

Karan(2301410015)

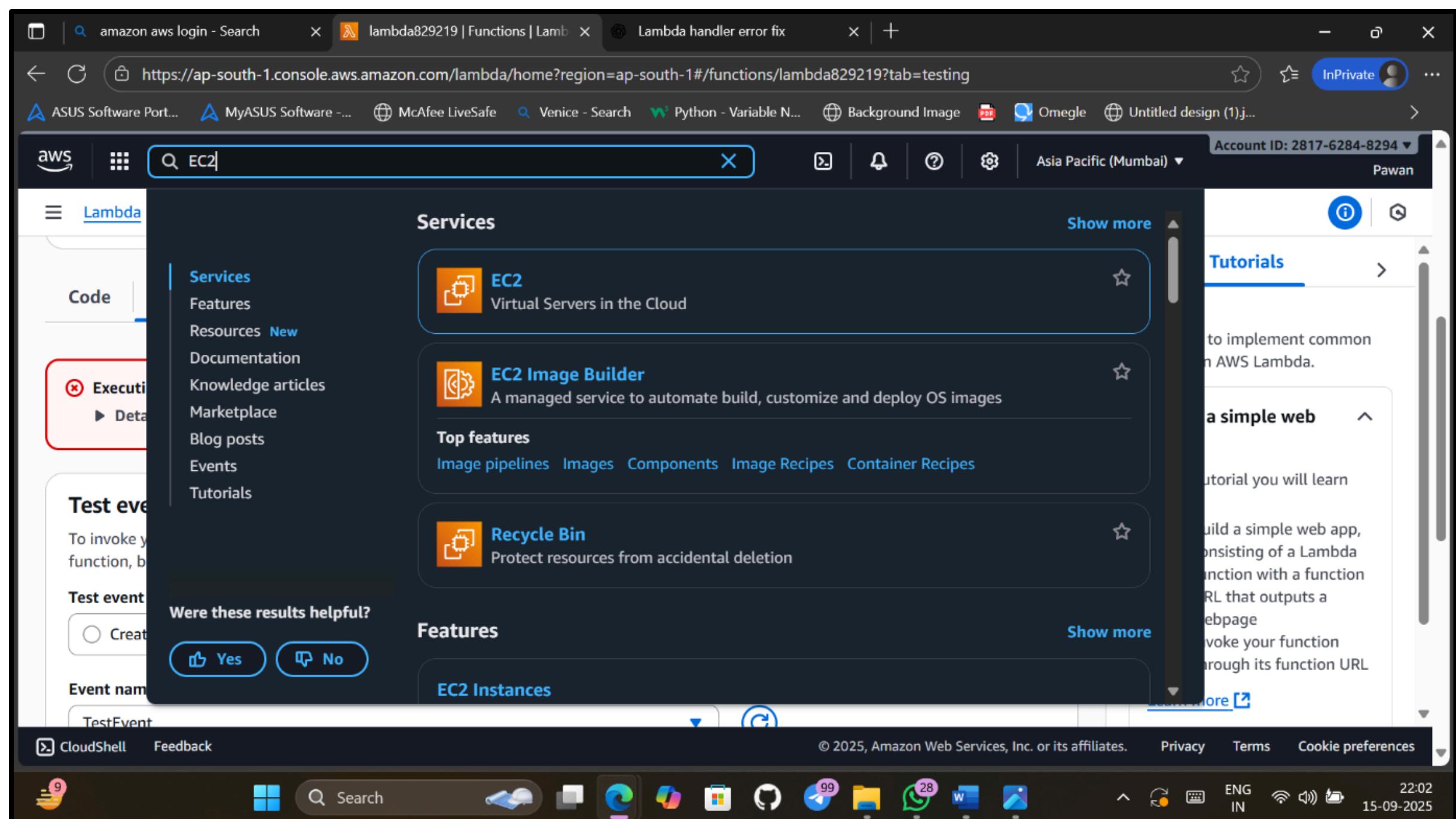
B.Tech CSE (Cyber security)

AWS CLOUD LAB ASSIGNMENT

1. Create a EC2 instance in Amazon AWS

Step -1 Navigate to the EC2 Dashboard

Once signed in, use the search bar at the top to find and select EC2. This will take you to the EC2 dashboard, where you can manage your virtual servers.



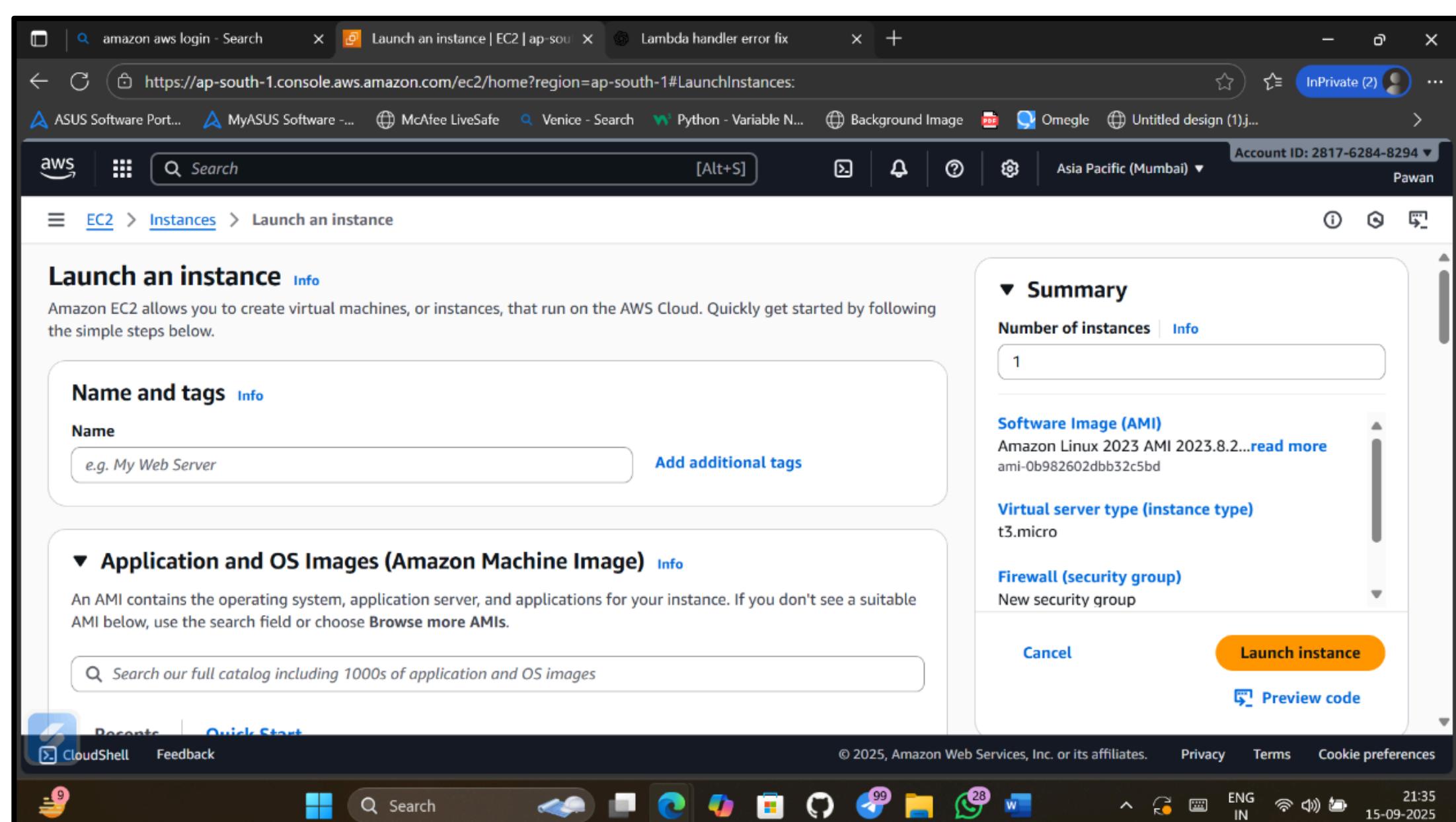
Step -2 Launch an Instance

On the EC2 dashboard, click the "Launch instance" button. This will start the wizard for creating a new EC2 instance.



Step-3 Choose an Amazon Machine Image (AMI)

Select an AMI, which is a template for your instance's operating system and software. For a basic setup, I choose a free tier eligible AMI like Amazon Linux 2023 AMI or Ubuntu Server 22.04 LTS.



Step-4 Choose an Instance Type

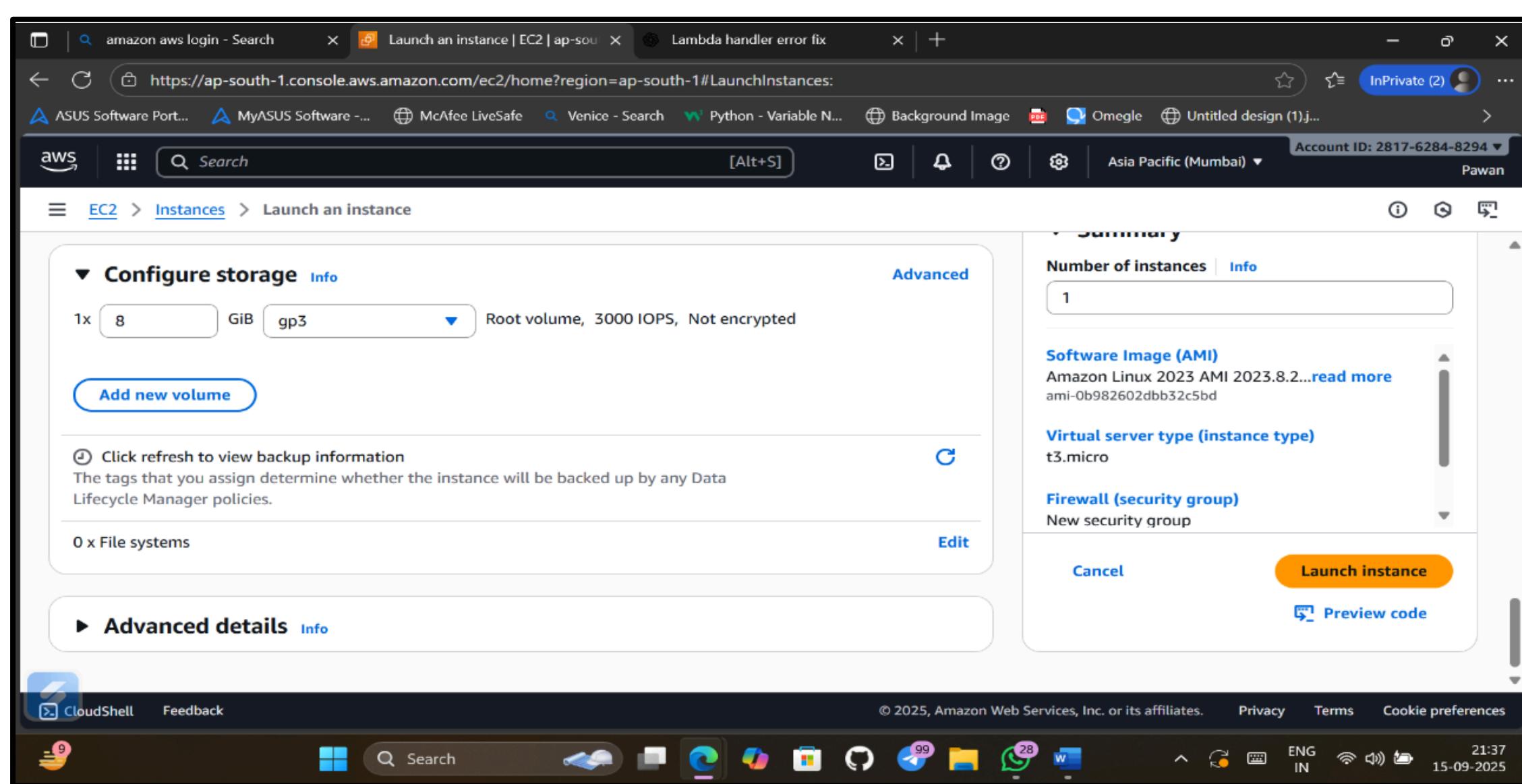
An instance type determines the CPU, memory, storage, and networking capacity of your instance. For most basic use cases, the t2.micro or t3.micro instance types are a great choice as they are free tier eligible.

