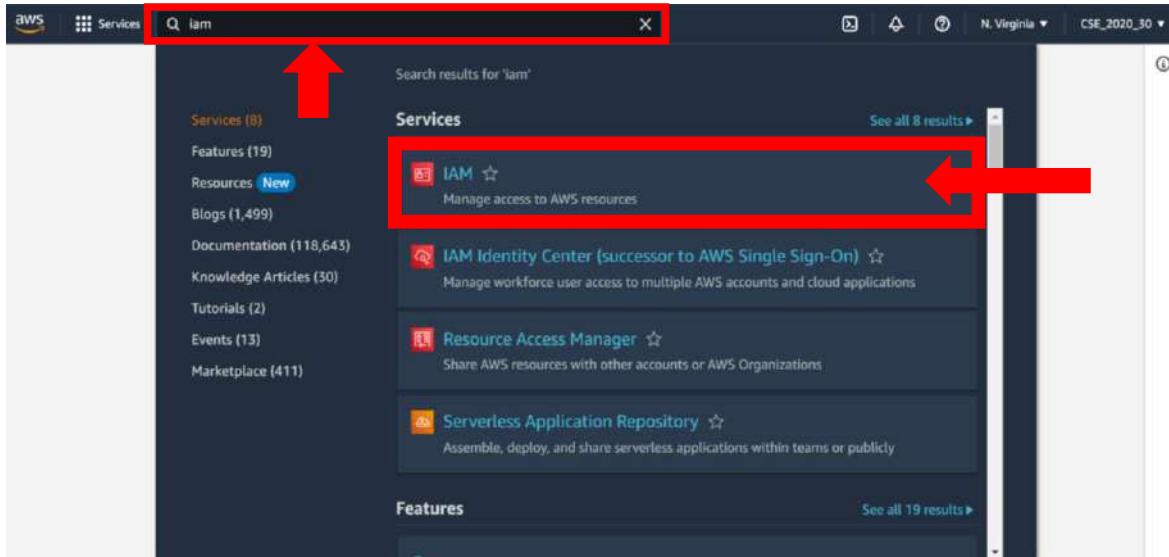


# ASSIGNMENT 3

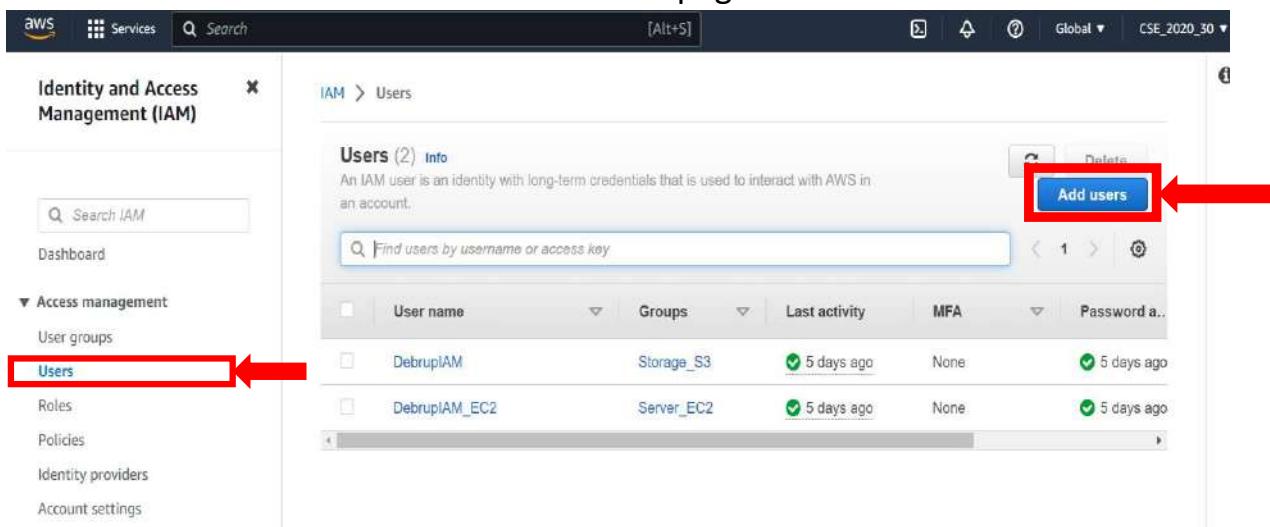
**Problem Statement:** Create IAM resource giving full access of S3(storage).

## Procedure:

1. Sign in to your console (as root user).
2. On the top side of the page go to the **Search bar** and type “IAM”.
3. Click on the first result showing “IAM”.



4. We are then redirected to the Identity and Access Management (IAM) dashboard. We then have to select the **user** option in the left side panel under **Access Management**.
5. Next click on **Add Users** button in the **Users** page.



6. After that you have to create a user and specify the details.
  - a. Specify the name of the user
  - b. **Check** the “Provide user access to the AWS Management Console” box
  - c. **Select** the option “I want to create an IAM user”.
  - d. Select custom password and enter it.
  - e. **Uncheck** the “Users must create a new password at next sign-in” box.
  - f. Then click on next

User details

User name: Test ←

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = . @ \_ - (hyphen)

Provide user access to the AWS Management Console - optional  
↑ If you're providing console access to a person, it's a best practice [to manage their access in IAM Identity Center.](#)

**Are you providing console access to a person?**

Specify a user in Identity Center - Recommended  
We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

I want to create an IAM user  
We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.

Console password:

Autogenerated password  
You can view the password after you create the user.

Custom password  
Enter a custom password for the user.  
..... ←

Users must create a new password at next sign-in (recommended).  
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel

Next →

7. Now under **Permissions Options**, select **Add user to Group** option.
8. Under **User Groups** click on **Create Group** button.

Permissions options

Add user to group  
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

Copy permissions  
Copy all group memberships, attached managed policies, and inline policies from an existing user.

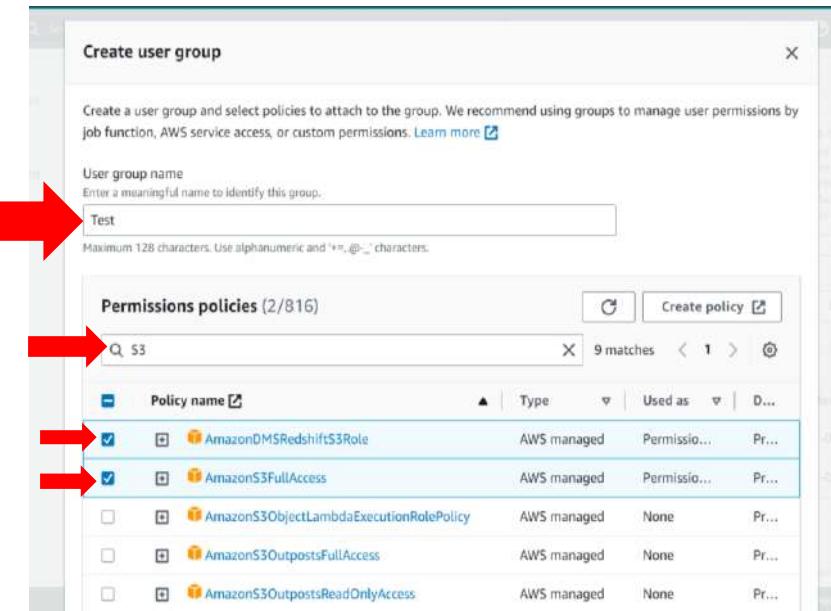
Attach policies directly  
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

User groups (2)

Group name	Users	Attached policies ...	Created
Server_EC2	1	AmazonEC2FullAcce...	2023-02-13 (5 days ...)
Storage_S3	1	AmazonDMSRedshift...	2023-02-13 (5 days ...)

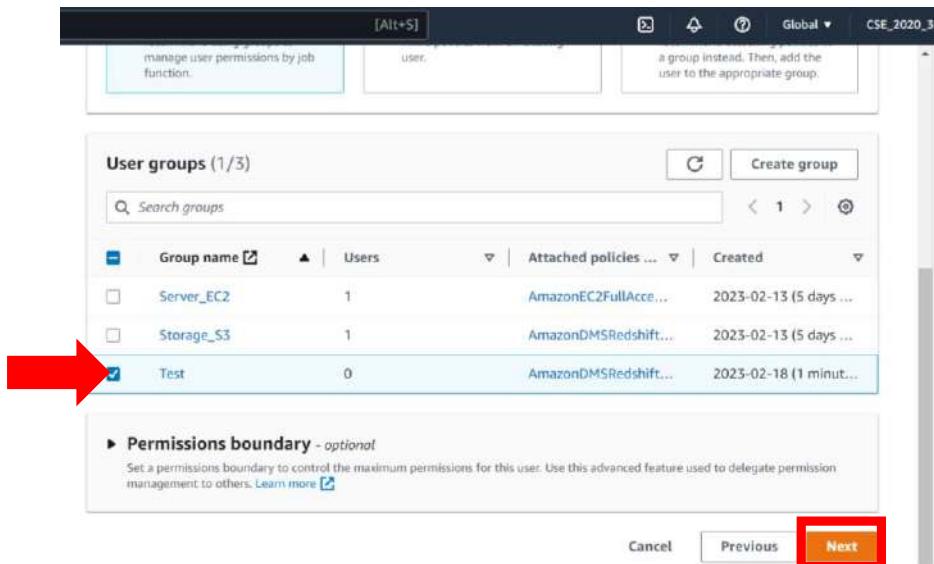
**Create group**

9. A pop-up will appear where you have to specify the new group name and edit the policies/permissions associated with it
  - a. Enter the **User Group Name**
  - b. Next in the find policies search bar type **S3** as we have to give permission only for S3.
  - c. Select the **first two** options
  - d. Then click on **Create User Group**

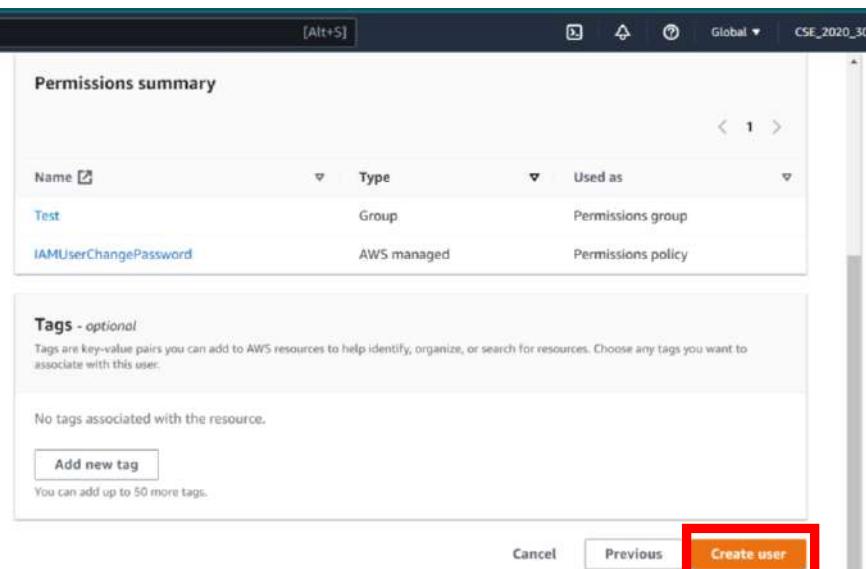


10. Now the pop-up closes and under the **User Groups** section our newly created group is visible in a table format. Select the group.

11. Then click on **Next**.



12. We arrive at the **Review and Create** page. After reviewing click on the **Create User** button.



13. Next, we arrive at the **Retrieve Password** page where we can download a .csv file or email the sign-in details of the newly created IAM user.

Step 1: Specify user details

Step 2: Set permissions

Step 3: Review and create

Step 4: Retrieve password

**Console sign-in details**

Console sign-in URL: https://728364961341.signin.aws.amazon.com/console

User name: Test

Console password: [REDACTED] [Show](#)

[Email sign-in instructions](#)

[Download .csv file](#) [Return to users list](#)

14. After that we can return to users list and see that our new user has been added to the users' table.

**User created successfully**

You can view and download the user's password and email instructions for signing in to the AWS Management Console.

**Users (3) Info**

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Groups	Last activity	MFA	Password a...
DebrupIAM	Storage_S3	5 days ago	None	5 days ago
DebrupIAM_EC2	Server(EC2)	5 days ago	None	5 days ago
<b>Test</b>	Test	Never	None	None

[View user](#) [Delete](#) [Add users](#)

15. Now we logout of our console.

16. Next, we again try to login to the console. But now we select **IAM user login**.

17. Here we have to enter **Account ID** of the root user. We can get that in the drop-down menu after logging in our root user account.

Alternatively, we can use the link in our **downloaded .csv file** or our **email** which if used in our **browser** will redirect use to the login page with the Account ID already entered!

Sign in as IAM user

Account ID (12 digits) or account alias  
728364961341

IAM user name

Password

Remember this account

[Sign in](#)

[Sign in using root user email](#)

[Forgot password?](#)

**ML Use Cases**  
Solve your most common challenges with practical ML

[LEARN MORE](#)

18. Enter the credentials.
19. Note the username in the top right corner. Also, you cannot access your account page as it is controlled only by your root user.
20. Next you can type S3 in the search box and select the first option.

The screenshot shows the AWS Console Home and the Amazon S3 service pages. At the top, there's a search bar with a red arrow pointing to it. To the right of the search bar, a red box highlights the 'Test @ 7283-6496-1341' link. On the S3 page, a green checkmark is placed over the 'Pricing' section, which states: 'With S3, there are no minimum fees. You only pay for what you use. Prices are based on the location of your S3 bucket.'

21. Here we get to Create Bucket. Hence we have full access of S3.
22. Now to check our limits let us search EC2 in search bar. Select the first choice.

The screenshot shows the EC2 Dashboard. On the left, there's a sidebar with 'EC2 Dashboard' and several menu items like 'Instances', 'Launch Templates', etc. The main area shows a table of resources with API error status. On the right, there's a 'Account attributes' section with two error messages: 'An error occurred' (An error occurred retrieving supported platforms) and 'An error occurred' (An error occurred checking for a default VPC). A large red X is drawn over this entire section.

23. Here, we encounter API error. This is proof that we do not have access to EC2. Hence, we have successfully restricted access to our IAM user.
24. Thus, we have successfully created an IAM user and given it only S3 access.
25. Now, we can logout.

# ASSIGNMENT 4

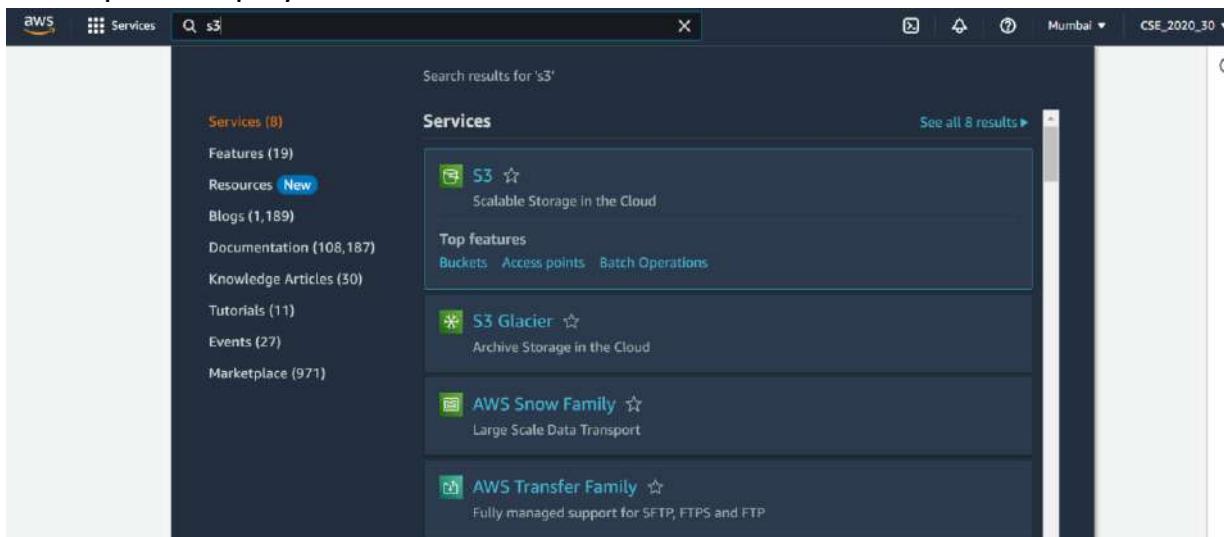
**Problem Statement:** Create a private bucket in AWS. Upload a file and check that through pre-signed URL whether you can access the file or not.

## Procedure:

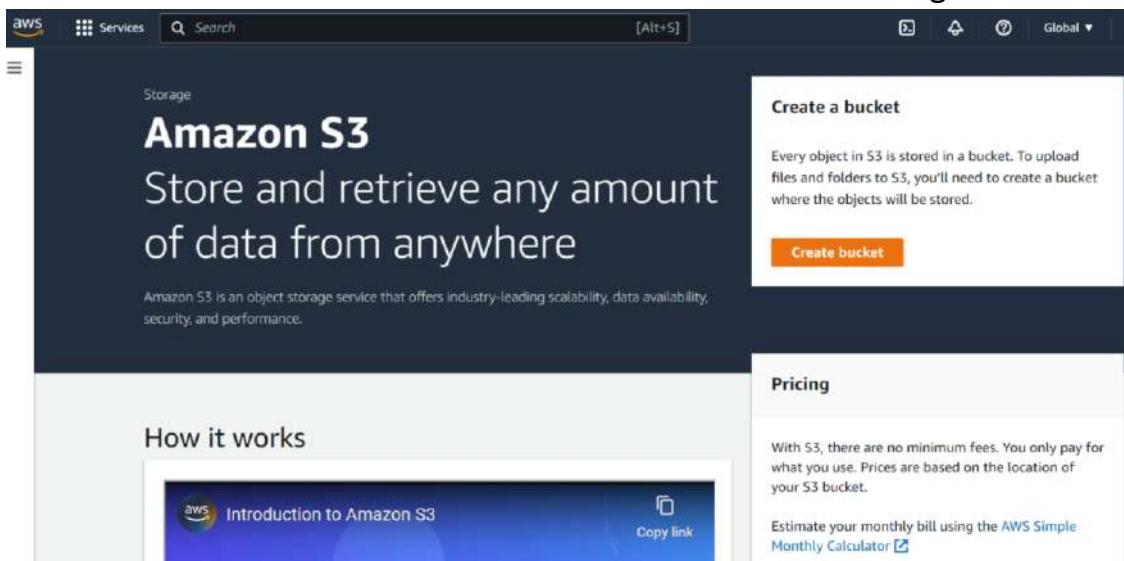
1. Sign in to your **AWS account** as root user.



2. Now in the **homepage** search for **S3** in the **search box** and then select the first option displayed.



3. After clicking on it, you will be redirected to the **Amazon S3** homepage. There we have to click on the **create bucket** button on the right hand side.



4. Next you will go to the **Create bucket screen** where you have to configure your bucket before creating it.

- Choose a globally unique name for your bucket. It should NOT contain any spaces or any uppercase letters.

- Select the **AWS Region** as **Asia Pacific (Mumbai) ap-south-1**.

**Remember** you can avail other options but each server region has **different pricing** associated with it. Since, we are **living in India**, we are choosing the one **closest to us** to remain fairly priced.

- Next we go to Object Ownership section where we keep ACLs disabled option checked (as it is).
- Next, we keep all public access blocked (as it is).
- Everything else remains unchanged.

f. Now click on the Create bucket button.

The screenshot shows the AWS S3 'Create bucket' interface. At the top, the navigation path is 'Amazon S3 > Buckets > Create bucket'. The main title is 'Create bucket' with an 'Info' link. Below it, a sub-section says 'Buckets are containers for data stored in S3. Learn more'.

**General configuration:** This section contains fields for 'Bucket name' (set to 'myawsbucket') and 'AWS Region' (set to 'Asia Pacific (Mumbai) ap-south-1'). There is also a 'Copy settings from existing bucket - optional' section with a 'Choose bucket' button.

**Object Ownership:** This section discusses object ownership rules. It shows two options: 'ACLs disabled (recommended)' (selected) and 'ACLs enabled'. A note states: 'All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.' Another note says: 'Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.'

**Object Ownership details:** It specifies 'Bucket owner enforced'.

**Upcoming permission changes to disable ACLs:** A note states: 'Starting in April 2023, to disable ACLs when creating buckets by using the S3 console, you will no longer need the s3:PutBucketOwnershipControls permission. Learn more.'

**Block Public Access settings for this bucket:** This section explains how to control public access. It notes: 'Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more.'

**Block all public access:** This setting is checked. A note says: 'Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.'

- Block public access to buckets and objects granted through new access control lists (ACLs)**  
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**  
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**  
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**  
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

**Default encryption** [Info](#)  
Server-side encryption is automatically applied to new objects stored in this bucket.

**Encryption key type** [Info](#)  
 Amazon S3-managed keys (SSE-S3)  
 AWS Key Management Service key (SSE-KMS)

**Bucket Key**  
When KMS encryption is used to encrypt new objects in this bucket, the bucket key reduces encryption costs by lowering calls to AWS KMS.  
[Learn more](#)

Disable  
 Enable

**Advanced settings**

[Info](#) After creating the bucket you can upload files and folders to the bucket, and configure additional bucket settings.

[Cancel](#) [Create bucket](#)

- After that we are redirected to the buckets page where we can see all our buckets in a table format.

[Amazon S3](#) > [Buckets](#)

**Account snapshot**  
Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

[View Storage Lens dashboard](#)

**Buckets (1) [Info](#)**  
Buckets are containers for data stored in S3. [Learn more](#)

Name	AWS Region	Access	Creation date
s3debruprivate1	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	February 24, 2023, 20:36:52 (UTC+05:30)

[Create bucket](#)

[Find buckets by name](#)

- Now we click on our newly selected bucket (on the name).
- Now we have successfully entered into our newly created bucket.

[Amazon S3](#) > [Buckets](#) > s3debruprivate1

**s3debruprivate1 [Info](#)**

[Objects](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

**Objects (0)**  
Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 Inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Name	Type	Last modified	Size	Storage class
No objects				

You don't have any objects in this bucket.

[Upload](#)

[Find objects by prefix](#)

- Click the Upload button to upload a file in our bucket.
- After clicking you will be redirected to the Upload page. Click on Add files button to add a file.

The screenshot shows the AWS S3 'Upload' interface. At the top, there's a navigation bar with 'Services', a search bar, and a global region selector set to 'Global'. Below the navigation is a breadcrumb trail: 'Amazon S3 > Buckets > s3debruprivate1 > Upload'. The main area is titled 'Upload' with a 'Info' link. A text instruction says: 'Add the files and folders you want to upload to S3. To upload a file larger than 150GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more'. Below this is a large input box with placeholder text: 'Drag and drop files and folders you want to upload here, or choose Add files, or Add folder.' A 'Files and folders (0)' section follows, containing a table header with columns: Name, Folder, Type, and Size. A search bar labeled 'Find by name' is above the table. The table body displays a message: 'No files or folders' and 'You have not chosen any files or folders to upload.'

10. You will open a pop up to browse from your pc to upload a file. After selection click on upload button.

This screenshot shows the 'Upload' dialog box. It lists a single file '927310.jpg' with a size of '1.1 MB' and type 'image/jpeg'. Below the file list is a 'Destination' section with the path 's3://s3debruprivate1'. Underneath is a 'Destination details' section with a note about bucket settings. At the bottom of the dialog are 'Permissions' and 'Properties' sections, and a prominent orange 'Upload' button.

11. You will then be redirected to the upload status page where a status bar will be present showing the progress of your upload. After completion it will look like this.

This screenshot shows the 'Upload: status' page after a successful upload. A green header bar indicates 'Upload succeeded' with a 'View details below.' link. The main content area has a summary table:

Destination	Succeeded	Failed
s3://s3debruprivate1	1 file, 1.1 MB (100.00%)	0 files, 0 B (0%)

Below the summary are tabs for 'Files and folders' (selected) and 'Configuration'.

12. Close your status page. Now in the bucket page you will see the file you have uploaded in the objects section.

The screenshot shows the AWS S3 console interface. At the top, the navigation bar includes 'Services', 'Search', and a global region selector 'Global ▾ CSE\_2020\_30 ▾'. Below the navigation, the path 'Amazon S3 > Buckets > s3debruprivate1' is displayed. The main content area is titled 's3debruprivate1 [Info](#)'. A horizontal menu bar at the top of the content area includes 'Objects' (which is highlighted in orange), 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. The 'Objects' section displays '(1) Objects'. A sub-section titled 'Objects (1)' provides information about objects in Amazon S3, mentioning the use of Amazon S3 inventory to get a list of all objects in your bucket. It also notes that others need explicit permissions to access your objects. Below this, there are several action buttons: 'Copy', 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', and 'Create folder'. An 'Upload' button is also present. A search bar labeled 'Find objects by prefix' is available. The object list table has columns for Name, Type, Last modified, Size, and Storage class. One object, '927310.jpg', is listed with the details: jpg, February 24, 2023, 21:18:02 (UTC+05:30), 1.1 MB, Standard.

13. Now click on the file.

14. Scroll down and copy the Object URL.

The screenshot shows the 'Properties' tab of the AWS S3 object details page for '927310.jpg'. The 'Object overview' section contains the following details:

- Owner: [REDACTED]
- S3 URI: [REDACTED]
- AWS Region: Asia Pacific (Mumbai) ap-south-1
- Last modified: February 24, 2023, 21:18:02 (UTC+05:30)
- Size: 1.1 MB
- Type: jpg
- Entity tag (Etag): [REDACTED]

A green message box with a checkmark icon appears, stating 'Object URL Copied' and providing a link: <https://s3debruprivate1.s3.ap-south-1.amazonaws.com/927310.jpg>.

15. Paste it in another browser.

16. IT WILL SHOW ERROR.

This is because your uploaded file is in a private bucket. Hence, it cannot be accessed by anyone other than you. Now, to let others access, you can only send them a pre-signed URL which remains active for a specific duration.

17. NOW WE WILL GENERATE A PRESIGNED URL

18. Go back to the previous page and select your file.

The screenshot shows the 'Properties' tab of the AWS S3 object details page for '927310.jpg'. The 'Actions' button is highlighted in orange. The 'Objects' section displays '(1) Objects' with the same details as before: Name (927310.jpg), Type (jpg), Last modified (February 24, 2023, 21:18:02 (UTC+05:30)), Size (1.1 MB), and Storage class (Standard).

19. Next click on the Actions button as shown above.

## 20. Select the “Share with presigned URL” option.

The screenshot shows the AWS S3 console interface. At the top, it displays 'Amazon S3 > Buckets > s3debruprivate1'. Below this, the bucket name 's3debruprivate1' is shown with a 'Info' link. The 'Objects' tab is selected, showing one object: '927310.jpg'. The object details are: Type: jpg, Last modified: February 24, 2023, 21:18:02 (UTC+05:30), Size: 1.1 MB, Storage class: Standard. To the right of the object list, a context menu is open, listing options like 'Download as', 'Share with a presigned URL', 'Calculate total size', 'Copy', 'Move', 'Initiate restore', 'Query with S3 Select', 'Edit actions', 'Rename object', and 'Edit storage class'. The 'Share with a presigned URL' option is highlighted with a yellow background.

21. Now a pop-up will appear as shown below. You have to specify the duration for which the link remains active. Next click on Create presigned URL.

Note that after creation the URL link automatically gets copied (in your clipboard). So you do not have to manually copy it. Just right click and paste it in another browser( Or use Ctrl+V shortcut in the browser search box)

The screenshot shows a modal dialog titled 'Share "927310.jpg" with a presigned URL'. It contains a note: 'Presigned URLs are used to grant access to an object for a limited time.' Below this, a message states: 'Anyone can access the object with this presigned URL until it expires, even if the bucket, and object are private.' Under the heading 'Time interval until the presigned URL expires', there are two radio buttons: 'Minutes' (selected) and 'Hours'. A dropdown menu labeled 'Number of minutes' shows '30' is selected. A note below says: 'Must be a whole number between 1 and 720.' At the bottom, it says: 'After you create the presigned URL, it's automatically copied to your clipboard.' There are 'Cancel' and 'Create presigned URL' buttons at the bottom. The background shows the AWS S3 interface with the same file '927310.jpg' listed.

