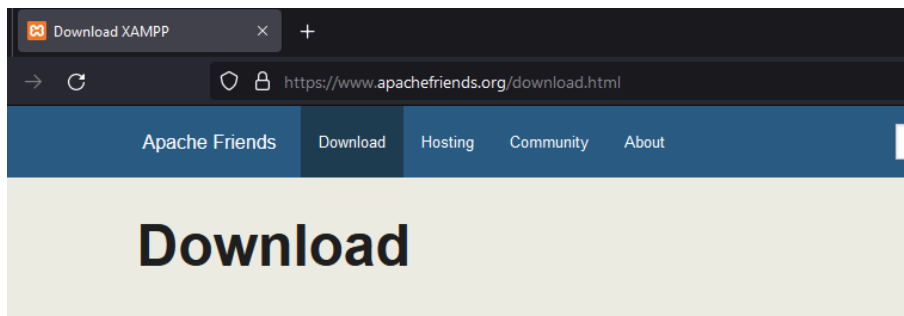


Experiment No. 1(A)

Aim: To develop a website and host it on local machine or virtual machine and Hosting a static website using Amazon S3 Bucket

1. To develop a website and host it on your local machine on a VM using XAMPP

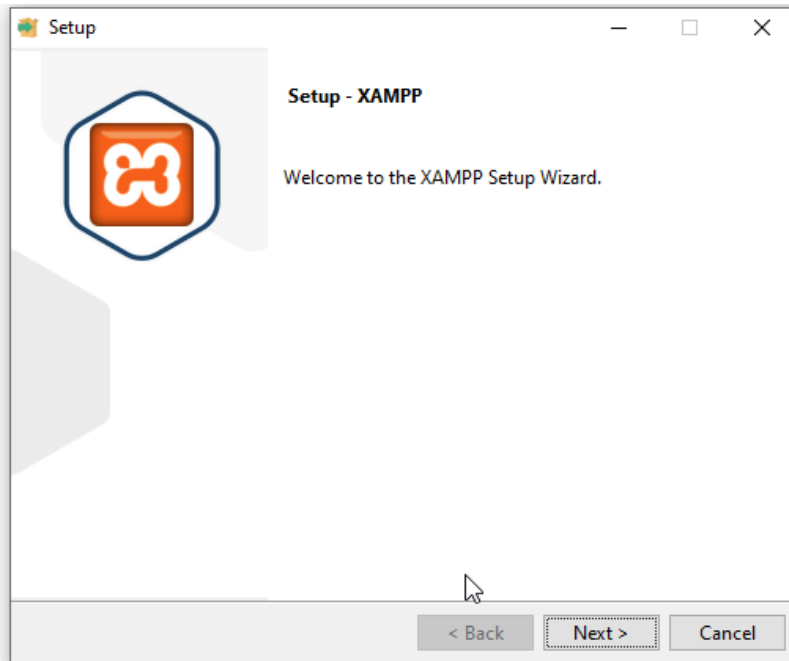
- 1) Goto official website of XAMPP and download the software as per your OS



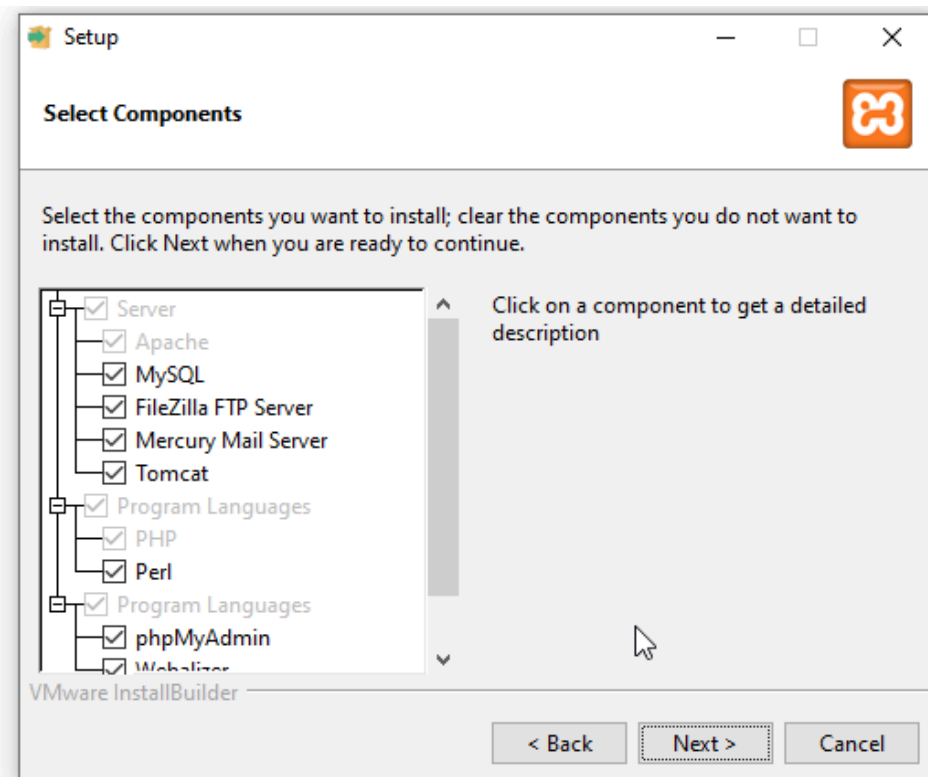
XAMPP is an easy to install Apache distribution containing MariaDB, PHP, and Perl. Just download and start the installer. It's that easy. Installers created using [InstallBuilder](#).

XAMPP for Windows 8.0.30, 8.1.25 & 8.2.12					
Version	Checksum		Size		
8.0.30 / PHP 8.0.30	What's Included?	md5 sha1	Download (64 bit)	144 Mb	
8.1.25 / PHP 8.1.25	What's Included?	md5 sha1	Download (64 bit)	148 Mb	
8.2.12 / PHP 8.2.12	What's Included?	md5 sha1	Download (64 bit)	149 Mb	

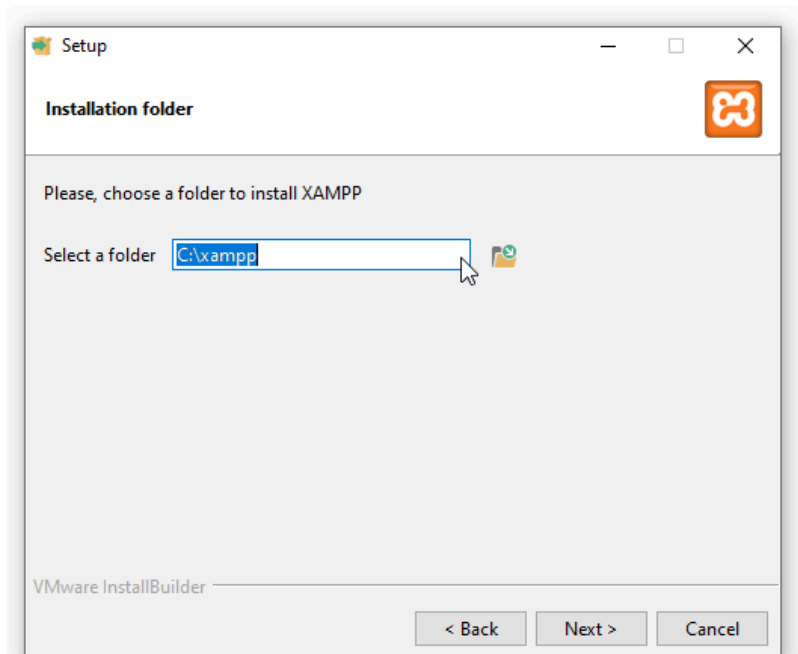
- 2) After Downloading, open it for installation, Click on next