Step-5 Configure Instance Details

On this page, you can configure network settings, assign an IAM role, and more. For a simple setup, you can often leave most of the settings at their defaults.

Step-6 Add Storage

Here you can specify the size and type of the root volume (the main disk) for your instance. The default is usually sufficient, but you can increase it if you need more space. Free tier includes up to 30 GB of EBS General Purpose SSD(gp2) or Magnetic storage.



Step
7

Configure Security Group

A security group acts as a virtual firewall for your instance,

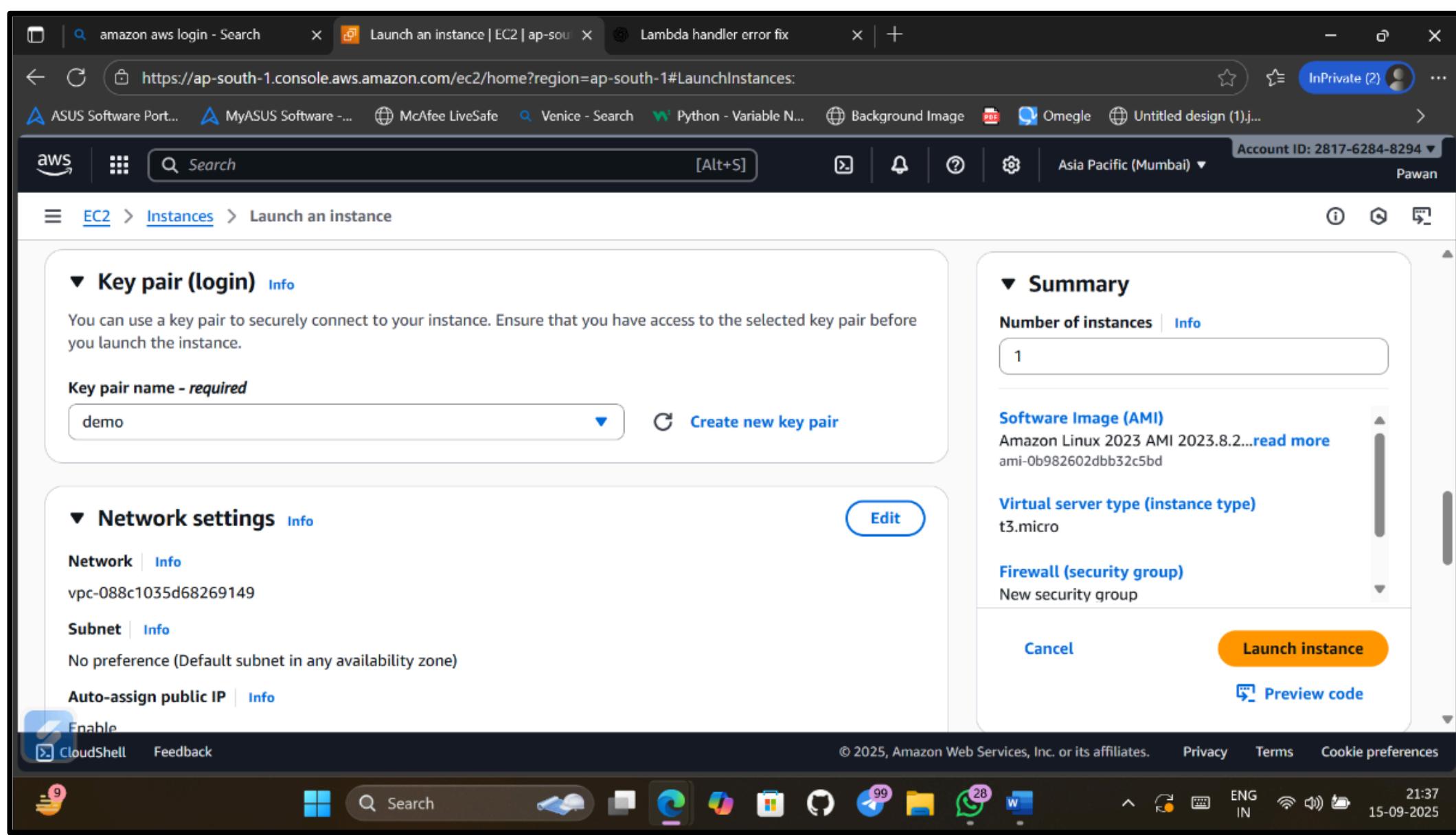
controlling inbound and outbound traffic. You can create a new security group or use an existing one. For a basic web server, you'll need to allow inbound traffic on port 80 (HTTP) and port 22 (SSH) to connect to the instance.

Step-8 Review and Launch

Review all the settings you've chosen. If everything looks correct, click the "Launch" button

Step-9 Create or Select a Key Pair

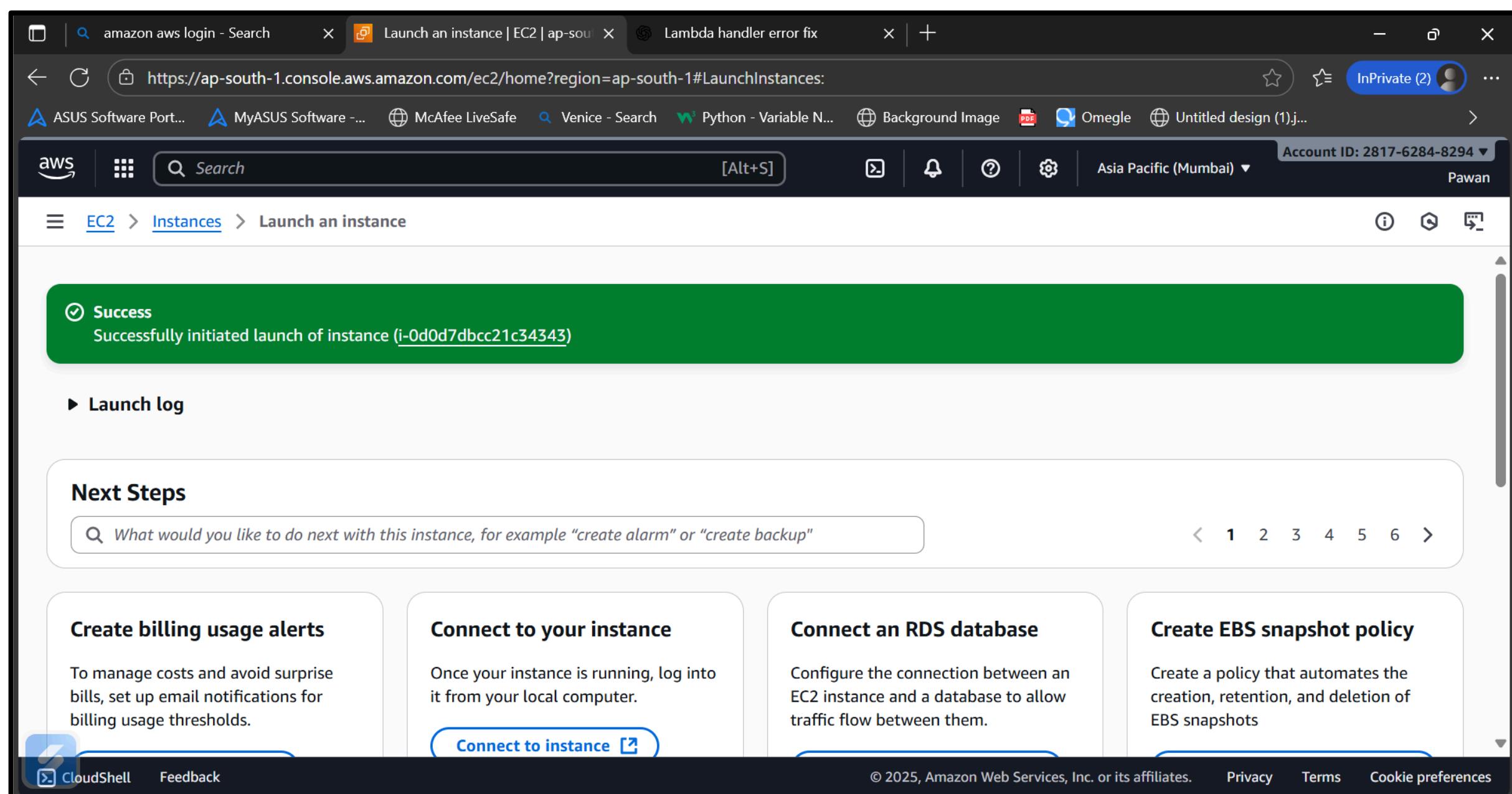
This is a crucial step. A **key pair** is used to securely connect to your instance via SSH. You must either **create a new key pair** or select an existing one. If you create a new one, you'll be prompted to download the **private key file (.pem)**. **Store this file securely**, as you will need it to connect to your instance and it cannot be re-downloaded later.



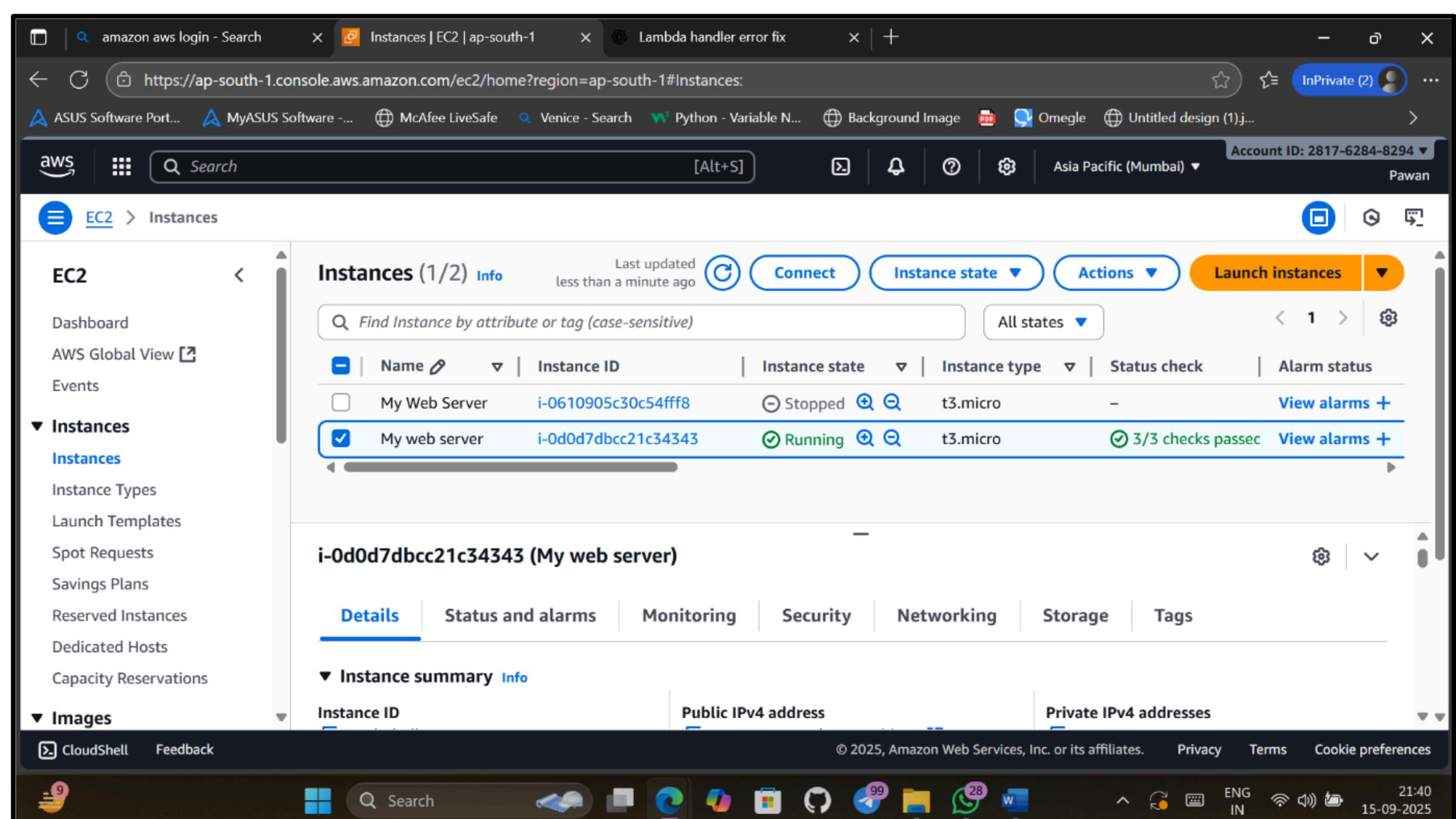
Ste
p-
10
La
un
ch
the
Ins
tan
ce

After selecting or creating the key pair, click "**Launch instance**". AWS will now provision your new EC2 instance. You can monitor its status on the EC2 dashboard, where it

will show up as "**Pending**" and then transition to "**Running**" when it's ready.



