

**ಬಿ.ಎಂ.ಎಸ್. ತಾಂತ್ರಿಕ ಮತ್ತು ವ್ಯವಸ್ಥಾಪನಾ ಮಹಾವಿದ್ಯಾಲಯ**  
**BMS Institute of Technology and Management**

(An Autonomous Institution Affiliated to VTU, Belagavi)  
Avalahalli, Doddaballapur Main Road, Bengaluru – 560119

**LAB RECORD**

**ON**

**CLOUD COMPUTING**

**(22MCA302)**

**III Semester**

*submitted in partial fulfilment of the requirement for the award of degree*

**MASTER OF COMPUTER APPLICATIONS**

**by**

**Trupti Yogish Kamat**

**1BY23MC096**



**Department of Master of Computer Applications**  
**(Accredited by NBA, New Delhi)**

**(2024-25 Odd Semester)**

**December-April 2025**

**BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT**  
(An Autonomous Institution, Affiliated to VTU, Belagavi)  
**Bengaluru – 560119**

**Department of MCA**

(Accredited by NBA, New Delhi)



**LABORATORY CERTIFICATE**

This is to certify that **Ms. Trupti Yogish Kamat** bearing **USN 1BY23MC096** has satisfactorily completed the course of experiments in **Java Programming Lab (22MCA302)** of II Semester MCA course at the **Department of MCA, BMS Institute of Technology and Management, Bengaluru**, during the period **July to Oct 2024**.

**Signature of the Course Co-ordinator**

**Dr. P Ganesh**

Associate Professor  
Department of MCA  
BMSIT&M  
Bengaluru.

**Signature of HoD**

**Dr. M Sridevi**

Assistant Professor & Head  
Department of MCA  
BMSIT&M  
Bengaluru

# **BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

**(An Autonomous Institution, Affiliated to VTU, Belagavi)**

**Bengaluru – 560119**

## **Department of MCA**



### **VISION**

To develop quality professionals in Computer Applications who can provide sustainable solutions to the societal and industrial needs.

### **MISSION**

Facilitate effective learning environment through quality education, state-of-the-art facilities, and orientation towards research and entrepreneurial skills.

### **Programme Educational Objectives (PEOs)**

**PEO 1:** Develop innovative IT applications to meet industrial and societal needs.

**PEO 2:** Adapt themselves to changing IT requirements through life-long learning.

**PEO 3:** Exhibit leadership skills and advance in their chosen career.



## BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(An Autonomous Institution, Affiliated to VTU, Belagavi)

Bengaluru – 560064

Department of MCA

### Programme Outcomes (POs)

**PO1 (Foundation Knowledge):** Apply knowledge of mathematics, programming logic and coding fundamentals for solution architecture and problem solving.

**PO2 (Problem Analysis):** Identify, review, formulate and analyse problems for primarily focussing on customer requirements using critical thinking frameworks.

**PO3 (Development of Solutions):** Design, develop and investigate problems with as an innovative approach for solutions incorporating ESG/SDG goals.

**PO4 (Modern Tool Usage):** Select, adapt and apply modern computational tools such as development of algorithms with an understanding of the limitations including human biases.

**PO5 (Individual and Teamwork):** Function and communicate effectively as an individual or a team leader in diverse and multidisciplinary groups. Use methodologies such as agile.

**PO6 (Project Management and Finance):** Use the principles of project management such as scheduling, work breakdown structure and be conversant with the principles of Finance for profitable project management.

**PO7 (Ethics):** Commit to professional ethics in managing software projects with financial aspects. Learn to use new technologies for cyber security and insulate customers from malware

**PO8 (Life-long learning):** Change management skills and the ability to learn, keep up with contemporary technologies and ways of working.



**BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

**(An Autonomous Institution, Affiliated to VTU, Belagavi)**

**Bengaluru – 560064**

**Department of MCA**

**Course Outcomes (COs)**

<b>CO1:</b>	Explore the evolution of cloud computing and enabling technologies.
<b>CO2:</b>	Analyze different computing environments.
<b>CO3:</b>	Classify various cloud service models and their providers.
<b>CO4:</b>	Compare various cloud deployment models.
<b>CO5:</b>	Deploy applications on real-time cloud platform(s).

## **Particulars of the Experiments Performed**

## **LAB PROGRAM 1: Creating an AWS account**

### Step 1: Open the AWS Free Tier Sign-Up Page

- Open a web browser and go to AWS Free Tier.
- Click on "Create an AWS Account".

### Step 2: Enter Account Information

- Email Address: Enter your email address (this will be your root user login).
- AWS Account Name: Provide a unique name for your AWS account.
- Click "Verify email address".
- Enter the verification code sent to your email and click "Verify".