3) Select components to install



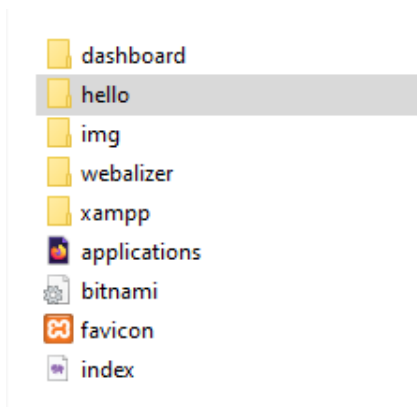
4) Select installation location, and confirm it



5) After installation, goto folder where XAMPP is installed, go to htdocs

name	last modified
anonymous	8/11/2024 2:29 AM
apache	8/11/2024 2:29 AM
cgi-bin	8/11/2024 2:33 AM
contrib	8/11/2024 2:29 AM
FileZillaFTP	8/11/2024 2:33 AM
htdocs	8/11/2024 2:29 AM
img	8/11/2024 2:29 AM
install	8/11/2024 2:33 AM
licenses	8/11/2024 2:29 AM
locale	8/11/2024 2:29 AM

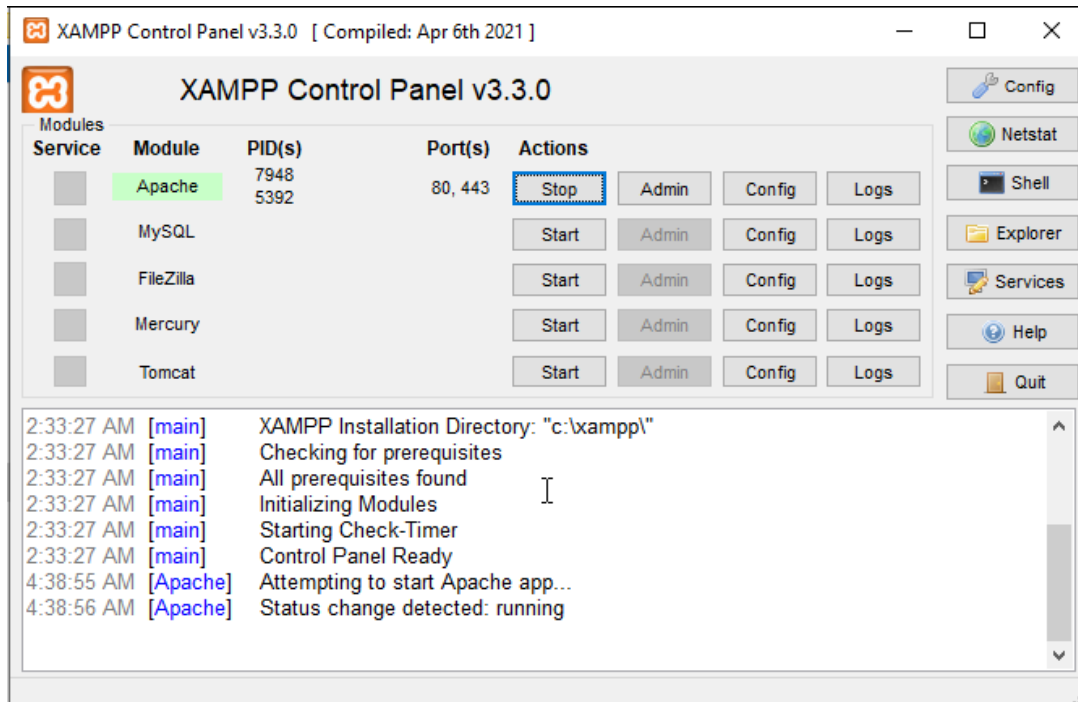
6) Create new folder named anything we want our URI to be.



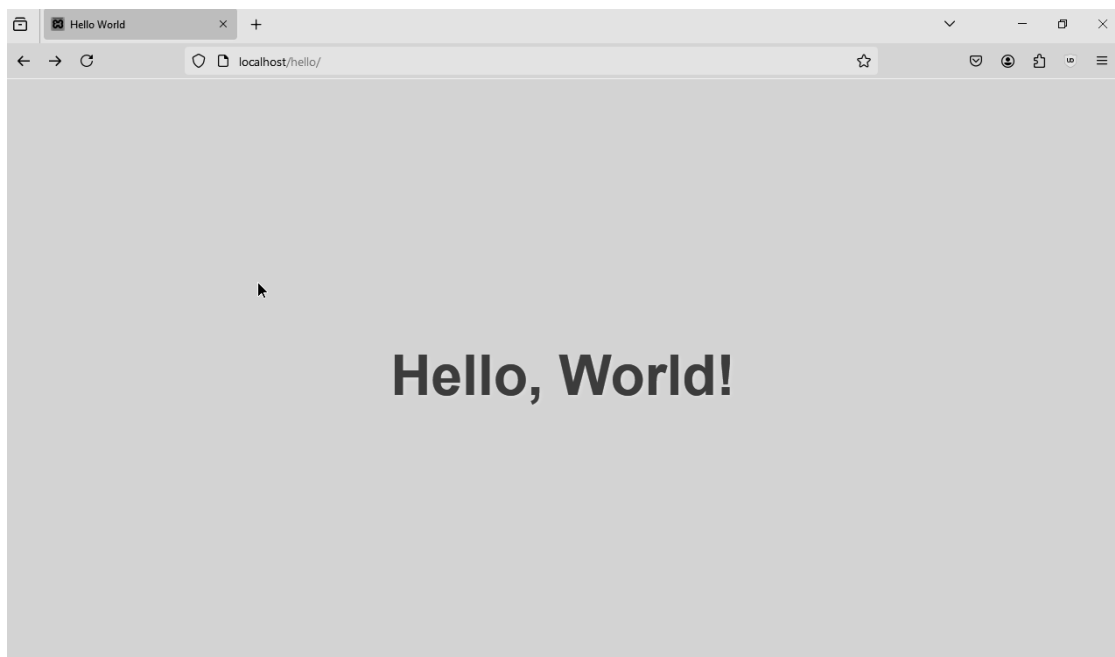
7) Create a file index.php and paste the following code.

```
*hello.php - Notepad
File Edit Format View Help
<title>Hello World</title>
<style>
  body {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background-color: #282c34;
    margin: 0;
    font-family: Arial, sans-serif;
    color: white;
  }
  h1 {
    font-size: 4rem;
    color: #61dafb;
    text-shadow: 2px 2px 5px rgba(0, 0, 0, 0.3);
  }
</style>
</head>
<body>
  <h1>Hello, World!</h1>
</body>
</html>";
?>
Ln 27, Col 27 100% Windows (CRLF) UTF-8
```