The screenshot shows the AWS EC2 'Launch an instance' page. At the top, there is a green success message box that says "Successfully initiated launch of instance (i-0d0d7dbcc21c34343)". Below this, there is a "Next Steps" section with several options: "Create billing usage alerts", "Connect to your instance", "Connect an RDS database", and "Create EBS snapshot policy". The "Connect to your instance" option is highlighted with a blue border. At the bottom of the page, there are links for "CloudShell" and "Feedback".



The screenshot shows the AWS EC2 'Instances' page. On the left, there is a navigation menu with 'EC2' selected. In the main area, there is a table titled 'Instances (1/2)'. The table has columns for 'Name', 'Instance ID', 'Instance state', 'Instance type', 'Status check', and 'Alarm status'. There are two rows: one for 'My Web Server' (stopped) and one for 'My web server' (running). The 'My web server' row is selected and highlighted with a blue border. Below the table, there is a detailed view for the selected instance, showing its ID as i-0d0d7dbcc21c34343 and the name 'My web server'. The 'Details' tab is selected, showing tabs for 'Status and alarms', 'Monitoring', 'Security', 'Networking', 'Storage', and 'Tags'. At the bottom of the page, there are links for "CloudShell" and "Feedback".

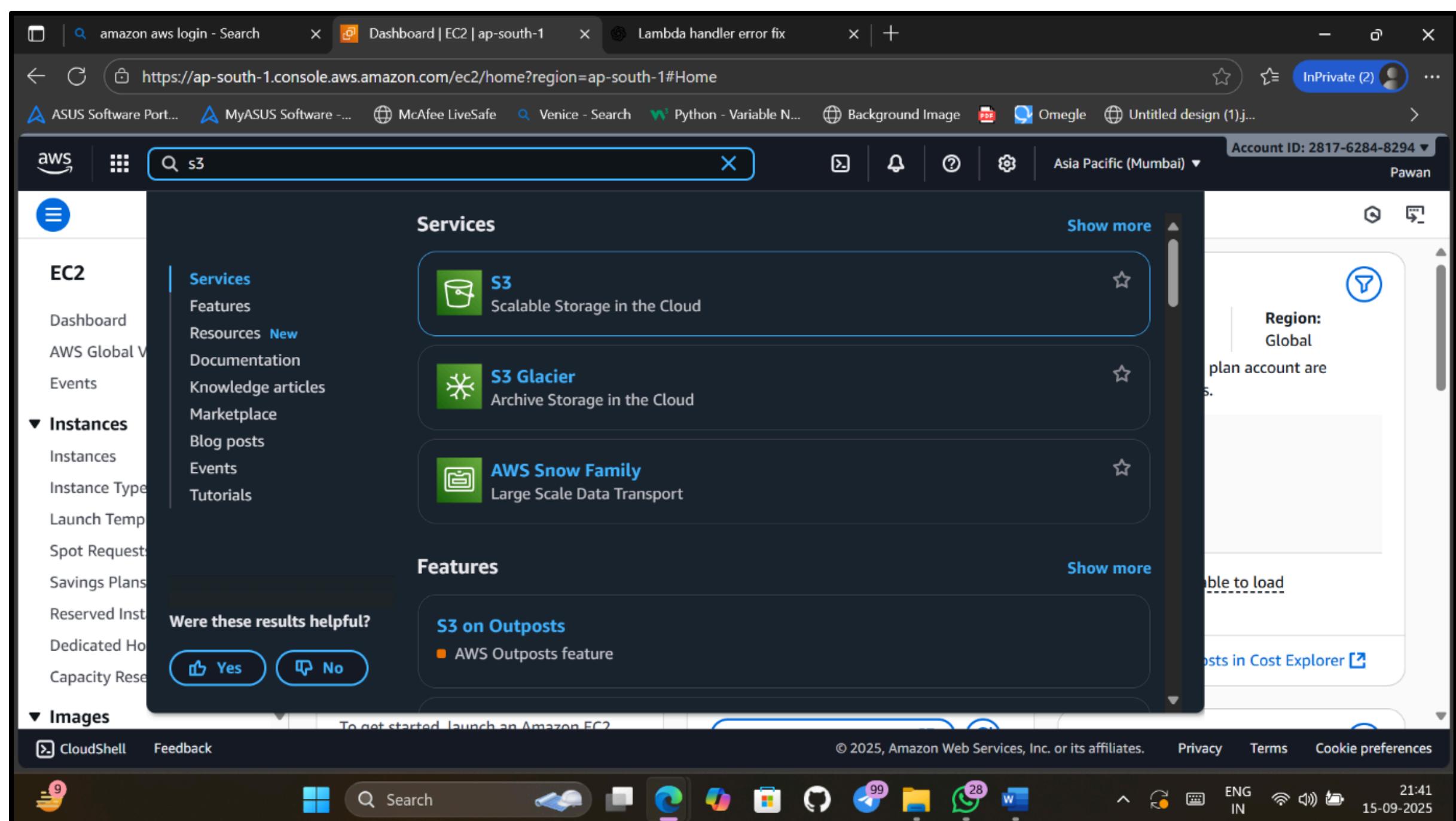
2. Create an S3 Bucket in AWS

1. Log into AWS Management Console

- Go to AWS Console.
- Sign in with your AWS account credentials.

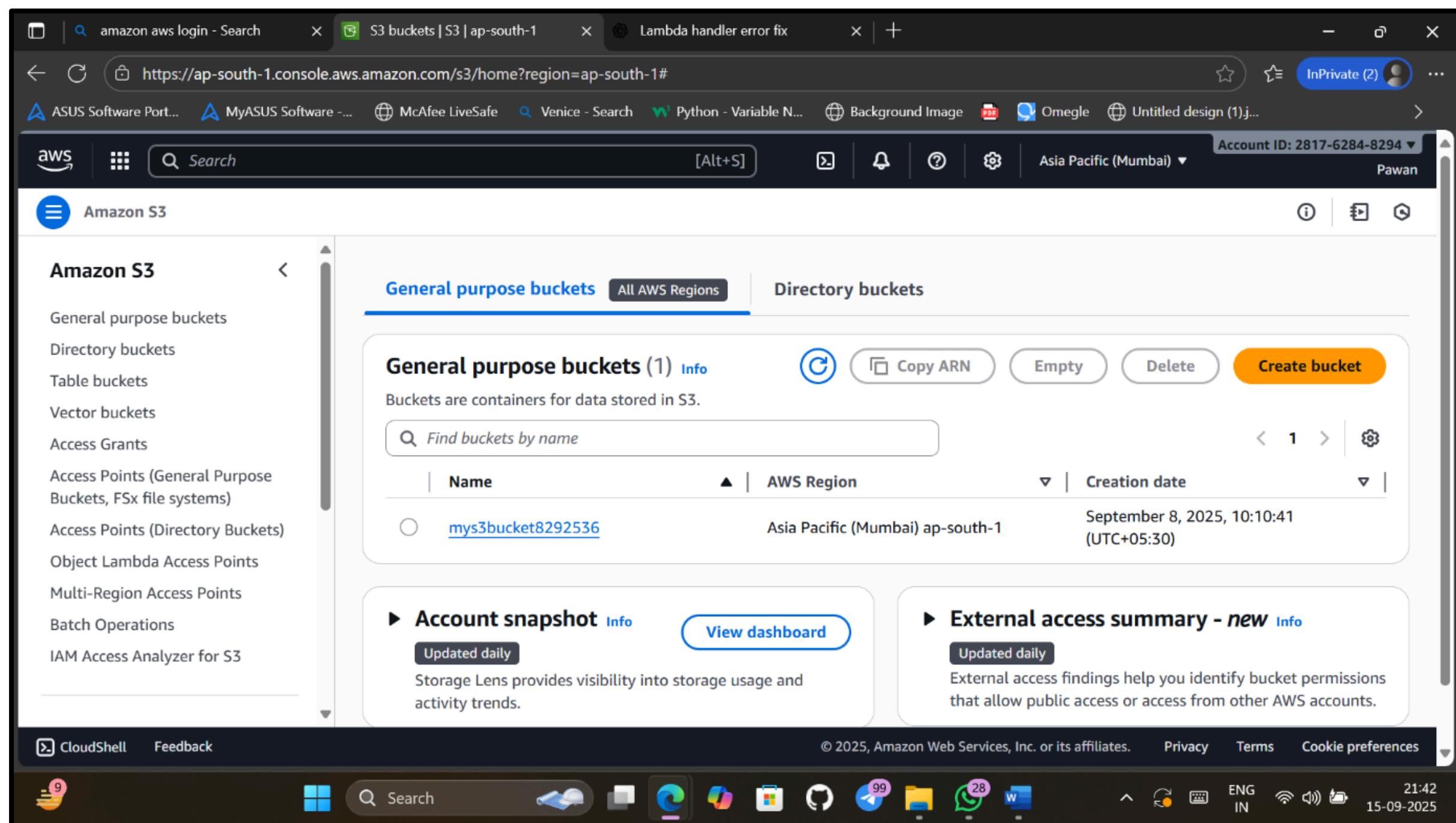
2. Navigate to S3

- In the search bar at the top, type S3.
- Click on **S3 (Scalable Storage in the Cloud)**.



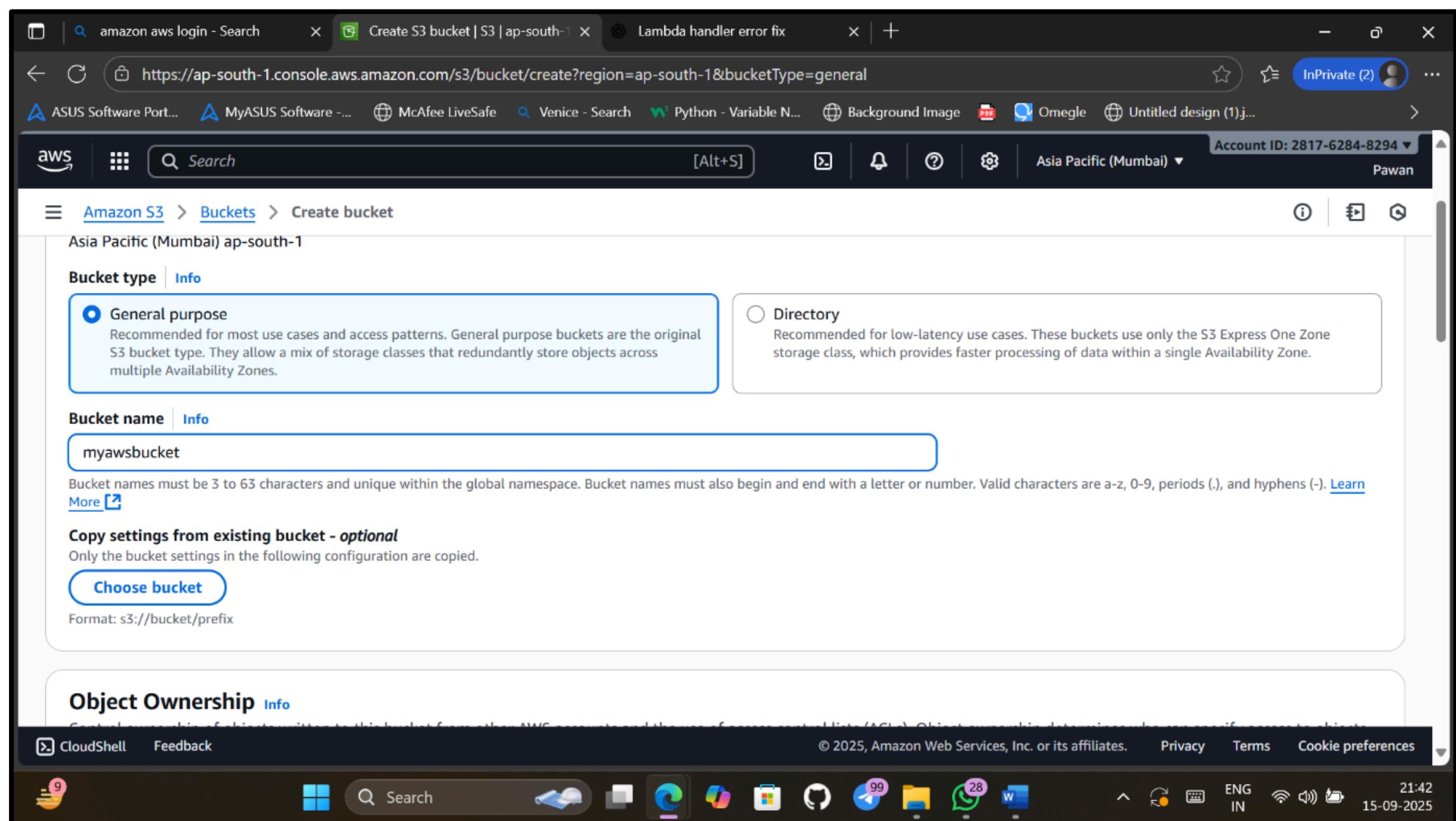
3. Create a New Bucket

- Click on "Create bucket" button.