### Step 3: Provide Personal and Contact Information

- Choose "Personal" as the account type.
- Enter your Full Name.
- Enter your Phone Number.
- Enter your Country/Region.
- Provide your Complete Address, including city, state, and postal code.
- Agree to AWS's Terms and Conditions by checking the box.
- Click "Continue".

### Step 4: Add Payment Information

- Enter your debit/credit card details.
- Provide the cardholder name and billing address.
- AWS will charge a small refundable amount (~₹2 INR or \$1 USD) to verify your card.
- Click "Continue".

### Step 5: Identity Verification

- Choose SMS Text Message or Voice Call for verification.
- Enter your phone number and select your country code.
- Click "Send SMS".
- Enter the OTP code received on your phone.
- Click "Verify".

### Step 6: Select Support Plan

- Choose the "Basic Support – Free" plan.
- Click "Continue".

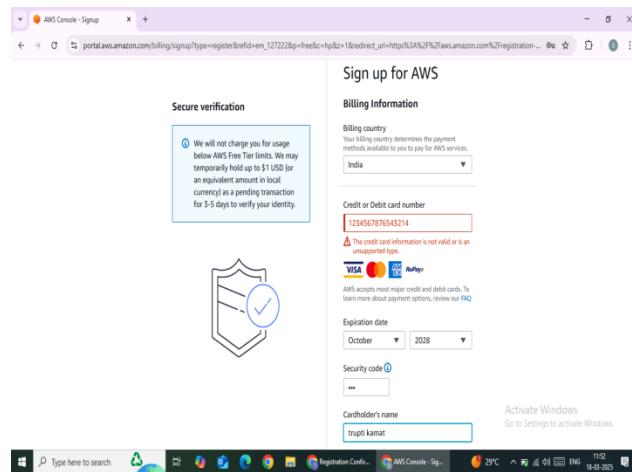
### Step 7: Sign in to AWS Console as the Root User

- Go to the AWS Management Console at <https://aws.amazon.com/console/>.
- Click on "Sign in to the Console".
- Enter your registered email address and click "Next".
- Enter your AWS root user password and click "Sign in".
- If MFA (Multi-Factor Authentication) is enabled, enter the verification code from your authenticator app.

### Step 8: Verify Successful Login

- Once logged in, you should see the AWS Management Console.
- Click on your account name (top right corner) → Select "Account".
- Ensure that your account details and Free Tier eligibility are correctly displayed.

## OUTPUT:



This screenshot shows the 'Console Home' page. On the left, the 'Sign up for AWS' form is partially visible. The main area displays the 'Recently visited' section with links to EC2, VPC, IAM, Lambda, CloudWatch, Billing and Cost Management, Amazon EventBridge, and Simple Notification Service. To the right, the 'Applications' section shows a message: 'No applications. Get started by creating an application.' A 'Create application' button is available. The bottom of the screen shows the Windows taskbar with various icons.

## 2. Creating a Virtual Machine using EC2

### Step 1: Open EC2 Dashboard

- Sign in to **AWS Management Console**.
- Navigate to **EC2**.
- Click on "**Launch Instance**".

### Step 2: Choose an Amazon Machine Image (AMI)

- Select an operating system (e.g., **Amazon Linux 2** or **Ubuntu**).

### Step 3: Choose an Instance Type

- Select **t2.micro** (Free Tier eligible).

### Step 4: Configure Instance Details

- Keep default settings.

### Step 5: Add Storage

- Default **8GB** is sufficient.

### Step 6: Configure Security Group

- Allow **SSH (port 22)** and **HTTP (port 80)** if needed.

### Step 7: Review and Launch

- Click "**Launch**".
- Create a new **key pair**, download it (.pem file), and click **Launch Instances**.

### Step 8: Verify the Running Instance

- Click **View Instances** to check the status.

The image is a composite screenshot of the AWS Cloud Computing interface, specifically the EC2 (Amazon Elastic Compute Cloud) section. It shows several browser tabs and windows side-by-side.

- Top Left:** A tab labeled "Launch an instance | EC2 | us-east-1" showing the EC2 Instances page. It features a large banner for "Amazon Elastic Compute Cloud (EC2)" with the subtext "Create, manage, and monitor virtual servers in the cloud." Below this are sections for "Benefits and features" and "Get started".
- Top Right:** A tab labeled "Connect to instance | EC2 | us-east-1" showing the "Connect to instance" details for an instance with ID i-0e30eb7d844aa710c. It includes options for "Session Manager" and "RDP client", and a link to "Download remote desktop file".
- Middle Left:** A tab labeled "Launch an instance | EC2 | us-east-1" showing the "Connect to instance" details for the same instance. This view is identical to the one in the top right.
- Middle Right:** A tab labeled "Launch an instance | EC2 | us-east-1" showing the "Launch an instance" process. It displays a green success message: "Successfully initiated launch of instance i-0e30eb7d844aa710c". Below this, there's a "Launch log" section and a "Next Steps" section with links to "Create billing and free tier usage alerts", "Connect to your instance", "Connect an RDS database", and "Create EBS snapshot policy".
- Bottom:** A dark navigation bar at the bottom of the screen showing various icons and the date/time (18-03-2025).