The screenshot shows a browser context menu with the 'Copy' option highlighted in yellow. Other options visible include 'Cut', 'Paste', 'Delete', 'Select all', and links to 'Manage search engines and site search' and 'Always show full URLs'. The background shows a dark-themed browser window with a tab labeled 'New Tab'.

22. After pasting the link in the bar, press Enter key. Now we can access our file using the presigned URL.



So, our file is private and can only be accessed by those with the pre-signed URL link for a limited duration.

# ASSIGNMENT 5

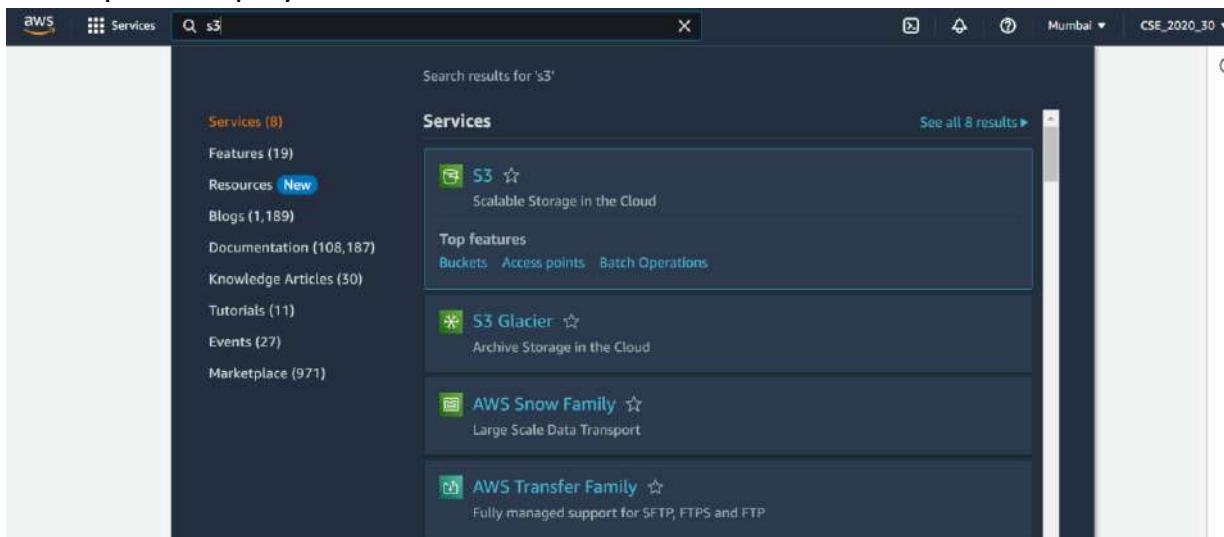
**Problem Statement:** Create a public bucket in AWS. Upload a file and give the necessary permission to check the file URL is working or not.

## Procedure:

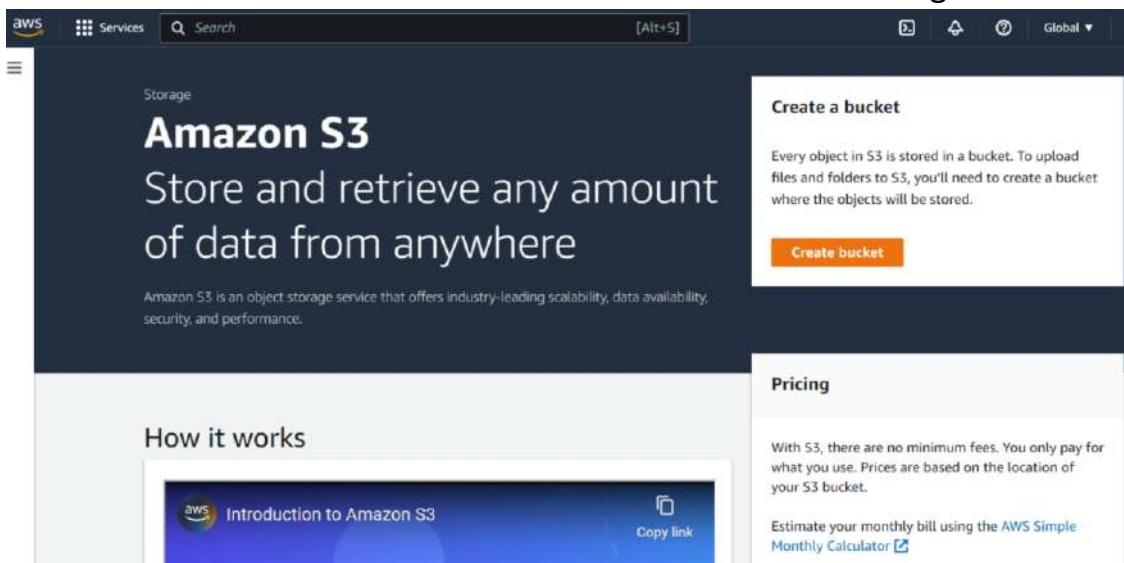
1. Sign in to your **AWS account** as root user.



2. Now in the **homepage** search for **S3** in the **search box** and then select the first option displayed.



3. After clicking on it, you will be redirected to the **Amazon S3** homepage. There we have to click on the **create bucket** button on the right-hand side.



4. Next you will go to the **Create bucket screen** where you have to configure your bucket before creating it.

- a. Choose a globally unique name for your bucket. It should NOT contain any spaces or any uppercase letters.
- b. Select the **AWS Region** as **Asia Pacific (Mumbai) ap-south-1**.  
**Remember** you can avail other options but each server region has **different pricing** associated with it. Since, we are **living in India**, we are choosing the one **closest to us** to remain fairly priced.
- c. Next, we go to Object Ownership section where we keep ACLs enabled option checked. We also keep “Bucket owner preferred” option checked in the object ownership choice
- d. Next, we UNCHECK the Block all public access option.
- e. Remember to check the “I acknowledge that the current settings might result in this bucket and the objects within becoming public” option just below.
- f. Everything else remains unchanged.
- g. Now click on the Create bucket button.

The screenshot shows the 'Create bucket' wizard in the AWS S3 console. The first section, 'General configuration', includes fields for 'Bucket name' (set to 's3debrupublic1'), 'AWS Region' (set to 'Asia Pacific (Mumbai) ap-south-1'), and a 'Copy settings from existing bucket - optional' section. The second section, 'Object Ownership', shows two options: 'ACLs disabled (recommended)' and 'ACLs enabled' (which is selected). Below this, the 'Bucket owner preferred' option is selected under 'Object Ownership'. A note at the bottom states: 'If you want to enforce object ownership for new objects only, your bucket policy must specify that the bucket-owner-full-control canned ACL is required for object uploads.' A 'Learn more' link is provided for this note.

**Block Public Access settings for this bucket**

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

- Block all public access**  
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
- Block public access to buckets and objects granted through new access control lists (ACLS)**  
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLS)**  
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**  
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**

**⚠️ Turning off block all public access might result in this bucket and the objects within becoming public.**  
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

I acknowledge that the current settings might result in this bucket and the objects within becoming public.

**Default encryption [Info](#)**  
Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption key type [Info](#)

- Amazon S3-managed keys (SSE-S3)
- AWS Key Management Service key (SSE-KMS)

Bucket Key  
When KMS encryption is used to encrypt new objects in this bucket, the bucket key reduces encryption costs by lowering calls to AWS KMS.  
[Learn more](#)

- Disable
- Enable

**▶ Advanced settings**

[ⓘ After creating the bucket you can upload files and folders to the bucket, and configure additional bucket settings.](#)

[Cancel](#) [Create bucket](#)

- After that we are redirected to the buckets page where we can see all our buckets in a table format.

**Successfully created bucket "s3debrupublic1"**  
To upload files and folders, or to configure additional bucket settings choose [View details](#).

**Account snapshot**  
Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

**Buckets (2) [Info](#)**  
Buckets are containers for data stored in S3. [Learn more](#)

Name	AWS Region	Access	Creation date
s3debruprivate1	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	February 24, 2023, 20:36:52 (UTC+05:30)
s3debrupublic1	Asia Pacific (Mumbai) ap-south-1	Objects can be public	February 25, 2023, 00:04:32 (UTC+05:30)

- Now we click on our newly selected bucket (on the name).
- Now we have successfully entered into our newly created bucket.
- Click the Upload button to upload a file in our bucket.

Amazon S3 > Buckets > s3debrupublic1

## s3debrupublic1 [Info](#)

- [Objects](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

**Objects (0)**

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 Inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

[Delete](#) [Actions ▾](#)

[Create folder](#) [Upload](#)

[Find objects by prefix](#)

Name	Type	Last modified	Size	Storage class
No objects				

You don't have any objects in this bucket.

9. After clicking you will be redirected to the Upload page. Click on Add files button to add a file.

Amazon S3 > Buckets > s3debrupublic1 > Upload

## Upload [Info](#)

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

Drag and drop files and folders you want to upload here, or choose [Add files](#), or [Add folders](#).

**Files and folders (0)**

All files and folders in this table will be uploaded.

[Find by name](#)

Name	Folder	Type	Size
No files or folders			

You have not chosen any files or folders to upload.

10. You will be shown a box to browse from your pc to upload a file. After selection click on upload button.

**Files and folders (1 Total, 4.0 MB)**

All files and folders in this table will be uploaded.

Name	Folder	Type	Size
912442.jpg	-	image/jpeg	4.0 MB

**Destination**

Destination  
s3://s3debrupublic1

**Destination details**  
Bucket settings that impact new objects stored in the specified destination.

**Permissions**  
Grant public access and access to other AWS accounts.

**Properties**  
Specify storage class, encryption settings, tags, and more.

[Cancel](#) [Upload](#)

11. You will then be redirected to the upload status page where a status bar will be present showing the progress of your upload. After completion it will look like the following.

The screenshot shows the 'Upload: status' page from the AWS S3 console. At the top, a green bar indicates 'Upload succeeded' with a link to 'View details below.' Below this, the title 'Upload: status' is displayed with a 'Close' button. A message box states: 'The information below will no longer be available after you navigate away from this page.' The main area is titled 'Summary' and shows the destination 's3://s3debrupublic1' with a status of 'Succeeded' and '1 file, 4.0 MB (100.00%)'. To the right, there is a section for 'Failed' files with '0 files, 0 B (0%)'. Below the summary, there are tabs for 'Files and folders' (which is selected) and 'Configuration'. Under 'Files and folders', it says '(1 Total, 4.0 MB)' and lists one item: '912442.jpg'.

12. Close your status page. Now in the bucket page you will see the file you have uploaded in the objects section.

The screenshot shows the 'Objects' tab of the 's3debrupublic1' bucket page. It displays a single object named '912442.jpg' with the following details:

Name	Type	Last modified	Size	Storage class
912442.jpg	jpg	February 25, 2023, 00:13:50 (UTC+05:30)	4.0 MB	Standard

13. Now click on the file.
14. Scroll down and copy the Object URL.

The screenshot shows the 'Properties' tab for the file '912442.jpg'. In the 'Object overview' section, the 'Object URL Copied' message is visible, along with the URL: <https://s3debrupublic1.s3.ap-south-1.amazonaws.com/912442.jpg>.

15. Paste it in another browser.
16. IT WILL SHOW ERROR.

This is because your uploaded file, though present in a public bucket, has to be given specific permission so that others can access it. Hence, it cannot

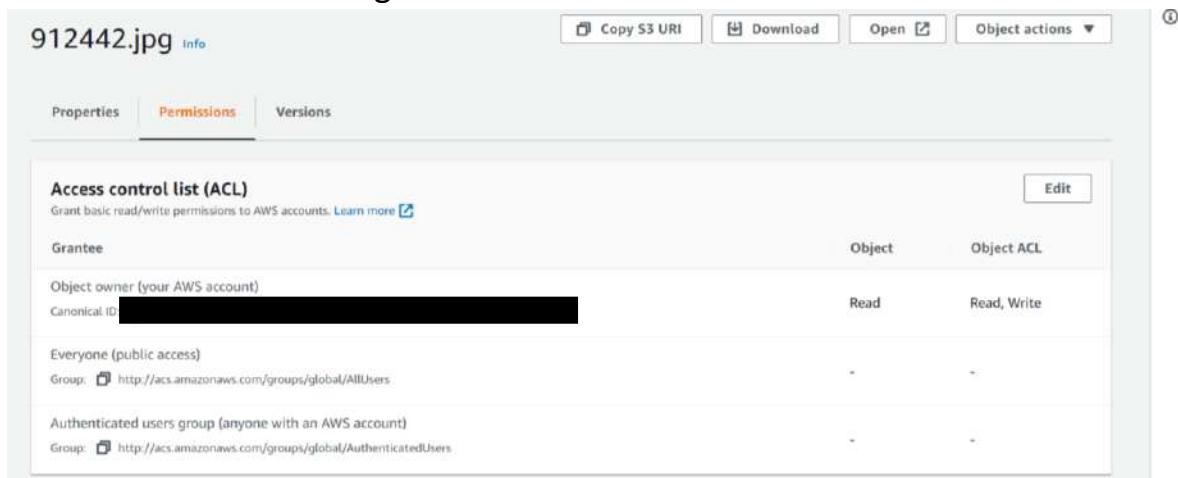
be accessed by anyone other than you. Now, to let others access, you can change the permissions associated with your file using the ACL.

## 17. NOW WE WILL GIVE PERMISSIONS.....

18. Scroll to the top and Click on the permissions bar on the top below the filename and beside the properties bar(in orange font).

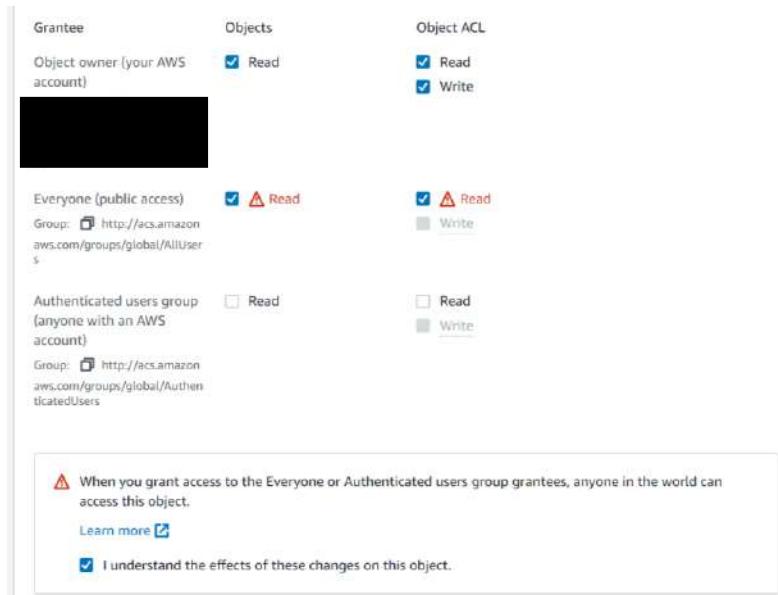


19. You will arrive in the permissions section of your file. You will clearly see the first section as the ACL or Access Control List of your file. Now click on the Edit button on the right hand side of the ACL section.



20. Now select CHECK both “Read”(Object & Object ACL) option for **Everyone(public access)**

Also, remember to check the I understand the effects of these changes on this object option, to further proceed.



21. Now, scroll down and click the save changes button.

**⚠️** When you grant access to the Everyone or Authenticated users group grantees, anyone in the world can access this object.

[Learn more](#)

I understand the effects of these changes on this object.

**Access for other AWS accounts**

No other AWS accounts associated with this resource.

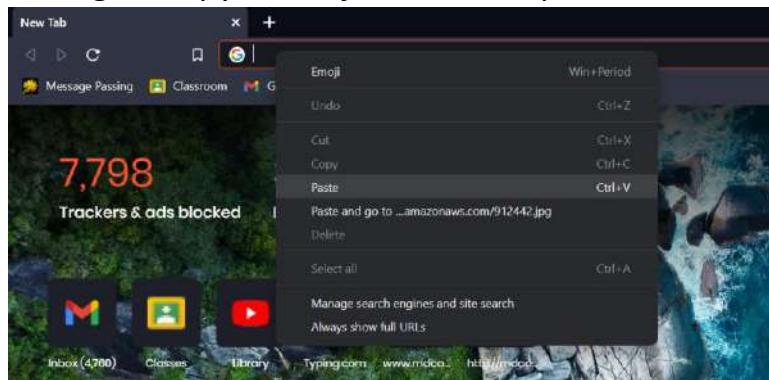
[Add grantee](#)

**Specified objects**

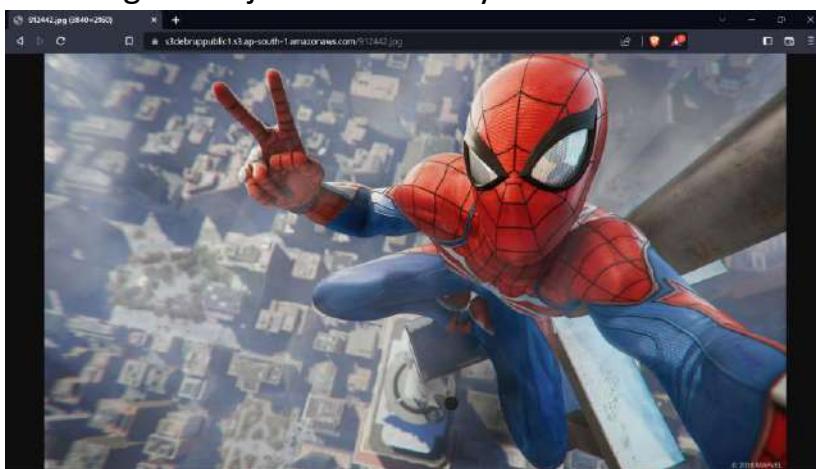
Name	Type	Last modified	Size
912442.jpg	jpg	February 25, 2023, 00:13:50 (UTC+05:30)	4.0 MB

[Cancel](#) [Save changes](#)

22. Now, again copy the object URL and paste it in a different browser.



23. After pasting the link in the bar, press Enter key. Now we can access our file using the object URL directly.



So, our file is public and can be accessed by those with the object URL(link) anytime anywhere. The URL is working perfectly as intended.

# ASSIGNMENT 6

**Problem Statement:** Upload a static website on S3.

**Procedure:**

1. Sign in to your AWS account as a root user.

The screenshot shows the AWS S3 service page. At the top right, there's a 'Create a bucket' button. Below it, a text box explains that every object in S3 is stored in a bucket. On the left, there's a 'How it works' section with a link to 'Introduction to Amazon S3'. On the right, there's a 'Pricing' section with a note about no minimum fees and a link to the 'AWS Simple Monthly Calculator'.

2. Then create a **public S3 bucket**. (Refer to Assignment 5 for full procedure)

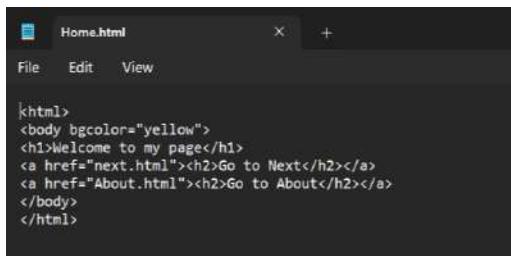
The screenshot shows the 'Buckets' list page. A green banner at the top indicates a successful creation of the bucket 's3depublic1'. The main table lists one bucket: 's3depublic1' located in 'Asia Pacific (Mumbai)' with 'Access' set to 'public'. There are buttons for 'Copy ARN', 'Empty', 'Delete', and 'Create bucket'.

3. Now create three html files anywhere in your computer (preferably in a folder in the Desktop).

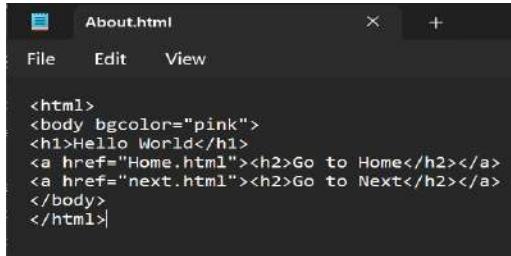
- a. The first one is named Home.html
- b. The second one is named About.html
- c. The third one is named next.html

Remember, you can give any name to the files but you have to modify the steps shown further accordingly. We are going to use the given file names to proceed.

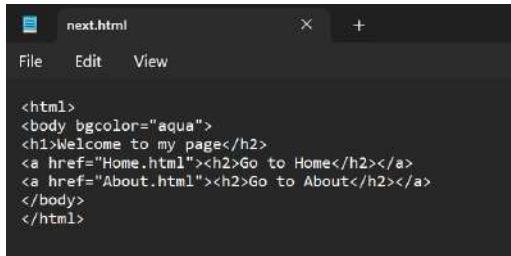
The contents of the files are shown in the given snapshots.....



```
<html>
<body bgcolor="yellow">
<h1>Welcome to my page</h1>
<a href="next.html"><h2>Go to Next</h2></a>
<a href="About.html"><h2>Go to About</h2></a>
</body>
</html>
```

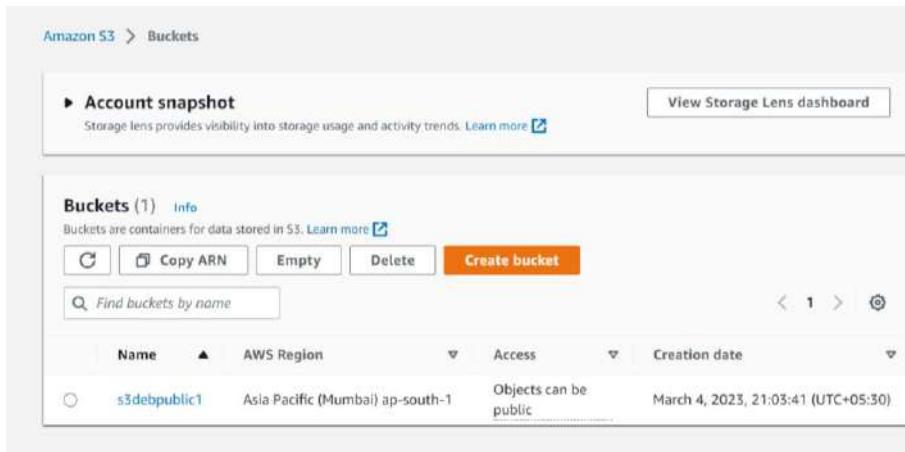


```
<html>
<body bgcolor="pink">
<h1>Hello World</h1>
<a href="Home.html"><h2>Go to Home</h2></a>
<a href="next.html"><h2>Go to Next</h2></a>
</body>
</html>
```



```
<html>
<body bgcolor="aqua">
<h1>Welcome to my page</h1>
<a href="Home.html"><h2>Go to Home</h2></a>
<a href="About.html"><h2>Go to About</h2></a>
</body>
</html>
```

4. Now let us go back to AWS. Select the newly created public bucket.



Amazon S3 > Buckets

**Account snapshot**  
Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

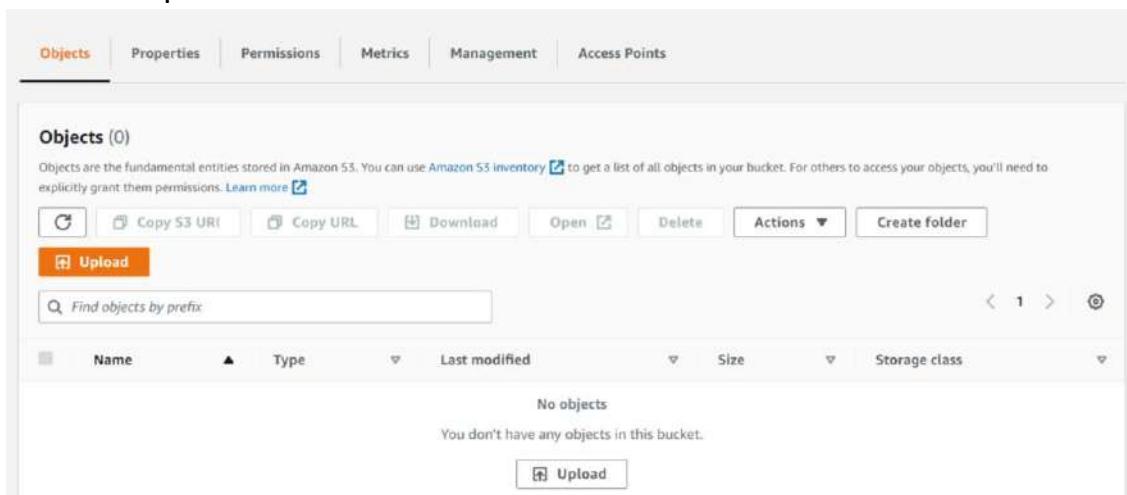
**Buckets (1)** [Info](#)  
Buckets are containers for data stored in S3. [Learn more](#)

[Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

Find buckets by name

Name	AWS Region	Access	Creation date
s3debpublic1	Asia Pacific (Mumbai) ap-south-1	Objects can be public	March 4, 2023, 21:03:41 (UTC+05:30)

5. Click the upload button.



Objects (0)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory [to get a list of all objects in your bucket](#). For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

[Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#)

**Upload**

Find objects by prefix

Name	Type	Last modified	Size	Storage class
No objects				

You don't have any objects in this bucket.

[Upload](#)

6. Next click on the add files button. Select all the three html files and upload them by pressing the upload button.

The screenshot shows the AWS S3 'Upload' interface. At the top, it says 'Amazon S3 > Buckets > s3debpublic1 > Upload'. Below that is a large central area with a placeholder message: 'Drag and drop files and folders you want to upload here, or choose Add files, or Add folder.' Underneath this is a table titled 'Files and folders (0)' with a single row: 'All files and folders in this table will be uploaded.' There are three buttons at the top of the table: 'Remove', 'Add files', and 'Add folder'. A search bar labeled 'Find by name' is present. The table has columns: Name, Folder, Type, and Size. A message 'No files or folders' is centered below the table, followed by the sub-instruction 'You have not chosen any files or folders to upload.'

The screenshot shows a Windows file explorer window with the path 'This PC > Local Disk (D:) > AWS html'. In the main pane, there are three files selected: 'About.html', 'Home.html', and 'next.html'. The status bar at the bottom shows the file names and their sizes: 'File name: "About.html" "Home.html" "next.html"' and 'Size: 1 KB'. The file explorer has a standard ribbon menu and navigation buttons.

The screenshot shows the AWS S3 'Upload' dialog. In the 'Files and folders' section, three files are listed: 'About.html', 'Home.html', and 'next.html', each with a checkbox next to its name. Below this is a 'Destination' section where the URL 's3://s3debpublic1' is entered. Under 'Destination details', it says 'Bucket settings that impact new objects stored in the specified destination.' At the bottom, there are sections for 'Permissions' (with a note about public access) and 'Properties' (with a note about storage class). At the very bottom right are 'Cancel' and 'Upload' buttons.

7. Now, after closing the upload status page after all files are updated, select all the files shown in the object table in the public bucket.