8) Now go to XAMPP control panel and start Apache server

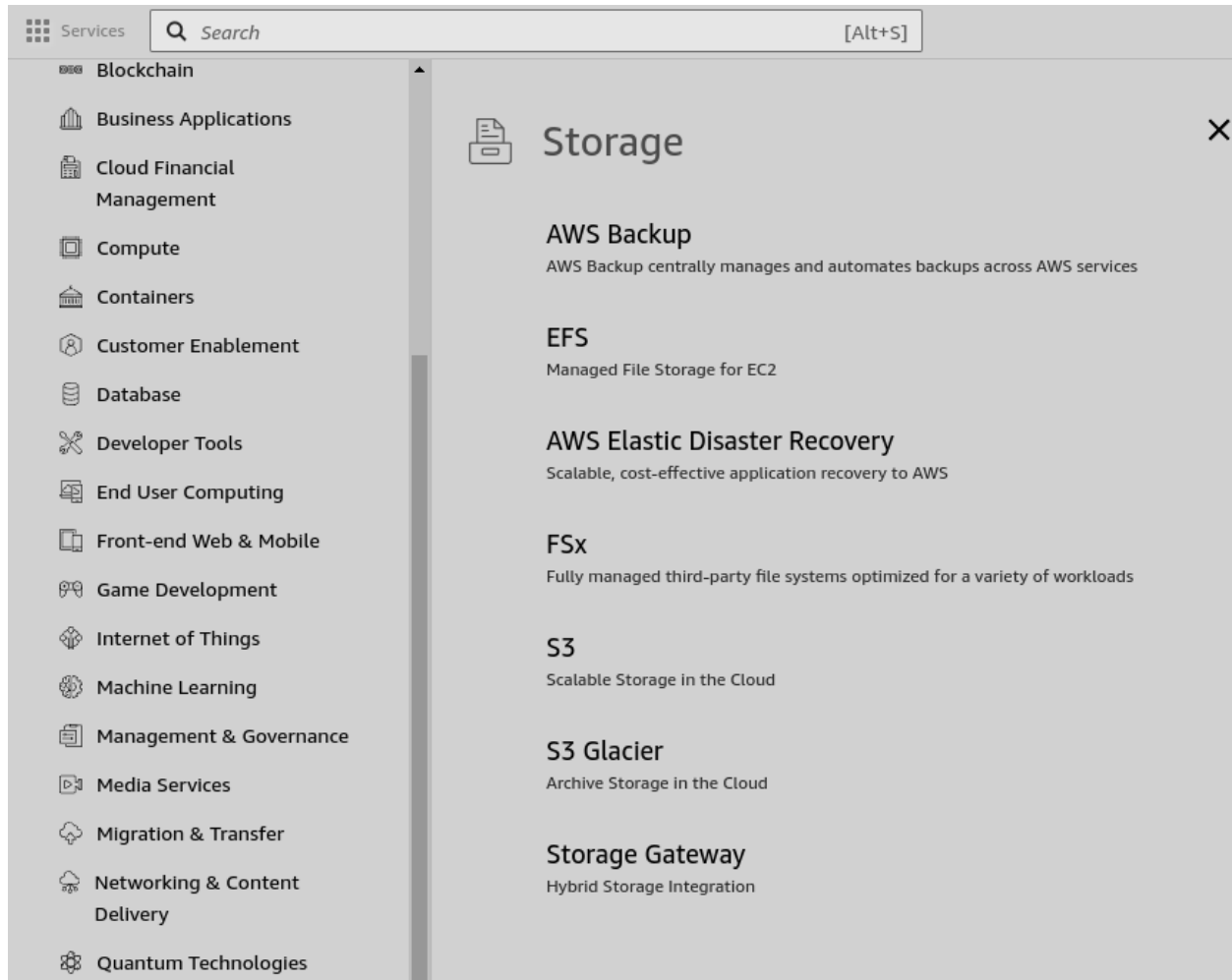


- 9) Now open browser and enter http://localhost/folder_name. This will open webpage locally on our machine



2. Hosting a static website on Amazon S3

1) Go to services then under Storage, select S3



2) Create a new bucket



3) Enter Name of the bucket

General configuration

AWS Region
Asia Pacific (Mumbai) ap-south-1

Bucket name Info

host-web-alok

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.

Choose bucket

Format: s3://bucket/prefix

- 4) Disable “Block all public access” to make the contents of bucket publicly accessible


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**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

- 5) Leaving the remaining configuration to it's default, click on Create Bucket

 After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

Cancel Create bucket

6) A new bucket will be created

General purpose buckets (1) [info](#) [All AWS Regions](#)

Refresh Copy ARN Empty Delete Create bucket

Buckets are containers for data stored in S3.

☐

Name

▲

AWS Region

▼

IAM Access Analyzer

Creation date

▼

<input type="radio"/>	host-web-alok	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1	August 7, 2024, 20:26:35 (UTC+05:30)
-----------------------	---------------	----------------------------------	--	--------------------------------------

7) Now to host a static web page, open the bucket we created, goto Properties, edit “Static Web hosting settings”

Static website hosting Edit

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

Static website hosting
Disabled

8) Enable Static web 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

Cancel Save changes

9) After enabling, we need to enter the names of files two files, one for main file that would be displayed when page loads and the other when some error occurs. Name those files and save it

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](#)

i 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](#)

Index document
Specify the home or default page of the website.

index.html

Error document - optional
This is returned when an error occurs.

error.html

Redirection rules – optional

10) After naming, go to objects and upload the files you mentioned in the static web hosting settings page

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 folder**.

Files and folders (2 Total, 1.3 KB) Remove Add files Add folder

All files and folders in this table will be uploaded.

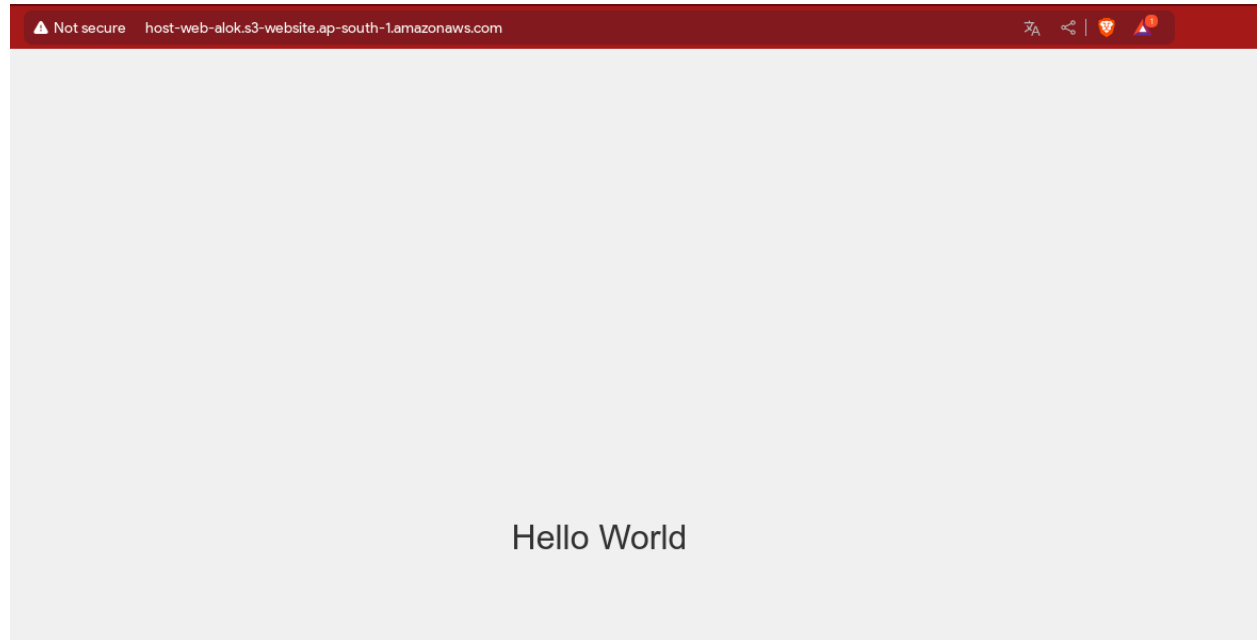
<input type="checkbox"/>	Name	Folder
<input type="checkbox"/>	index.html	-
<input type="checkbox"/>	error.html	-

11) Now to make sure our page is accessible to public, we need to change bucket policy. To change it, go to permissions and edit Bucket policy.