### **3. Run a Web Application on AWS**

Steps to Install and Configure IIS on a Windows Server:

1. Open Server Manager:

Click on the Start Menu.

Open "Server Manager."

2. Add Roles and Features:

Click on "Manage" in the Server Manager.

Select "Add Roles and Features."

Click "Next" to proceed.

3. Select Installation Type:

Choose "Role-based or Feature-based installation."

Click "Next."

4. Select Destination Server:

Choose the local server from the list.

Click "Next."

5. Select Server Roles:

Scroll down and check "Web Server (IIS)."

Click "Next."

6. Select Features:

Keep the default features or select additional ones if required.

Click "Next."

7. Confirm Installation:

Review the selected roles and features.

Click "Install" and wait for the installation to complete.

#### Configuring IIS:

##### 1. Open IIS Manager:

Click on "Start" and search for "IIS Manager."

Open the "Internet Information Services (IIS) Manager."

##### 2. Browse to Website Settings:

Expand the server node in the left panel.

Click on "Sites" and select "Default Web Site."

##### 3. Configure Website:

Click on "Browse" to view the default IIS page.

If needed, modify settings such as binding, authentication, and security.

##### 4. Manage IIS Features:

Go to "Advanced Settings" for deeper configuration.

Modify settings such as physical path, port, and logging.

##### 5. Testing:

Open a browser and type <http://localhost>.

Ensure the IIS default page loads successfully.

The image displays a series of screenshots illustrating the process of launching an EC2 instance and attempting to connect to it via RDP.

**Initial Launch:**

- The top-left screenshot shows the AWS EC2 Instances page with a "Launch instance" button highlighted.
- The top-right screenshot shows the "Launch an instance" wizard step 1 of 6, titled "Launch an instance (EC2) us-east-1". It shows a green success message: "Successfully initiated launch of instance i-0c35eb7844a710c".
- The bottom-left screenshot shows the "Connect to instance | EC2 | us-east-1" page, where a "Remote Desktop Connection" dialog box is open. It displays certificate verification errors: "The identity of the remote computer cannot be verified. Do you want to connect anyway?".
- The bottom-right screenshot shows the Windows taskbar with the "CloudShell Feedback" icon and the date/time "12:45 23-03-2025".

**Subsequent Attempts:**

The middle section shows two more attempts to connect to the same instance (i-0c35eb7844a710c) using the "Connect to instance" feature. Both attempts result in the same certificate verification dialog box, indicating that the connection attempt failed due to certificate issues.

## 4. Create Storage using S3

### Step 1: Open AWS S3 Console

- Navigate to S3 → Click **Create Bucket**.