Name	Type	Last modified	Size	Storage class
About.html	html	March 4, 2023, 21:32:56 (UTC+05:30)	159.0 B	Standard
Home.html	html	March 4, 2023, 21:32:57 (UTC+05:30)	170.0 B	Standard
next.html	html	March 4, 2023, 21:32:57 (UTC+05:30)	168.0 B	Standard

8. Click on the actions button and from the subsequent dropdown menu select Make public using ACL.

Name	Type	Last modified	Size	Storage class
About.html	html	March 4, 2023, 21:32:56 (UTC+05:30)	159.0 B	Standard
Home.html	html	March 4, 2023, 21:32:57 (UTC+05:30)	170.0 B	Standard
next.html	html	March 4, 2023, 21:32:57 (UTC+05:30)	168.0 B	Standard

9. Click on Make public button to make all the selected files public.

Name	Type	Last modified	Size
About.html	html	March 4, 2023, 21:32:56 (UTC+05:30)	159.0 B
Home.html	html	March 4, 2023, 21:32:57 (UTC+05:30)	170.0 B
next.html	html	March 4, 2023, 21:32:57 (UTC+05:30)	168.0 B

10. Now, we after making public we close the status page and we are redirected to our public bucket page. Now we select the properties tab of the bucket which is located in the right side of the objects tab (in orange font).

**Objects (3)**

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory [\[Learn more\]](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [\[Learn more\]](#)

Name	Type	Last modified	Size	Storage class
About.html	html	March 4, 2023, 21:32:56 (UTC+05:30)	159.0 B	Standard
Home.html	html	March 4, 2023, 21:32:57 (UTC+05:30)	170.0 B	Standard
next.html	html	March 4, 2023, 21:32:57 (UTC+05:30)	168.0 B	Standard

11. After arriving in the properties tab. Scroll down all the way to the bottom. We will focus only on the Static website hosting section. By default, it will show disabled. We have to enable it. In order to do so click on the Edit button on the right-hand side of the section.

Disabled

Amazon S3 currently does not support enabling Object Lock after a bucket has been created. To enable Object Lock for this bucket, contact Customer Support [\[Learn more\]](#)

**Requester pays**

When enabled, the requester pays for requests and data transfer costs, and anonymous access to this bucket is disabled. [\[Learn more\]](#)

Requester pays  
Disabled

**Static website hosting**

Use this bucket to host a website or redirect requests. [\[Learn more\]](#)

Static website hosting  
Disabled

12. Now we will choose the enable option. After choosing it multiple options will appear. Just follow the snapshot provided below and make the same changes.

Amazon S3 > Buckets > s3depublic1 > Edit static website hosting

**Edit static website hosting** [\[Info\]](#)

**Static website hosting**

Use this bucket to host a website or redirect requests. [\[Learn more\]](#)

Static website hosting

Disable

Enable

**Hosting type**

Host a static website

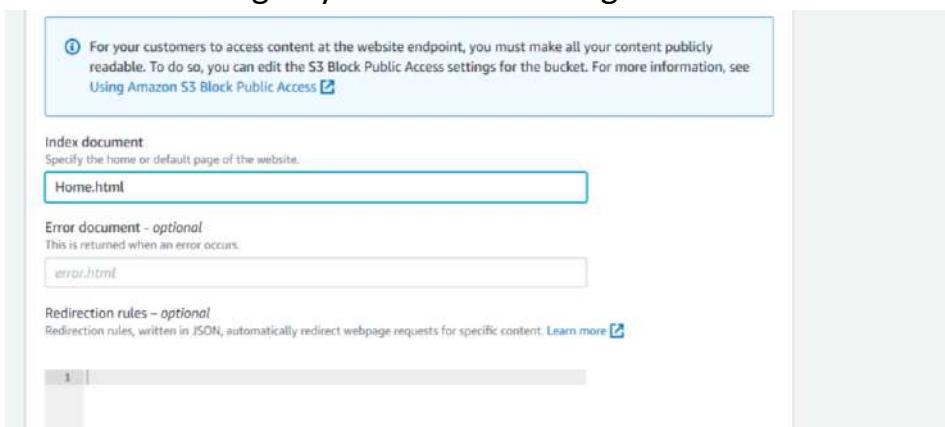
Use the bucket endpoint as the web address. [\[Learn more\]](#)

Redirect requests for an object

Redirect requests to another bucket or domain. [\[Learn more\]](#)

For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see [Using Amazon S3 Block Public Access](#) [\[Learn more\]](#)

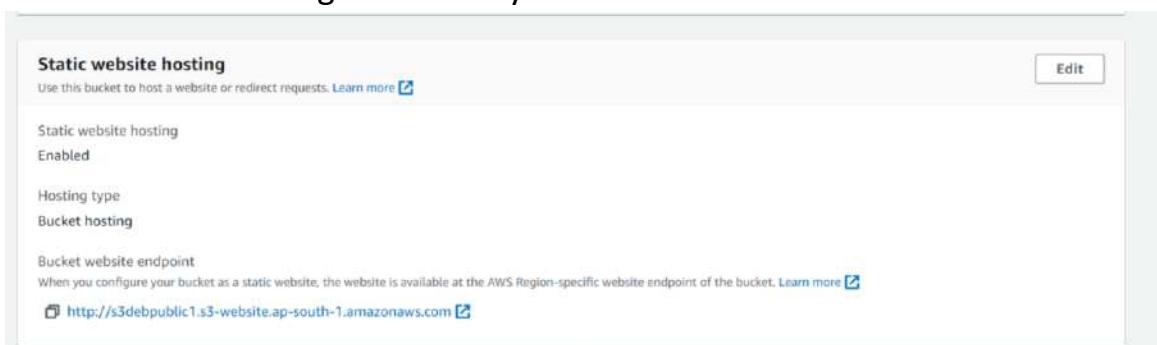
13. Next scroll down. Now we have to mention the html document our link will show. This is the one which anyone can access using the bucket link. In our case we will mention Home.html as our index or main html file. You can choose according to your wish and design.



14. Next scroll down and click on save changes button.



15. Now, you will again arrive in the bucket's properties tab. Scroll down to the static website hosting area. Now you can see a link has arrived.



16. Copy the link and paste it in another browser.



17. We are now viewing the Home.html page. And as scripted, we can click on the Go to Next to arrive at the next.html page. Similarly, we can click on the Go to About page to go the About.html page.



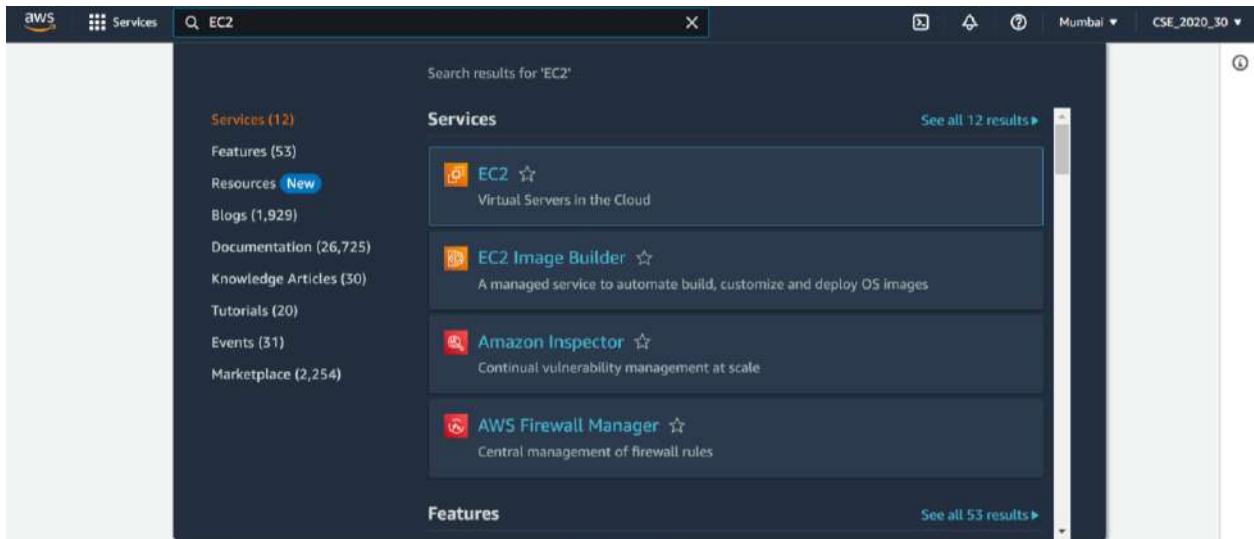
**So, we have successfully hosted a static website in amazon S3.**

# ASSIGNMENT-7

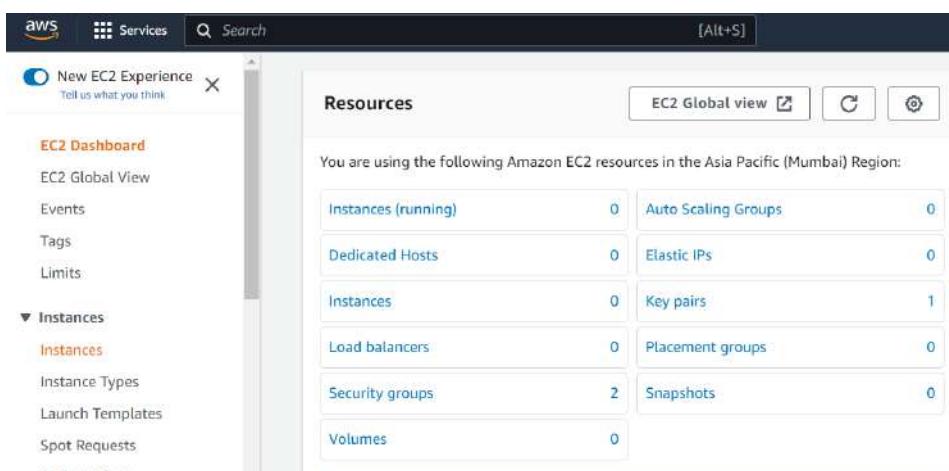
**Problem Statement:** Upload a static website on EC2.

## Procedure:

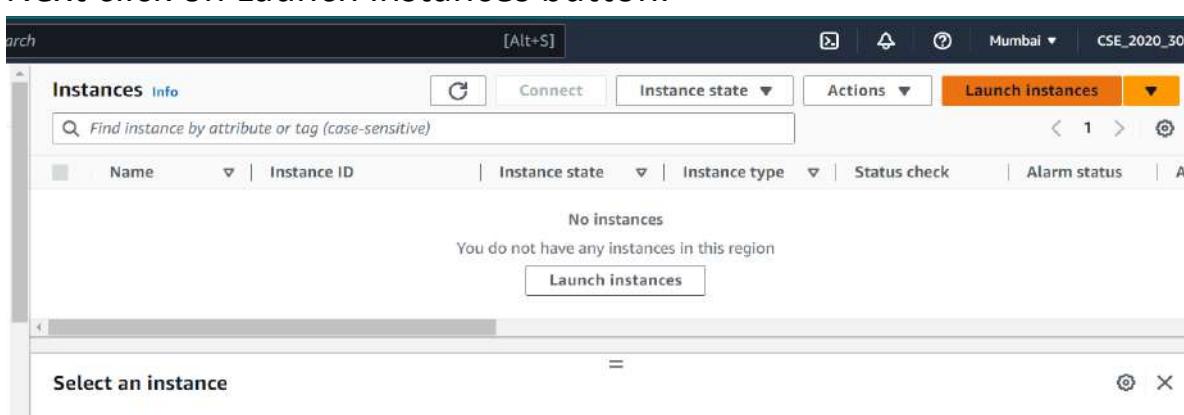
1. Login to your AWS account as root user. Then search “EC2” in the search box. Click on the first result that appears.



2. Click on Instances dropdown menu on the left sidebar. Then again click on instances.



3. Next click on Launch instances button.



4. Now customize the instance you want to launch.
  - a. Set the unique instance name.
  - b. Select Ubuntu as OS.
  - c. Next go to key pair(login) section.
    - i. Click on create new key pair
    - ii. Enter the name of key pair.
    - iii. Select RSA as Key pair type.
    - iv. Select “. pem” as file format.
    - v. Create the key pair.
    - vi. Save the automatically downloaded file. It will be required later.
  - d. Now select the newly created key pair from the dropdown selection.
  - e. Go at the bottom of the network settings section and check the
    - i. Allow HTTP traffic box.
    - ii. Allow HTTPS traffic box.
  - f. Next Click on Launch Instance button on the right side.

The screenshot shows the AWS Lambda console interface. At the top, there's a search bar and a navigation menu with options like 'Compute', 'Logs', 'Metrics', 'CloudWatch Metrics', 'CloudWatch Logs', and 'Lambda'. Below the search bar, there's a 'Create New Function' button. The main area is titled 'HelloWorld' and shows the following details:

- Runtime:** Node.js 12.x
- Description:** A simple Hello World application that prints "Hello World" to the CloudWatch logs.
- Code entry type:** Lambda@Edge
- Code:** A small snippet of Node.js code that prints "Hello World".
- Test:** A button to run a test event.
- Environment variables:** A table with one entry: 'LOG\_GROUP' set to '/aws/lambda/helloworld'.
- Role:** arn:aws:lambda:us-east-1:123456789012:role/lambdaBasicExecutionRole
- Tags:** None
- Deployment:** A table showing deployment history with two rows: 'Deployed' and 'Deployed'.
- Logs:** A link to view CloudWatch Logs.
- Metrics:** A link to view CloudWatch Metrics.
- CloudWatch Metrics:** A link to view CloudWatch Metrics.
- CloudWatch Logs:** A link to view CloudWatch Logs.
- Version:** \$LATEST
- Invoke URL:** https://123456789012.execute-api.us-east-1.amazonaws.com/edge/helloworld

- 5.** Now check whether your newly created instance is running or not in the instances page. Note it will take a few seconds to show the running status. (From Pending)

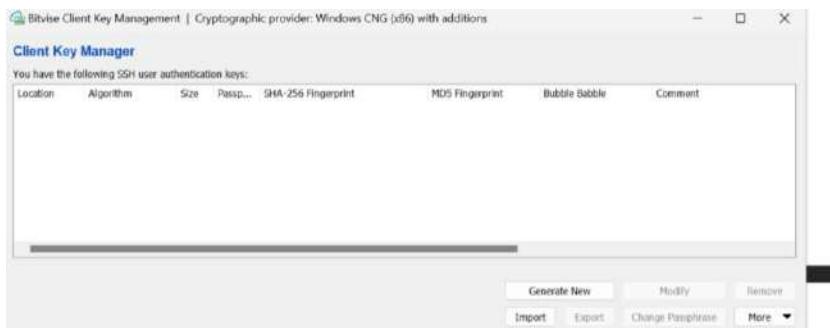
- 6.** Now click on the Instance ID of the server.

- 7.** Copy the Public IPv4 address.

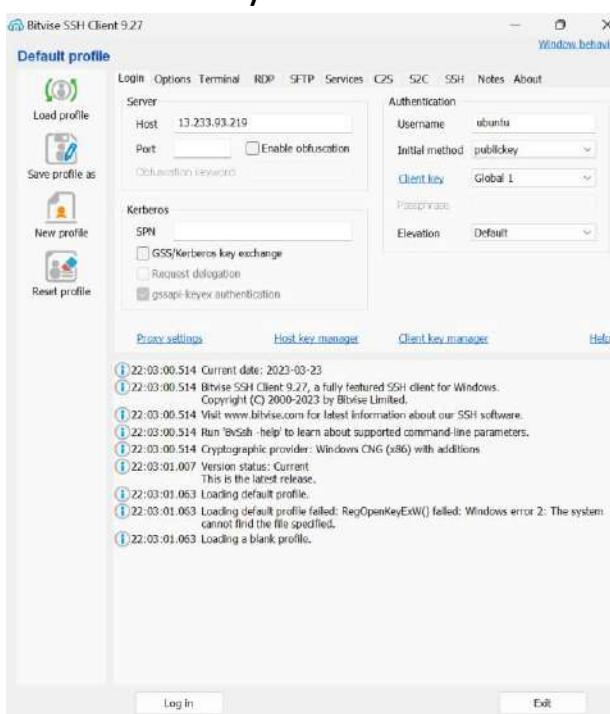
- 8.** Now for the next steps we require Bitvise SSH client. Download it and install in your local pc.

- 9.** Now open the Bitvise SSH Client.

- 10.** Paste the copied IPv4 address in the Host section.
- 11.** Set user name to ubuntu.
- 12.** Click on the client key manager link below the authentication section.  
It will open another pop-up window. There click on import button.  
Select the previously downloaded .pem file. Click on import. Then close the Client key manager window.



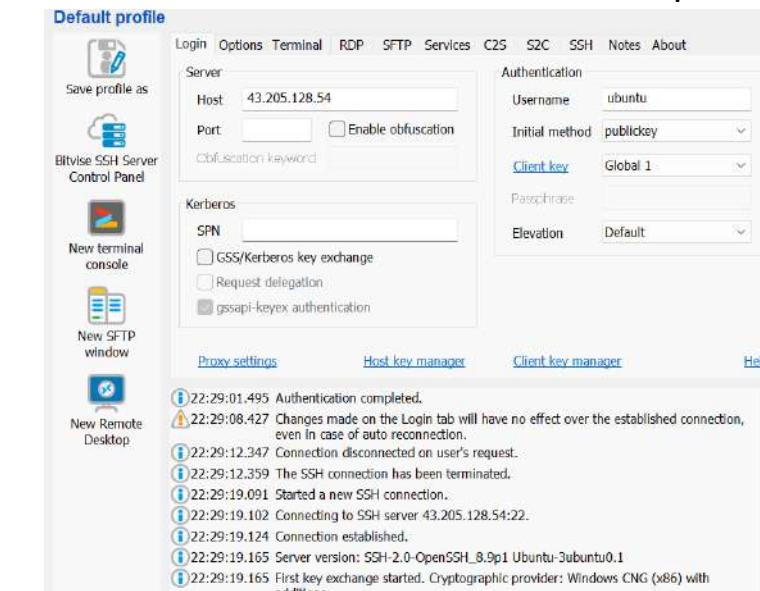
- 13.** Now set initial method to public key.
- 14.** Set Client Key to Global 1.



- 15.** Now click on the Log In button at the bottom of the Window.  
Click on Accept and Save button on the pop-up.  
One of many ways in which you can know that whether you have successfully logged in is if your Log In button has changed to Log Out.

**16.** Now newly created options will arise on the left sidebar on successful login.

Click on the new terminal console to open terminal of our server.



**17.** Enter the following commands:

a. **sudo apt-get update**

b. **sudo apt-get upgrade**

(Remember to press Y and then Enter when prompted)

(After the process is completed a new box/window appears. But just press Enter to continue.)

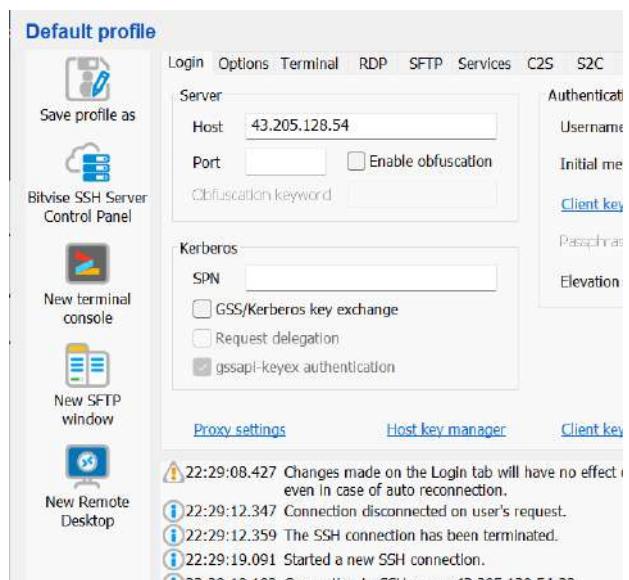
c. **sudo apt-get install nginx**

(Remember to press Y and then Enter when prompted)

(After the process is completed a new box/window appears. But just press Enter to continue.)

**18.** Now minimize the console.

**19.** Click on the new SFTP window icon on the left sidebar.



**20.** Select the folder where you have kept HTML files of your website on the local files section. Just keep it open.

Name	Type	Date Modified
about.html	175 Chrome HT...	23-03-2023 08
index.html	176 Chrome HT...	23-03-2023 08
next.html	175 Chrome HT...	23-03-2023 08

Name	Type	Date Modified
.cache	4,096 File folder	23-03-2023
ssh	4,096 File folder	23-03-2023
bash_logout	220 Bash Logout...	06-01-2022
bashrc	3,771 Bash RC So...	06-01-2022
profile	807 Profile Sourc...	06-01-2022
sudo_as_admin_successful	0 SUDO_AS_...	23-03-2023

**21.** Now click the Up button (2 times) on the Remote Files section. You will be able to see a bunch of folders. Scroll down and open the last folder named “var”.

Name	Type	Date Modified
about.html	175 Chrome HT...	23-03-2023 08
index.html	176 Chrome HT...	23-03-2023 08
next.html	175 Chrome HT...	23-03-2023 08

Name	Type	Date Modified
lib32	10 File folder	08-02-2C
lost-found	16,384 File folder	08-02-2C
media	4,096 File folder	08-02-2C
mnt	4,096 File folder	08-02-2C
opt	4,096 File folder	08-02-2C
proc	0 File folder	23-03-2C
root	4,096 File folder	23-03-2C
run	920 File folder	23-03-2C
sbin	8 File folder	08-02-2C
snap	4,096 File folder	08-02-2C
srv	4,096 File folder	08-02-2C
sys	0 File folder	23-03-2C
tmp	4,096 File folder	23-03-2C
usr	4,096 File folder	08-02-2C
var	4,096 File folder	23-03-2C

**22.** Now again open the last folder in it named “www”.

Name	Type	Date Modified
about.html	175 Chrome HT...	23-03-2023 08
index.html	176 Chrome HT...	23-03-2023 08
next.html	175 Chrome HT...	23-03-2023 08

Name	Type	Date Modified
backups	4,096 File folder	18-04-2022
cache	4,096 File folder	23-03-2023
crash	4,096 File folder	08-02-2023
lib	4,096 File folder	23-03-2023
local	4,096 File folder	18-04-2022
lock	9 File folder	08-02-2023
log	4,096 File folder	23-03-2023
mail	4,096 File folder	08-02-2023
opt	4,096 File folder	08-02-2023
run	4 File folder	08-02-2023
snap	4,096 File folder	08-02-2023
spool	4,096 File folder	08-02-2023
tmp	4,096 File folder	23-03-2023
www	4,096 File folder	23-03-2023

**23.** Open the only folder named “html” and keep it open. You will see a default html already present.

You can check whether nginx is working by pasting our previously copied IPv4 address of our server instance in a different browser. It will show something like this.



### Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support, please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

Thank you for using nginx.

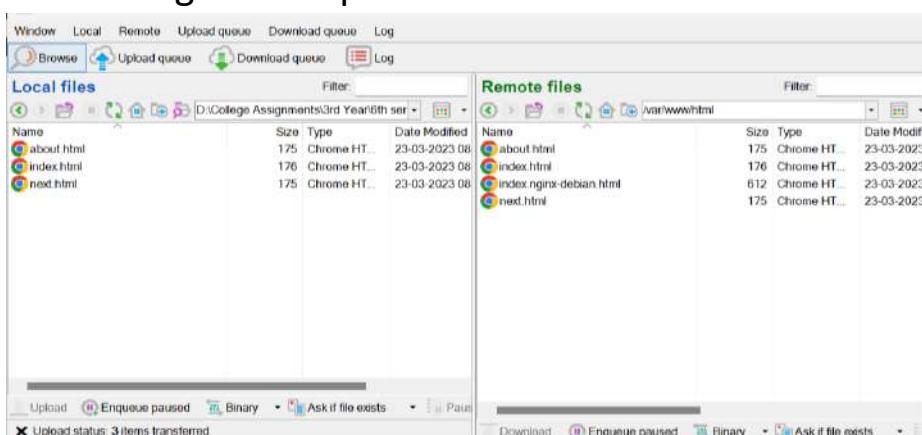
**24.** We actually need to transfer our local html file here in this open folder of the remote server. However, we do not have such permissions for this folder. To give such permission we need to go back to the terminal console and give the required permissions to the folder.

**25.** Now type the following commands in the terminal.

- a. **cd /**
- b. **cd var/www/**
- c. **sudo chmod 777 html**

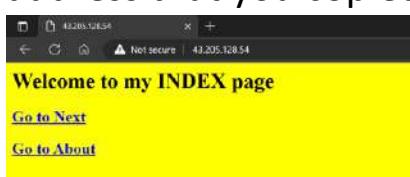
Now the permission (Read, Write, Execute) of the folder is successfully granted.

**26.** Now drag and drop all the files from local to remote.



Remember you must have the opening html named “index.html” in order to show the opening html page by the web server.

**27.** Finally open the website from any browser or device by using the public IPv4 address that you copied.



We now have successfully hosted a static website on an AWS EC2 sever.

# ASSIGNMENT-8

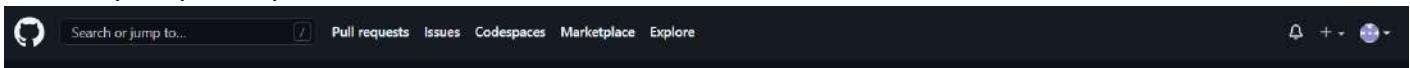
**Problem Statement:** Deploy a project from Local Machine to GitHub and vice-versa.

## Procedure:

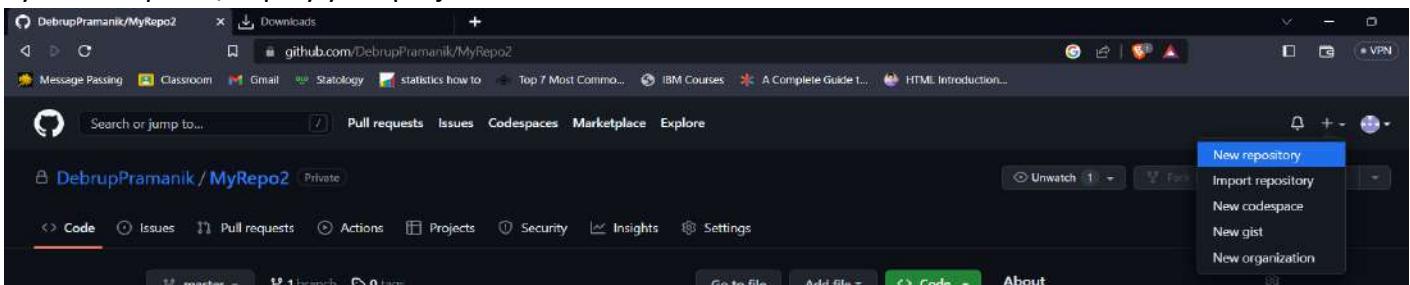
1. Make sure you have installed Git for Windows Application if you are using a Windows machine. If not then download the same from <https://gitforwindows.org/> website. Install and make sure you have integrated it with the Windows Shell. (It is checked on by default so no need to change anything.)
2. Now go to GitHub Website <https://github.com/> and Sign In to your account.  
If you do not have an account then click on Sign Up button to Create an account for yourself. Then just follow the on-screen instructions to complete your registration. Now finally you will Sign-In to your account.



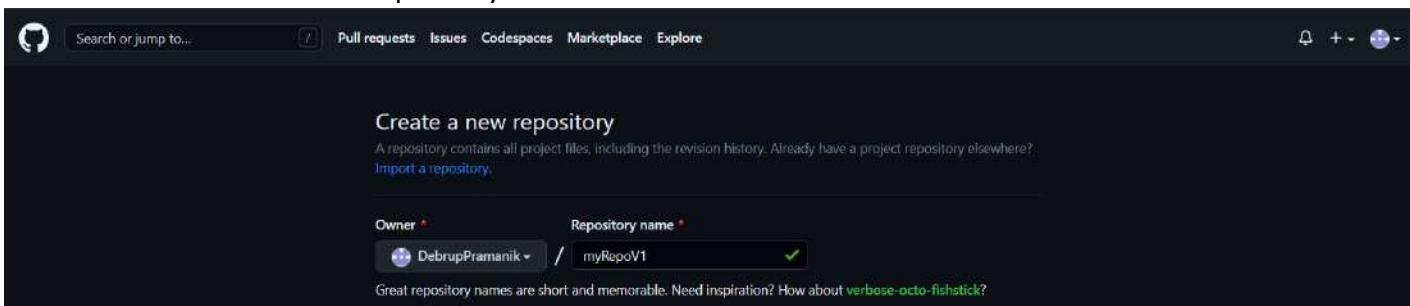
3. After successfully signing in GitHub, click on the + button situated on the top right corner of the website beside your profile picture.