Change resource arn as per your bucket name

```
Policy
1
2 {
3   "Version": "2012-10-17",
4   "Statement": [
5     {
6       "Sid": "PublicReadGetObject",
7       "Effect": "Allow",
8       "Principal": {
9         "AWS": "*"
10      },
11      "Action": "s3:GetObject",
12      "Resource": "arn:aws:s3:::host-web-alok/*"
13    }
14  ]
15 }
```

12) Now we can go to home page of our bucket and our static page will be loaded

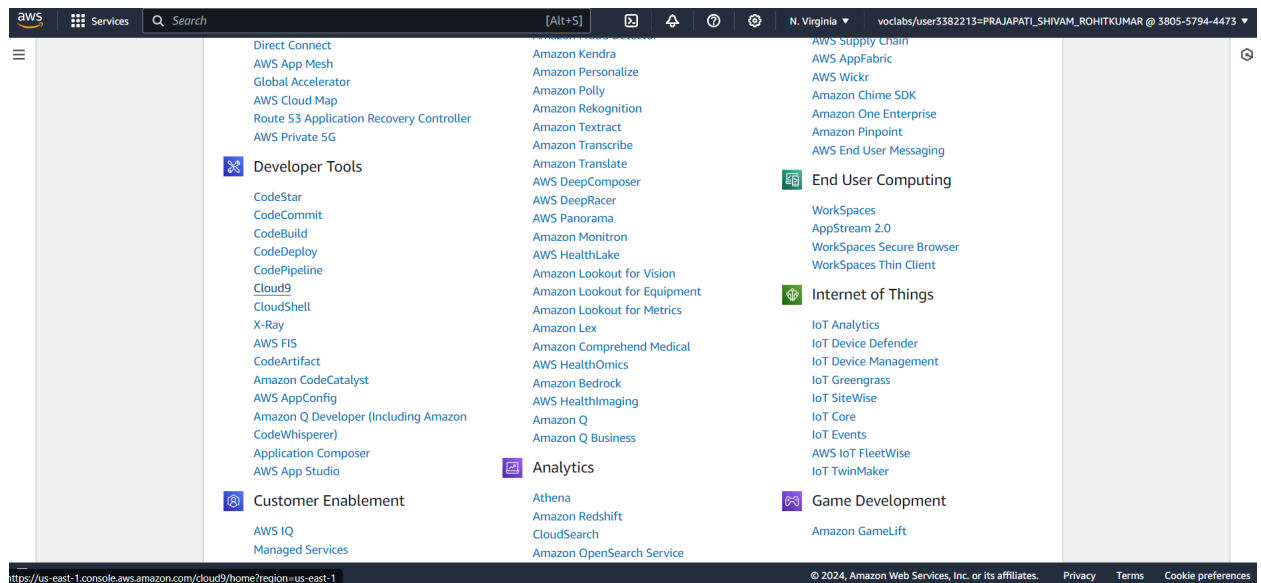


Experiment No: 1(B)

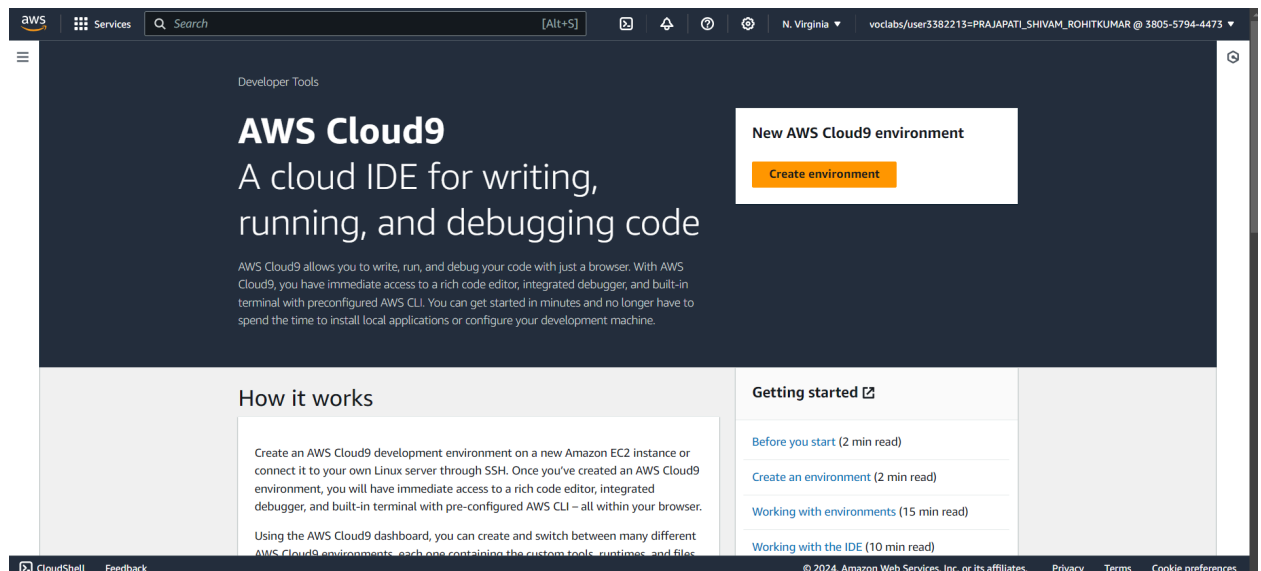
Aim: To understand the benefits of Cloud Infrastructure and Setup AWS Cloud9 IDE, Launch AWS Cloud9 IDE and Perform Collaboration Demonstration.

Step 1: Set up a Cloud9 environment.

- 1) Go to Cloud9 services under developers tool in All services



- 2) Click on create environment



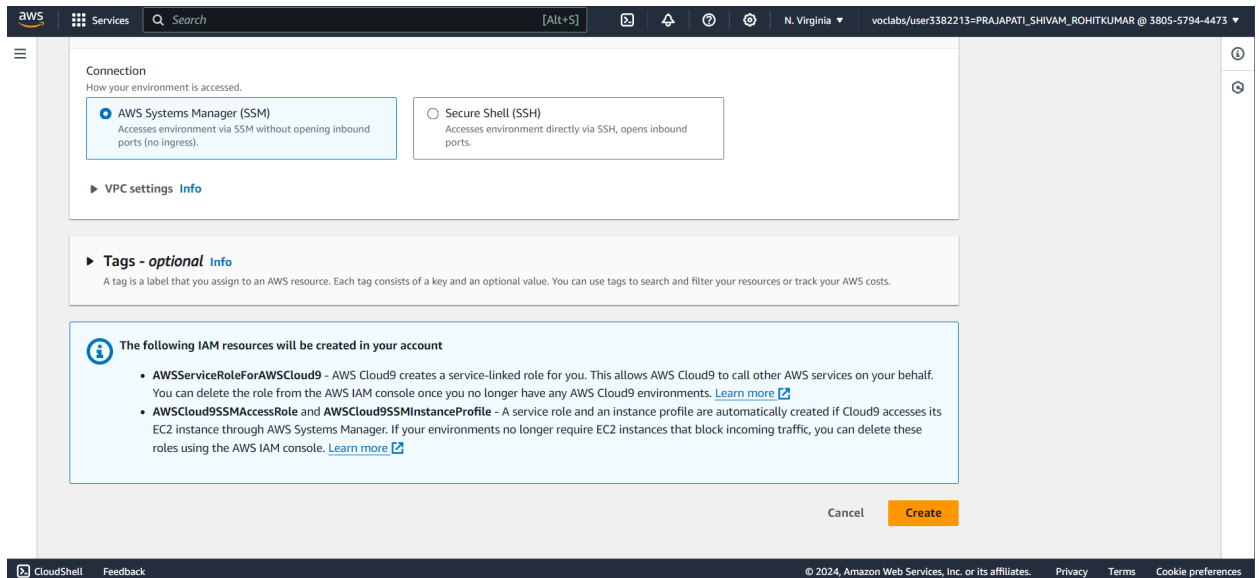
3) Give the name to your Environment ,keeping the other settings as default like environment type should be New EC2 instance