### Step 2: Configure Bucket Settings

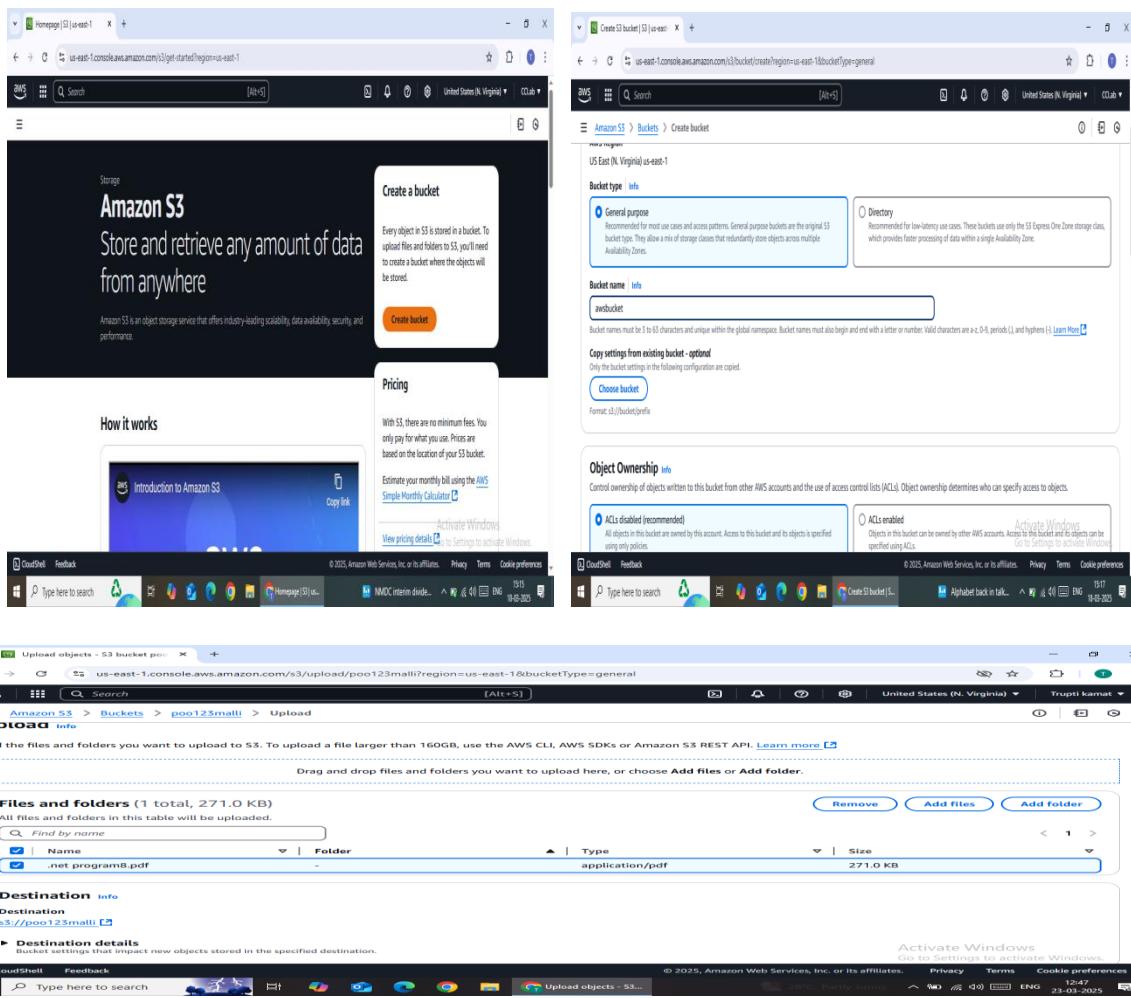
- Enter a **unique bucket name**.
- Choose an AWS **region**.
- Uncheck **Block all public access** (if public access is needed).

### Step 3: Create the Bucket

- Click **Create Bucket**.

### Step 4: Upload Files

- Click on the bucket → **Upload** → Choose files → Click **Upload**.



## 5. Manage Billing and CloudWatch

### Step 1: Create a Launch Template

1. Navigate to the AWS EC2 Dashboard.
2. Click on Launch Templates → Create Launch Template.
3. Provide a Launch Template Name (e.g., ASETemp).
4. (Optional) Provide a Version Description (e.g., Demo).
5. AMI Selection:
  - Choose an Amazon Machine Image (AMI).
  - Select Windows Server 2019 Base (Free Tier Eligible).
6. Instance Type: Choose t2.micro (Free Tier Eligible).
7. Key Pair:
  - Create a new key pair (e.g., asg) or use an existing one.
8. Security Group: Select an existing security group or use the default one.
9. Click Create Launch Template.
10. After creation, click View Template.

### Step 2: Create an Auto Scaling Group

1. Go to Auto Scaling Groups.
2. Click Create Auto Scaling Group.
3. Provide Name: (e.g., ASEDemo).
4. Select Launch Template: Choose the previously created launch template (ASETemp).
5. Choose Availability Zones:
  - Select at least two availability zones for better fault tolerance.
6. Click Next.

### Step 3: Configure Load Balancer & Health Checks

1. Attach a Load Balancer (Optional):
  - Choose Default Routing.
  - Create a Target Group (for default routing).
2. Enable Health Checks:
  - Turn on EC2 status checks.
  - Enable ELB (Elastic Load Balancing) health checks if using a load balancer.
3. Click Next.

### Step 4: Configure Scaling Policies

1. Desired Capacity: Set the number of instances to launch initially (e.g., 1).
2. Scaling Limits:
  - Minimum Capacity: 1
  - Maximum Capacity: 6
3. Target Tracking Scaling Policy:
  - Enable automatic scaling based on CPU utilization or other metrics.
4. Click Next.

### Step 5: Add Tags and Review

- 1.(Optional) Add tags to manage instances efficiently.
- 2.Click Next → Create Auto Scaling Group.

**Step 6: Verify Instance Creation**

1. The instance will automatically launch as per the defined scaling rules.
2. Navigate to EC2 Instances to verify the running instances.
3. Check Load Balancers → Auto Scaling Groups to ensure proper configuration.