4. Configure Bucket Settings

- **Bucket name:** Enter a unique name (e.g., my-first-s3-bucket-2025).
⚠ Bucket names must be globally unique across AWS.
- **AWS Region:** Select a region (choose one closest to your users for better performance).



5. Set Bucket Options

- **Object Ownership:**
 - Choose **ACLs disabled (recommended)** for most use cases.
- **Block Public Access:**
 - By default, all public access is blocked (recommended for private buckets).
 - If you need a public bucket (for hosting static websites), uncheck the option and acknowledge the warning.

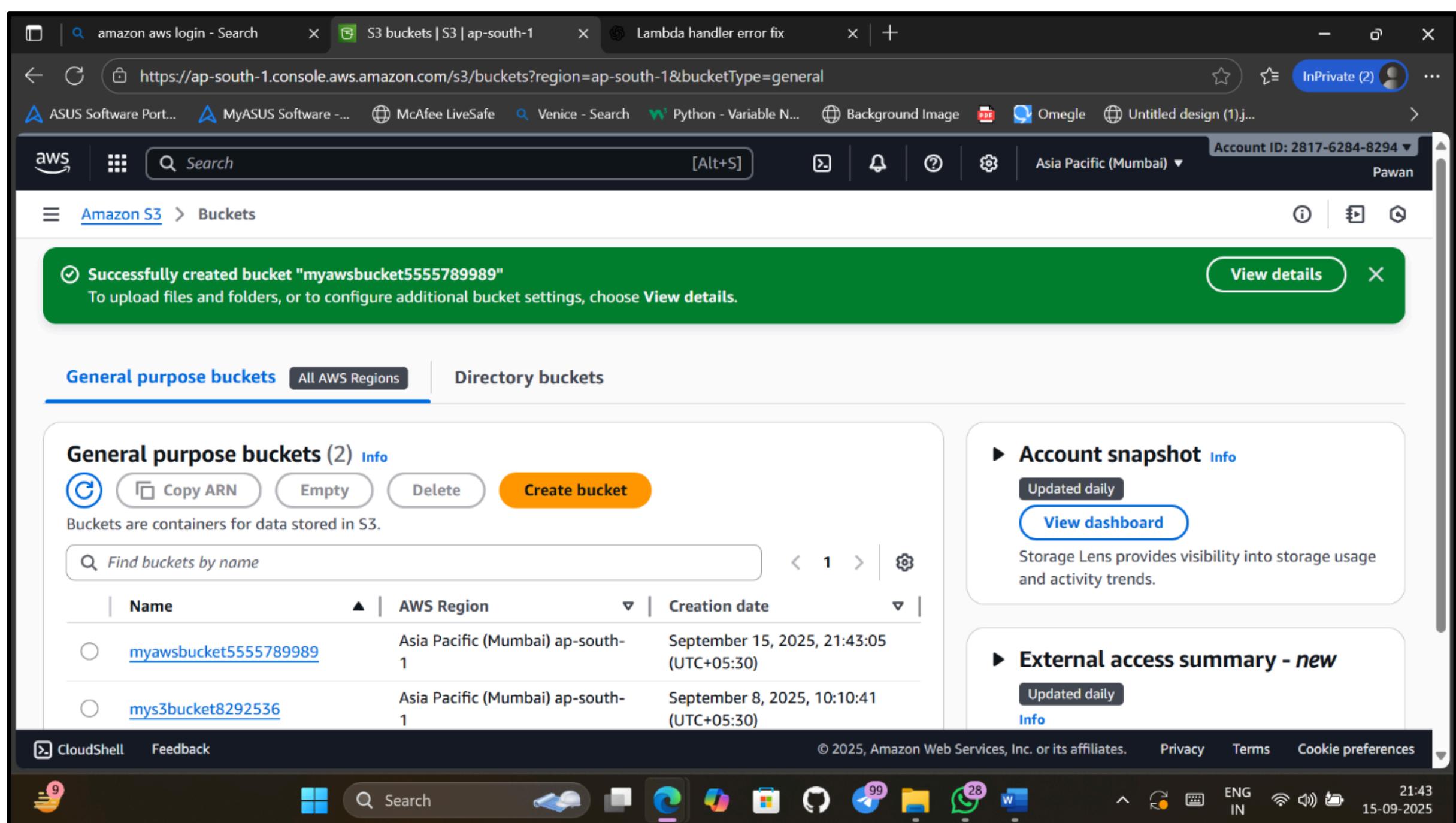
6. Configure Bucket Settings (Optional)

- **Versioning:** Enable if you want to keep multiple versions of an object.

- **Encryption:** Enable default encryption if required.
- **Tags:** Add key-value tags if you want to manage cost tracking or organization.

7. Review and Create

- Review your settings.
- Click **Create bucket**.



8. Upload Objects (Optional)

- After creation, open the bucket.
- Click **Upload → Add files**.
- Select your file(s), then click **Upload**.

The screenshot shows the 'Upload' page for an S3 bucket named 'mys3bucket8292536'. At the top, there's a large input field with the placeholder 'Drag and drop files and folders you want to upload here, or choose Add files or Add folder.' Below this is a table titled 'Files and folders (0)' with a single row: 'All files and folders in this table will be uploaded.' The table includes columns for Name, Folder, Type, and Size, with sorting and filtering options. A message at the bottom states 'No files or folders' and 'You have not chosen any files or folders to upload.' The browser's address bar shows the URL: https://ap-south-1.console.aws.amazon.com/s3/upload/mys3bucket8292536?region=ap-south-1&bucketType=general.

The screenshot shows the 'Objects' tab for the same S3 bucket 'mys3bucket8292536'. The top navigation bar includes tabs for Objects, Properties, Permissions, Metrics, Management, and Access Points. The 'Objects' tab is selected. The main area displays a table of objects with one item: 'Resume_(1).pdf'. The table has columns for Name, Type, Last modified, Size, and Storage class. The object 'Resume_(1).pdf' is a PDF file, last modified on September 8, 2025, at 10:11:23 (UTC+05:30), with a size of 46.8 KB and a storage class of Standard. Action buttons for Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload are available above the table. The browser's address bar shows the URL: https://ap-south-1.console.aws.amazon.com/s3/buckets/mys3bucket8292536?region=ap-south-1&tab=objects.

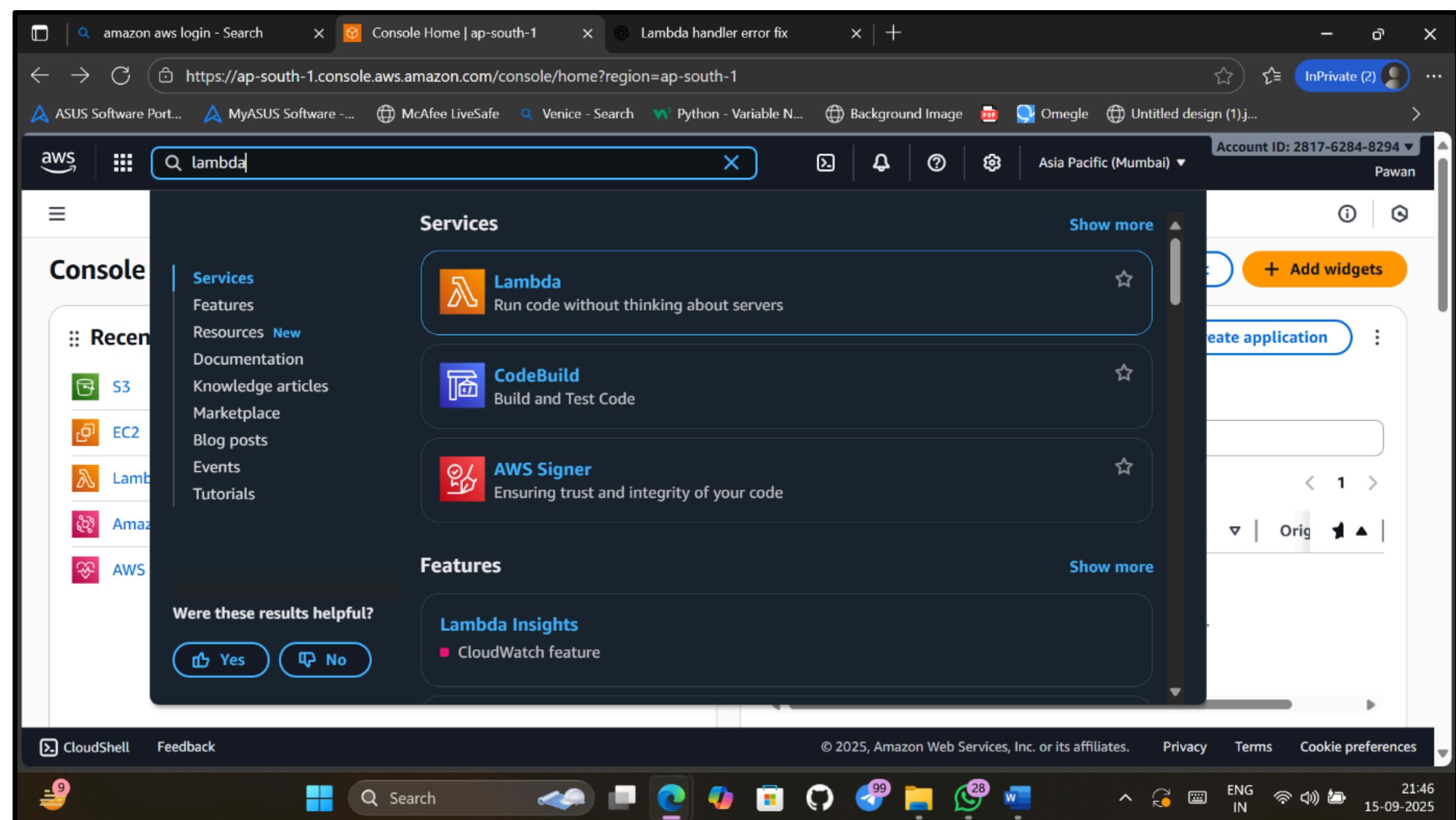
3. Steps to Create an AWS Lambda Function

1. Log into AWS Management Console

- Go to AWS Console.
- Sign in with your credentials.

2. Navigate to AWS Lambda

- In the search bar, type **Lambda**.
- Click on **Lambda** service.



3. Create a New Lambda Function

- Click **Create function**.