The screenshot shows the 'Create environment' page in the AWS Cloud9 console. The 'Name' field is filled with 'WebAppIDE'. The 'Description' field is empty. The 'Environment type' is set to 'New EC2 instance'. The 'New EC2 instance' section is expanded, showing the 'Instance type' as 't2.micro (1 GiB RAM + 1 vCPU)'. The 'Platform' is set to 'Amazon Linux 2023'. The 'Timeout' is set to '30 minutes'.

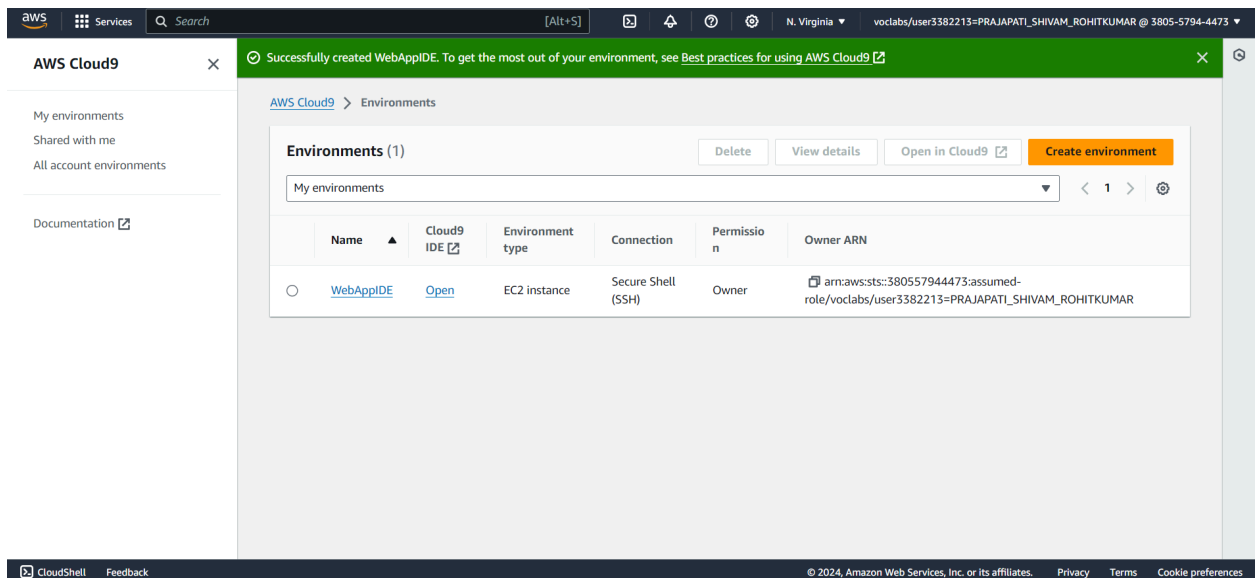
4) Select the correct platform type as shown below and keep the others details as default like instance type as t2.micro which gives the user 1GB RAM + 1 Virtual CPU

The screenshot shows the 'New EC2 instance' page in the AWS Cloud9 console. The 'Instance type' is set to 't2.micro (1 GiB RAM + 1 vCPU)'. The 'Platform' is set to 'Amazon Linux 2023'. The 'Timeout' is set to '30 minutes'.

5) Click on SSH under connection type in network settings if we go for AWS Manager(SSM) then it won't allow to create an environment then click on Create