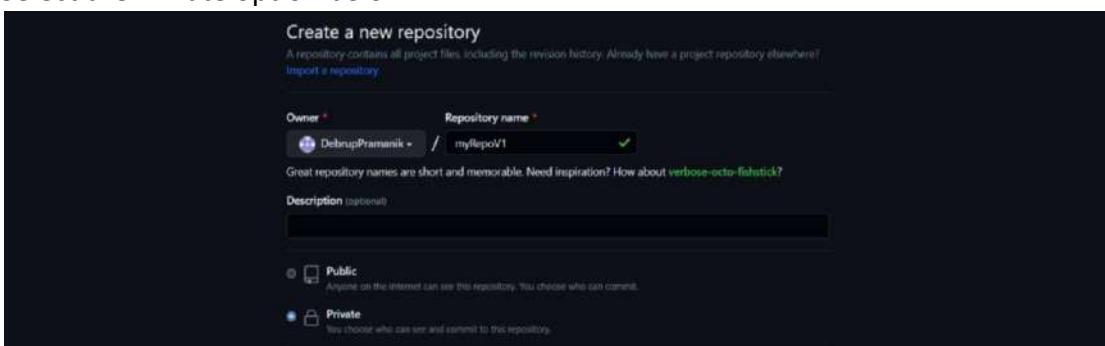
4. Now after clicking a menu will appear. Click on New Repository. This will create a New Repository where you can upload/deploy your project folders and files.



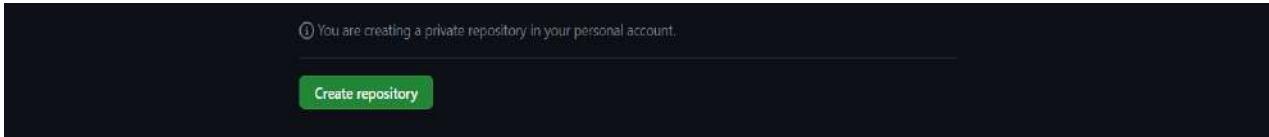
5. Now Enter the name of the Repository.



6. Select the Private option below.



## 7. Next scroll-down and click on Create Repository.

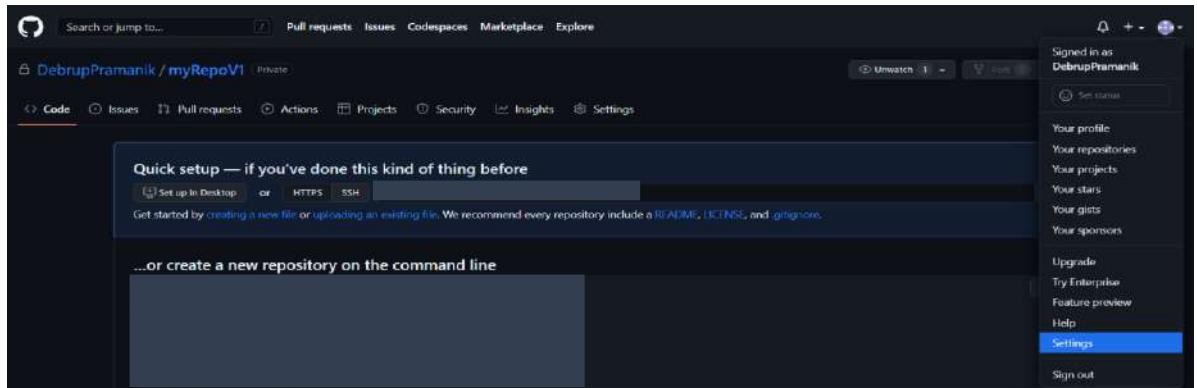


## 8. You will be redirected to the Repository code page.

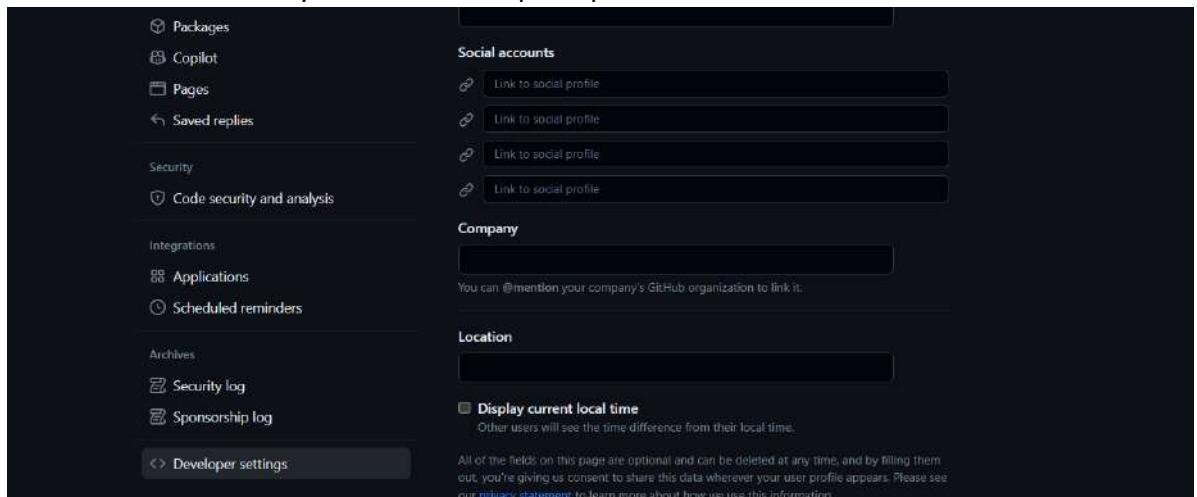
Now while adding your project to your GitHub repository, you need to provide credentials of your account and sign-in every time you are doing this, through the Git bash terminal. Or there is another way, and that is by using Tokens generated for your account.

## 9. So, for generating token for your account follow the steps:

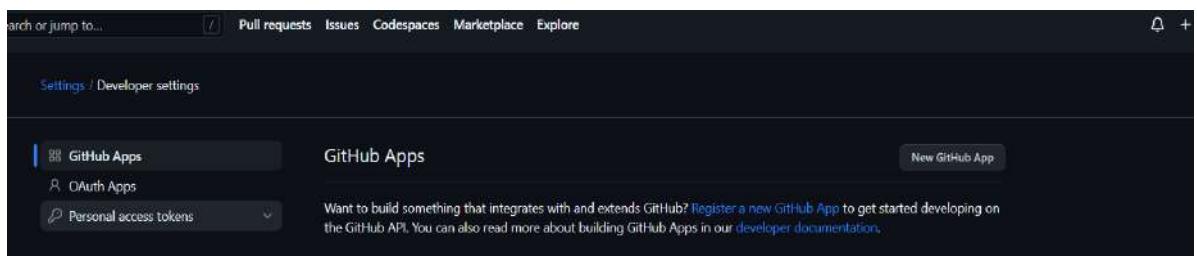
- Click on your profile on the top right corner of the web page.
- A drop-down menu will appear. Click on the Settings option.



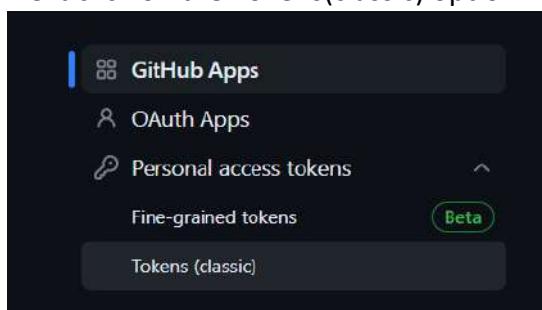
- Now scroll down until you see a Developer Options button on the left Nav bar. Click on it.



- Now click on the Down arrow beside Personal Access Tokens in the left Nav bar.



- Next click on the Tokens(classic) option.



f. Next click on Generate New Token button

The screenshot shows the GitHub Developer settings interface. On the left, there's a sidebar with options like GitHub Apps, OAuth Apps, Personal access tokens (with sub-options: Fine-grained tokens and Tokens (classic)), and Marketplace. The main area is titled "Personal access tokens (classic)" and contains a sub-header "Tokens you have generated that can be used to access the GitHub API." At the top right, there are two buttons: "Generate new token" and "Revoke all".

g. Again, click on Generate New Token(classic).

h. Give the Token Name.

Select the Expiration time (Preferably 90 Days).

Now check all the parent boxes below.

Then Click on Generate Token Button.

**New personal access token (classic)**

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API](#) over Basic Authentication.

**Note**

What's this token for?

**Expiration \***

90 days     The token will expire on Wed, Jul 5 2023

**Select scopes**

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> <b>repo</b>	Full control of private repositories
<input checked="" type="checkbox"/> <b>repostatus</b>	Access commit status
<input checked="" type="checkbox"/> <b>repo_deployment</b>	Access deployment status
<input checked="" type="checkbox"/> <b>public_repo</b>	Access public repositories
<input checked="" type="checkbox"/> <b>reposinvite</b>	Access repository invitations
<input checked="" type="checkbox"/> <b>security_events</b>	Read and write security events
<input checked="" type="checkbox"/> <b>useremail</b>	Access user email addresses (read-only)
<input checked="" type="checkbox"/> <b>userfollow</b>	Follow and unfollow users
<input checked="" type="checkbox"/> <b>delete_repo</b>	Delete repositories
<input checked="" type="checkbox"/> <b>written:discussion</b>	Read and write team discussions
<input checked="" type="checkbox"/> <b>read:discussion</b>	Read team discussions
<input checked="" type="checkbox"/> <b>admin:enterprise</b>	Full control of enterprises
<input checked="" type="checkbox"/> <b>manage_runner:enterprise</b>	Manage enterprise runners and runner groups
<input checked="" type="checkbox"/> <b>manage_billing:enterprise</b>	Read and write enterprise billing data
<input checked="" type="checkbox"/> <b>read:enterprise</b>	Read enterprise profile data
<input checked="" type="checkbox"/> <b>audit:log</b>	Full control of audit log
<input checked="" type="checkbox"/> <b>read:audit:log</b>	Read access of audit log
<input checked="" type="checkbox"/> <b>codespace</b>	Full control of codespaces
<input checked="" type="checkbox"/> <b>codespace:secrets</b>	Ability to create, read, update, and delete codespace secrets
<input checked="" type="checkbox"/> <b>project</b>	Full control of projects
<input checked="" type="checkbox"/> <b>read:project</b>	Read access of projects
<input checked="" type="checkbox"/> <b>admin:gpg_key</b>	Full control of public user GPG keys
<input checked="" type="checkbox"/> <b>write:gpg_key</b>	Write public user GPG keys
<input checked="" type="checkbox"/> <b>read:gpg_key</b>	Read public user GPG keys
<input checked="" type="checkbox"/> <b>admin:ssh_signing_key</b>	Full control of public user SSH signing keys
<input checked="" type="checkbox"/> <b>write:ssh_signing_key</b>	Write public user SSH signing keys
<input checked="" type="checkbox"/> <b>read:ssh_signing_key</b>	Read public user SSH signing keys

**Generate token**    **Cancel**

i. Now a Token will be generated. Copy it and save it in a text file. Keep it somewhere safe as we will need it later.

**10.** Now scroll-up and click on the icon on the top left corner of the web page. This will redirect you to your home page.

The screenshot shows the top navigation bar of the GitHub website. It includes a user icon, a search bar with placeholder text "Search or jump to...", and navigation links for Pull requests, Issues, Codespaces, Marketplace, and Explore. Below the search bar, it says "Settings / Developer settings".

11. Now on the left nav bar you will find your Top Repositories You are currently working/contributing to. You can also locate your newly created Repository here. Click on it.

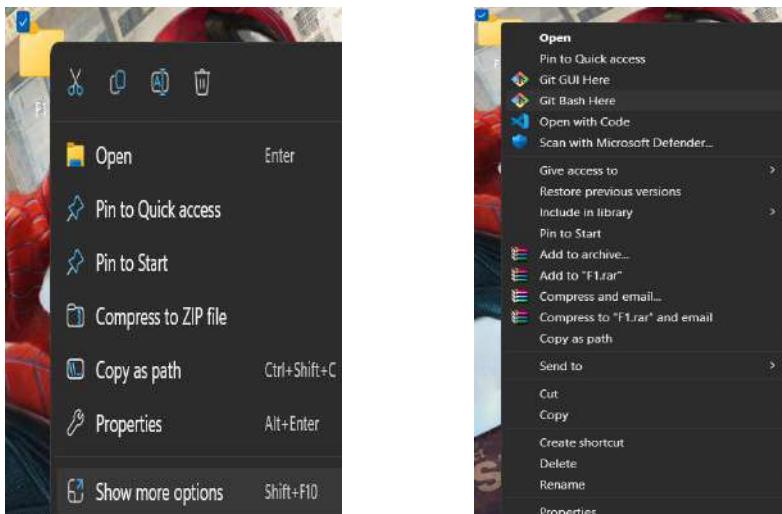
The screenshot shows the GitHub homepage with the navigation bar at the top. Below it, the 'Top Repositories' section is displayed, featuring a list of repositories. One repository, 'DebrupPramanik/myRepoV1', is highlighted as private. To the right of this list is a 'Discover interesting projects and people' feed, which includes a button to 'Explore GitHub'. Below the main content area, there's a 'Recent activity' sidebar and some tips about news feeds.

12. Go to the code section and keep it open. Now minimize your browser.

The screenshot shows the GitHub repository page for 'myRepoV1'. The top navigation bar is visible, and the repository name 'DebrupPramanik / myRepoV1' is shown. Below the repository name, there are several tabs: 'Code' (which is selected), 'Issues', 'Pull requests', 'Actions', 'Projects', 'Security', 'Insights', and 'Settings'. A 'Quick setup' message is displayed above the main content area.

13. Now create a folder anywhere in your computer. Give it a name. Now Right Click on it and select Git Bash here.

If you are in Win 11 then after Right click go to Show more Options and then select Git Bash here.

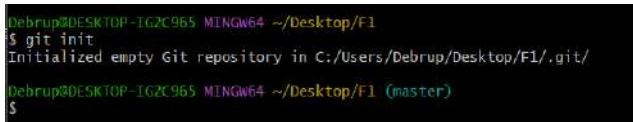


14. It will open the Git Bash Terminal



15. Now type the following to clone the project provided/required (Link will be provided or specified or can be obtained from visiting the required project's GitHub page and copying it):

→ **git init**



→ **git clone https://github.com/sudip7407/New-Repo1.git**

```
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1
$ git init
Initialized empty Git repository in C:/Users/Debrup/Desktop/F1/.git/
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1 (master)
$ git clone https://github.com/sudip7407/New-Repo1.git
Cloning into 'New-Repo1'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 15 (delta 6), reused 4 (delta 0), pack-reused 0
receiving objects: 100% (15/15), done.
resolving deltas: 100% (6/6), done.

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1 (master)
$ |
```

→ **ls -A**

```
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1
$ git init
Initialized empty Git repository in C:/Users/Debrup/Desktop/F1/.git/
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1 (master)
$ git clone https://github.com/sudip7407/New-Repo1.git
Cloning into 'New-Repo1'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 15 (delta 6), reused 4 (delta 0), pack-reused 0
receiving objects: 100% (15/15), done.
resolving deltas: 100% (6/6), done.

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1 (master)
$ ls -A
.git/ New-Repo1/
```

(We see all the files downloaded from the repository we just cloned. Now our job is to remove/delete the .git/ files from all the folders, including nested ones. Also, we have to remove. gitignore files from wherever it is present.)

→ **rm -r .git/**

→ **ls -A**

```
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1 (master)
$ rm -r .git/

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1
$ ls -A
New-Repo1/

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1
$
```

(We have successfully removed the .git/ file from the main directory. However we have to check the New-Repo1/ directory to see if it has no unwanted files.)

→ **cd New-Repo1/**

→ **ls -A**

```
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1
$ cd New-Repo1/
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ ls -A
.git/ .gitignore 'New Text Document.txt' index.js package.json

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
```

(We now see we have to remove the .git/ and .gitignore files. Then we need to go back to the main directory)

(Sometimes permissions are asked when removing some files. Just type y when prompted and press enter and repeat it when asked every time.)

→ **rm -r .git/**

```
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ rm -r .git/
rm: remove write-protected regular file '.git/objects/pack/pack-32a7e09ea210097af1de5d09c501a57c3528ef13.idx'? y
rm: remove write-protected regular file '.git/objects/pack/pack-32a7e09ea210097af1de5d09c501a57c3528ef13.pack'? y
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1
```

→ **rm -r .gitignore**

→ **ls -A**

```
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1
$ rm -r .gitignore
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1
$ ls -A
'New Text Document.txt' index.js package.json
```

(Our work here is done. Now we need to return to our main directory)

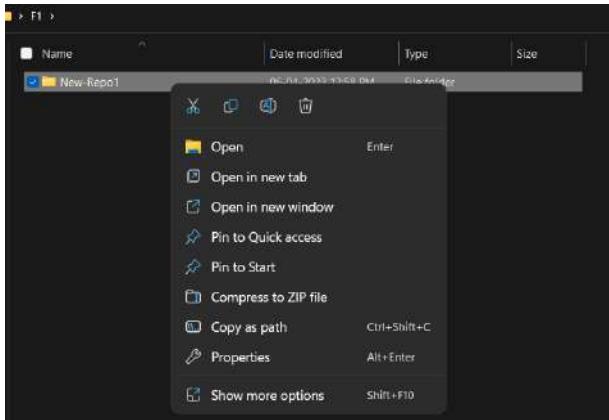
→ cd ..

```
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1
$ cd ..
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1
$ ls -A
New-Repo1/
```

We now have successfully cloned our required project.

16. Now close the terminal and open the folder in which you just cloned the project.

17. Single Click on the New-Repo1 file and right click and just like explained above again select Git bash here option.



18. Again, a Git bash terminal will open.

19. Again, type the following commands but now we will now upload this cloned project to our created repository on our GitHub:

→ git init

```
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1
$ git init
Initialized empty Git repository in C:/Users/Debrup/Desktop/F1/New-Repo1/.git/
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$
```

→ ls -A

→ git status

```
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1
$ git init
Initialized empty Git repository in C:/Users/Debrup/Desktop/F1/New-Repo1/.git/
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ ls -A
.git/ 'New Text Document.txt' index.js package.json

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    New Text Document.txt
    index.js
    package.json

nothing added to commit but untracked files present (use "git add" to track)
```

→ git config --global user.email "Your email here"

→ git config --global user.name "Your GitHub account username here"

```
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ git config --global user.email "debrup202002@gmail.com"

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ git config --global user.name "DebrupPramanik"
```

→ git config user.name

→ git config user.email

```
Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ git config user.name
DebrupPramanik

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ git config user.email
debrup202002@gmail.com

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
```

(this is required to set as it will commit the files in your repository using the set username and email.)

→ git add .

→ git status

```
Debrup@DESKTOP-TG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ git add .
warning: in the working copy of 'cls', LF will be replaced by CRLF the next time
Git touches it

Debrup@DESKTOP-TG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:  New Text Document.txt
    new file:  cls
    new file:  index.js
    new file:  package.json

Debrup@DESKTOP-TG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$
```

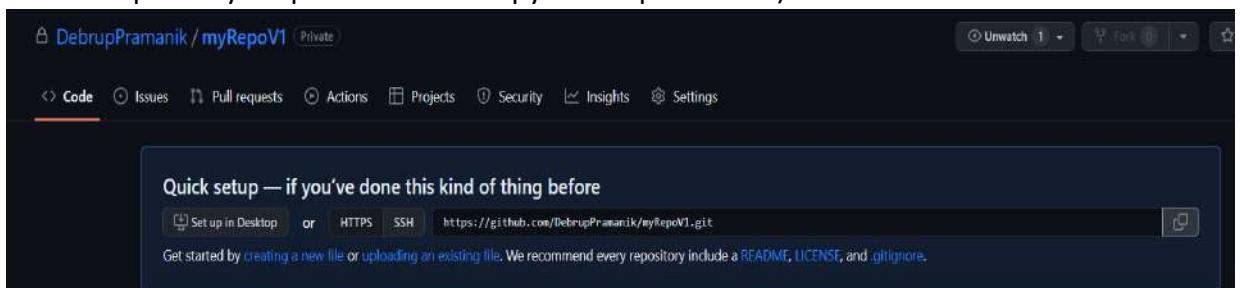
(Notice all the files that was colored red previously is now green, indicating that they have been added to your local repository. We now have to commit the changes we made and then push it to our GitHub repository)

→ git commit -m “type your own message here”

```
Debrup@DESKTOP-TG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ git commit -m "Hello World. My first ever commit"
[master (root-commit) 36046dc] Hello World. My first ever commit
 4 files changed, 49 insertions(+)
 create mode 100644 New Text Document.txt
 create mode 100644 cls
 create mode 100644 index.js
 create mode 100644 package.json

Debrup@DESKTOP-TG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$
```

→ git remote add origin [\(The https address is the address of your repository. To obtain it maximize your browser where your GitHub repository is open and there copy the https address\)](https://github.com/yourusername>Yourrepositoryname.git</a></p></div><div data-bbox=)



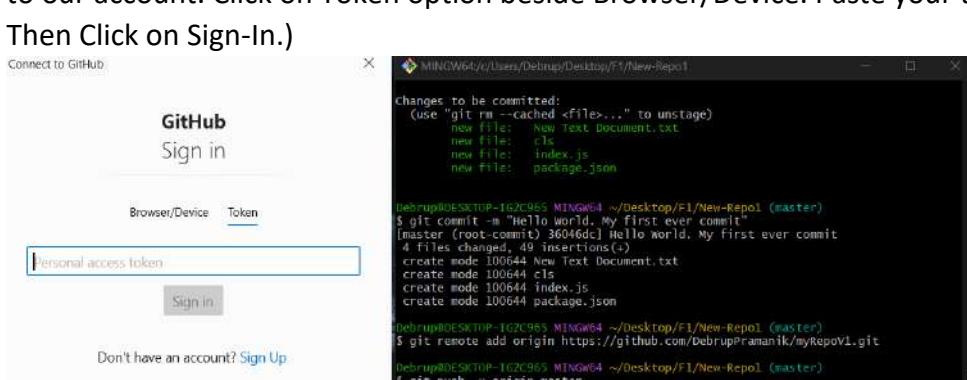
```
Debrup@DESKTOP-TG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ git commit -m "Hello World. My first ever commit"
[master (root-commit) 36046dc] Hello World. My first ever commit
 4 files changed, 49 insertions(+)
 create mode 100644 New Text Document.txt
 create mode 100644 cls
 create mode 100644 index.js
 create mode 100644 package.json

Debrup@DESKTOP-TG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ git remote add origin https://github.com/DebrupPramanik/myRepoV1.git

Debrup@DESKTOP-TG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$
```

→ git push -u origin master

(This is the final command. A pop-up window will open named Connect to GitHub. You will have several options to Sign-In. However, we will be using our Generated token for your account to Sign-In to our account. Click on Token option beside Browser/Device. Paste your token in the placeholder. Then Click on Sign-In.)



(After Successful sign-in the following will show up in the terminal. The Pop-up Window will close automatically)

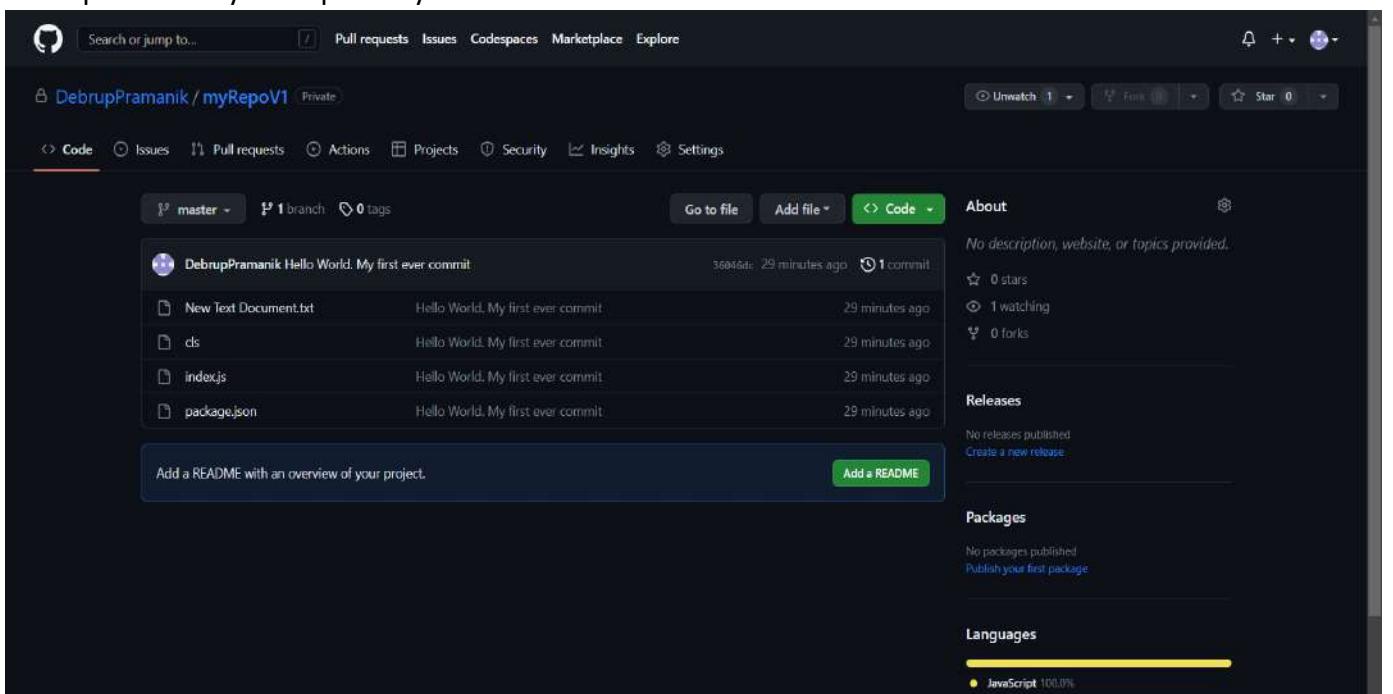
```
[master (root-commit) 36046dc] Hello World. My first ever commit
 4 files changed, 49 insertions(+)
 create mode 100644 New Text Document.txt
 create mode 100644 cls
 create mode 100644 index.js
 create mode 100644 package.json

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ git remote add origin https://github.com/DebrupPramanik/myRepoV1.git

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ git push -u origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 6 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 1.05 KiB | 1.05 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/DebrupPramanik/myRepoV1.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

Debrup@DESKTOP-IG2C965 MINGW64 ~/Desktop/F1/New-Repo1 (master)
$ |
```

20. Now go to your browser where your GitHub repository is open. Refresh the page. Now you will see the files uploaded in your repository!



We have successfully Cloned and Uploaded our project to GitHub using Git and Git Bash terminal.

21. You can also add collaborators who can access your private repository and can contribute to it.
- Go to the Setting section of your repository (It is in the same nav bar as the code section)
  - Then select the collaborators option in the left nav bar.  
It will ask you to enter your password.  
Enter it.
  - Now you can click on add people button to add others as collaborators of your repository. You have to search by their Username or Email.