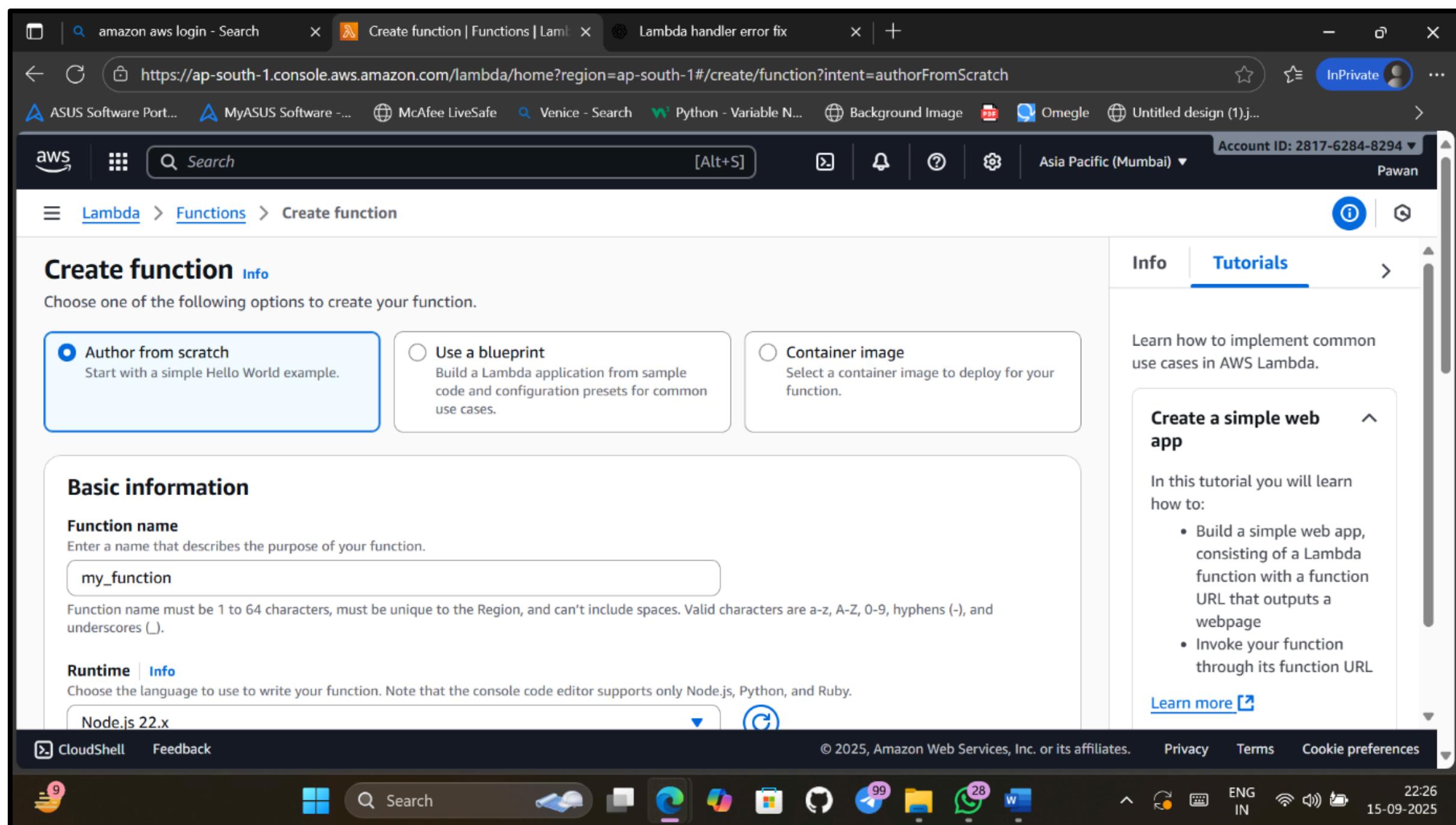
4. Choose a Creation Method

You'll see 3 options:

1. **Author from scratch** → (most common, start fresh).
2. **Use a blueprint** → predefined templates.
3. **Container image** → deploy code as Docker container.

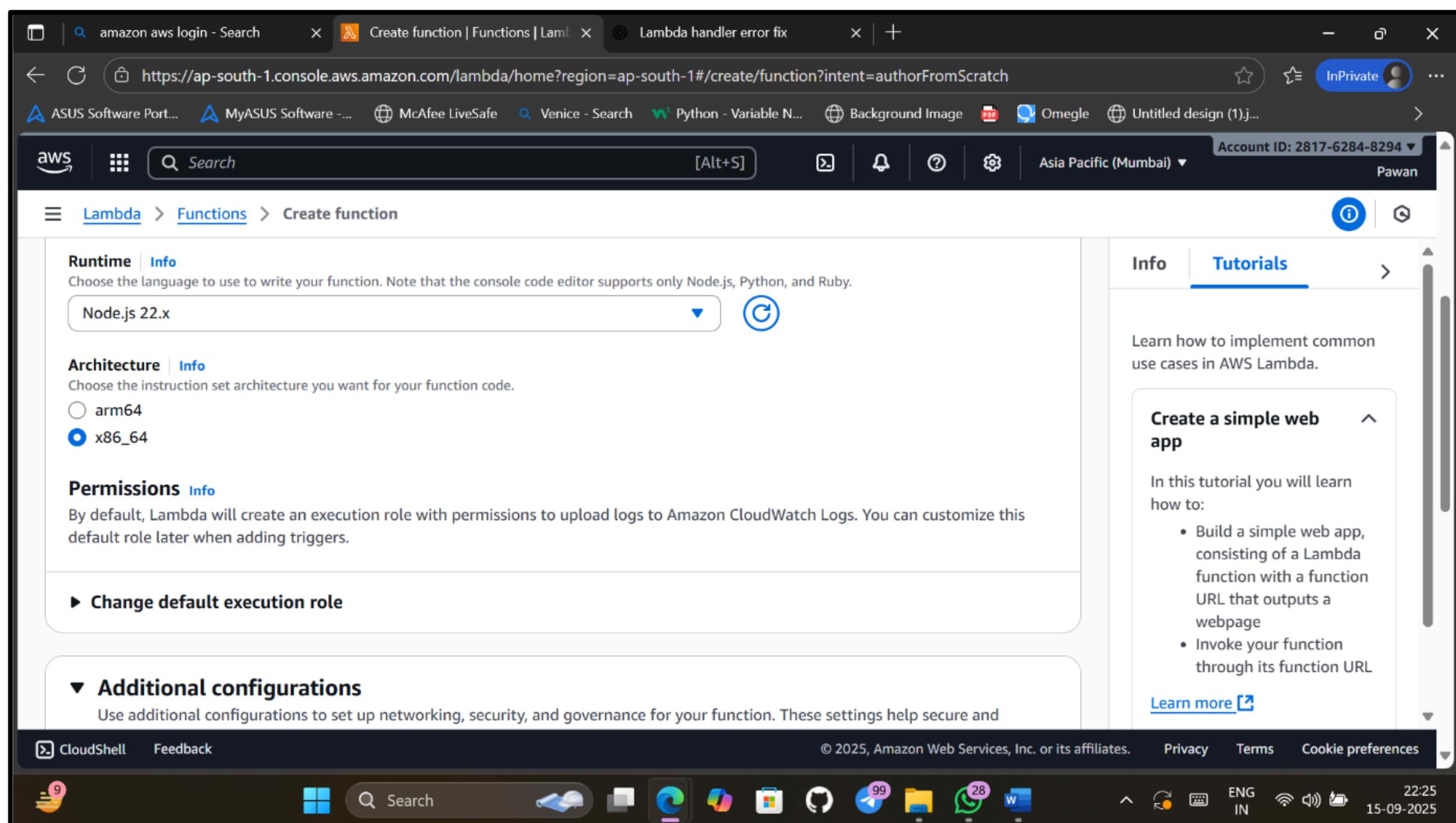
👉 Select **Author from scratch**.

The screenshot shows the AWS Lambda Functions console in a web browser. The URL in the address bar is <https://ap-south-1.console.aws.amazon.com/lambda/home?region=ap-south-1#/functions>. The browser's toolbar includes various tabs and icons. On the left, a sidebar menu is open under the 'Lambda' section, showing 'Dashboard', 'Applications', 'Functions' (which is selected), 'Additional resources' (including 'Code signing configurations', 'Event source mappings', 'Layers', and 'Replicas'), and 'Related AWS resources' (including 'Step Functions state machines'). The main content area displays a table titled 'Functions (1)'. The table has columns for 'Function name', 'Description', 'Package type', 'Runtime', and 'Last modified'. A single row is shown for a function named 'lambda829219', which was created '1 week ago'. The 'Actions' button is visible next to the table. To the right of the table, there are 'Info' and 'Tutorials' tabs. The 'Tutorials' tab is active, showing a section titled 'Create a simple web app' with a brief description and a 'Learn more' link. The status bar at the bottom shows the URL again, along with copyright information, privacy terms, cookie preferences, and system status indicators like battery level, signal strength, and date/time (15-09-2025, 21:46).

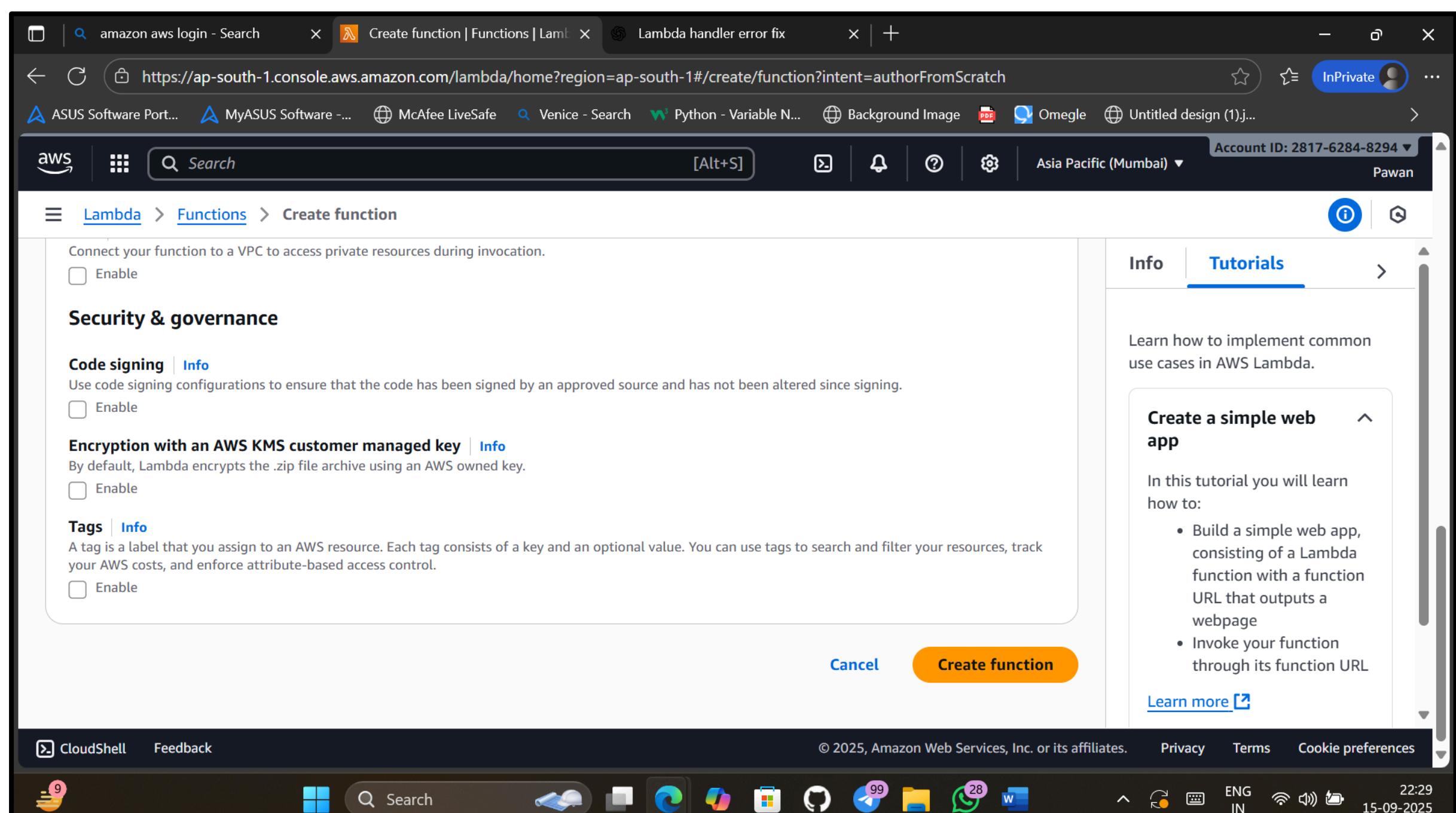


5. Configure Basic Settings

- **Function name:** Enter a unique name (e.g., MyFirstLambda).
- **Runtime:** Choose a runtime (Node.js, Python, Java, Go, etc. → example: Python 3.9).
- **Permissions (Execution Role):**
 - Create a new role with basic Lambda permissions (recommended if you're new).
 - Or choose an existing IAM role if you already have one.



6. Click Create Function



Wait a few seconds while AWS provisions the function.