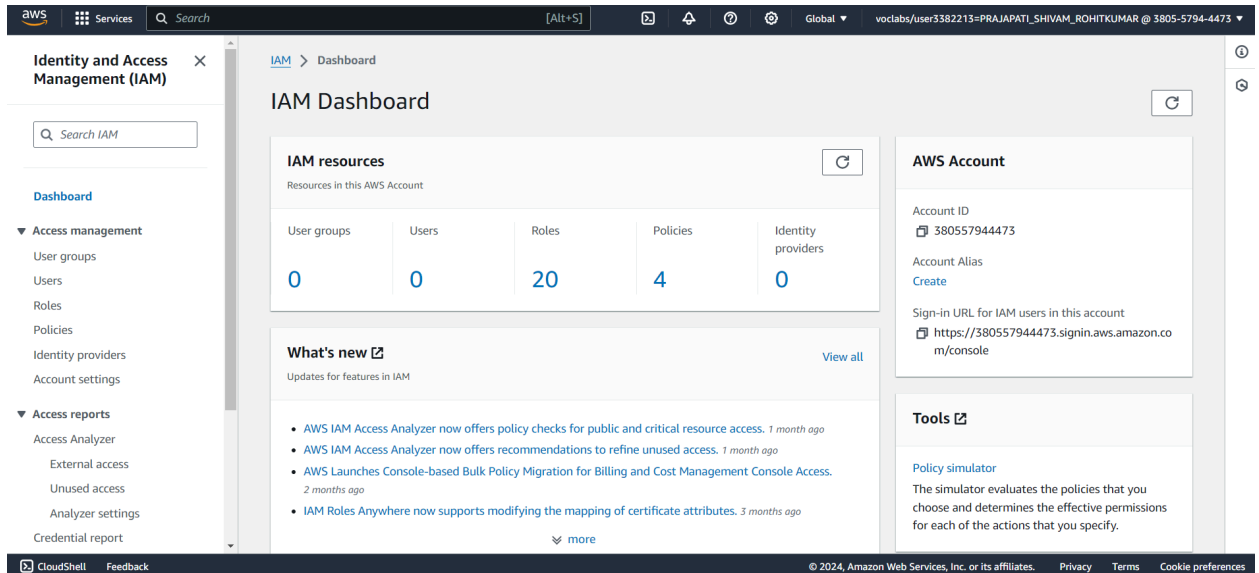


6) Successfully created the environment so now click on open

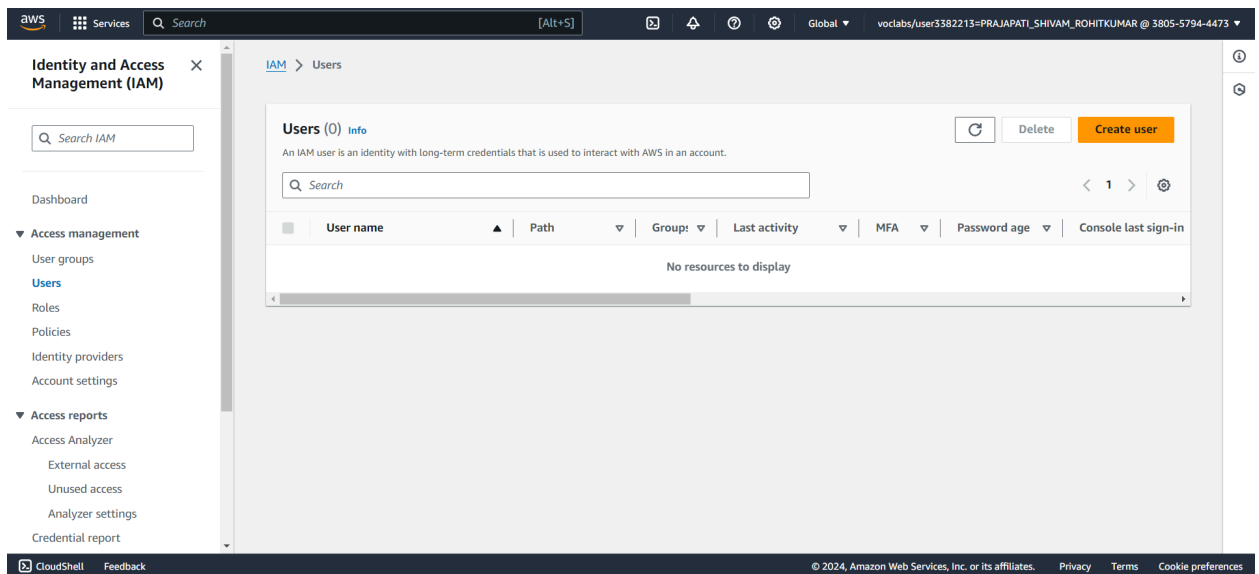


Step 2: Creating IAM user.

1) Search IAM on the services search bar and open it. Click on Create User



2) Click on the create user



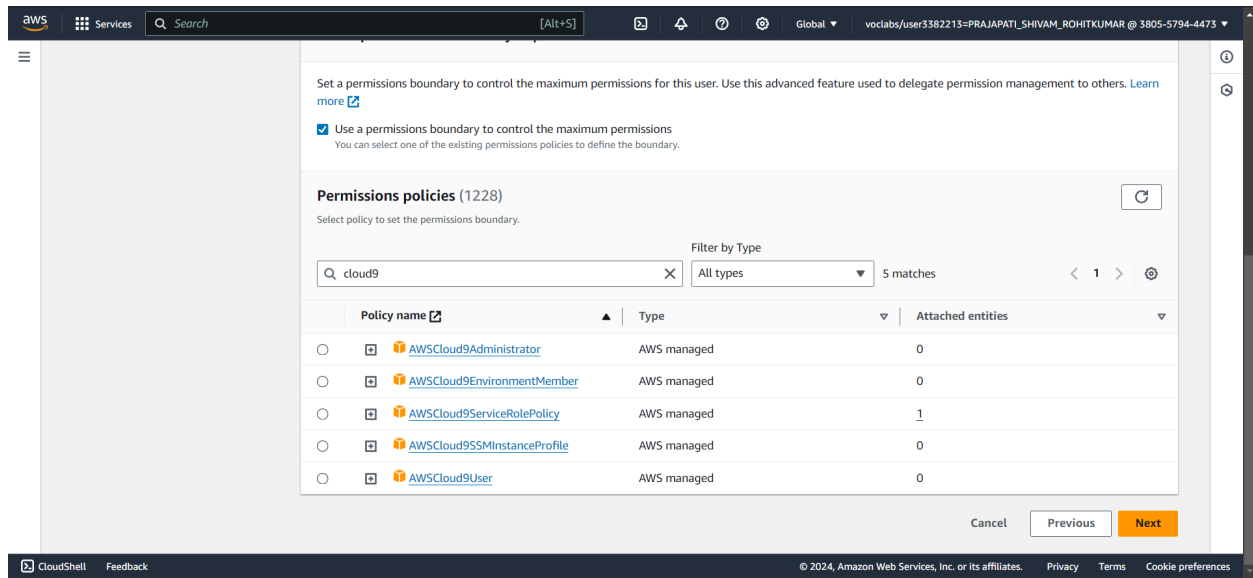
3) Write the name of the user you want to add and click on next

The screenshot shows the AWS IAM console's 'Create user' wizard, specifically the 'Specify user details' step. The left sidebar indicates the current step is Step 1, with Step 2 'Set permissions' and Step 3 'Review and create' also visible. The main content area is titled 'Specify user details' and contains a 'User details' section. In this section, the 'User name' field is populated with 'apsiti'. Below the field, a note states: 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and +, -, @, _ (hyphen)'. There is an unchecked checkbox for 'Provide user access to the AWS Management Console - optional' with a sub-note: 'If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.' A blue information box contains a tip: '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'. At the bottom right of the form are 'Cancel' and 'Next' buttons. The footer of the console shows 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. or its affiliates.

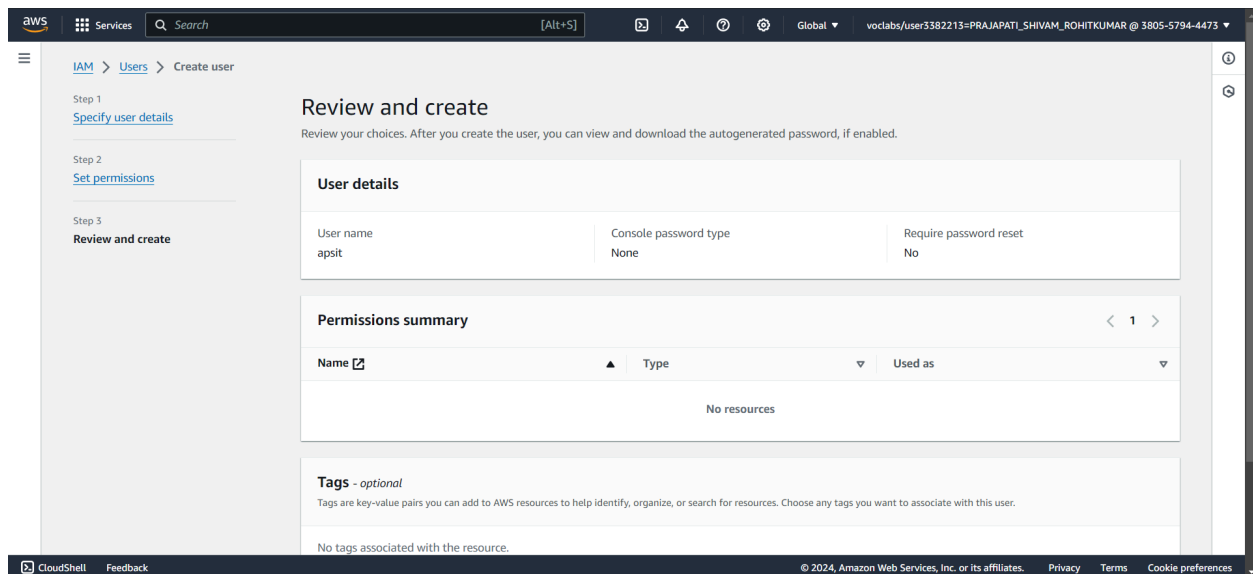
4) Select add User to Group. If there are no user groups on your accounts, you will have to create one. Click on Create Group. Click on the drop down menu of the set permissions boundary