# ASSIGNMENT-9

**Problem Statement:** Deploy a project from GitHub to EC2.

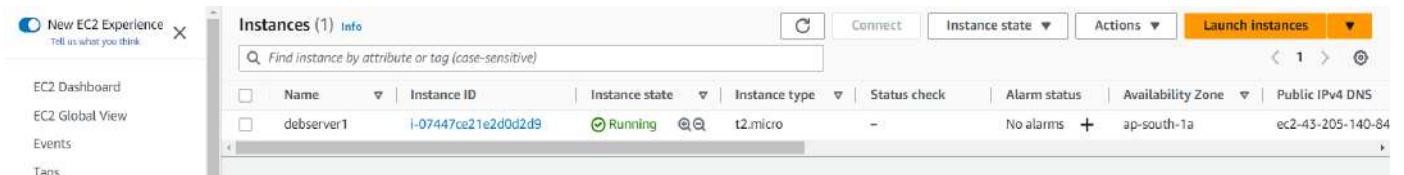
**Procedure:**

1. Go to GitHub Website <https://github.com/> and Sign In to your account.

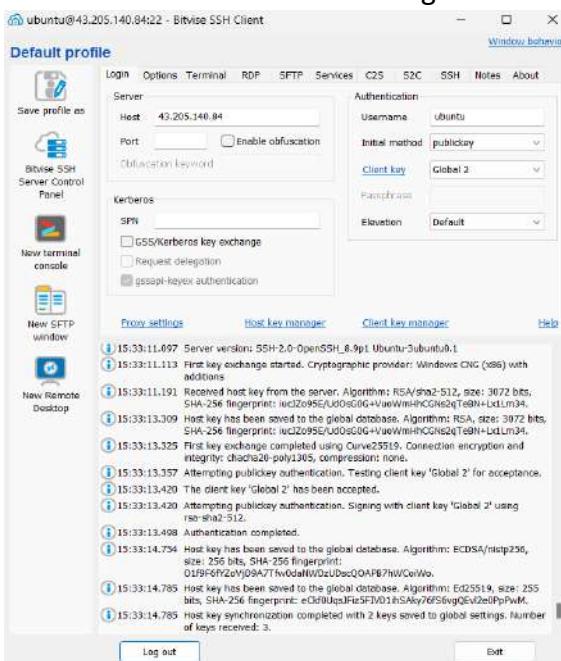


2. Also, Sign-In to your AWS account.

3. Create an EC2 instance (Refer to Ass7)



4. Connect the to the instance using the Bitvise SSH Client (Refer to Ass7)



5. Now Click on New Terminal Console option in the Left Sidebar of the Bitvise Client.

6. A terminal window will open and in it type the following commands:-

→ **sudo apt-get update & sudo apt-get upgrade**

(After few steps of progression, in case of any prompts asking (Y/N) press 'y' button and then press 'Enter' to continue. At the last stages if a UI appears on the screen, just press 'Enter' to continue. After the whole process is complete enter the next command as mentioned below)

→ **sudo apt-get install nginx**

(After few steps of progression, in case of any prompts asking (Y/N) press 'y' button and then press 'Enter' to continue. At the last stages if a UI appears on the screen, just press 'Enter' to continue. After the whole process is complete enter the next command as mentioned below)

→ **nginx -v**

```
ubuntu@ip-172-31-38-127:~$ nginx -v
nginx version: nginx/1.18.0 (Ubuntu)
```

(This command displays the nginx version installed in the server system)

→ **curl -Sf https://deb.nodesource.com/setup\_18.x | sudo -E bash -**

(This command downloads NodeJS files with all dependencies in our server system)

→ **sudo apt install nodejs**

(Press ‘Enter’ to continue when any UI appears on screen)

(This command installs NodeJS in our server system)

→ **node -v**

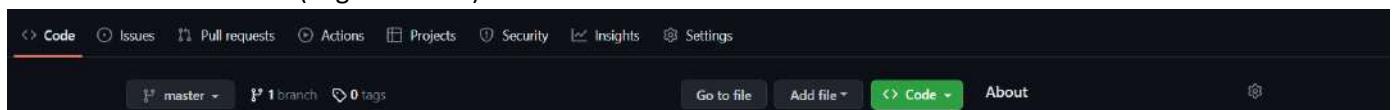
```
ubuntu@ip-172-31-38-127:~$ node -v  
v18.15.0
```

(This command displays the version of NodeJS installed in our server system)

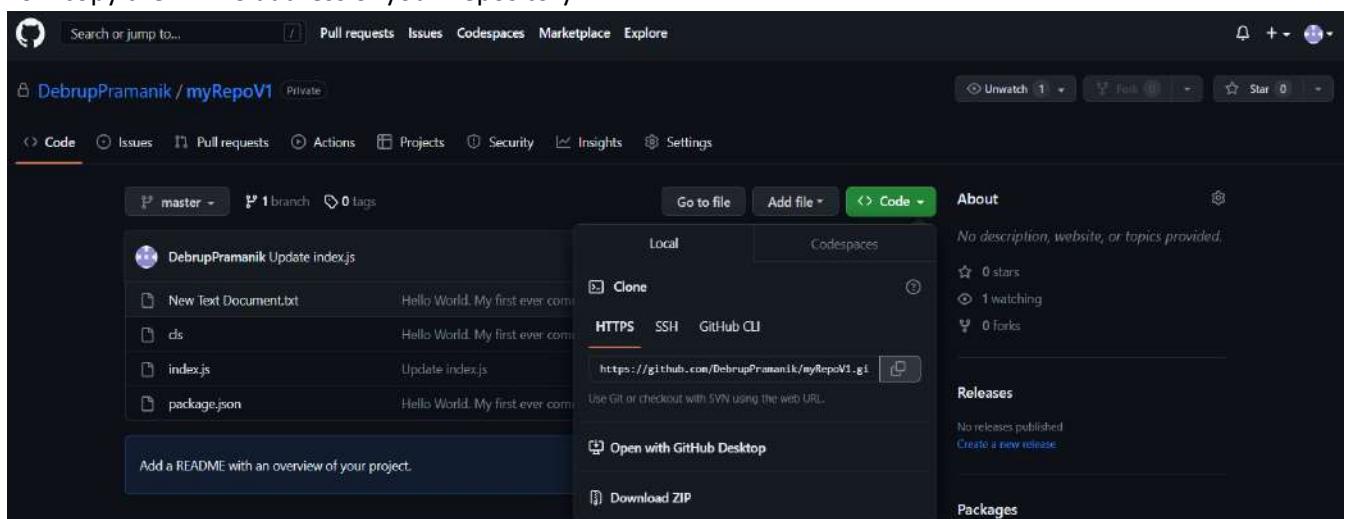
Now, minimize the terminal window. Go to the browser where our GitHub is Logged-In.

7. Go to your GitHub Repository which you want to upload in your EC2 server.

8. Click on the code button (in green color).



9. Now copy the HTTPS address of your Repository



10. Now return to the minimized terminal window and enter the following commands:-

→ **git clone https-address-you-just-copied-in-step-10**

```
ubuntu@ip-172-31-38-127:~$ git clone https://github.com/DebrupPramanik/myRepoV1.git  
Cloning into 'myRepoV1'...  
Username for 'https://github.com':
```

(Remember to paste your own https address that you copied in the above command in place of the one given in the screenshot)

(As shown in the screenshot, you will be asked to enter your username for GitHub. So mention your username there.)

After that you will be requested to provide your password. However, you have to enter your Account Token you generated instead of your password. If you don't have a Account Token then refer to Ass7 and create one for your GitHub account. Now copy-paste the Account Token (from the text file you haved saved it) where it wants to mention your password. For pasting just Right click for a single time on the terminal where you want to paste (Note you won't be able to see your pasted token on the terminal as it is hidden by default. So just press ‘Enter’ to continue)

```
ubuntu@ip-172-31-38-127:~$ git clone https://github.com/DebrupPramanik/myRepoV1.git  
Cloning into 'myRepoV1'...  
Username for 'https://github.com': DebrupPramanik  
Password for 'https://DebrupPramanik@github.com':  
remote: Enumerating objects: 9, done.  
remote: Counting objects: 100% (9/9), done.  
remote: Compressing objects: 100% (8/8), done.  
remote: Total 9 (delta 2), reused 4 (delta 0), pack-reused 0  
Receiving objects: 100% (9/9), done.  
Resolving deltas: 100% (2/2), done.
```

→ **dir**

```
ubuntu@ip-172-31-38-127:~$ dir  
myRepoV1
```

(As seen this is the name of our cloned repository. This means a new directory has been created in our present working directory which has been named automatically to match the name of our Repository.)

→ cd myRepoV1/

```
ubuntu@ip-172-31-38-127:~$ dir  
myRepoV1  
ubuntu@ip-172-31-38-127:~$ cd myRepoV1/  
ubuntu@ip-172-31-38-127:~/myRepoV1$ █
```

(Now we enter into the directory)

→ ls -A

```
ubuntu@ip-172-31-38-127:~/myRepoV1$ ls -A  
.git 'New Text Document.txt' cls index.js package.json  
ubuntu@ip-172-31-38-127:~/myRepoV1$ █
```

(This command displays all the files in the current directory)

(We observe that we have all the files that we have in our Repository has been cloned in our directory in the server system)

→ npm install

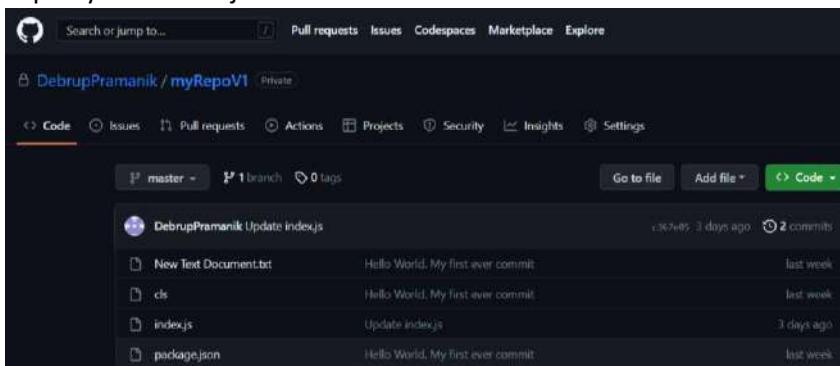
```
ubuntu@ip-172-31-38-127:~/myRepoV1$ npm install  
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.  
  
added 258 packages, and audited 259 packages in 10s  
  
18 packages are looking for funding  
  run `npm fund` for details  
  
found 0 vulnerabilities  
npm notice  
npm notice New minor version of npm available! 9.5.0 -> 9.6.4  
npm notice Changelog: https://github.com/npm/cli/releases/tag/v9.6.4  
npm notice Run npm install -g npm@9.6.4 to update!  
npm notice  
ubuntu@ip-172-31-38-127:~/myRepoV1$ █
```

(This command installs the npm package manager)

Now before proceeding further we need to return back to GitHub. Minimize the terminal for now.

11. Go back to your Repository in Github.

12. Open your “index.js” file



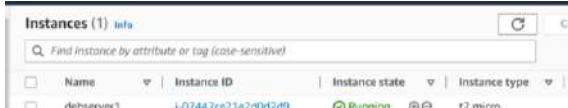
13. Check the port no. specified in the program. It is specified in the app.listen() method as the first parameter. Here it is ‘4000’. Copy or remember this no. as it is the port no. and will be required to connect to our website.

```
11 lines (9 sloc) | 205 Bytes  
  
1 const express = require('express')  
2 const app = express()  
3  
4 app.get('/', function (req, res) {  
5   res.send('Hello. My Name is Spider-Man!!!')  
6 })  
7  
8 app.listen(4000, ()=>{  
9   console.log("Started server");  
10 }  
11 )
```

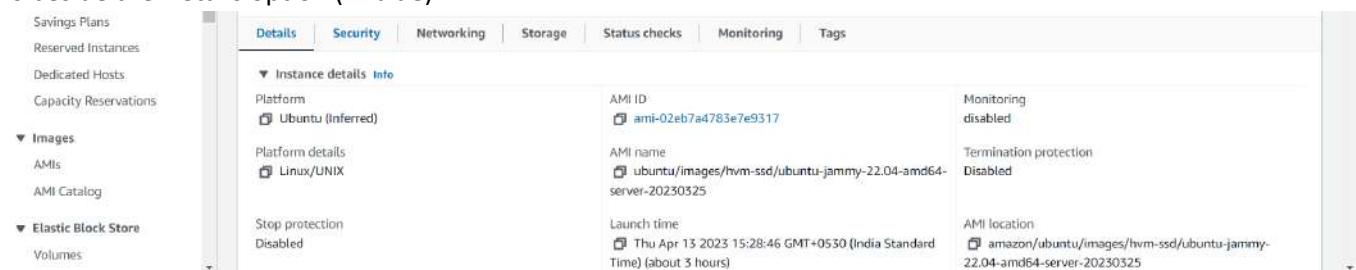
We have to add this port no. to our EC2 instance security group rule otherwise we won't be able to access the website from anywhere.

14. Now go back to your AWS EC2 instances page.

15. Click on the instance that is being used.



16. Scroll down until you find a section bar where by default the details option is selected. Select the Security option. It is beside the Details option (in blue).



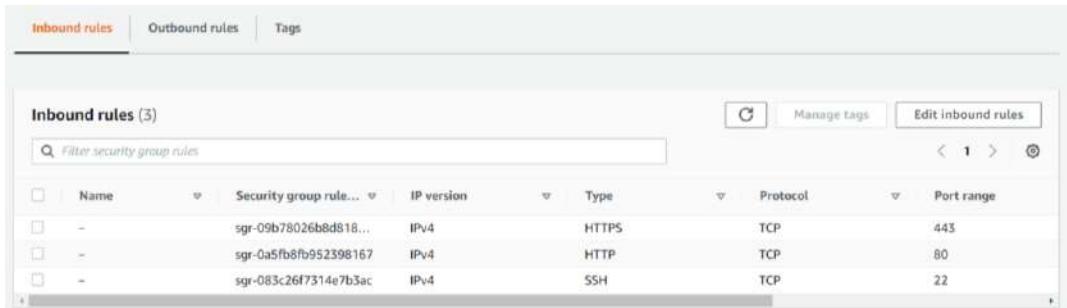
Name	Value	Description
Platform	Ubuntu (Inferred)	AMI ID: ami-02eb7a4783e7e9317
Platform details	Linux/UNIX	AMI name: ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20230525
Stop protection	Disabled	Launch time: Thu Apr 13 2023 15:28:46 GMT+0530 (India Standard Time) (about 3 hours)
		AMI location: amazon/ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20230525

17. Then click on the security groups link under security groups



Name	Security group rule ID	Port range	Protocol	Source	Security groups
-	sgr-09b78026b8d8185ec	443	TCP	0.0.0.0/0	launch-wizard-5
-	sgr-0a5fb8fb952398167	80	TCP	0.0.0.0/0	launch-wizard-5
-	sgr-083c26f7314e7b3ac	22	TCP	0.0.0.0/0	launch-wizard-5

18. Then click on Edit Inbound Rules button.

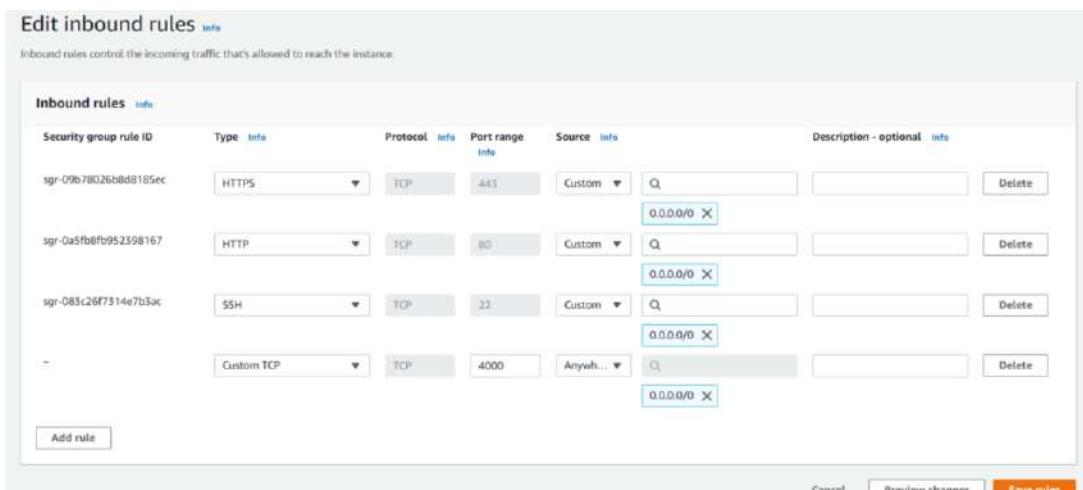


19. Click on the Add Rule button.



20. A new Row will be generated. Let the type remain Custom TCP. Under Port Range write your Port no. you want to open. In this case we have 4000 port no. as we found out earlier in our index.js code. Next in source click on the search box and the first option with value 0.0.0.0/0 should be selected.

21. Now click on the save rules button.



Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-09b78026b8d8185ec	HTTPS	TCP	443	Custom	0.0.0.0/0
sgr-0a5fb8fb952398167	HTTP	TCP	80	Custom	0.0.0.0/0
sgr-083c26f7314e7b3ac	SSH	TCP	22	Custom	0.0.0.0/0
-	Custom TCP	TCP	4000	Anywh...	0.0.0.0/0

Cancel   Preview changes   Save rules

We have successfully added the Port No. to our Inbound rules. Now we can access our website. But first we need to start our server.

22. We return back to the terminal and type:-

→ node index.js

```
ubuntu@ip-172-31-38-127:~/myRepoV1$ node index.js
Started server
```

Our server has started and it is also reflected by the terminal prompt. Now to check we need to open another browser and type in the IPv4 address of our EC2 server to access our website.

23. Now copy the IPv4 address of your EC2 server and paste it in another browser. But before pressing Enter add a colon (:) and then mention the Port No. mentioned in the index.js file. For our case it is 4000. In our case the full address resulted in the one in the below screenshot.

```
43.205.140.84:4000
```

24. Now press Enter to load the website.



25. We have successfully deployed our project from GitHub to our EC2 server.

Now if we want to change something in our project or file or code we will follow these general steps:

26. Suppose we want to modify the displayed message. So open the index.js file in GitHub.  
27. After opening we click on the pen icon on the right corner side of the code viewer.

```
11 lines (9 sloc) 205 Bytes
1 const express = require('express')
2 const app = express()
3
4 app.get('/', function (req, res) {
5   res.send('Hello. My Name is Spider-Man!!!')
6 })
7
8 app.listen(4000, ()=>{
9   console.log("Started server")
10 })
11 )
```

28. We then modify the string passed through the res.send() method.

```
11 lines (9 sloc) 225 Bytes
1 const express = require('express')
2 const app = express()
3
4 app.get('/', function (req, res) {
5   res.send('Hello. My Name is Spider-Man!!! Nice to meet You!!!')
6 })
7
8 app.listen(4000, ()=>{
9   console.log("Started server")
10 })
11 )
```

29. Now after editing, scroll-down and click the Commit changes button.

30. Our changes have finally been committed in our Repository in GitHub.

However our changes have not been reflected in our Remote Server. For that we have to ‘pull’ the new updated files into the repository directory in our Remote Server.

31. We have to Close our already running Server. For this, go to the Terminal which we have been working with. Then press <CTRL + C> shortcut to close the server.

```
Started server
```

```
^C
```

```
ubuntu@ip-172-31-38-127:~/myRepoV1$
```

Our server has been stopped.

32. Now type:-

→ git pull

(Enter the username when asked)

(Enter your account Token as your Password when asked for password)

(Right click once to paste, then press Enter)

```
ubuntu@ip-172-31-38-127:~/myRepoV1$ git pull
Username for 'https://github.com': DebrupPramanik
Password for 'https://DebrupPramanik@github.com':
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 685 bytes | 228.00 KiB/s, done.
From https://github.com/DebrupPramanik/myRepoV1
  c367e05..188da63  master      -> origin/master
Updating c367e05..188da63
Fast-forward
 index.js | 2 ++
 1 file changed, 1 insertion(+), 1 deletion(-)
ubuntu@ip-172-31-38-127:~/myRepoV1$
```

Now we have to restart the server.

→ **node index.js**

(We restarted the server)

33. We now have to Refresh our browser where we have our website open.



The changes have been successfully reflected. This is how we have to edit and update our project if required.

We have successfully completed our task of Deploying our project from GitHub to our EC2 server.

# ASSIGNMENT – 10

**Problem Statement:** Deploy project from GitHub to EC2 by creating new security group and user data.

## Procedure:

1. Sign in to your AWS account.
2. Go to your EC2 dashboard
3. Scroll down and Click on Security Groups option on the left side nav bar under Network & Security option.

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with options like Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, and Key Pairs. The main area is titled 'Resources' and displays various Amazon EC2 resources in the Asia Pacific (Mumbai) Region. It includes sections for Instances (running), Auto Scaling Groups, Dedicated Hosts, Elastic IPs, Instances, Key pairs, Load balancers, Placement groups, Security groups, Snapshots, and Volumes. A callout box at the bottom right of the main area provides information about easily sizing, configuring, and deploying Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server.

4. Select all the Security Groups other than the one named “default”.

The screenshot shows the AWS Security Groups list. There are two entries: 'default' (Security group ID: sg-0e4361e5c76c8c036, VPC ID: vpc-0a33deec3fd6dc096, Owner: 728364961341) and 'mysecgr1' (Security group ID: sg-0abe126418b4ea2a8, VPC ID: vpc-0a33deec3fd6dc096, Owner: 728364961341). The 'mysecgr1' row is selected, indicated by a blue border around its columns.

5. Then Click on the Actions button.

The screenshot shows the AWS Security Groups list with the Actions button clicked, opening a dropdown menu. The menu includes options: Manage tags, Manage stale rules, Copy to new security group, and Delete security groups. The 'Delete security groups' option is highlighted with a blue border.

6. Scroll-Down the dropdown list until you find the “delete all security groups” option. Click on it.

This is a close-up screenshot of the Actions dropdown menu from the previous step. The 'Delete security groups' option is clearly visible and highlighted with a blue border.

7. Now only the “default” security group remains and we keep it that way.

The screenshot shows the AWS Security Groups list again, but now only the 'default' security group (sg-0e4361e5c76c8c036) is listed. The 'mysecgr1' entry is no longer present, indicating it has been deleted.

8. Now click on the “Create Security Group” button.

The screenshot shows the AWS Security Groups list with the 'Create security group' button highlighted in orange at the top right of the page.

9. Now start by giving a name to the security group and giving its description (anything).  
Let the VPC remain unchanged.

The screenshot shows the 'Create security group' page. In the 'Basic details' section, the 'Security group name' field contains 'mysec1'. Below it, a note says 'Name cannot be edited after creation.' The 'Description' field contains 'mysec1'. Under 'VPC', a dropdown menu shows 'vpc-0a33deec5fd6dc096'.

10. Next, we will add Inbound Rules. Start adding by clicking the Add rule button. These include:

a) SSH

The screenshot shows the 'Inbound rules' section for an existing security group. It lists one rule: Type: SSH, Protocol: TCP, Port range: 22, Source: Anywhere, Description: optional (empty), and IP range: 0.0.0.0/0.

b) HTTP

The screenshot shows the 'Inbound rules' section for an existing security group. It lists one rule: Type: HTTP, Protocol: TCP, Port range: 80, Source: Anywhere, Description: optional (empty), and IP range: 0.0.0.0/0.

c) HTTPS

The screenshot shows the 'Inbound rules' section for an existing security group. It lists one rule: Type: HTTPS, Protocol: TCP, Port range: 443, Source: Anywhere, Description: optional (empty), and IP range: 0.0.0.0/0.

d) Custom TCP

The screenshot shows the 'Inbound rules' section for an existing security group. It lists one rule: Type: Custom TCP, Protocol: TCP, Port range: 4000, Source: Anywhere, Description: optional (empty), and IP range: 0.0.0.0/0.

The last one with custom TCP has a specific port range that we require to connect to our project. It has been specified in our index.js file (refer Ass9).

Now the final Inbound Rules section should look like this.

The screenshot shows the 'Inbound rules' section for an existing security group. It lists four rules: 1. SSH (Type: SSH, Protocol: TCP, Port range: 22, Source: Anywhere, IP range: 0.0.0.0/0). 2. HTTP (Type: HTTP, Protocol: TCP, Port range: 80, Source: Anywhere, IP range: 0.0.0.0/0). 3. HTTPS (Type: HTTPS, Protocol: TCP, Port range: 443, Source: Anywhere, IP range: 0.0.0.0/0). 4. Custom TCP (Type: Custom TCP, Protocol: TCP, Port range: 4000, Source: Anywhere, IP range: 0.0.0.0/0). An 'Add rule' button is visible at the bottom left.

11. Next outbound rules and all other sections remain unchanged. Now Click on the create security group button.

The screenshot shows the 'Outbound rules' section. It lists one rule: All traffic (Protocol: All, Port range: All, Destination: Custom, IP range: 0.0.0.0/0). Below this, there's a 'Tags - optional' section with a note about tags and a 'Create security group' button at the bottom right.

12. Now go back to the security groups list and click on the security group ID of the newly created Security Group.

Name	Security group ID	Description	VPC ID
mysec1	sg-0493398d43b761e55	mysec1	vpc-0a33dec3fd6dc096
default	sg-0e4361e5c76c8c036		

Inbound rules (4)

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-0677b32c36bf02194	IPv4	HTTPS	TCP	443
-	sgr-02f914343500960...	IPv4	SSH	TCP	22
-	sgr-0ff95fa0debaeccc64	IPv4	HTTP	TCP	80
-	sgr-0d92a3e25bf3ad37	IPv4	Custom TCP	TCP	4000

After clicking we can view the inbound rules that we added during its creation.

13. Now we go to the instances section from the left side nav bar.

14. Now we Create a new EC2 instance. Click on the Launch Instance button.

Instances Info

Find instance by attribute or tag (case-sensitive)

Actions ▾ Launch instances ▾

No instances  
You do not have any instances in this region

Now,

a) Give the name

Name and tags

Name: definst1

Add additional tags

b) Select Ubuntu as OS.

Application and OS Images (Amazon Machine Image)

Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Linux	rmeOS	Ubuntu	Windows	Red Hat
AWS	Mac	Microsoft	Red Hat	

Amazon Machine Image (AMI)

c) Select a keypair or generate a new one if none is available.

Key pair (login)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name (required): diskkey2

Create new key pair

d) Then under Network settings select the Select Existing Security Group option.

Network

vpc-0a33dec3fd6dc096

Subnet

No preference (Default: subnet in any availability zone)

Auto-assign public IP

Enable

Firewall (security groups)

Select existing security group

Security groups

Select security groups

- e) Now under the security groups dropdown menu select the one we just created.

It should look like this.....

- f) Now scroll down and click on the Advanced Details option.

- g) Now again scroll-down to the newly appeared sub-sections until you find User Data section.

- h) Write the following commands in the given box. Remember this user data is given to execute the given commands once the server starts. So essentially, we can provide all commands that we entered in our Assignment 9 previously and execute them without connecting to our server itself!! They will be executed sequentially.

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
```

Now, here is a caveat. We have created a private repository in GitHub. So, whenever we run the git clone command it asks for our username and password. Hence this cannot be executed directly through our User Data instructions. We have to connect manually and enter all commands starting from the git clone command.

- i) Now we click on the launch instance button.



## 15. Now we Click on the 'Instance Id' link of our newly created server in our Instances list.

Instances (1) <a href="#">Info</a>		<a href="#">C</a>	Connect	Instance state	Actions	<a href="#">Launch in</a>	
		<input type="text"/> Find instance by attribute or tag (case-sensitive)					
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	debserver1	i-0a6ab24417f81fffb	<span>Running</span>	t2.micro	<span>Initializing</span>	No alarms	ap-south-1a

## 16. Now click on the connect button

Instance summary for i-0a6ab24417f81fffb (debserver1) <a href="#">Info</a>		<a href="#">C</a>	Connect	Instance state	Actions
Updated less than a minute ago					
Instance ID	i-0a6ab24417f81fffb (debserver1)	Public IPv4 address	3.110.154.34   <a href="#">open address</a>	Private IPv4 addresses	172.31.41.246
IPv6 address	-	Instance state	<span>Running</span>	Public IPv4 DNS	ec2-3-110-134-34.ap-south-1.compute.amazonaws.com   <a href="#">open address</a>

## 17. Again, click on the connect button

Connect to instance [Info](#)  
Connect to your instance i-0a6ab24417f81fffb (debserver1) using any of these options

<a href="#">EC2 Instance Connect</a>	<a href="#">Session Manager</a>	<a href="#">SSH client</a>	<a href="#">EC2 serial console</a>
Instance ID	i-0a6ab24417f81fffb (debserver1)		
Public IP address	3.110.154.34		
User name	Enter the User name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name.		
ubuntu	<input type="text"/>		
<span>(i) Note:</span> In most cases, the default user name, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.			
<a href="#">Cancel</a> <a href="#">Connect</a>			

## 18. After this anew Tab will open with a Bash Terminal that is of our remote EC2 server!

Here we can type all our required commands that we used to type in a similar terminal by connecting to our remote server through our Bitvise SSH client software in our previous assignments.

The terminal window shows the AWS welcome message and a prompt for the user to run sudo apt update. The user has run the command and is now at the root prompt, with the full command history visible.