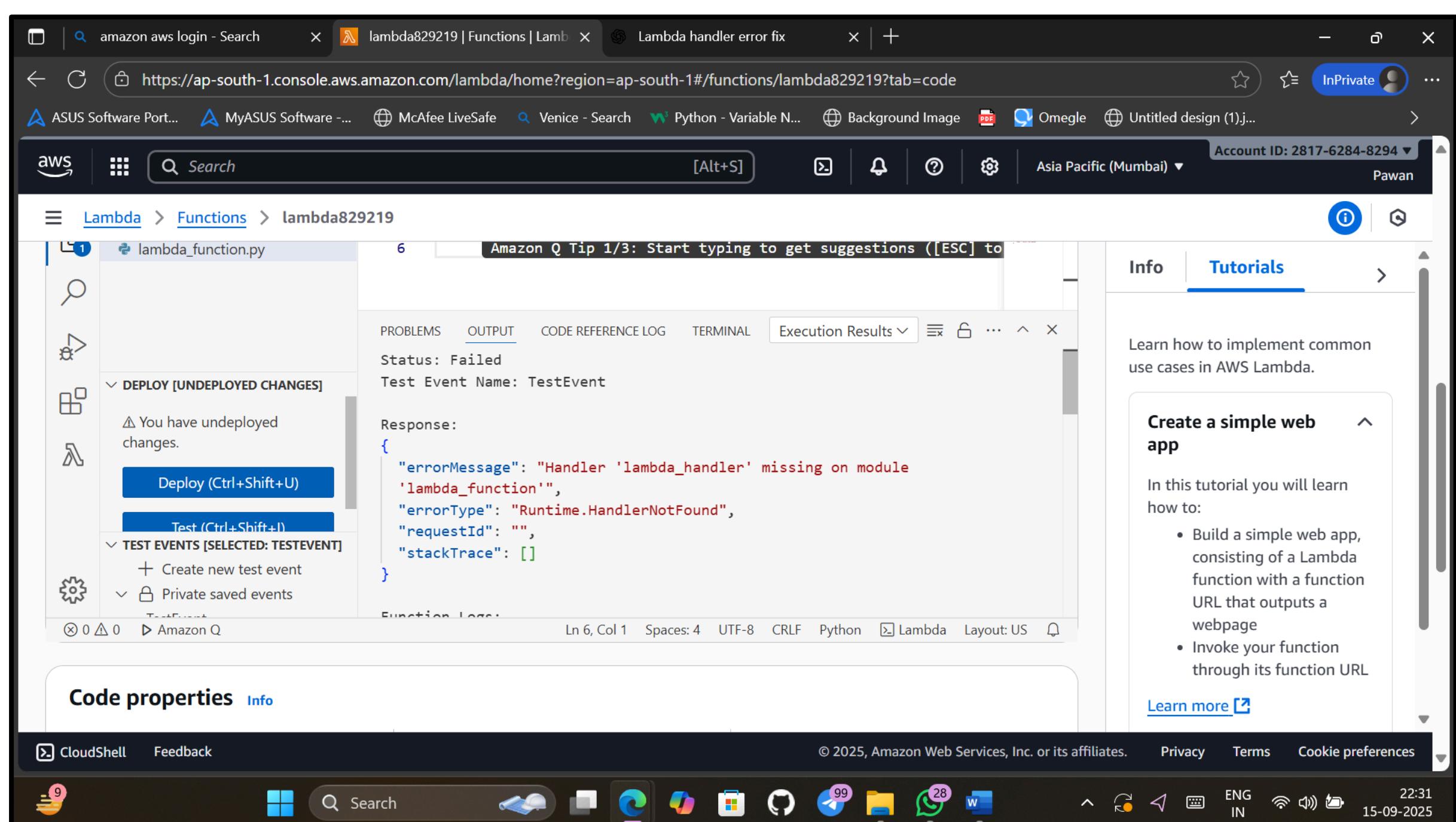
7. Add Your Code

- In the **Function code** section, you can:
 - Write inline code in the editor.
 - Or upload a .zip file.
 - Or use **Amazon S3** (if your code is stored there).

Example default code in Python:

```
def lambda_handler(event, context):
```

```
    return {
        'statusCode': 200,
        'body': 'Hello from Lambda!'
    }
```



8. Configure TestEvent

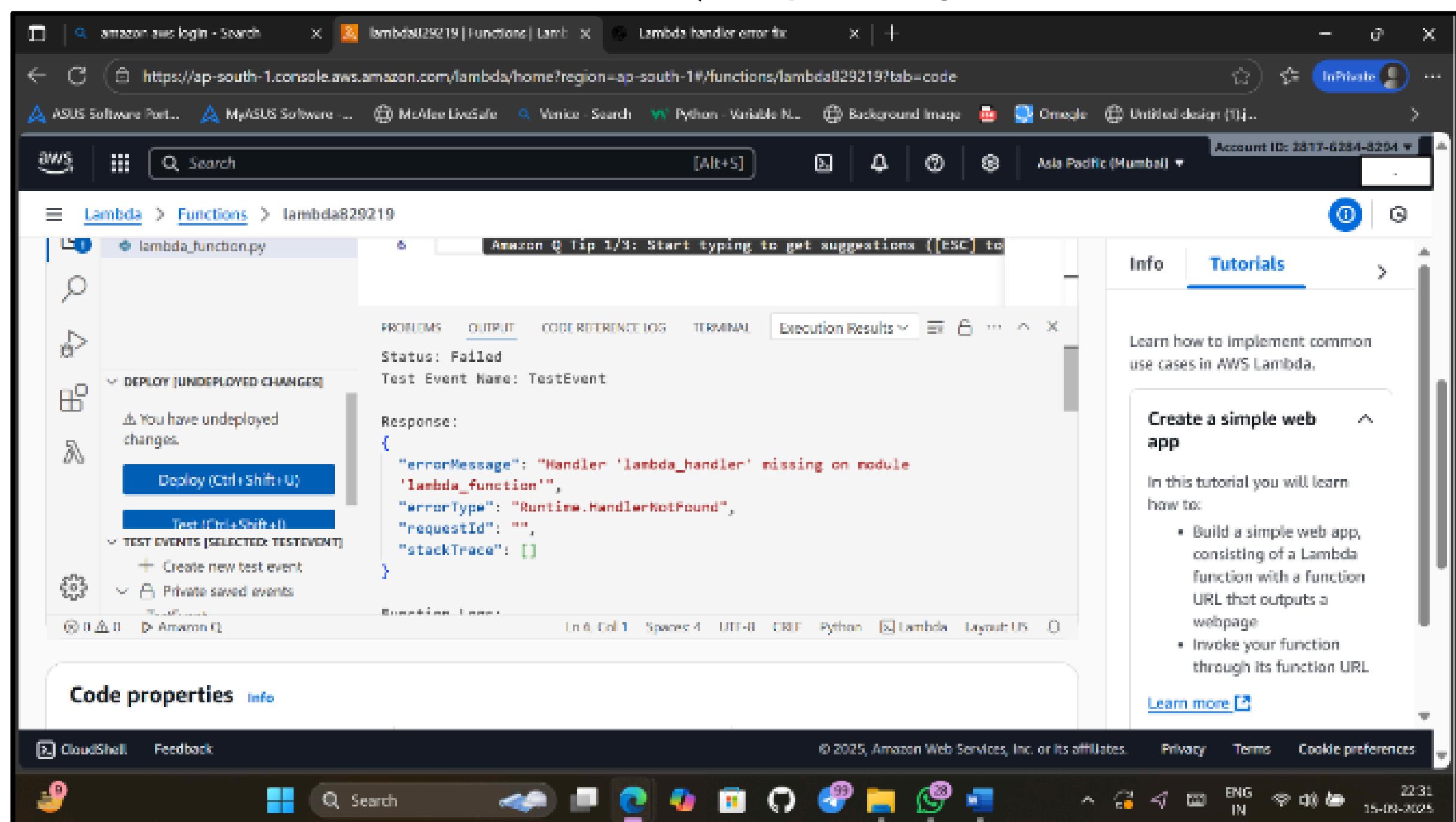
- Click **Test** → **Configure test event**.
- Give it a name (e.g., `TestEvent`).

- Keep the default event JSON or modify as needed.
- Save.

9. Run the Function

- Click **Test** again.

Check the execution results (output, logs, and status)

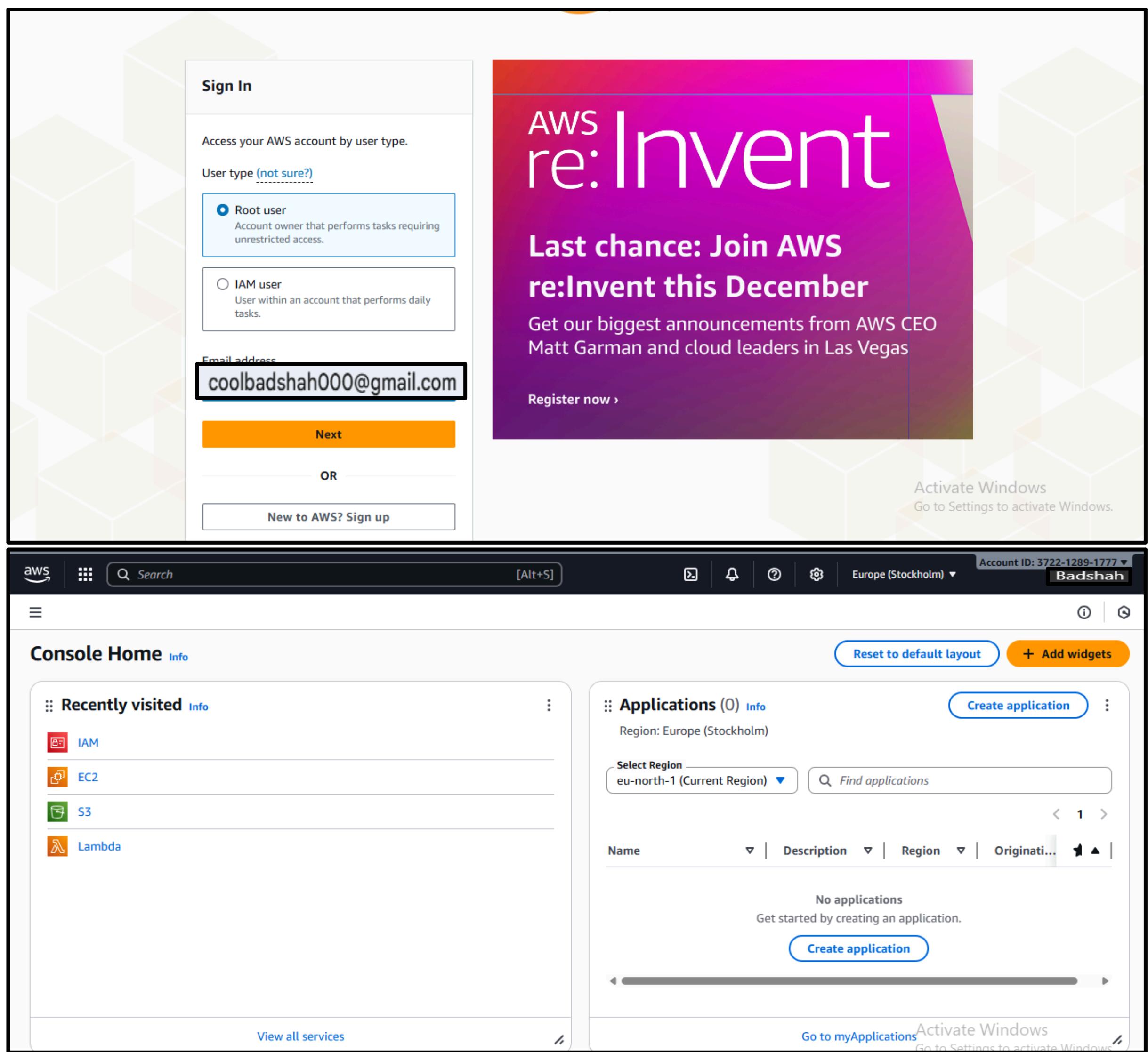


Experiment:-4

Objective:- creating an IAM (Identity and Access Management) user in AWS is to provide secure and controlled access to AWS resources for individuals or applications without using the root account.