**AWS Billing and Cost Management - Zero Budget Setup****Step 1: Enable Billing Alerts**

1. Go to AWS Management Console → Search for Billing Dashboard.
2. Click on Billing Preferences.
3. Enable Receive Billing Alerts.
4. Click Save Preferences.

**Step 2: Create a Billing Alarm in CloudWatch**

1. Go to AWS CloudWatch.
2. Click on Alarms → Create Alarm.
3. Click Select Metric → Choose Billing Metrics.
4. Click Total Estimated Charge.
5. Click Select Metric → Set the following:
  - Threshold Type: Static
  - Condition: Greater than
  - Threshold Value: \$0 (or any budget limit you set).
6. Click Next.

**Step 3: Configure Alarm Notifications**

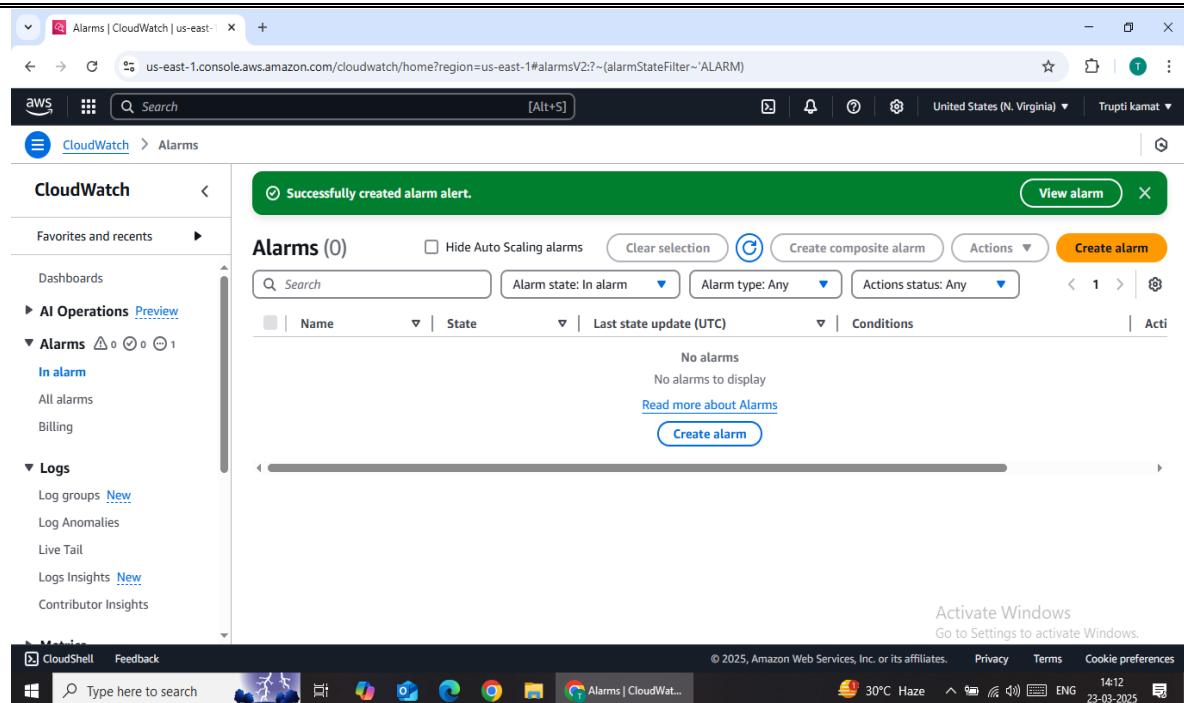
1. Choose Create New SNS Topic.
2. Enter a Topic Name (e.g., BillingAlert).
3. Enter your email address to receive alerts.
4. Click Create Topic and confirm the subscription from your email.