```

Doing Planck's MyReport ...
Connecting to instance i-0a6ab24417f81fffb (debserver1) ...
AWS Instance Connect ...
+ https://ubuntu.com/ubuntu/
Ubuntu gets updates to over 25,000 software packages with your
Ubuntu Pro subscription. Free for personal use.
https://ubuntu.com/ubuntu/pro

Expanded Security Maintenance for Applications is not enabled.
33 updates can be applied immediately.
18 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

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

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@i-0a6ab24417f81fffb:~$ i-0a6ab24417f81fffb (debserver1)
Ubuntu@i-0a6ab24417f81fffb:~$ PublicIP: 3.110.154.34 PrivateIP: 172.31.41.246

```

## 19. Now type the following commands in the terminal:-

→ git clone <https://github.com/> ..... //Your GitHub Repository URL

Give your Username of GitHub when asked.

Give your account Token when your Password is asked.

```
ubuntu@ip-172-31-41-246:~$ git clone https://github.com/DebrupPramanik/myRepoV1.git
Cloning into 'myRepoV1'...
Username for 'https://github.com': DebrupPramanik
Password for 'https://DebrupPramanik@github.com':
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 15 (delta 6), reused 4 (delta 0), pack-reused 0
Receiving objects: 100% (15/15), done.
Resolving deltas: 100% (6/6), done.
```

→ cd YourRepositoryname/

```
ubuntu@ip-172-31-41-246:~$ cd myRepoV1/
ubuntu@ip-172-31-41-246:~/myRepoV1$ █
```

→ npm install

```
ubuntu@ip-172-31-41-246:~/myRepoV1$ npm install
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. See https://v8.dev/blog/math-random for details.

added 258 packages, and audited 259 packages in 15s

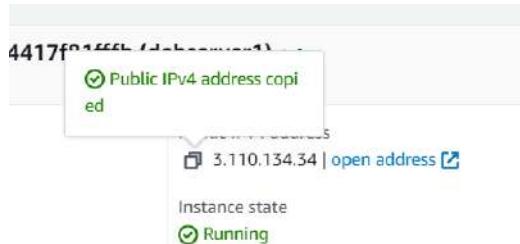
18 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
npm notice
npm notice New minor version of npm available! 9.5.1 -> 9.6.5
npm notice Changelog: https://github.com/npm/cli/releases/tag/v9.6.5
npm notice Run npm install -g npm@9.6.5 to update!
npm notice
```

→ node index.js

```
ubuntu@ip-172-31-41-246:~/myRepoV1$ node index.js
Started server
```

20. Now copy and paste the Public IPv4 address of your EC2 instance in another browser.



21. Now append the port no. 4000 (for our case) to the IP address in the browser with a ":" sign.



We have successfully Deployed a project from GitHub to EC2 by creating a new Security group and User Data.

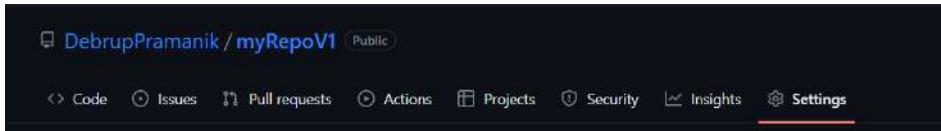
# ASSIGNMENT – 11

**Problem Statement:** Build Scaling plans in AWS that balance load on different EC2 instances.

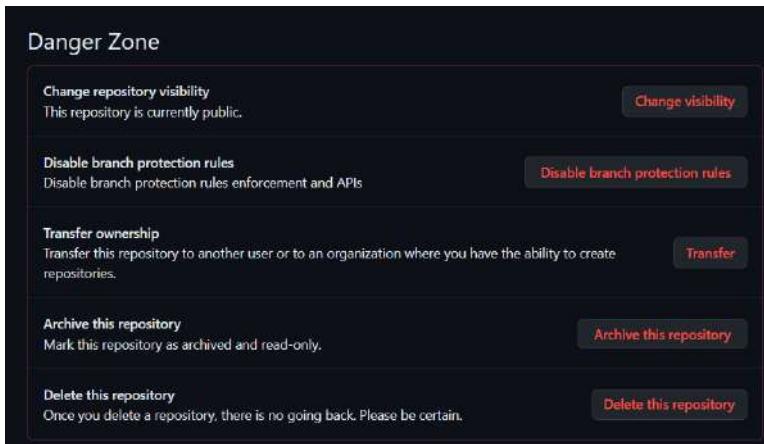
## Procedure:

1. Sign-in to GitHub. Make sure the Repository which will be cloned is made public or not.

- For This, select the “settings” tab of your repository.



- Next Scroll Down until you reach the danger-zone section.



- In here, check The Change Repository visibility option. Here, we can see the repository is currently Public. If it shows Private then click on the Change Visibility option and follow the on-screen Instructions to make the repository Public.

2. Now in another tab open AWS and Sign-in to your console.

3. Now go to your EC2 dashboard.

4. From the Left Side Nav-bar Go to your Instances Section. Under it Click the launch templates button.

A screenshot of the AWS EC2 Dashboard. On the left, a navigation bar includes "Instances" under the "Launch Templates" section. The main area shows "Resources" with a table of EC2 resources: Instances (running) 0, Auto Scaling Groups 0, Dedicated Hosts 0, Elastic IPs 0, Instances 0, Key pairs 2, Load balancers 0, Placement groups 0, Security groups 2, Snapshots 0, and Volumes 0. Below this are sections for "Launch instance" (with a "Launch instance" button) and "Service health" (showing Region: Asia Pacific (Mumbai)). On the right, there are "Account attributes" and "Explore AWS" sections.

5. Now click on the Create Launch Templates button.

A screenshot of the "EC2 launch templates" page. The title is "Streamline, simplify and standardize instance launches". It includes a "New launch template" button. Below it are sections for "Benefits and features": "Streamline provisioning" (Minimize steps to provision instances. With EC2 Auto Scaling, updates to a launch template can be automatically passed to an Auto Scaling group.), "Simplify permissions" (Create shorter, easier to manage IAM policies.), "Documentation" (links to Documentation and API reference), and "API reference".

6. Now, give a name and description for your EC2 template you are about to create. Here we gave the same for both the fields.

Next, Check the “Provide Guidance” box.

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name - required

debtemp1

Must be unique to this account. Max 128 chars. No spaces or special characters like %, \*, @, #.

Template version description

debtemp1

Max 256 chars

Auto Scaling guidance [Info](#)

Select this if you intend to use this template with EC2 Auto Scaling

Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

▶ Template tags

▶ Source template

7. Next, under the quick start menu select Ubuntu as the OS.

Recents | Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | S |

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type

ami-02eb7a47b3e7e9317 (64-bit (x86)) / ami-05dcff6fb7af5fc9 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-03-25

Architecture

AMI ID

64-bit (x86) | ami-02eb7a47b3e7e9317 | Verified provider

8. Under Instance type select t2.micro type of configuration.

▼ Instance type [Info](#)

Advanced

Instance type

t2.micro

Family: t2 - 1 vCPU - 1 GiB Memory Current generation: true

Free tier eligible

On-Demand Linux pricing: 0.0124 USD per Hour

On-Demand Windows pricing: 0.017 USD per Hour

On-Demand RHEL pricing: 0.0724 USD per Hour

On-Demand SUSE pricing: 0.0124 USD per Hour

All generations

Compare instance types

9. Select Existing Key-Pair and Security Group and if not applicable then Generate or Create a Key-Pair or Security Group wherever required.

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name

debkey2

Create new key pair

▼ Network settings [Info](#)

Subnet [Info](#)

Don't include in launch template

When you specify a subnet, a network interface is automatically added to your template.

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Select existing security group

Create security group

Security groups [Info](#)

Select security groups

mysec1 sg-0493398d43b761e55 X

VPC: vpc-0a33deec3fd6dc096

Compare security group rules

## 10. Now, Click on the Advanced Group Section at the bottom.

Scroll Down to User Data Section and paste the following commands in the provided box.

User data - optional [Info](#)

Enter user data in the field.

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
git clone YourRepositoryURLhere
cd YourRepositoryNamehere/
npm install
node index.js
```

After adding the commands, it will look like this.....

User data - optional [Info](#)

Enter user data in the field.

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
git clone https://github.com/DebrupPramanik/myRepoV1.git
cd myRepoV1/
npm install
node index.js
```

## 11. Now click on the Create Launch Template button.

Don't include in launch template

Metadata response hop limit: [Info](#)

Don't include in launch template

Allow tags in metadata: [Info](#)

Don't include in launch template

User data - optional [Info](#)

Enter user data in the field.

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
git clone https://github.com/DebrupPramanik/myRepoV1.git
cd myRepoV1/
npm install
node index.js
```

User data has already been base64 encoded

▼ Summary

Software Image (AMI)  
Canonical, Ubuntu, 22.04 LTS, ...[read more](#)  
ami-02eb7a4785e7e9517

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
mysec1

Storage (volumes)  
1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

Launch templates (1) <small>info</small>				
Launch template ID	Launch template name	Default version	Latest version	Actions
lt-0d93dae5bfe8072d5	debtemp1	1	1	<a href="#">Create launch template</a>

12. Now from the Left side Nav Bar go to Auto Scaling Groups under Auto-Scaling section.

The screenshot shows the AWS Management Console with the 'Auto Scaling' section selected in the left navigation bar. The main content area displays the 'Launch templates' page, which lists a single entry: 'debtemp1' with a 'Launch template ID' of 'lt-0d93dae5bfe8072d5'. The navigation bar also includes tabs for 'Services' and 'Search', and shows the location 'EC2 > Launch templates'. The top right corner displays the user's name 'Mumbai' and the date 'CSE\_2020\_30'.

13. Now click on Create Auto Scaling Group.

This screenshot shows the first step of the 'Create Auto Scaling group' wizard. The title reads 'Amazon EC2 Auto Scaling helps maintain the availability of your applications'. Below the title is a description of what Auto Scaling groups are: 'Auto Scaling groups are collections of Amazon EC2 instances that enable automatic scaling and fleet management features. These features help you maintain the health and availability of your applications.' A prominent orange button at the bottom right says 'Create Auto Scaling group'.

14. Now, Give a unique name to your new Auto Scaling Group. Also select the Launch Template that we recently created by using the drop-down menu under Launch Templates section.

This screenshot shows the second step of the 'Create Auto Scaling group' wizard. The 'Name' field is filled with 'deb\_asg1'. The 'Launch template' dropdown is set to 'debtemp1'. Below the dropdown is a link 'Create a launch template'. At the bottom, there is a 'Version' dropdown with 'Latest (1)' selected. The 'Switch to launch configuration' link is visible above the dropdown.

15. Under the selected Launch Template click on the version option and select Latest.

This screenshot shows a close-up of the 'Version' dropdown menu. The 'Latest (1)' option is highlighted with a blue selection bar. The dropdown also contains 'Default (1)' and 'Default (1)' again, with the number '1' below it.

**16.** Now click on the Next button.

**17.** After that, Under Availability Zones and Subnets select all the zones that appear.

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC

Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-0a33deec3fd6dc096  
172.31.0.0/16 Default

Create a VPC 

Availability Zones and subnets

Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets  

ap-south-1a | subnet-0bbe74a9835a07e58 X  
172.31.32.0/20 Default

ap-south-1b | subnet-0916d7caf57f5d661 X  
172.31.0.0/20 Default

ap-south-1c | subnet-09ddf920c63dcde50 X  
172.31.16.0/20 Default

Create a subnet 

**18.** Again, click on the Next button.

**19.** Now Under Load Balancers select the Attach to a New Load balancer option.

Load balancing 

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

No load balancer  
Traffic to your Auto Scaling group will not be fronted by a load balancer.

Attach to an existing load balancer  
Choose from your existing load balancers.

Attach to a new load balancer  
Quickly create a basic load balancer to attach to your Auto Scaling group.

**20.** Now select Internet-Facing under Load balancer scheme.

Load balancer scheme

Scheme cannot be changed after the load balancer is created.

Internal

Internet-facing

**21.** Under Listeners and Routing enter the port no. of the project and select Create target group followed by giving the target group a name.

Listeners and routing

If you require secure listeners, or multiple listeners, you can configure them from the [Load Balancing console](#) after your load balancer is created.

Protocol	Port	Default routing (forward to)
HTTP	4000	<input type="button" value="Create a target group"/> 

New target group name  
An instance target group with default settings will be created.

debashg1

**22.** Now click on the next button.

**23.** After clicking on the Next button, a new page will open. Under Group Size mention:

- a) Desired Capacity = 2
- b) Minimum Capacity = 2
- c) Maximum Capacity = 3

Group size - optional 

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity  
2

Minimum capacity  
2

Maximum capacity  
3

**24.** Now under Scaling policies Choose the Target Tracking Scaling policy option.

Select the metric type as Average CPU utilization.

Set Target Value to 50.

Set Warm-Up time to 300 seconds under Instances Need.

**Scaling policies - optional**

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. [Info](#)

**Target tracking scaling policy**  
Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.

**None**

Scaling policy name

Metric type

Target value

Instances need  
 seconds warm up before including in metric

Disable scale in to create only a scale-out policy

**25.** Then click on next.

**26.** Nothing to do in Notifications page. So again, click on the Next button.

**27.** No tags needed. Again, click on the Next button.

**28.** Now Review your Auto-Scaling Group you are going to create. Now click on the Create Auto-Scaling Group button.



**29.** Now we can go to the Auto Scaling Groups section and find our newly created Auto Scaling Group.

Auto Scaling groups (1) <a href="#">Info</a>						
<input type="checkbox"/> Name		Launch template/configuration	Instances	Status	Desired capacity	Min
<input type="checkbox"/>	debashg1	debtemp1   Version Latest	2	-	2	2

**30.** Return to the Instances Page using the Left side Nav bar.

Name	Instance ID	Instance state	Instance type	Status check
-	i-0a7cf75ace548840f	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>
-	i-0c51d43594a967f2b	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>

**31.** Click on the First instance. Copy its public IPV4 DNS.

Instance summary for i-0a7cf75ace548840f <a href="#">Info</a>				
Updated less than a minute ago				
Instance ID <a href="#">i-0a7cf75ace548840f</a>	Public IPv4 address <a href="#">13.126.127.63</a> <a href="#">open address</a>	Private IPv4 addresses <a href="#">172.31.14.78</a>		
IPv6 address -	Instance state <span>Running</span>	Public IPv4 DNS <a href="#">ec2-13-126-127-63.ap-south-1.compute.amazonaws.com</a> <a href="#">open address</a>		

32. Paste it in another browser.



## Welcome to nginx!

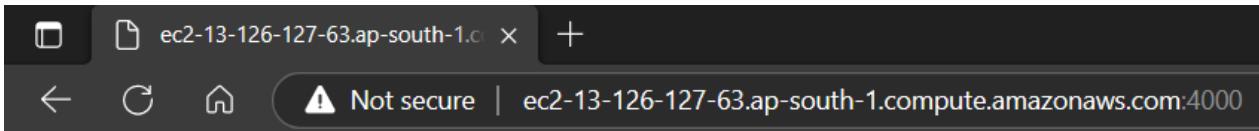
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

Thank you for using nginx.

We can successfully access the webpage.

33. Now to access our project webpage we need to append the port no. (4000) of our project with a ":"



Hello. My Name is Spider-Man!!! Nice to meet You!!!

34. To test our Auto-Scaling Group actually works we need to crash or overload the existing instance servers. Then only our Auto-Scaling Group will provide fresh instance servers automatically in case of crash; or it can provide extra servers to handle overloads.

35. We will now **CRASH THE SERVER INSTANCES** by terminating them.

36. Go to the instances page. Select the server instances.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
-	i-0a7cf75ace548840f	Running	t2.micro	2/2 checks passed	No alarms	+ ap-south-1b	ec2-13-126-127-63
-	i-0c51d43594a967f2b	Running	t2.micro	2/2 checks passed	No alarms	+ ap-south-1a	ec2-3-109-184-169

37. Now click on the Instance State button up top. From the drop-down select the Terminate instance option.

Instances (2/2) <a href="#">Info</a>		Connect	Instance state	Actions	Launch instances
<input type="checkbox"/>	Find instance by attribute or tag (case-sensitive)	<a href="#">Stop instance</a>		<a href="#">Actions</a>	<a href="#">Launch instances</a>
<input checked="" type="checkbox"/>	Instance state = running	<a href="#">Start instance</a>		<a href="#">Actions</a>	<a href="#">Launch instances</a>
<input checked="" type="checkbox"/>	Instance state = running	<a href="#">Reboot instance</a>		<a href="#">Actions</a>	<a href="#">Launch instances</a>
<input checked="" type="checkbox"/>	i-0a7cf75ace548840f	<a href="#">Hibernate instance</a>		<a href="#">Actions</a>	<a href="#">Launch instances</a>
<input checked="" type="checkbox"/>	i-0c51d43594a967f2b	<a href="#">Terminate instance</a>		<a href="#">Actions</a>	<a href="#">Launch instances</a>

38. Refresh the instances the page from time to time. After few seconds the instances will get terminated.

39. Wait for some time. Keep refreshing using the refresh button on the left side of Instance state button.

40. After some you will notice a new instance server will appear automatically! To help finding it more easily we need to activate the instance running filter. Click on the search box below the Instances Section Heading. Start typing running.

Select the option "Instance state = running option in the suggestion dropdown. The filter will be activated.

You can alternatively type the command directly in the search-box directly.

Now, only the running instances will be shown in the list.

Instances (1) <a href="#">Info</a>		Connect	Instance state	Instance type
<input type="checkbox"/>	running	<a href="#">Stop instance</a>		<a href="#">Actions</a>
<input type="checkbox"/>	Use: "running"	<a href="#">Start instance</a>		<a href="#">Actions</a>
<input type="checkbox"/>	filters	<a href="#">Reboot instance</a>		<a href="#">Actions</a>
<input type="checkbox"/>	API filters values	<a href="#">Hibernate instance</a>		<a href="#">Actions</a>
<input type="checkbox"/>	Instance state = running	<a href="#">Terminate instance</a>		<a href="#">Actions</a>

41. After some few seconds of refreshing we will be able to see two new servers are running.

Instances (2) <a href="#">Info</a>		Connect	Instance state	Actions	Launch instances
<input type="checkbox"/>	Find instance by attribute or tag (case-sensitive)	<a href="#">Stop instance</a>		<a href="#">Actions</a>	<a href="#">Launch instances</a>
<input type="checkbox"/>	Instance state = running	<a href="#">Start instance</a>		<a href="#">Actions</a>	<a href="#">Launch instances</a>

Now again copy paste the new public IPv4 DNS of the first instance and port no. in the other browser to see if the instances are working. It will be working.

So, our Auto-Scaling Group can handle instance crashing by providing new fresh instances.

**42.** We will now **OVERLOAD THE SERVER INSTANCES** by running scripts and increasing CPU utilization value above the threshold that we specified during Configuration of the Auto scaling group.

**43.** For it we will use:

- a) Use Bitvise SSH client for instance 1.
- b) Use direct connect terminal in AWS for instance 2.

**44. For Instance-1:**

- a) Copy the public IPv4 address
- b) Open Bitvise SSH client.
- c) Paste the IP and select/specify the necessary options. (**Refer Ass7**)
- d) Now Log-In to your server.
- e) Open the new Terminal.
- f) Now enter the command:

→ **nano infil.sh**

- g) After the command a new nano Editor window will open. Type the following in it.

```
#!/bin/bash
while true
do
    echo "Loop running"
done
```

```
GNU nano 6.2
#!/bin/bash
while true
do
    echo "Loop running"
done
```

- h) Now, to save and close the shell script we need to press the following shortcuts and keys sequentially:

**Ctrl+X**

**Y**

**Enter**

- i) Now you will be returned back to the terminal.

- j) Now type the following commands:

→ **chmod +x infil.sh**

(Used to give the execute permission for the infil.sh file)

```
ubuntu@ip-172-31-13-76:~$ nano infil.sh
ubuntu@ip-172-31-13-76:~$ chmod +x infil.sh
```

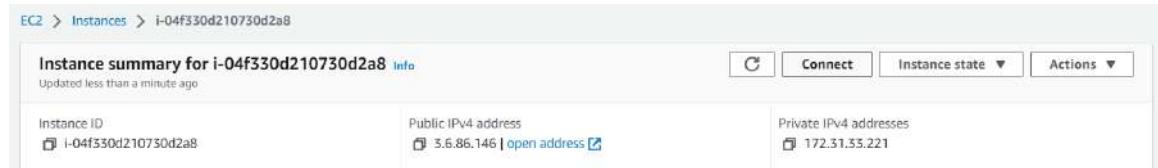
→ **./infil.sh**

(Used to execute the infil.sh script)

- k) Now the script will start running infinitely!
  - l) DO NOT CLOSE THE TERMINAL. Keep it minimized.

#### **45. For Instance-2:**

- a) Click on the instance 2.
  - b) Now click on the connect button



- c) Click on connect again
  - d) After the terminal gets opened we again follow the same steps as we did for instance-2
  - e) Enter the command:  
→ **nano infil.sh**
  - f) After the command a new nano Editor window will open. Type the following in it.

```
#!/bin/bash
while true
do
    echo "Loop running"
done
```

- g) Now, to save and close the shell script we need to press the following shortcuts and keys:  
**Ctrl+X**  
**Y**  
**Enter**
  - h) Now you will be returned back to the terminal.
  - i) Now type the following commands:

→ chmod +x infil.sh

(Used to give the execute permission for the infil.sh file)

→ ./infil.sh

(Used to execute the `infil.sh` script)

- j) Now the script will start running infinitely!
  - k) DO NOT CLOSE THE TERMINAL TAB. Go back to the previous tab to keep working in AWS.

**46.** Now go to the instances page.

**47.** Select both the instances.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
✓ -	i-0f5106bab83bb8ec1	Running	t2.micro	2/2 checks passed	No alarms	+ ap-south-1b	ec2-43-204-214-47
✓ -	i-0naef30d210730d2a8	Running	t2.micro	2/2 checks passed	No alarms	+ ap-south-1a	ec2-3-5-86-146.ap-south-1.amazonaws.com

48. Click on the instances white bar at the bottom of the page.

The screenshot shows the AWS CloudWatch Metrics Instances page. At the top, there are buttons for 'Instances (2/2)', 'Info', 'Connect', and 'Instance state'. Below is a search bar with placeholder text 'Find instance by attribute or tag (case-sensitive)'. A table lists two instances:

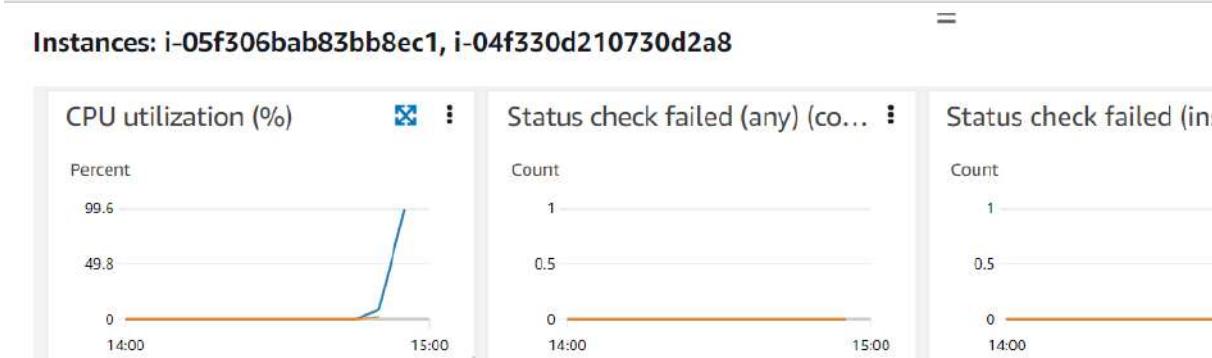
Name	Instance ID	Instance state	Instance type	Status check	Alarm status
-	i-05f306bab83bb8ec1	Running	t2.micro	2/2 checks passed	No alarms
-	i-04f330d210730d2a8	Running	t2.micro	2/2 checks passed	No alarms

At the bottom of the page, a grey bar displays the text 'Instances: i-05f306bab83bb8ec1, i-04f330d210730d2a8'.

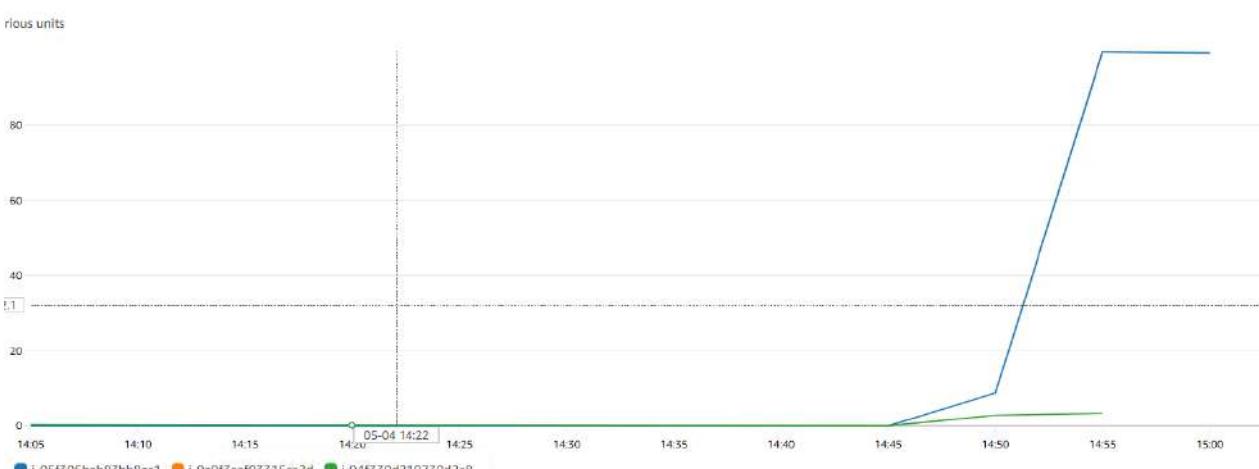
49. Now drag the two bars to expand the view.

The screenshot shows the same AWS CloudWatch Metrics Instances page, but the view has been expanded. The top bar now includes a maximize icon (two horizontal lines) and a close icon (an 'X').

50. We are interested only in the CPU utilization graph. Click on the maximize icon by hovering over the graph as shown in the fig to maximize this graph.



51. Our 1<sup>st</sup> instance has already crossed over 50% utilization. That's why we can see already a new third instance has been initiated by our auto-scaling group to compensate for the overload.



52. There can only be 3 servers running at a time for us as specified in our Auto-Scaling group when we were creating it. Hence, we have reached our maximum limit of instances running concurrently.

Name	Instance ID	Instance state	Security group name	Key name	Launch time
-	i-05f306bab83bb8ec1	Running	mysec1	debkey2	2023/05/04 16:36 GMT+5:30
-	i-04f330d210730d2a8	Running	mysec1	debkey2	2023/05/04 16:38 GMT+5:30
-	i-0a0f3aaaf93315ca2d	Running	mysec1	debkey2	2023/05/04 20:33 GMT+5:30

Hence, our Auto-Scaling Group can handle instance overloading by providing new instances to handle the overloading.