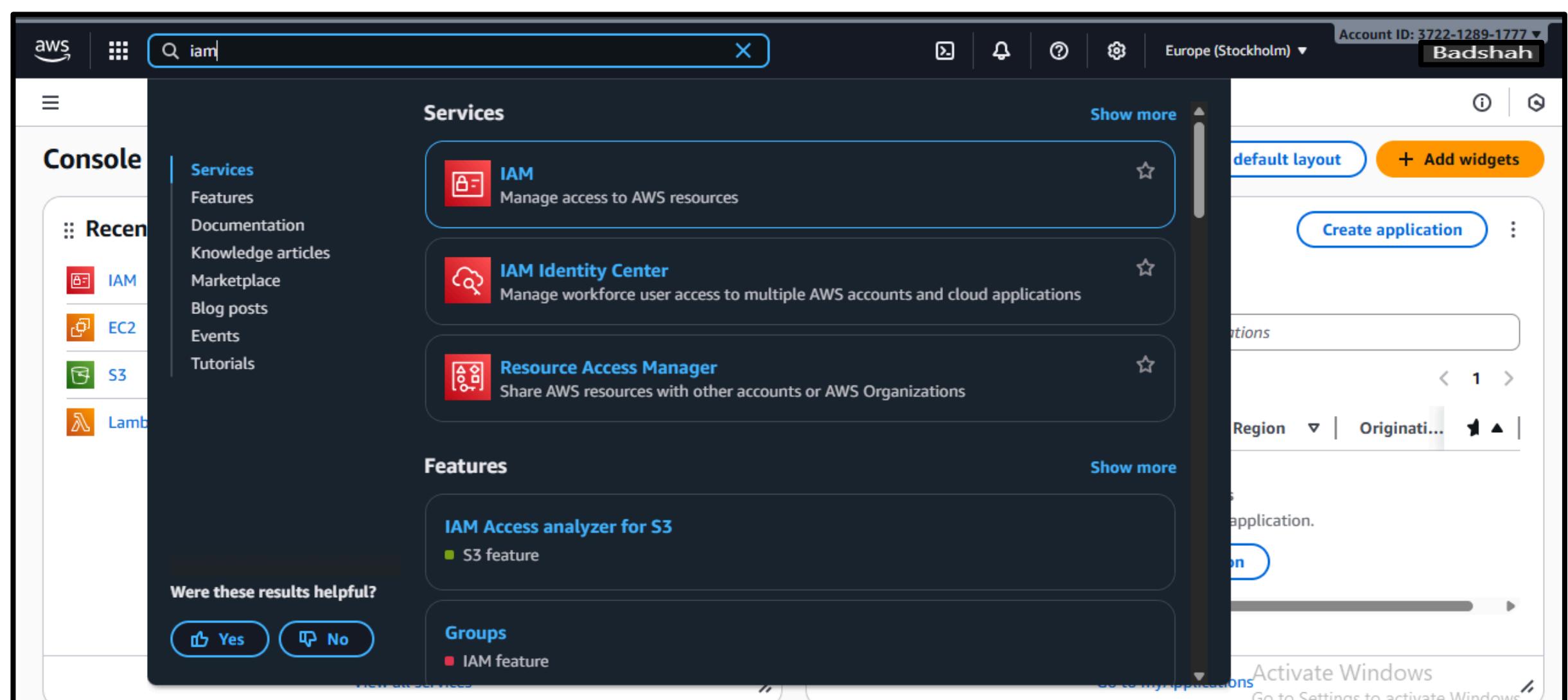
Step 1: Sign in to AWS Console

Log in to your **AWS Management Console** using your **root account** or an **IAM user** who has **Administrator privileges**.



Step 2: Open IAM Service

In the search bar at the top, type **IAM**. Select **IAM (Identity and Access Management)**.



The screenshot shows the IAM Dashboard. On the left, a sidebar includes "Identity and Access Management (IAM)" and sections for "Access management" (User groups, Users, Roles, Policies, Identity providers, Account settings, Root access management) and "Access reports" (Access Analyzer, Resource analysis). The main area displays "Security recommendations" (Root user has MFA, Root user has no active access keys), "AWS Account" (Account ID: 372212891777, Account Alias: Create, Sign-in URL: https://372212891777.signin.aws.amazon.com/console), and "Quick Links" (My security credentials, Manage your access keys, multi-factor authentication (MFA) and other credentials). A watermark for "Badshah" is visible.

Step 3: Go to Users Section

In the left sidebar, click on **Users**. You will see a list of all existing users. Click “Create user” to add a new one.

The screenshot shows the "Users" section of the IAM console. The left sidebar includes "Identity and Access Management (IAM)" and sections for "Access management" (User groups, Users, Roles, Policies, Identity providers, Account settings, Root access management) and "Access reports" (Access Analyzer, Resource analysis). The main area shows a table of users: Sanchi and Srishti. The table columns include User name, Path, Group, Last activity, MFA, Password age, and Console. A watermark for "Badshah" is visible.

Step 4: Enter User Details

Enter a **User name**. Choose the type of access: **Password access** → if the user needs to log in to the AWS Console. Click **Next**.

The screenshot shows the 'Create user' wizard at Step 4. The 'User name' field is filled with 'Badshah'. Under 'Console password', 'Custom password' is selected, and a password is entered. A checkbox for 'Users must create a new password at next sign-in - Recommended' is checked. The left sidebar shows steps 1 through 4.

Step 5: Set Permissions

Choose **Attach policies directly** → assign permissions manually (e.g., AmazonS3FullAccess, AdministratorAccess, etc.).

Then click **Next**.

The screenshot shows the 'Create user' wizard at Step 5, 'Set permissions'. The 'Permissions options' section shows 'Attach policies directly' selected. The 'Permissions policies' section shows a search bar with 's3', a filter by type dropdown, and a 'Create policy' button.

The screenshot shows the 'Permissions policies' section of the 'Create user' wizard. A search bar at the top left contains 's3'. A filter bar below it says 'Filter by Type' with 'All types' selected. A table lists several AWS managed policies:

Policy name	Type	Count
AmazonDMSRedshiftS3Role	AWS managed	0
AmazonS3FullAccess	AWS managed	5
AmazonS3ObjectLambdaExecutionRolePolicy	AWS managed	0
AmazonS3OutpostsFullAccess	AWS managed	0
AmazonS3OutpostsReadOnlyAccess	AWS managed	0
AmazonS3ReadOnlyAccess	AWS managed	7
AmazonS3TablesFullAccess	AWS managed	0
AmazonS3TablesLakeFormationServiceRole	AWS managed	0

Step 6: Review and Create

Review all details carefully. Click **Create user**.

The screenshot shows the 'Review and create' step of the 'Create user' wizard. On the left, a vertical navigation bar indicates the current step: Step 1 (Specify user details), Step 2 (Set permissions), Step 3 (Review and create), and Step 4 (Retrieve password). The 'Review and create' step is highlighted with a blue circle.

Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details

Name: Badshah	Console password type: Custom password	Require password reset: Yes
---------------	--	-----------------------------

Permissions summary

Name	Type	Used as
AmazonS3FullAccess	AWS managed	Permissions policy
IAMUserChangePassword	AWS managed	Permissions policy

Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

Permissions summary

Name	Type	Used as
AmazonS3FullAccess	AWS managed	Permissions policy
IAMUserChangePassword	AWS managed	Permissions policy

Tags - optional
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel Previous Create user

Step 7: Save Login Details

- Once the user is created, AWS will show:
 - User ARN (Amazon Resource Name)**
 - Console login link**
 - Password or Access key/Secret key** (Download the .csv file – it won't be shown again).

User created successfully

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

[View user](#)

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Step 4
Retrieve password

Retrieve password

You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

Console sign-in details

Console sign-in URL
<https://372212891777.signin.aws.amazon.com/console>

U

Console password
 [Show](#)

Email sign-in instructions ↗

Activate Windows
Go to Settings to activate Windows

EXPERIMENT:-5

Objective: To create a user group in AWS IAM in order to manage permissions collectively for multiple users having similar roles or responsibilities.

Step1:- Open the IAM Service:
In the search bar at the top of the console, type **IAM**, then select **Identity and Access Management** from the results.

The screenshot shows the AWS IAM Dashboard. On the left sidebar, under 'Access management', 'User groups' is selected. The main area displays 'Security recommendations' with two items: 'Root user has MFA' and 'Root user has no active access keys'. Below that is the 'IAM resources' section, which shows 3 User groups, 3 Users, 7 Roles, 1 Policies, and 0 Identity providers. On the right side, there's an 'AWS Account' summary with the Account ID (372212891777), Account Alias (Create), and a Sign-in URL (https://372212891777.signin.aws.amazon.com/console). A 'Quick Links' section includes links for 'My security credentials' and 'Activate Windows'.

Step2:- Go to User Groups Section:
In the left-hand sidebar, click on **User groups**.

The screenshot shows the AWS IAM Dashboard. On the left sidebar, under 'Access management', 'User groups' is selected. The main area displays 'Security recommendations' with two items: 'Root user has MFA' (green checkmark) and 'Root user has no active access keys' (green checkmark). Below this is the 'AWS Account' section with account ID 372212891777 and a 'Sign-in URL for IAM users in this account' link. A 'Quick Links' section at the bottom right includes 'My security credentials' and a note to activate Windows.

The screenshot shows the 'User groups' page. The left sidebar shows 'User groups' is selected under 'Access management'. The main area lists three user groups: 'Administrator', 'Administrators', and 'Group1'. Each group has a status of 'Defined' and was created within the last few days. A 'Create group' button is located in the top right corner.

Step3:- Click on “ Create group” :

On the User Groups page, click the “ Create group” button to start creating a new group.

The screenshot shows the 'Create user group' wizard. The left sidebar shows 'User groups' is selected under 'Access management'. The main area is titled 'Create user group' and contains a 'Name the group' section with a 'User group name' input field. Below it is an 'Add users to the group - Optional' section showing two users: 'Sanchi' and 'Srishti'. Both users have a status of '5 days ago'.

Step4:- Enter Group Name:

Type a **unique name** for your group (for example, Developers, Admins, or ReadOnlyUsers).

The screenshot shows the 'Create user group' interface in the AWS IAM console. The 'User group name' field is filled with 'Developer'. In the 'Add users to the group' section, three users are listed: Sanchi, Srishti, and srishti2. Each user has a checkbox next to their name, and the 'Group' column shows '2' for each.

Step5:- Attach Permissions Policies (Optional):

You can choose policies to attach to this group, such as:

- AmazonS3FullAccess
- AmazonEC2ReadOnlyAccess
- AdministratorAccess

If you want to add permissions later, you can **skip this step** and click **Next**.

The screenshot shows the 'Attach permissions policies' step in the 'Create user group' wizard. The 'Policy name' column lists various AWS managed policies, with 'AdministratorAccess' being selected. Other policies shown include 'AdministratorAccess-Amplify', 'AdministratorAccess-AWSElasticBeanstalk', 'AIOpsConsoleAdminPolicy', 'AmazonAPIGatewayAdministrator', 'AmazonNimbleStudio-StudioAdmin', 'AmazonSageMakerAdmin-ServiceCatalogProduct...', and 'AmazonSageMakerHyperPodObservabilityAdmin...'. The 'Used as' column indicates that 'AdministratorAccess' is used as a 'Permissions policy (8)'.

Step 6:- Add Users to the Group (Optional):

You can select existing IAM users to include in this group now, or you can add users later after creating the group.

Identity and Access Management (IAM)

Group name: Developer

Add users to the group - Optional (3) Info

User name	Group	Last activity	Creation time
Sanchi	2	5 days ago	4 weeks ago
Srishthi	2	33 days ago	4 weeks ago
srishthi2	0	-	19 minutes ago

Attach permissions policies - Optional (3/1082) Info

Filter by Type

admin

48 matches

Step 7:- Review and Create Group:

Review the group details and attached policies, then click **Create group**.

Identity and Access Management (IAM)

Create user group

AmazonWorkSpacesApplicationManagerAdminAccess

AWS-SSM-DiagnosisAutomation-AdministrationRole

AWS-SSM-DiagnosisAutomation-OperationalAccess

AWS-SSM-RemediationAutomation-AdministrationRole

AWS-SSM-RemediationAutomation-OperationalAccess

AWSAppSyncAdministrator

AWSAuditManagerAdministratorAccess

AWSBackupOrganizationAdminAccess

AWSBudgetsActions_RolePolicyForResourceAdministrator

AWSCloud9Administrator

Cancel

Create user group

Identity and Access Management (IAM)

User groups (4) Info

Developer user group created.