**Step 4: Set Up Free Tier Usage Monitoring**

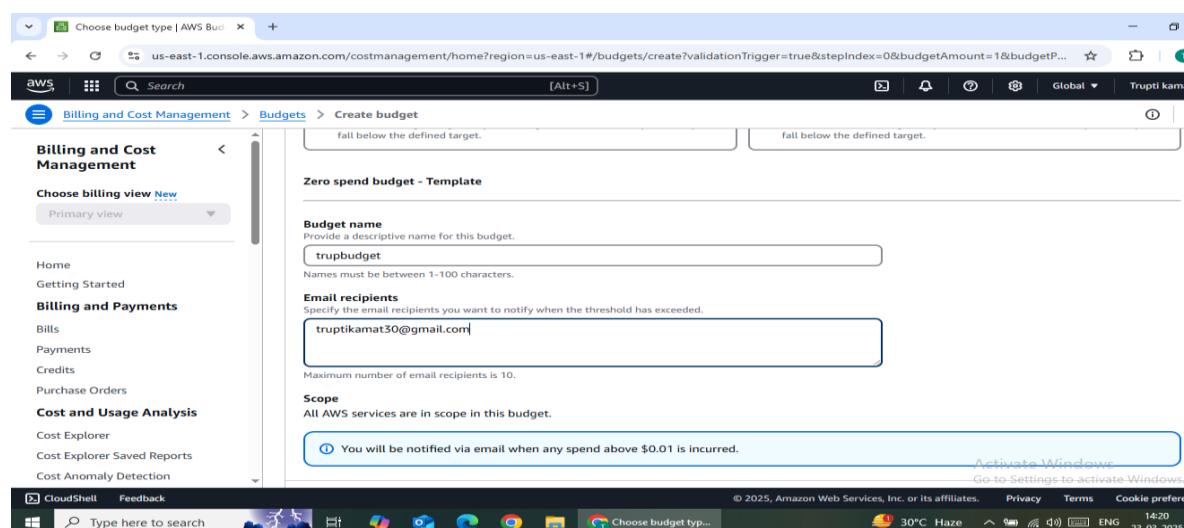
1. Go to AWS Cost Management Dashboard.
2. Click on Cost Anomaly Detection → Create an Anomaly Detector.
3. Select Service Category: Choose All Free Tier Services.
4. Click Next → Name the detector (e.g., FreeTierUsage).
5. Set an alert to notify when free tier limits are exceeded.

The image displays five distinct screenshots of the AWS CloudWatch service, arranged in a grid-like layout. Each screenshot shows a different aspect of CloudWatch's functionality:

- Top Left:** Shows the 'Overview' page with various metrics and CloudWatch Metrics.
- Top Right:** Shows the 'Select metric' interface for creating alarms.
- Middle Left:** Shows the 'Create log group' interface.
- Middle Right:** Shows the 'Create alarm' configuration interface.
- Bottom Left:** Shows the 'Create alarm' configuration interface for Step 1.



The screenshot shows the AWS CloudWatch Alarms console. A green success message at the top says "Successfully created alarm alert." with a "View alarm" button. Below it, a table header for "Alarms (0)" includes columns for Name, State, Last state update (UTC), and Conditions. A "Create alarm" button is prominently displayed. The left sidebar shows navigation links for AI Operations, Alarms (with 1 item), Logs, Metrics, and CloudShell.

The screenshot shows the AWS Billing and Cost Management Budgets console. It displays a "Zero spend budget - Template" form. Fields include "Budget name" (trupbudget), "Email recipients" (truptikamat30@gmail.com), and a note about email notifications. The left sidebar lists categories like Billing and Payments, Cost and Usage Analysis, and Global.