Note that our project webpage was not at all disconnected because of this.

**We have successfully created, configured and tested our Auto-Scaling Group.**

Now observe that whenever we close or terminate any instance then a new instance gets created. Hence, we cannot delete them if we want to delete them finally.

Follow these steps to cleanly remove Auto-Scaling Groups and everything related to it.

1) Go to Auto-Scaling groups and select the one which we are using.

The screenshot shows the AWS Auto Scaling Groups page. At the top, there is a header with the title 'Auto Scaling groups (1/1)' and a 'Create an Auto Scaling group' button. Below the header is a search bar labeled 'Search your Auto Scaling groups'. The main table lists one item: 'debasg1' with a status of 'debttemp1 | Version Latest'. The table includes columns for Name, Launch template/configuration, Instances, Status, Desired capacity, and Min.

2) Now click on the delete button.

3) Type delete and finally delete it.

4) Notice the status changes to Deleting. It will take some minutes to fully delete. Now go to the Load balancer section from the left side nav bar.

The screenshot shows the AWS Auto Scaling Groups and Load Balancers pages. On the left, the navigation menu is visible with the 'Auto Scaling Groups' option selected under the 'Auto Scaling' section. The top navigation bar shows the AWS logo, services, search bar, and region 'Mumbai'. The main content area shows the 'Auto Scaling groups' page with 'debasg1' in 'Deleting' status. Below it, the 'Auto Scaling group: debasg1' details page is shown with tabs for Details, Activity, Automatic scaling, Instance management, Monitoring, and Instance refresh. The 'Group details' section shows the group name 'debasg1', desired capacity '5', and status '-'. The 'Amazon Resource Name (ARN)' field contains 'arn:aws:autoscaling:ap-south-1:7285'. The bottom part of the screenshot shows the 'Load balancers' page with 'debasg1-1' listed as active with 3 availability zones and type 'application'. The navigation menu on the left also has the 'Load Balancers' option selected under the 'Load Balancing' section.

5) Now select the load balancer and click on the action button on the top.

The screenshot shows the AWS EC2 Load balancers page. A single load balancer, 'debasg1-1', is listed in the table. The table columns include Name, DNS name, State, VPC ID, and Availability. The 'Actions' menu on the right provides options like Edit IP address type, Edit subnets, Edit instances, Edit health check settings, Edit listener, Edit security groups, Edit load balancer attributes, Manage tags, and Delete load balancer.

Now select the Delete Load balancer option to delete it.

6) Now go to the Target Groups section.

7) Select the target group and click on the action button on the top. Select the delete option.

The screenshot shows the AWS EC2 Target groups page. A single target group, 'debasg1-1', is listed in the table. The table columns include Name, ARN, Port, Protocol, and Targets. The 'Actions' menu on the right provides options like Delete, Register targets, Edit health check settings, Edit target group attributes, Manage tags, Associate with a new load balancer, and Associate with an existing load balancer.

8) Now go to the instances page.

9) You will find that all the instances created by the Auto-Scaling group will automatically be terminated.

(If not wait for some time. Check if the Auto Scaling Group has been deleted by now.)

The screenshot shows the AWS EC2 Instances page. Three instances are listed, all in the 'Terminated' state. The table columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS.

10) Finally check the Auto-Scaling Group section to see if it is completely removed/deleted.

The screenshot shows the AWS EC2 Auto Scaling groups page. It displays a message stating 'No Auto Scaling groups found in the current region'. A 'Create an Auto Scaling group' button is visible at the bottom.

Everything was deleted successfully.

**Hence, we successfully deleted our Auto-Scaling Group and all the load balancers, target groups, instances that was created.**

# ASSIGNMENT-12

**Problem Statement:** Deploy a project from GitHub to EC2 without using Port.

## Procedures:

1. Sign-in to AWS console.
2. Go to the EC2 dashboard. Now go to the instances page.
3. Click on the create new instance button.
4. Now create an EC2 server using the Security Group created earlier and enter the user data  
**(Refer to Ass10)**

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

**Name and tags** [Info](#)

Name  Add additional tags

**Recent** **Quick Start**

Amazon Linux  macOS  Ubuntu  Windows  Red Hat  >  Browse more AMIs  
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type Free tier eligible

ami-02eb7a4783e7e9317 (64-bit (x86)) / ami-0a5dcfffb7af3fc9 (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Root device type: ebs

Description  
Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-03-25

Architecture  AMI ID: ami-02eb7a4783e7e9317 Verified provider

**Instance type** [Info](#)

Instance type  Free tier eligible

t2.micro 1 vCPU 1 GiB Memory Current generation: true  
Family: t2 On-Demand Linux pricing: 0.0124 USD per Hour  
On-Demand Windows pricing: 0.017 USD per Hour  
On-Demand RHEL pricing: 0.0724 USD per Hour  
On-Demand SUSE pricing: 0.0124 USD per Hour

All generations [Compare instance types](#)

**Key pair (login)** [Info](#)  
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required  [Create new key pair](#)

**Network settings** [Info](#)

Network [Info](#)  
vpc-0a33deec3fd6dc096

Subnet [Info](#)  
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)  Enable

**Firewall (security groups)** [Info](#)  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group  Select existing security group

Security groups [Info](#)  
 [Compare security group rules](#)

mysec1 sg-0493398d43b761e55 X  
VPC: vpc-0a33deec3fd6dc096

**Advanced details** [Info](#)

User data - optional [Info](#)  
Enter user data in the field.

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
git clone https://github.com/DebrupPramanik/myRepoV1.git
cd myRepoV1
npm install
node index.js
```

[Launch instance](#)

User data has already been base64 encoded

[Review commands](#)

## 5. Create the instance and click on the instance after creation.

Instances (1) <a href="#">Info</a>		<a href="#">C</a>	Connect	Instance state	Actions	<a href="#">Launch instances</a>	▼
<a href="#">Find instance by attribute or tag (case-sensitive)</a>							
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	debser1	i-0204a46d8ffd384cb	<span style="color: green;">Running</span>	t2.micro	<span style="color: gray;">Initializing</span>	No alarms +	ap-south-1a

## 6. Copy the public IPv4 address and paste it in another browser. The nginx homepage will show up.

**Instance summary for i-0204a46d8ffd384cb (debser1) [Info](#)**  
Updated less than a minute ago

Instance ID <a href="#">i-0204a46d8ffd384cb (debser1)</a>	Public IPv4 address <a href="#">open address</a>	Private IPv4 addresses <a href="#">172.31.38.17</a>
IPv6 address -	Instance state <span style="color: green;">Running</span>	Public IPv4 DNS <a href="#">ec2-65-2-169-220.ap-south-1.compute.amazonaws.com</a>   <a href="#">open address</a>
Hostname type IP name: ip-172-31-38-17.ap-south-1.compute.internal	Private IP DNS name (IPv4 only) <a href="#">ip-172-31-38-17.ap-south-1.compute.internal</a>	



### Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](#). Commercial support is available at [nginx.com](#).

*Thank you for using nginx.*

Our server is working perfectly. Note, in previous assignments we used to connect to our project webpages using port no. However, in this exercise we are going to access our project webpage without using any port no.

## 7. Copy the Public IPv4 address of the server instance and use this to connect it to the server using Bitvise SSH client. (Refer Ass7)

**Default profile**

- [Save profile as](#)
- [Bitvise SSH Server Control Panel](#)
- [New terminal console](#)

Login	Options	Terminal	RDP	SFTP	Services	C2S	S2C	SSH	Notes	About	
<b>Server</b>											
Host <input type="text" value=""/>											
Port <input type="text" value=""/>	<input type="checkbox"/> Enable obfuscation										
Obfuscation keyword <input type="text" value=""/>											
<b>Kerberos</b>											
SPN <input type="text" value=""/>											
<input type="checkbox"/> GSS/Kerberos key exchange											
<input type="checkbox"/> Request delegation											
<input checked="" type="checkbox"/> gssapi-keyex authentication											
<b>Authentication</b>											
Username <input type="text" value="ubuntu"/>											
Initial method <input type="text" value="publickey"/>											
Client key <input type="text" value="Global 2"/>											
Passphrase <input type="text" value=""/>											
Elevation <input type="text" value="Default"/>											

**8. Now open the terminal in Bitvise.**

```
ubuntu@65.2.169.220:22 - Bitvise xterm - ubuntu@ip-172-31-38-17: ~
Last login: Wed May 10 12:14:04 2023 from 150.129.133.232
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-38-17:~$
```

**9. Enter the following commands in it.**

→ **pwd**

```
ubuntu@ip-172-31-38-17:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-38-17:~$
```

(To check current directory)

→ **cd /**

```
ubuntu@ip-172-31-38-17:~$ cd /
```

(To go to root folder)

→ **pwd**

```
ubuntu@ip-172-31-38-17:/~$ pwd
/
```

→ **cd /etc/nginx/sites-available/**

```
ubuntu@ip-172-31-38-17:/~$ cd /etc/nginx/sites-available/
ubuntu@ip-172-31-38-17:/etc/nginx/sites-available$
```

(To open the sites-available directory under nginx)

→ **sudo nano default**

```
ubuntu@ip-172-31-38-17:/etc/nginx/sites-available$ sudo nano default
GNU nano 6.2                               default
##
## You should look at the following URL's in order to grasp a solid understanding
## of Nginx configuration files in order to fully unleash the power of Nginx.
## https://www.nginx.com/resources/wiki/start/
## https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
## https://wiki.debian.org/Nginx/DirectoryStructure
##
## In most cases, administrators will remove this file from sites-enabled/ and
## leave it as reference inside of sites-available where it will continue to be
## updated by the nginx packaging team.
##
## This file will automatically load configuration files provided by other
## applications, such as Drupal or Wordpress. These applications will be made
## available underneath a path with that package name, such as /drupal18.
##
## Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##
```

```
# Default server configuration
#
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
```

(To open the default file in the nano editor)

**10. After opening the default file in the nano editor, search for the location / {}. It should be after server\_name\_;**

```
server_name _;

location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $uri $uri/ =404;
}
```

11. Comment out the location block and each and every line inside the block.

```
server_name _;

#location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    #try_files $uri $uri/ =404;
}
```

12. Now paste the following code just under the closing curly bracket.

```
location / {
    proxy_pass http://localhost:4000;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'Upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
}
```

```
server_name _;

#location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    #try_files $uri $uri/ =404;
}
location / {
    proxy_pass http://localhost:4000;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'Upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
}
```

13. Now save it by Ctrl+X and exit nano editor.

14. You will be reverted back to the terminal. Type the following command.....

→ sudo systemctl restart nginx

```
ubuntu@ip-172-31-2-192:/etc/nginx/sites-available$ sudo systemctl start nginx
```

15. Now paste the public IPv4 address in your browser. Now press Enter. Our project page will show up without entering our port no.

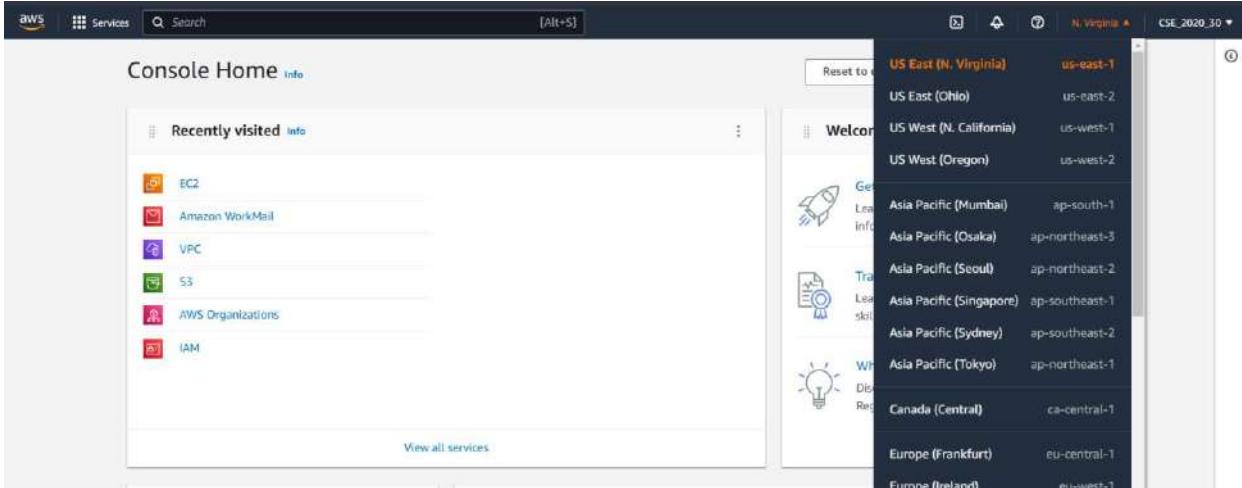


We have successfully deployed a project from GitHub to EC2 without using port.

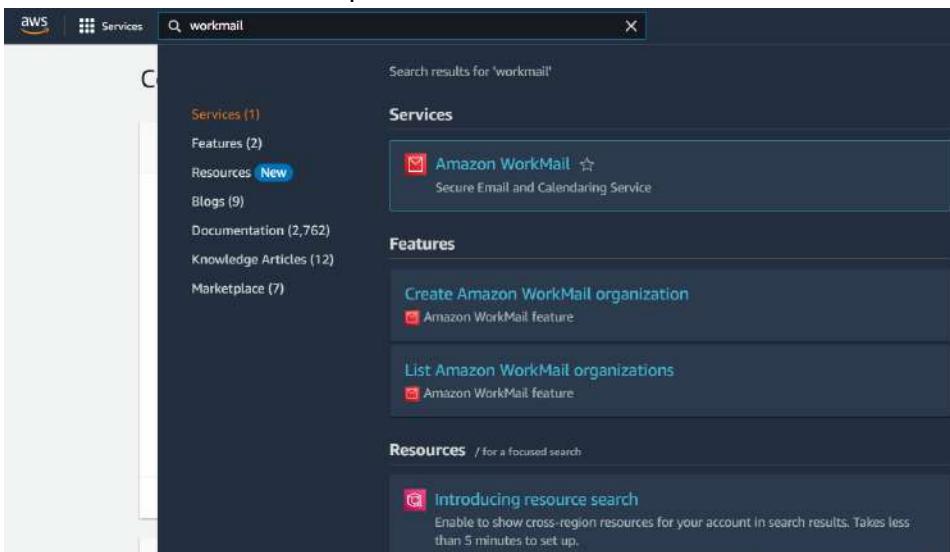
# ASSIGNMENT-13

**Problem Statement:** Create a workmail for your profile.

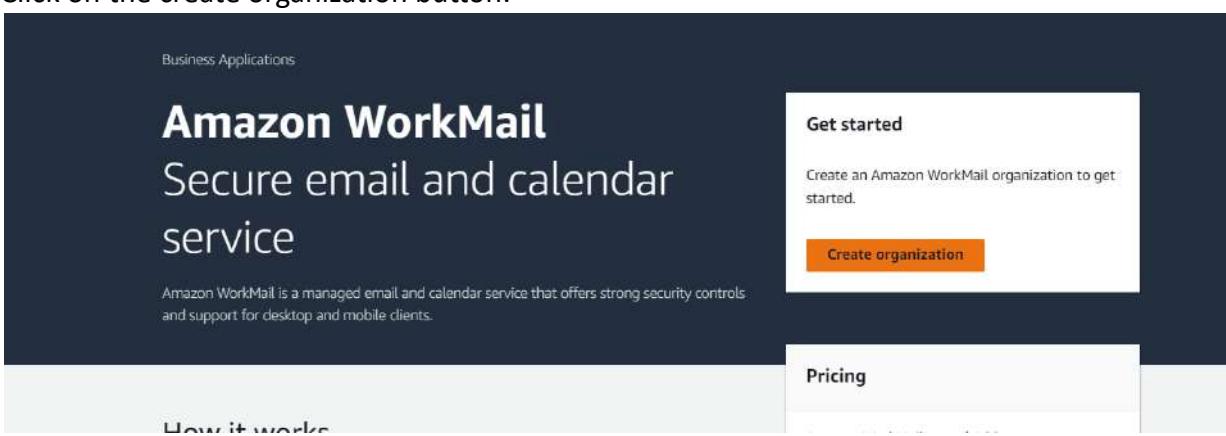
1. Sign-in to your AWS console.
2. Change your region to N.Virginia from the dropdown on the left-side of the username in the top bar of the console homepage.



3. Search for workmail.
4. Select the AWS workmail option.



5. Click on the create organization button.



## 6. Select the free test domain option under email domain.

Email domain [Info](#)  
Select the domain to use for email addresses in your organization.

- Existing Route 53 domain  
Select a domain name that you manage with a Route 53 hosted zone.
- New Route 53 domain  
Register a new Route 53 domain name to use with Amazon WorkMail.
- External domain  
Enter a domain name that you manage with an external DNS provider.
- Free test domain  
Use a free testing domain provided by Amazon WorkMail. You can add a domain later.

7. Give the organization name/ alias. Make sure it is unique in the region.

8. Next click on create organization button.

9. Wait for two minutes.

10. After waiting the organization will become active. Now click on the organization name.

Amazon WorkMail > Organizations

Organizations (1) [Info](#)

Find organizations

papinoob	<input type="radio"/>
Organization ID	m-86566ac1147549f187484e2793ace4ad
Default domain	papinoob.awsapps.com
State	<span style="color: green;">Active</span>

You are using the test domain as your default domain. We recommend that you add a custom domain and set it as the default domain.

Manage domains

Amazon WorkMail > Organizations > papinoob

**papinoob**

**Organization details** [Info](#)

Organization ID	<a href="#">m-86566ac1147549f187484e2793ace4ad</a>	State	<span style="color: green;">Active</span>	Directory type	WorkMail directory
ARN	<a href="#">arn:aws:workmail:us-east-1:728364961341:organization/m-86566ac1147549f187484e2793ace4ad</a>	Date created	May 10, 2023 at 20:51 (UTC+5:30)	Directory ID	<a href="#">d-906794a9b0</a>
		Default domain	<a href="#">Info</a>	<a href="#">papinoob.awsapps.com</a>	

**User login**

Desktop or mobile apps	<a href="#">WorkMail documentation for setting up email clients</a>	Amazon WorkMail web application	<a href="https://papinoob.awsapps.com/mail">https://papinoob.awsapps.com/mail</a>
------------------------	---	---------------------------------	---

[Delete organization](#)

11. Now go to the Users section under Organization tab on the left side nav bar.

Amazon WorkMail

Organizations

What's new

**Organization**

- [Users](#)
- [Groups](#)
- [Resources](#)

**12. Click on the create user button.**

Amazon WorkMail > Organizations > papinoob > Users

Users (0) Info

Search users

Create user

Display name	User name	Primary email address	State
No users to display.			

**13. Give username. Then display name.**

Create a user Info

Add a user to your Amazon WorkMail organization.

User details

User name  
The user name enables the user to login to the Amazon WorkMail webmail.  
david

User name can only contain the following characters: a-z, A-Z, 0-9, \_, (underscore), - (hyphen) and @.

First name - optional

Last name - optional

Display name  
The name by which the user is presented in the system.  
david

Email setup

Email address  
Primary email address to be used for this user.  
david @ papinoob.awsapps.com

Password  
Password for the user to log in with.  
\*\*\*\*\*

Passwords must have an 8-character minimum with at least one character from three of these four categories: lowercase, uppercase, numeric, and special characters.

Repeat password  
\*\*\*\*\*

**14. Then provide the password.**

Email setup

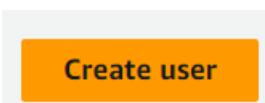
Email address  
Primary email address to be used for this user.  
david @ papinoob.awsapps.com

Password  
Password for the user to log in with.  
\*\*\*\*\*

Passwords must have an 8-character minimum with at least one character from three of these four categories: lowercase, uppercase, numeric, and special characters.

Repeat password  
\*\*\*\*\*

**15. Next click on the Create User button.**



**16. After the user is created return back to your organization page using the following link.**

aws Services Search [Alt+S]

N. Virginia CSE\_2020

Amazon WorkMail > Organizations > papinoob > Users

Users (1) Info

Search users

Create user

Display name	User name	Primary email address	State
david	david	david@david@papinoob.awsapps.com	Enabled

Successfully created david.

**17. Click on the organization name.**

papinoob

Organization ID  
m-86566ac1147549f187484e2793ace4ad

Default domain  
papinoob.awsapps.com

State  
Active

**18.** Now click on the Amazon Workmail web application link.

The screenshot shows the 'Organization details' section of the AWS WorkMail console. It includes fields for Organization ID (m-86566ac1147549f187484e2793ace4ad), State (Active), Directory type (WorkMail directory), ARN (arn:aws:workmail:us-east-1:728364961341:organization/m-86566ac1147549f187484e2793ace4ad), Date created (May 10, 2023 at 20:51 (UTC+5:30)), Directory ID (d-906794a9b0), and Default domain (papinoob.awsapps.com).

**User login**

Desktop or mobile apps | Amazon WorkMail web application  
<https://papinoob.awsapps.com/mail>

**19.** Enter the credentials of the user you just created in it.



Please log in with your papinoob credentials

Username (not email address)

Username

Remember username

Password

Password

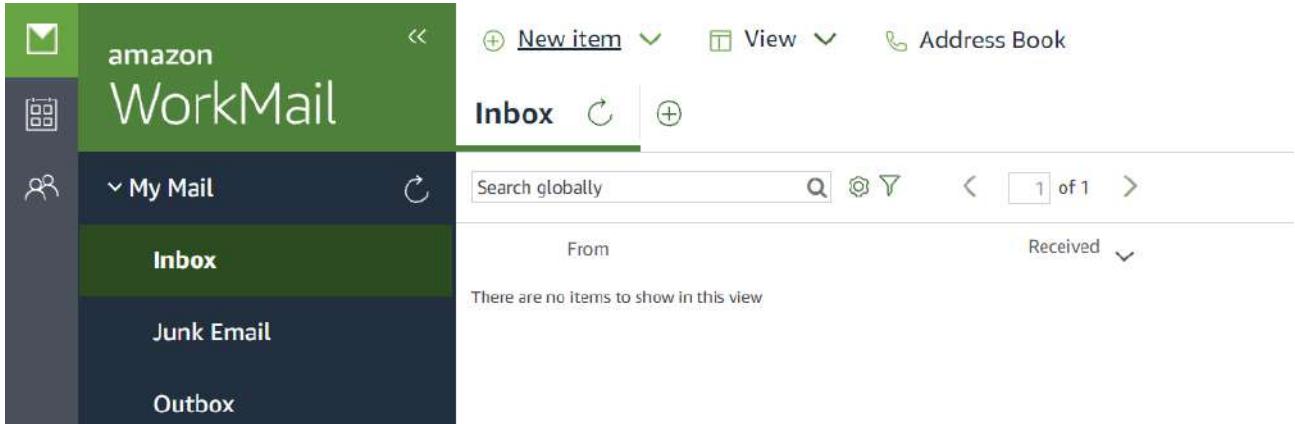
**Sign In**

By continuing, you agree to the AWS Customer Agreement or other agreement for AWS services, and the Privacy Notice. This site uses essential cookies. See our Cookie Notice for more information.

**20.** Our mail will be opened as the entered user.

The screenshot shows the AWS WorkMail inbox interface. The left sidebar lists 'My Mail' sections: Inbox (selected), Junk Email, Outbox, Drafts, Sent Items, Deleted Items, and RSS Feeds. The main area displays the 'Inbox' view with a search bar and a message stating 'There are no items to show in this view'. The top right corner shows the user's name (david) and various navigation icons.

- 21.** Now let's test our mail server by seeing whether it can send and receive mail. Send some mail to your Gmail from this by creating a mail using the new item option.



- 22.** Select new email. Now write an email to your Gmail address.

david Sent on: Today, 9:11 pm

Hello World

Hello,  
This is David from AWS Workmail.

- 23.** Check your Gmail.

Hello World Inbox

 david <david@papinoob.awsapps.com>  
to me ▾

Hello,  
This is David from AWS Workmail.

← Reply → Forward

We received it.

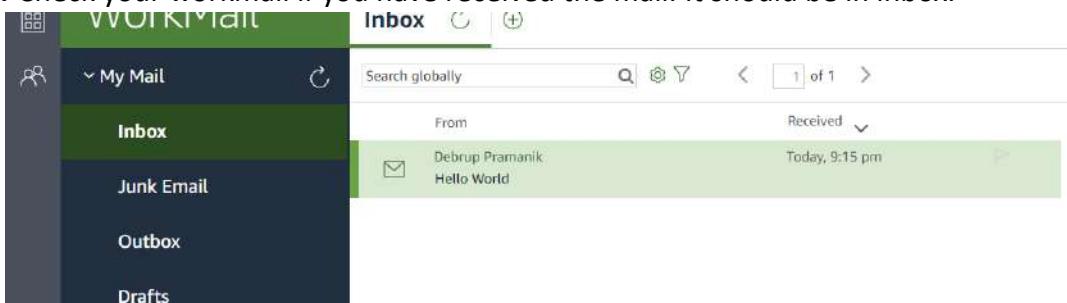
- 24.** Now send something back to the workmail.

 Debrup Pramanik <debrup202002@gmail.com>  
to david ▾

Hello,  
This is Debrup from Gmail. I received your request.

← Reply → Forward

- 25.** Check your workmail if you have received the mail. It should be in inbox.



Hello World

To david

---

Hello,  
This is Debrup from Gmail. I received your request.

We have successfully received a mail.

**Hence our mail is working perfectly fine.**

**We have successfully created a workmail for our profile.**

# ASSIGNMENT – 14

## Problem Statement: Create an elastic IP for an instance.

1. Sign-in to your AWS console.
2. Create an EC2 instance. (We do not need any user-data or any custom security group for this assignment)

The screenshot shows the AWS EC2 Instances list. There is one instance named "debserver1" which is "Running". The Public IPv4 address is listed as 3.110.83.71. The "Actions" dropdown menu is open, showing options like "Launch instances", "Stop instance", "Start instance", "Reboot instance", "Hibernate instance", and "Terminate instance".

3. After the instance gets created click on it. Copy the public IPv4 address and paste it in a simple text file anywhere in your pc.

The screenshot shows the Instance summary for the instance "debserver1". It displays the Public IPv4 address as 3.110.83.71. A large black box highlights this address. The "Actions" dropdown menu is open, showing options like "Launch instances", "Stop instance", "Start instance", "Reboot instance", "Hibernate instance", and "Terminate instance".

4. Now go back to the instances list and select our instance.

The screenshot shows the AWS EC2 Instances list with the instance "debserver1" selected. The "Actions" dropdown menu is open, showing options like "Launch instances", "Stop instance", "Start instance", "Reboot instance", "Hibernate instance", and "Terminate instance".

5. After selection click on the Instance state button and click on the Stop Instance option.

The screenshot shows the AWS EC2 Instances list with the instance "debserver1" selected. The "Instance state" dropdown menu is open, showing options: "Stop instance", "Start instance", "Reboot instance", "Hibernate instance", and "Terminate instance".

6. Wait for few seconds.

7. Now again select the instance and click on the Instance state button. Now click on the start instance button.

The screenshot shows the AWS EC2 Instances list with the instance "debserver1" selected. The "Instance state" dropdown menu is open, showing options: "Force stop instance", "Start instance", "Reboot instance", "Hibernate instance", and "Terminate instance".

8. Click on the instance and copy the IPv4 address again and paste it in the same text file.

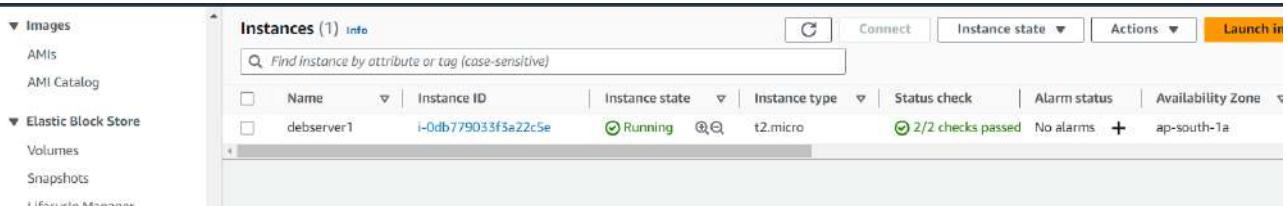
The screenshot shows the Instance summary for the instance "debserver1". It displays the Public IPv4 address as 43.205.95.224. A large black box highlights this address. The "Actions" dropdown menu is open, showing options like "Launch instances", "Stop instance", "Start instance", "Reboot instance", "Hibernate instance", and "Terminate instance".

## 9. Now compare both the new and old IP address and notice that they are not the same.