Group name	Users	Permissions	Creation time
Administrator	2	Defined	5 days ago
Administrators	0	Defined	5 days ago
Developer	3	Defined	Now
Group1	2	Defined	4 weeks ago

View group

Delete

Create group

Administrator

Administrators

Developer

Group1

5 days ago

5 days ago

Now

4 weeks ago

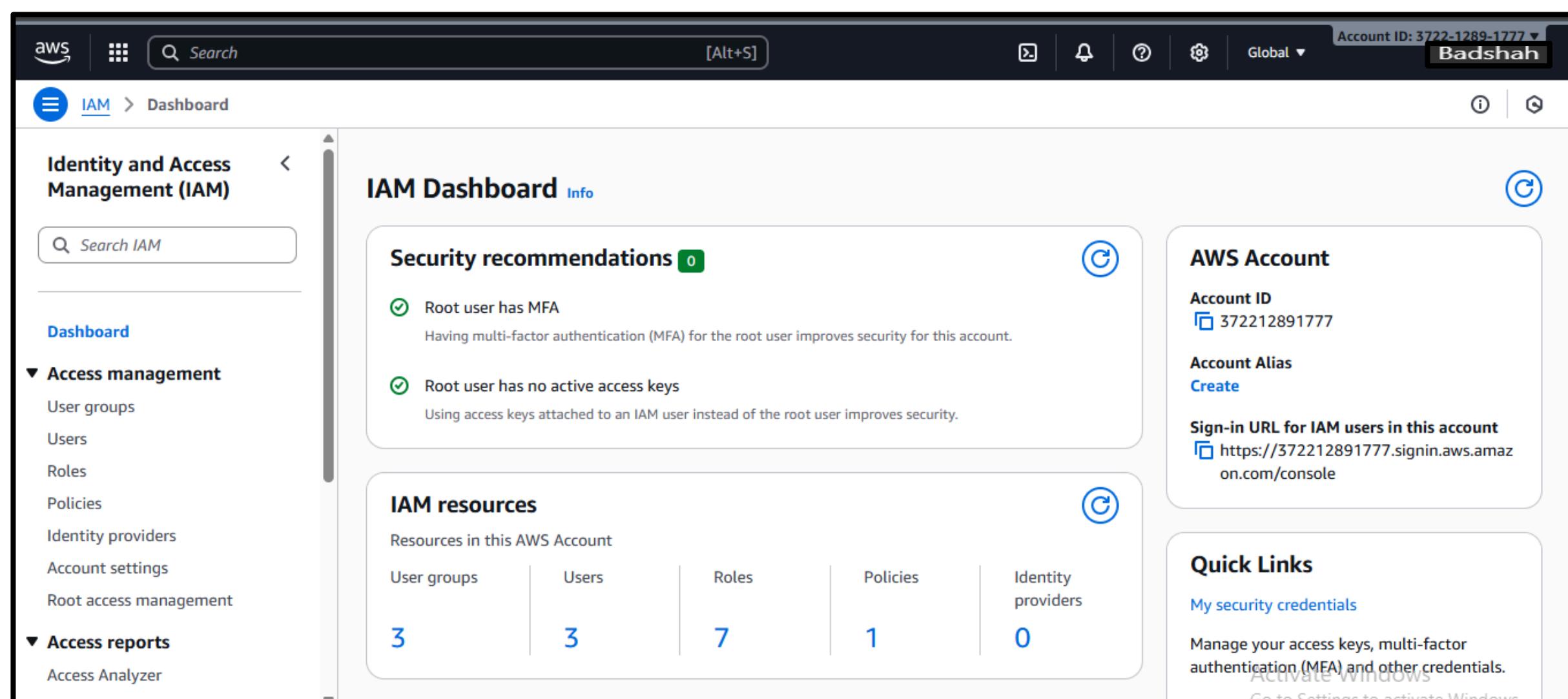
Activate Windows

Go to Settings to activate Windows.

EXPERIMENT:-6

Objective:-To create a security role in AWS IAM that allows AWS services or users to securely access specific AWS resources with defined permissions, ensuring controlled and temporary access without sharing long-term credentials.

Step1:- Open the IAM Service:
In the search bar at the top of the console, type **IAM**, then select **Identity and Access Management** from the results.



Step 2:- Go to Roles Section:

In the left-hand navigation pane, click on **Roles**.

The screenshot shows the AWS IAM Roles page. On the left, there's a navigation pane with 'Identity and Access Management (IAM)' selected. Under 'Access management', 'Roles' is also selected. The main area displays a table of roles with columns for 'Role name', 'Trusted entities', and 'Last activity'. There are four entries:

Role name	Trusted entities	Last activity
AWSServiceRoleForResourceExplorer	AWS Service: resource-explorer-2 (Service-Linked)	13 minutes ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linked)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked)	-
myfirstlambda-role-x54bi11s	AWS Service: lambda	47 days ago

Below the table, there's a section titled 'Roles Anywhere' with options for 'Access AWS from your non-AWS' and 'X.509 Standard'. A 'Manage' button is available for this section.

**Step 3:- Click on "Create role" :
On the Roles page, click the " Create role" button to start the process.**

The screenshot shows the 'Create role' wizard at Step 1: 'Select trusted entity'. The left sidebar shows steps: Step 1 (selected), Step 2, Step 3. The main area is titled 'Select trusted entity' and contains a 'Trusted entity type' section with five options:

- AWS service**: Allows AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account**: Allows entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity**: Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation**: Allows users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy**: Create a custom trust policy to enable others to perform actions in this account.

At the bottom, there's a 'Use case' link and an 'Activate Windows' message.

Step4:- Select Trusted Entity Type:

Choose who will use the role, such as:

- **AWS Service** (e.g., EC2, Lambda)
- **Another AWS Account**
- **Web Identity or SAML 2.0 Federation**

Click **Next** after selecting the appropriate option.

The screenshot shows the AWS IAM 'Create role' interface. The top navigation bar includes the AWS logo, search bar, and account information (Account ID: 3722-1289-1777). The main navigation path is IAM > Roles > Create role. The current step is 'Use case'. A dropdown menu titled 'Service or use case' is set to 'EC2'. Below it, a list of 'Use case' options is shown, with 'EC2' selected (indicated by a blue circle).

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case

EC2

Choose a use case for the specified service.

Use case

EC2
Allows EC2 instances to call AWS services on your behalf.

EC2 Role for AWS Systems Manager
Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.

EC2 Spot Fleet Role
Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.

EC2 - Spot Fleet Auto Scaling
Allows Auto Scaling to access and update EC2 spot fleets on your behalf.

EC2 - Spot Fleet Tagging
Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.

EC2 - Spot Instances
Allows EC2 Spot Instances to launch and manage spot instances on your behalf.

EC2 - Spot Fleet
Allows EC2 Spot Fleet to launch and manage spot fleet instances on your behalf.

Step:-5 Attach Permissions Policies:

Select the **permissions policies** that define what actions the role can perform (for example, AmazonS3FullAccess or AmazonEC2FullAccess).

The screenshot shows the AWS IAM 'Create role' wizard at Step 2: 'Add permissions'. The left sidebar indicates the process flow: Step 1 (Select trusted entity) is completed (dark grey dot), Step 2 (Add permissions) is currently selected (blue dot), and Step 3 (Name, review, and create) is pending (light grey dot). The main content area is titled 'Add permissions' and displays a list of 'Permissions policies (1082)'. A search bar and a 'Filter by Type' dropdown are present. The table lists several AWS managed policies, each with a checkbox, policy name, type, and description. Policies listed include 'AdministratorAccess', 'AdministratorAccess-Amplify', 'AdministratorAccess-AWSElasti...', 'AIOpsAssistantIncidentReportP...', 'AIOpsAssistantPolicy', and 'AIOpsConsoleAdminPolicy'. The descriptions for these policies mention full access to AWS services, account administrative permissions, and specific incident reporting or console access.

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

Add permissions Info

Permissions policies (1082) Info

Choose one or more policies to attach to your new role.

Filter by Type

<input type="checkbox"/>	Policy name	Type	Description
<input type="checkbox"/>	AdministratorAccess	AWS managed - job function	Provides full access to AWS services an...
<input type="checkbox"/>	AdministratorAccess-Amplify	AWS managed	Grants account administrative permis...
<input type="checkbox"/>	AdministratorAccess-AWSElasti...	AWS managed	Grants account administrative permis...
<input type="checkbox"/>	AIOpsAssistantIncidentReportP...	AWS managed	Provides permissions required by the A...
<input type="checkbox"/>	AIOpsAssistantPolicy	AWS managed	Provides ReadOnly permissions requir...
<input type="checkbox"/>	AIOpsConsoleAdminPolicy	AWS managed	Grants full access to Amazon AI Opera...

aws | Search [Alt+S] | Global ▾ | Badshah

IAM > Roles > Create role

<input type="checkbox"/>	 AmazonEC2RoleforAWSCodeDeployLimited	AWS managed	Provides EC2 limited access to AWS CodeDeploy
<input type="checkbox"/>	 AmazonEC2RoleforDataPipelineRole	AWS managed	Default policy for the Amazon EC2 Data Pipeline
<input type="checkbox"/>	 AmazonEC2RoleforSSM	AWS managed	This policy will soon be deprecated
<input type="checkbox"/>	 AmazonEC2RolePolicyForLaunchWizard	AWS managed	Managed policy for the Amazon EC2 Launch Wizard
<input type="checkbox"/>	 AmazonEC2SpotFleetAutoscaleRole	AWS managed	Policy to enable Autoscaling for EC2 Spot Fleets
<input type="checkbox"/>	 AmazonEC2SpotFleetTaggingRole	AWS managed	Allows EC2 Spot Fleet to request tags
<input type="checkbox"/>	 AmazonElasticMapReduceforEC2Role	AWS managed	Default policy for the Amazon Elastic Map Reduce (EMR) on Amazon EC2
<input type="checkbox"/>	 AmazonSSMManagedEC2InstanceDefaultPolicy	AWS managed	This policy enables AWS Systems Manager (SSM) to manage EC2 instances

◀ Set permissions boundary - optional ▶

Cancel Previous Next

Activate Windows
Go to Settings to activate Windows

Step 6:- Name and Review the Role:

Enter a **role name** (for example, *EC2SecurityRole* or *LambdaAccessRole*) and review all selected settings.

The screenshot shows the 'Name, review, and create' step of the IAM role creation wizard. On the left, a vertical navigation bar lists 'Step 1: Select trusted entity', 'Step 2: Add permissions', and 'Step 3: Name, review, and create' (which is highlighted). The main area is titled 'Role details' and contains fields for 'Role name' (set to 'Developer') and 'Description' (set to 'Allows EC2 instances to call AWS services on your behalf'). Below this is 'Step 1: Select trusted entities', which includes a 'Trust policy' section. A note on the right says 'Activate Windows Go to Settings to activate Windows.'

The screenshot shows the 'Permissions policy summary' step. It displays a table of attached policies:

Policy name	Type	Attached as
AmazonEC2FullAccess	AWS managed	Permissions policy
AmazonS3FullAccess	AWS managed	Permissions policy

Below this is 'Step 3: Add tags', which includes an 'Add tags - optional' section with a note about key-value pairs and a button to 'Add new tag'. At the bottom are 'Cancel', 'Previous', and 'Create role' buttons, along with the 'Activate Windows' note.

Step 7:- Create the Role:

Click **Create role** to finish.

aws | Search [Alt+S] Account ID: 3722-1289-1777 Global Badshah

IAM > Roles

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users
- Roles**
- Policies
- Identity providers
- Account settings
- Root access management

Access reports

- Access Analyzer
- Resource analysis

Role Developer created.

Roles (5) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Search

Role name	Trusted entities	Last activity
AWSServiceRoleForResourceExplorer	AWS Service: resource-explorer-2 (Service-Linked Role)	22 minutes ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-
Developer	AWS Service: ec2	-
myfirstlambda-role-x54bi11s	AWS Service: lambda	47 days ago

Roles Anywhere Info

Activate Windows Manage Go to Settings to activate Windows

The screenshot shows the AWS IAM Roles page. A green success message at the top right says "Role Developer created." with a "View role" button. Below it, a table lists five roles: "AWSServiceRoleForResourceExplorer", "AWSServiceRoleForSupport", "AWSServiceRoleForTrustedAdvisor", "Developer", and "myfirstlambda-role-x54bi11s". Each role entry includes its last activity time. On the left sidebar, the "Roles" option under "Access management" is selected. At the bottom right, there's a "Manage" button for activating Windows.