## 6. Using Lambda Service in AWS

Steps to Create an AWS Lambda Function

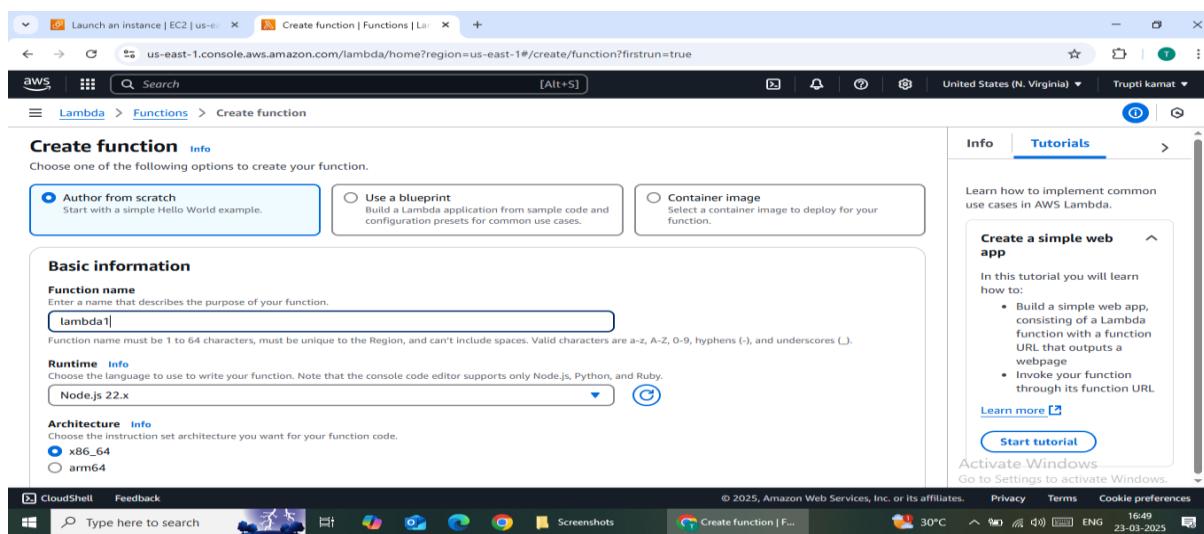
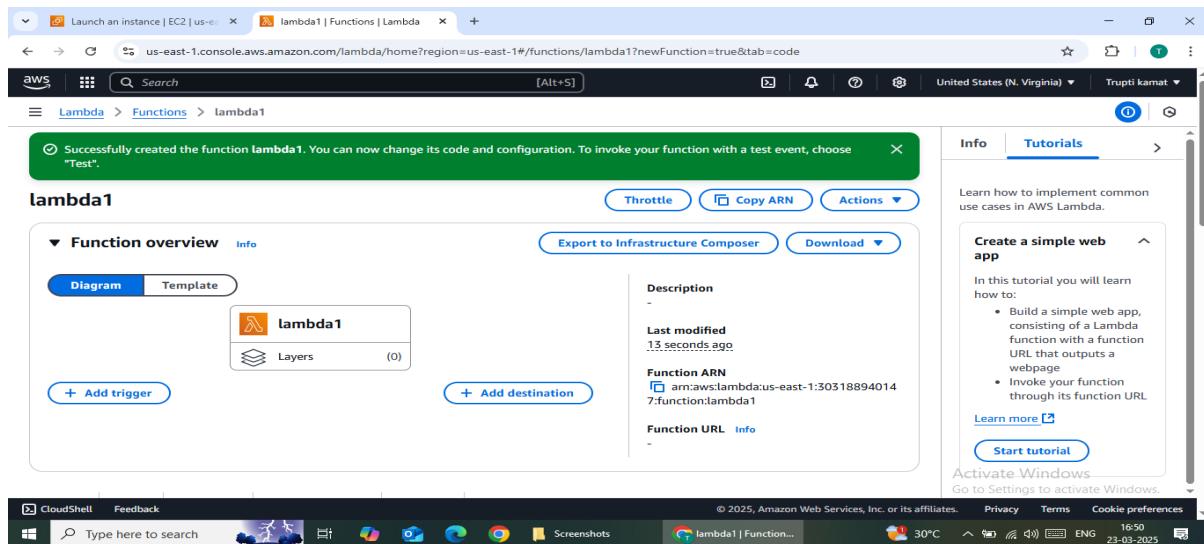
Method 1: Create a Lambda Function from Scratch