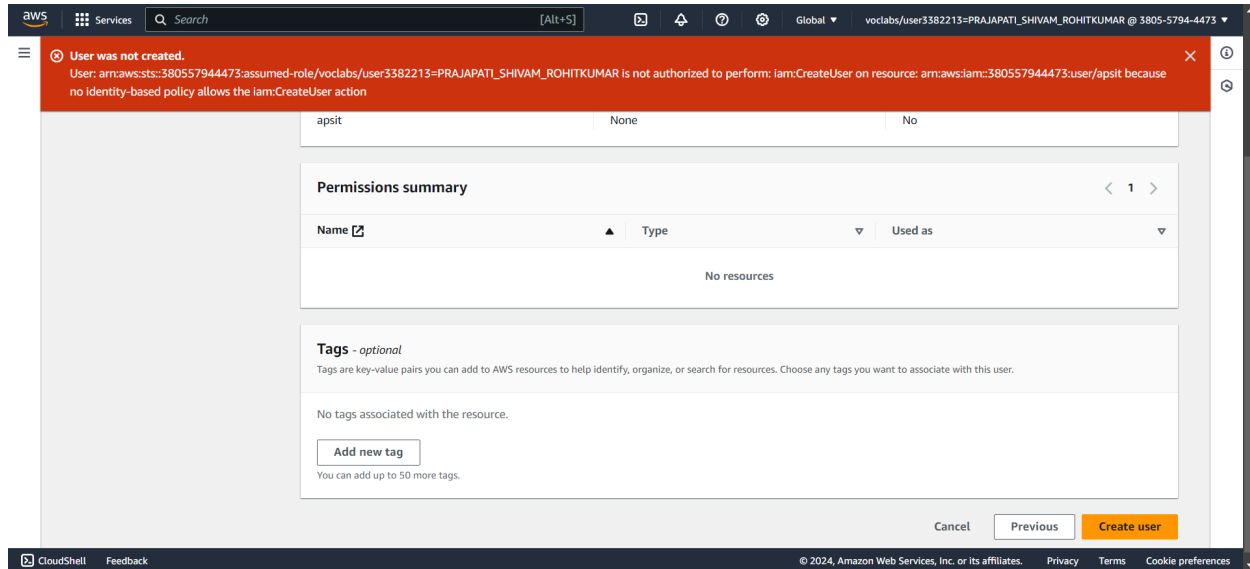
The screenshot shows the AWS IAM console's 'Set permissions' step of the 'Create user' wizard. The left sidebar shows Step 1 'Specify user details' as completed and Step 2 'Set permissions' as the current step. The main content area is titled 'Set permissions' and includes a sub-header 'Permissions options'. Three radio buttons are present: 'Add user to group' (which is selected), 'Copy permissions', and 'Attach policies directly'. The 'Add user to group' option has a description: 'Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.' The 'Copy permissions' option has a description: 'Copy all group memberships, attached managed policies, and inline policies from an existing user.' The 'Attach policies directly' option has a description: '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.' Below these options is a blue information box titled 'Get started with groups' with the text: 'Create a group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. Learn more'. A 'Create group' button is located to the right of this box. At the bottom, there is a section for 'Set permissions boundary - optional' with a dropdown menu. At the bottom right of the form are 'Cancel', 'Previous', and 'Next' buttons. The footer of the console shows 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. or its affiliates.

5) Click on the checkbox and search for cloud9 under permissions policies, click on next



6) Scroll down and click on create user

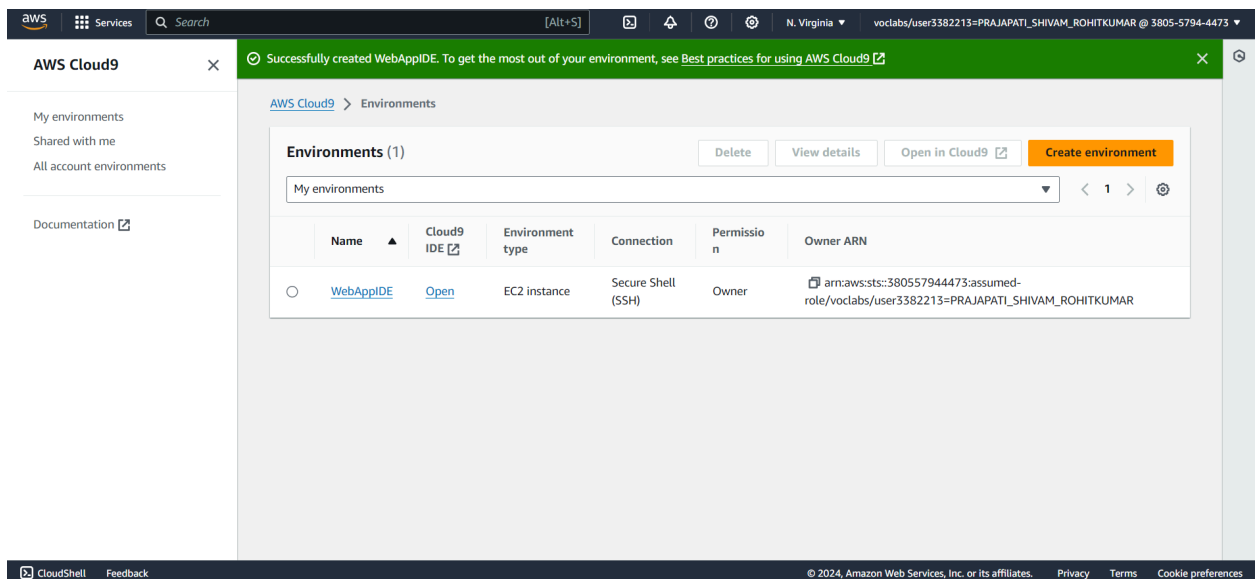




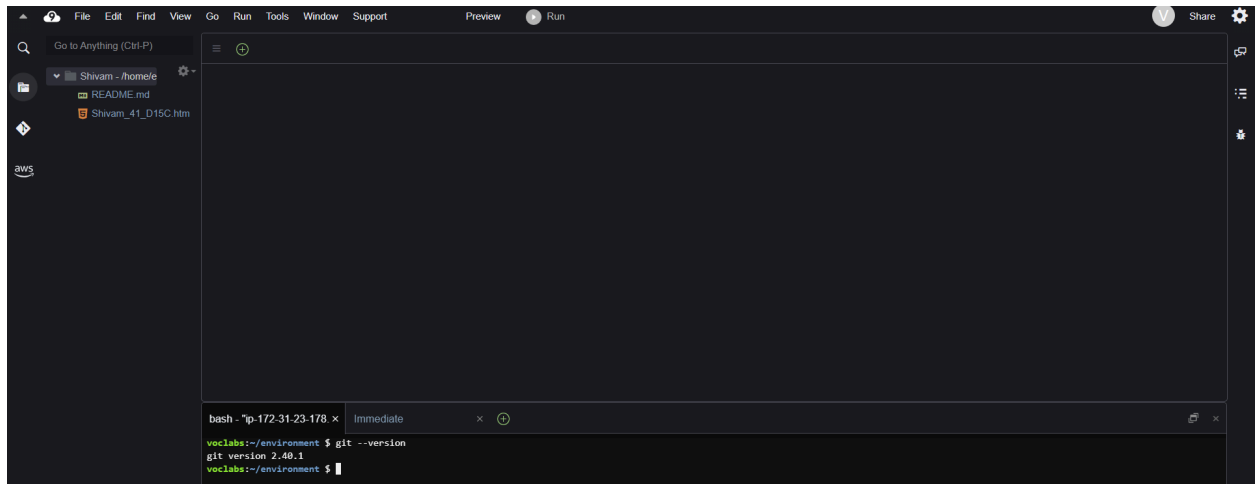
When we go to add user to a group, the AWS Academy account throws an error as we do not have the permissions to create a group. So we have to use our personal AWS account for this part.