```
3.110.83.71  
43.205.95.224
```

So even if we stop and restart our same instance it changes its public IPv4 address. This may not be desirable in some situations. So, to ensure that our instance does not change its public IPv4 address under any circumstances, we need to create an Elastic IP and associate/bind the instance to it. After that it will always be assigned the same Elastic IP as its public IPv4 address (static) all the time.

## 10. For creating an Elastic IP, we need to go scroll down the left side Nav bar and find the Network and security section.



## 11. Under it click on the Elastic IPs option.

### ▼ Network & Security

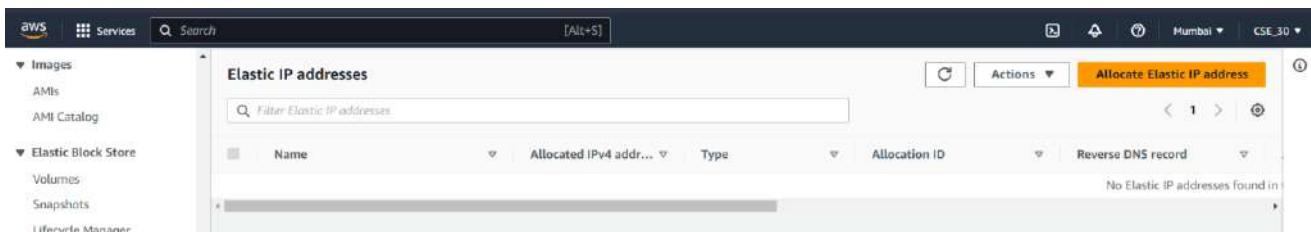
Security Groups

Elastic IPs

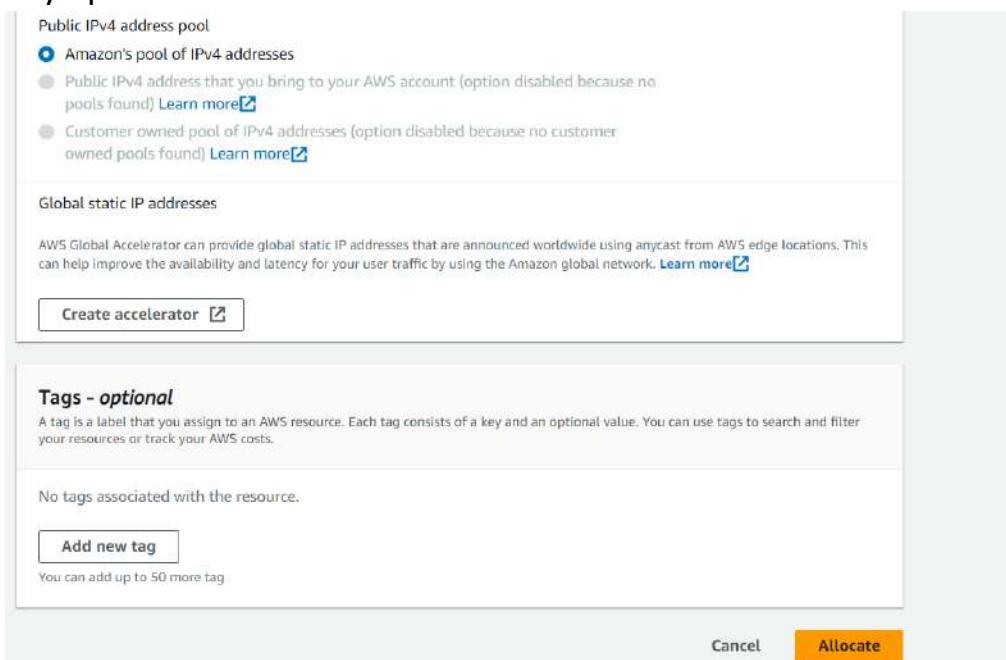
Placement Groups

Key Pairs

Network Interfaces



## 12. Now, click on the Allocate Elastic IP address button on the right side. No need to change any options. Just click on the Allocate button.



**13. Now click on the Elastic IP address (in blue).**

Elastic IP addresses (1/1)					
Name		Allocated IPv4 addr...	Type	Allocation ID	Reverse DNS record
<input checked="" type="checkbox"/>	-	<a href="#">3.7.231.115</a>	Public IP	eipalloc-0864617bd007a2f9b	-

**14. Next click on the Associate Elastic IP address button.**

EC2 > Elastic IP addresses > 3.7.231.115

**3.7.231.115**

**Actions** **Associate Elastic IP address**

**Summary**

Allocated IPv4 address <a href="#">3.7.231.115</a>	Type <a href="#">Public IP</a>	Allocation ID <a href="#">eipalloc-0864617bd007a2f9b</a>	Reverse DNS record -
Association ID -	Scope <a href="#">VPC</a>	Associated instance ID -	Private IP address -
Network interface ID -	Network interface owner account ID -	Public DNS -	NAT Gateway ID -
Address pool <a href="#">Amazon</a>	Network Border Group <a href="#">ap-south-1</a>		

**Tags (0)** **Manage tags**

Key	Value

**15. Choose your instance you want to associate with it.**

**16. Keep the Private IP address as specified in the dropdown when clicking for the Private Address.**

**17. Select the Allow Elastic IP to be reassociated option if we want to reuse it again for another instance.**

**Elastic IP address: 3.7.231.115**

Resource type  
Choose the type of resource with which to associate the Elastic IP address.

Instance  
 Network interface

**⚠ If you associate an Elastic IP address with an instance that already has an Elastic IP address associated, the previously associated Elastic IP address will be disassociated, but the address will still be allocated to your account. [Learn more](#)**

If no private IP address is specified, the Elastic IP address will be associated with the primary private IP address.

**Instance**

**Private IP address**  
The private IP address with which to associate the Elastic IP address.

**Reassociation**  
Specify whether the Elastic IP address can be reassigned to a different resource if it is already associated with a resource.

Allow this Elastic IP address to be reassigned

18. Now click the associate button.
19. The Elastic IP should have been successfully associated with the instance.
20. To check it go back to the instances page. Click on the Instance and see the Public IPv4 address and the Elastic IP address. They should be same. Also notice that the public IPv4 address has turned into a hyperlink to the Elastic IP page.

Instance summary for i-0db779033f3a22c5e (debserver1) <a href="#">Info</a>		Actions
Updated less than a minute ago		
Instance ID <a href="#">i-0db779033f3a22c5e (debserver1)</a>	Public IPv4 address <a href="#">3.7.231.115   open address</a>	Private IPv4 addresses <a href="#">172.31.40.195</a>
IPv6 address -	Instance state <span style="color: green;">Running</span>	Public IPv4 DNS <a href="#">ec2-3-7-231-115.ap-south-1.compute.amazonaws.com   open address</a>
Hostname type IP name: ip-172-31-40-195.ap-south-1.compute.internal	Private IP DNS name (IPv4 only) <a href="#">ip-172-31-40-195.ap-south-1.compute.internal</a>	Elastic IP addresses <a href="#">3.7.231.115 [Public IP]</a>
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	AWS Compute Optimizer finding <a href="#">Opt-in to AWS Compute Optimizer for recommendations.</a>   Learn more
Auto-assigned IP address -	VPC ID <a href="#">vpc-0a33deec3fd6dc096</a>	Auto Scaling Group name -
IAM Role -	Subnet ID <a href="#">subnet-0bbe74a9835a07e38</a>	

**Now stop and restart the instance and see if the public IPv4 address changes or not. It will not change.**

**Hence, we have successfully created an Elastic IP for an instance.**

To delete the Elastic IP, follow these steps:

1. Click on the Elastic IP.
2. Click on the actions button.

EC2 > Elastic IP addresses > 3.7.231.115

3.7.231.115			Actions									
<b>Summary</b> <table border="1"> <tr> <td>Allocated IPv4 address <a href="#">3.7.231.115</a></td> <td>Type <a href="#">Public IP</a></td> <td>Allocation ID <a href="#">eipalloc-0864617bd007a2f9b</a></td> </tr> <tr> <td>Association ID <a href="#">eipassoc-053b5539deac45cb4</a></td> <td>Scope <a href="#">VPC</a></td> <td>Associated instance ID <a href="#">i-0db779033f3a22c5e</a></td> </tr> <tr> <td>Network interface ID <a href="#">eni-0d87560629582fe63</a></td> <td>Network interface owner account ID <a href="#">728364961341</a></td> <td>Public DNS <a href="#">ec2-3-7-231-115.ap-south-1.compute.amazonaws.com</a></td> </tr> </table>			Allocated IPv4 address <a href="#">3.7.231.115</a>	Type <a href="#">Public IP</a>	Allocation ID <a href="#">eipalloc-0864617bd007a2f9b</a>	Association ID <a href="#">eipassoc-053b5539deac45cb4</a>	Scope <a href="#">VPC</a>	Associated instance ID <a href="#">i-0db779033f3a22c5e</a>	Network interface ID <a href="#">eni-0d87560629582fe63</a>	Network interface owner account ID <a href="#">728364961341</a>	Public DNS <a href="#">ec2-3-7-231-115.ap-south-1.compute.amazonaws.com</a>	<a href="#">Associate Elastic IP address</a> <ul style="list-style-type: none"> <li><a href="#">Release Elastic IP addresses</a></li> <li><a href="#">Disassociate Elastic IP address</a></li> <li><a href="#">Update reverse DNS</a></li> <li><a href="#">Enable transfers</a></li> <li><a href="#">Disable transfers</a></li> <li><a href="#">Accept transfers</a></li> </ul>
Allocated IPv4 address <a href="#">3.7.231.115</a>	Type <a href="#">Public IP</a>	Allocation ID <a href="#">eipalloc-0864617bd007a2f9b</a>										
Association ID <a href="#">eipassoc-053b5539deac45cb4</a>	Scope <a href="#">VPC</a>	Associated instance ID <a href="#">i-0db779033f3a22c5e</a>										
Network interface ID <a href="#">eni-0d87560629582fe63</a>	Network interface owner account ID <a href="#">728364961341</a>	Public DNS <a href="#">ec2-3-7-231-115.ap-south-1.compute.amazonaws.com</a>										

3. From the drop-down menu select Disassociate Elastic IP address. Then again click on disassociate on the pop-up.
4. Next again click on the Actions button and this time select Release Elastic IP address.

EC2 > Elastic IP addresses > 3.7.231.115

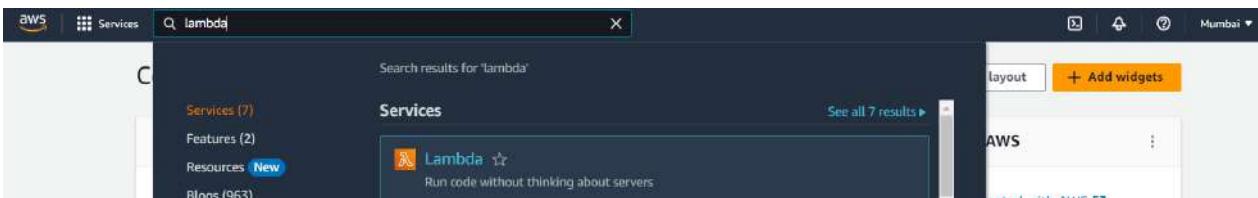
3.7.231.115		Actions
		<a href="#">Associate Elastic IP address</a> <ul style="list-style-type: none"> <li><a href="#">Release Elastic IP addresses</a></li> </ul>

5. Now you can go back to your instance and see that the IPv4 address has already changed to a random one and it has no Elastic IP address associated with it. Now you can terminate the instance.

# ASSIGNMENT – 15

## **Problem Statement: Create a serverless computing service.**

1. Sign-in to your AWS console.
2. Search for Lambda



- Click on the first result named Lambda.
3. Now click on the Create Function button on the top right corner.



4. Select Author from scratch option.



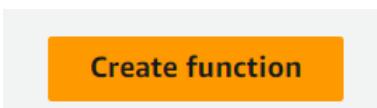
5. Give the name of the function.



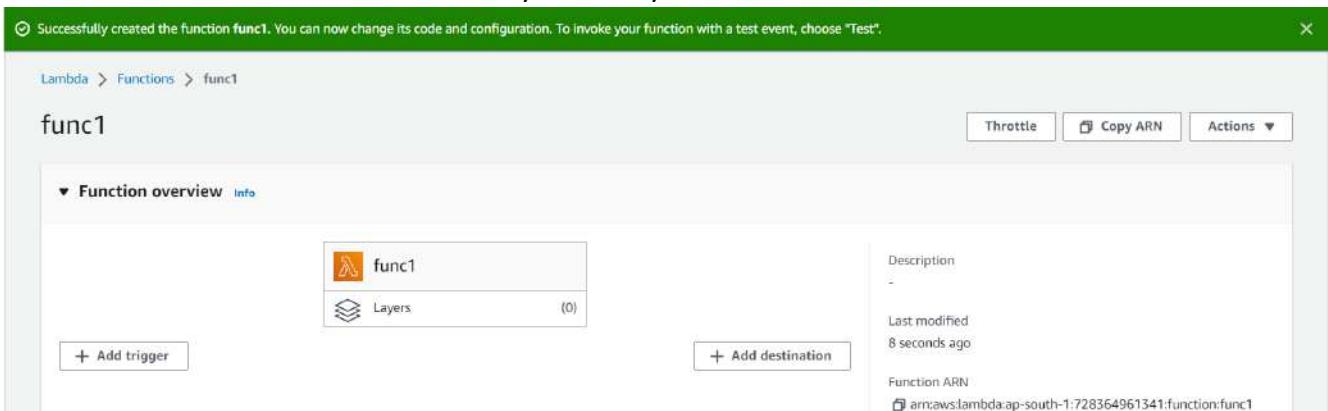
6. Choose Node.js as the Runtime. (No need to change architecture.)



7. Now click on the create function button



8. Now scroll-down to the code section of your newly created function.



9. Change the string in the code to be displayed.

```

1 export const handler = async(event) => {
2     // TODO implement
3     const response = {
4         statusCode: 200,
5         body: JSON.stringify('Hello from Lambda!'),
6     };
7     return response;
8 };
9

```

We changed it to this.....

```

1 export const handler = async(event) => {
2     // TODO implement
3     const response = {
4         statusCode: 200,
5         body: JSON.stringify('Hello from AWS !!!!!'),
6     };
7     return response;
8 };
9

```

10. Now go to File option and click on save to save the changes.

11. Now click on the Test button.

12. Select Create New Event. Then give a name. Then click on save.

**Configure test event**

A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

To invoke your function without saving an event, configure the JSON event, then choose Test.

**Test event action**

Create new event     Edit saved event

**Event name**

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

**Event sharing settings**

Private

This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)

Shareable

This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

Template - optional

hello-world

**Event JSON**

[Format JSON](#)

1  
2  
3

Cancel

[Save](#)

13. After saving. Now click on the Deploy button.

**Code source** [Info](#)

File Edit Find View Go Tools Window

[Test](#)

[Deploy](#)

14. After successful deployment a message will pop as mentioned below and the deploy button will be locked out, indicating that our Function has been successfully deployed.

 Successfully updated the function func1. X

15. Now go to the configuration tab.

Code Test Monitor **Configuration** Aliases Versions

**General configuration**

Triggers

Permissions

Destinations

Function URL

Environment variables

**General configuration** [Info](#)

Description

-

Memory

128 MB

Timeout

0 min 3 sec

SnapStart [Info](#)

None

16. Click on the Function URL option in the left side Bar.

**General configuration**

Triggers

Permissions

Destinations

Function URL

Environment variables

Tags

VPC

Monitoring and operations tools

17. Click on Create function URL.

[Function URL](#) [Info](#)

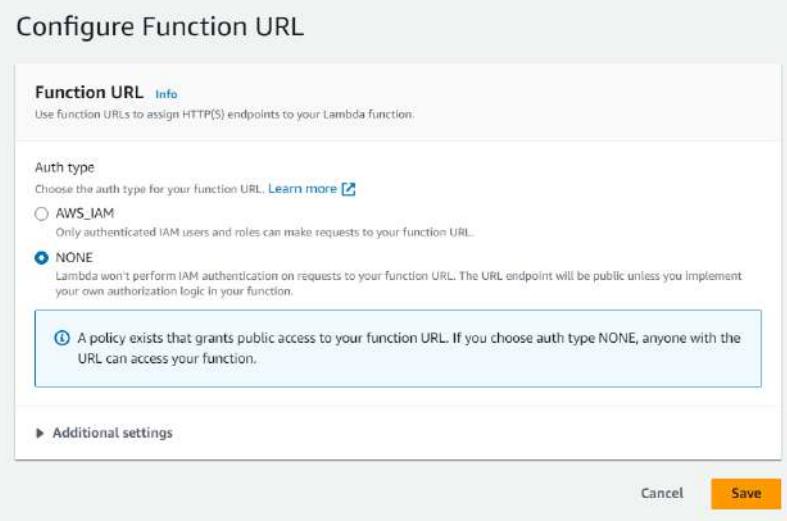
[Create function URL](#)

No Function URL

No Function URL is configured.

[Create function URL](#)

**18. Choose NONE and click on the save button.**



**19. Now copy the newly created Function URL and paste it in a different browser.**

Your changes have been saved.

Function overview

func1

Description

Last modified

6 minutes ago

Function ARN

arn:aws:lambda:ap-south-1:728364961341:function:func1

Function URL

https://ejkt6cr7wkr5hqrb2iavadyhe0thqki.lambda-url.ap-south-1.on.aws/

https://ejkt6cr7wkr5hqrb2iavadyhe0thqki.lambda-url.ap-south-1.on.aws

"Hello from AWS !!!!"

We have successfully Created a Serverless Computing service.

**To delete the Lambda Function, follow these steps:**

1. Click on the Actions button on the top right side.

Your changes have been saved.

Lambda > Functions > func1

func1

Throttle

Copy ARN

Actions

Function overview

func1

Description

Last modified

13 minutes ago

Function ARN

arn:aws:lambda:ap-south-1:728364961341:function:func1

Function URL

https://ejkt6cr7wkr5hqrb2iavadyhe0thqki.lambda-url.ap-south-1.on.aws/

2. Select the Delete function option and then click on delete button in the pop-up.

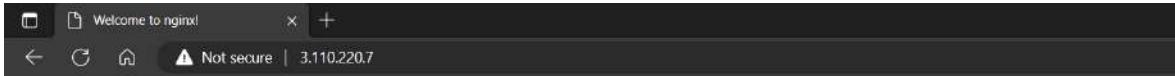
# ASSIGNMENT – 16

## Problem Statement: Manage Amazon DNS service and run a project using domain-name and URL

1. Sign-in to your AWS console.
2. Create an instance with custom security group and user data (Refer Ass10)
3. Click on the instance

The screenshot shows the AWS EC2 Instances page. At the top, there are buttons for 'Info', 'Connect', 'Instance state', 'Actions', and 'Launch instances'. Below is a search bar and a 'Clear filters' button. A table lists one instance: 'Name' (debserverX), 'Instance ID' (i-0da0c914af8583268), 'Instance state' (Running), 'Status check' (2/2 checks passed), 'Alarm status' (No alarms), 'Availability Zone' (ap-south-1a), and 'Public IPv4 DN' (ec2-3-110-220-). There are also 'Edit' and 'Delete' icons for the instance.

4. Copy the public IPv4 address and paste it in another browser.



### Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

5. Check if the project webpage is accessible by appending :4000 to your address.



Our EC2 instance works as intended. However, to access our webpage one always requires the public IPv4 address of our server instance which is very complicated/less accessible for end-users of our webpage/web application.

So, to make it easier for our end-users, we need to bind a domain name to the server instance. Now anyone can use the domain name and the URL to access our project.

6. Search Route 53 in the search bar of AWS console. Select the first result.

A screenshot of the AWS Route 53 dashboard. The search bar at the top contains 'route 53'. The left sidebar shows 'EC2 Dashboard' and 'Instances' under 'Services'. The main content area shows 'Search results for "route"'. The first result is 'Route 53' (Scalable DNS and Domain Name Registration), which is selected. Other results include 'Route 53 Resolver' (Resolve DNS queries in your Amazon VPC and on-premises network), 'Route 53 Application Recovery Controller' (Monitor application recovery readiness and manage failovers), and 'Amazon Location Service' (Securely and easily add location data to applications). On the right, there's a panel for 'Account attributes' showing 'Supported platforms' (VPC) and 'Default VPC' (vpc-0a33dec3fd6dc096). There are also sections for 'Settings', 'EBS encryption', 'Zones', 'EC2 Serial Console', 'Default credit specification', and 'Console experiments'. At the bottom, there's an 'Explore AWS' section.

We require a registered Domain name for this assignment. So, after obtaining one (free or paid) go to the Webpage of your Domain provider and log-in to your account where you can find all the details of your purchased Domains.

This may vary from site to site, so you will have to do this based on what site you are using.  
We (for now) will be using GoDaddy.com, because we have purchased a Domain from them.

7. After Reaching the Route 53 dashboard click on the Create Hosted Zone button.

## Route 53 Dashboard [Info](#)

**DNS management**  
A hosted zone tells Route 53 how to respond to DNS queries for a domain such as example.com.

**Traffic management**  
A visual tool that lets you easily create policies for multiple endpoints in complex configurations.

**Availability monitoring**  
Health checks monitor your applications and web resources, and direct DNS queries to healthy resources.

**Domain registration**  
A domain is the name, such as example.com, that your users use to access your application.

**Create hosted zone** | **Create policy**

**Create health check** | **Register domain**

**Register domain**

Find and register an available domain, or transfer your existing domains to Route 53.

Enter a domain name

Each label (each part between dots) can be up to 63 characters long and must start with a-z or 0-9. Maximum length: 255 characters, including dots. Valid characters: a-z, 0-9, and - (hyphen).

**Check**

Alternatively, you can go to hosted zones from the left-side bar and then select create hosted zone option.

- Now, copy your Domain name from your Domain providers website. Here we used GoDaddy.com. Paste the domain name in the given field in Hosted Zone configuration page.

1 domain

**Domain Name** ↑

[debrup.co.in](#) ...

**Create hosted zone [Info](#)**

**Hosted zone configuration**  
A hosted zone is a container that holds information about how you want to route traffic for a domain, such as example.com, and its subdomains.

**Domain name [Info](#)**  
This is the name of the domain that you want to route traffic for.

Valid characters: a-z, 0-9, ! " # \$ % & ' ( ) \* + , - / ; : < = > ? @ [ \ ] ^ \_ [ ] , ~

**Description - optional [Info](#)**  
This value lets you distinguish hosted zones that have the same name.

The description can have up to 256 characters. 0/256

**Type [Info](#)**  
The type indicates whether you want to route traffic on the internet or in an Amazon VPC.

**Public hosted zone**  
A public hosted zone determines how traffic is routed on the internet.

**Private hosted zone**  
A private hosted zone determines how traffic is routed within an Amazon VPC.

- Now scroll-down and click on the Create Hosted Zone button.

- Now click on the Create a record button.

Route 53 > Hosted zones > debrup.co.in

**Public debrup.co.in [Info](#)**

**Hosted zone details**

**Records (2)** [Info](#)

Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

Record ...	Type	Routing policy	Alias	Value/Route traffic to	TTL (\$...)	Health ...
debrup.co.in	NS	Simple	-	No ns-758.awsdns-30.net. ns-1483.awsdns-57.org. ns-2015.awsdns-59.co.uk. ns-327.awsdns-40.com.	172800	-
debrup.co.in	SOA	Simple	-	No ns-758.awsdns-30.net. awsd...	900	-

**Delete zone** | **Test record** | **Configure query logging**

**Edit hosted zone**

**Records (2)** [Info](#)

**Import zone file** | **Create record**

**Filter records by property or value**

- Follow these Steps:

- Do not give any name. Keep the record name blank.
- Keep record type as it is. No change required.
- Under the value, copy and paste your server instance public IPv4 address which you want to route to using your DNS.
- Then click on create records button

The screenshot shows the 'Quick create record' interface. It displays a single record entry for 'Record 1'. The 'Record name' field contains 'subdomain' and the 'Value' field contains '192.0.2.235'. The 'Record type' is set to 'A' (Routestraffic to an IPv4 address and some AWS resources). The 'TTL (seconds)' is set to 500. The 'Routing policy' is 'Simple routing'. At the bottom right, there is a prominent orange 'Create records' button.

## 12. Now again click on the Create Record button like the previous step.

- But this time give the record name as → **www**
- Select Record type as **CNAME**
- In the text box under value, write the full domain-name there. (For example: example.com)
- Click on create records button

The screenshot shows the 'Quick create record' interface. It displays a single record entry for 'Record 1'. The 'Record name' field contains 'www' and the 'Value' field contains 'debrup.co.in'. The 'Record type' is set to 'CNAME' (Routestraffic to another domain name and to some AWS resources). The 'TTL (seconds)' is set to 500. The 'Routing policy' is 'Simple routing'. At the bottom right, there is a prominent orange 'Create records' button.

## 13. Now select the record with type nameserver (NS).

The screenshot shows the 'Hosted zone details' page for the 'debrup.co.in' zone. On the left, a table lists existing records: one SOA record for 'debrup.co.in' and three NS records for 'debrup.co.in'. On the right, a detailed view of the selected NS record for 'debrup.co.in' is shown. The record has a TTL of 172800 seconds and four listed values: ns-758.awsdns-30.net., ns-1483.awsdns-57.org, ns-2015.awsdns-59.co.uk, and ns-327.awsdns-40.com.

The values seen on the right-hand side are required for the next steps.

## 14. Now go to your Domain providers webpage. Go to your purchased Domains settings.

< Domain Portfolio

debrup.co.in

Overview

DNS

Products

15. Click on DNS section. (This may vary from provider to provider)

16. Click on the nameservers option.

< Domain Portfolio

debrup.co.in

Overview

**DNS**

Products

DNS Records

Forwarding

**Nameservers**

Premium DNS

Hostnames

17. Click on the Change nameservers and add here all the values opened in the Route 53 page.

- a. Select use my own nameservers option.
- b. Add nameservers.
- c. Then click on the save button.

Nameservers determine where your DNS is hosted and where you add, edit or delete your DNS records.

Using default nameservers

Change Nameservers

Nameservers ⓘ

X

Edit nameservers

Choose nameservers for debrup.co.in

GoDaddy Nameservers (recommended)

I'll use my own nameservers

ns-758.awsdns-30.net

ns-1483.awsdns-57.org

ns-2015.awsdns-59.co.uk

ns-327.awsdns-40.com

Add Nameserver

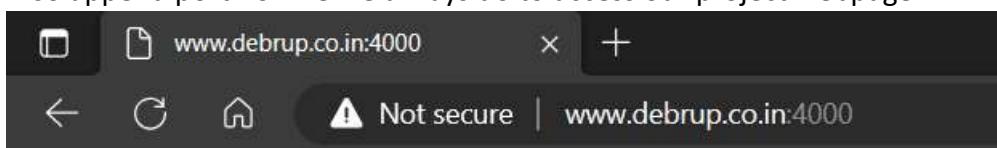
Save Cancel

18. Wait for few minutes.

19. Now try searching from any browser using your domain name with www.

(For example: [www.example.com](http://www.example.com))

20. Also append port no. like we always do to access our project webpage.



Hello. My Name is Spider-Man!!! Nice to meet You!!!

We have successfully run our project using our custom domain-name and URL.