Step 3: Working on Cloud9 IDE

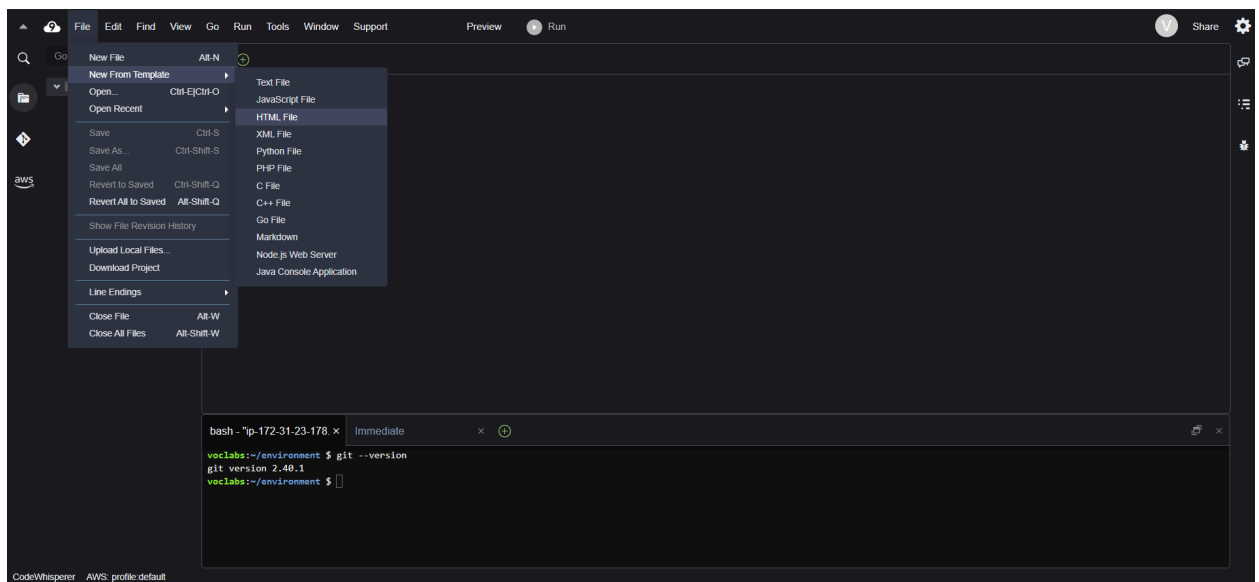
1) Go to Cloud9 services. Click on Open under Cloud9 IDE



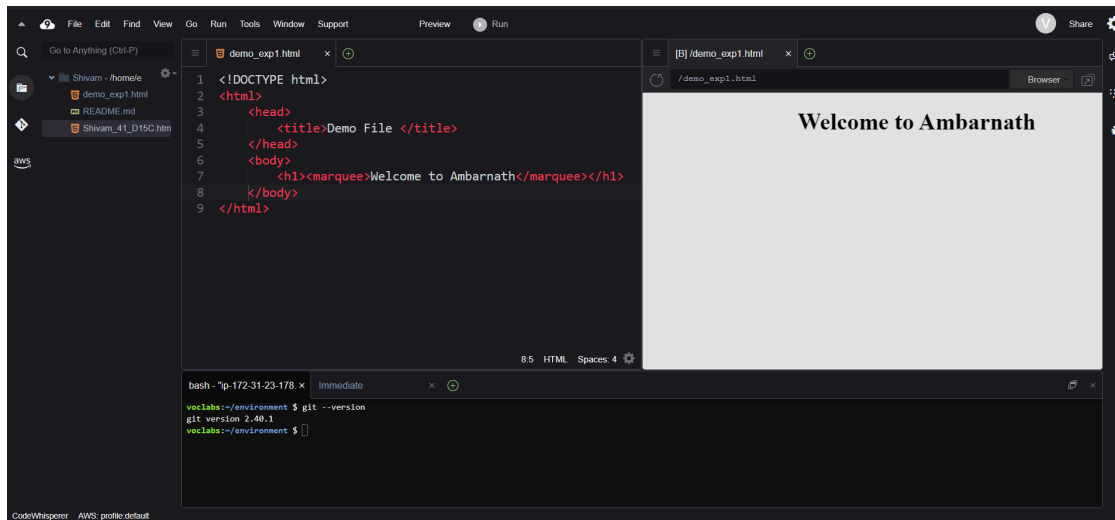
2) This is the Cloud9 IDE interface. The major part of the screen is the coding IDE. There is a command console just below it. For example, the command `git --version` is run.



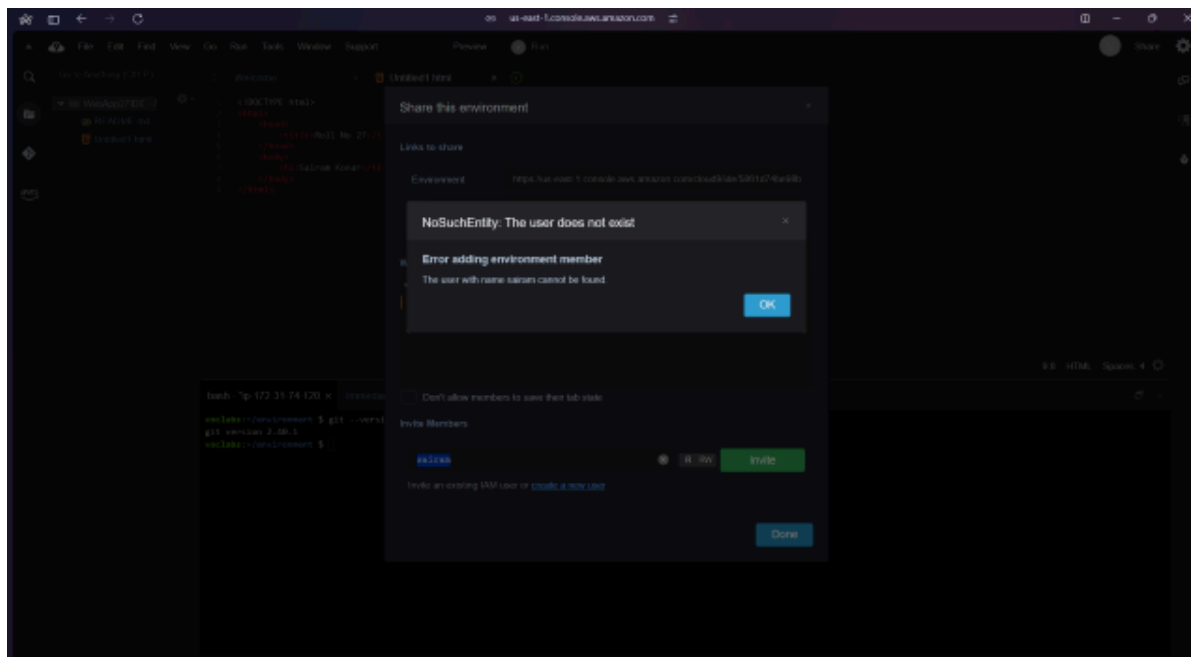
3) To add a file, click on file. For this experiment, we are to add an HTML file. So go to File → New From Template → HTML file. This gives a basic HTML template on the coding IDE



4) Make a basic website on the HTML template and save it.



After saving, on the toolbar towards the far right, click on Share. Then put the username that you had put during creating IAM user.



Here, it gives an error as Cloud9 was created on the academy account where creating an IAM group is not available, meanwhile on the personal account, the services of Cloud9 have been deprecated. So currently, it is not possible to integrate the cloud9 and IAM parts of the experiment.