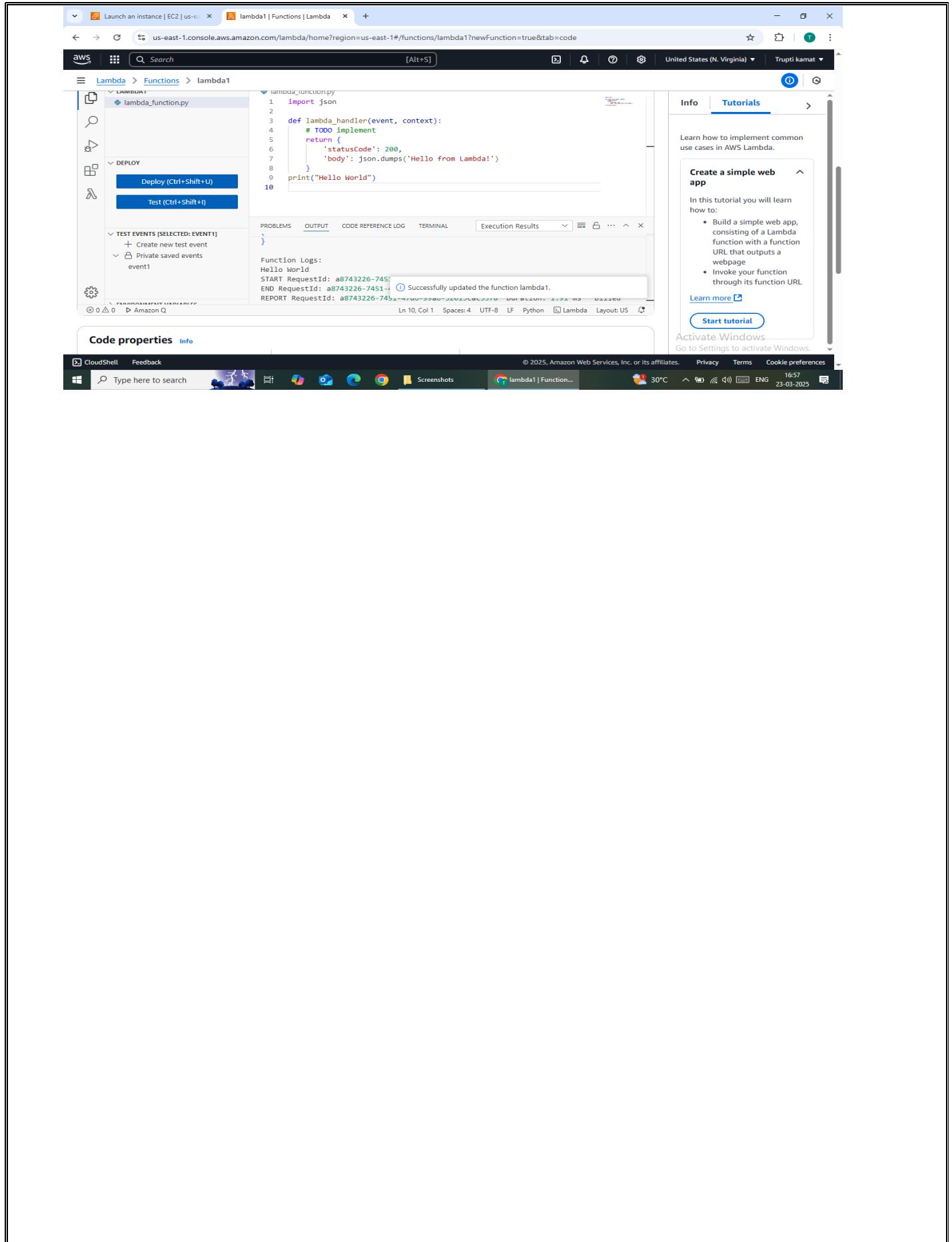
1. Search for Lambda in the AWS Management Console.
2. Click on Create Function.
3. Select “Author from Scratch”.
4. Enter Function Name (e.g., lambdaDemo).
5. Choose Runtime (e.g., Python 3.13).
6. Select Architecture (x86-64).
7. Click on Create a New Role.
8. Click on Create Function.
9. Once created, go to the Code section.
10. Enter the required code in the editor.
11. Click Deploy.
12. Go to Test, create a test event with any name, and save it.
13. Click Test to execute the function.

Method 2: Create a Lambda Function Using a Blueprint

1. Go to AWS Lambda and click Create Function.
2. Select “Use a Blueprint”.
3. In Basic Information, search for a suitable blueprint (e.g., canary).
4. Enter a Function Name (e.g., canaryLambda).
5. Create a New Role with basic Lambda permissions.
6. Create a New Rule for scheduling in AWS EventBridge.
7. Enter Rule Name (e.g., rate).
8. Provide a Description (e.g., rate (1 minute)).
9. Select Event Pattern or Schedule Type (e.g., rate-based invocation).
10. Click Create Function.

11. After function creation, go to Test and create an event.
12. Click Deploy → Test to validate execution.
13. Configure Environment Variables if needed.
14. Edit Online Triggers such as SNS or S3.
15. Go to EventBridge to delete the scheduled rule if no longer needed.





## 7. Launch a Linux Instance using EC2

### 1. Launch an EC2 Instance

- Log in to your AWS Console.
- Navigate to EC2 and click Launch Instance.
- Choose an Amazon Linux or Ubuntu AMI.
- Select an instance type (e.g., t2.micro for free tier).
- Configure security groups (allow SSH on port 22).
- Launch the instance and connect via SSH.

### 2. Connect to EC2 Instance via Terminal

- Use the following command to connect to your instance:

### 3. Install Required Packages

- Update package manager and install essential tools

### 4. Create and Run a C Program

Write your C code inside program.c, then save and exit (press ESC, then type :wq and hit Enter).

- Compile the C program

### 5. Create and Run a Shell Script

- Open Vim and create a shell script:

Add shell commands inside the file.

- Save and exit (:wq).

- Make the script executable

Run the shell script:

## 8. Create an Auto Scaling Group in AWS EC2

### Step 1: Create a Launch Template

1. Navigate to the AWS EC2 Dashboard.
2. Click on Launch Templates → Create Launch Template.
3. Provide a Launch Template Name (e.g., ASETemp).
4. (Optional) Provide a Version Description (e.g., Demo).
5. AMI Selection:
  - Choose an Amazon Machine Image (AMI).
  - Select Windows Server 2019 Base (Free Tier Eligible).
6. Instance Type: Choose t2.micro (Free Tier Eligible).
7. Key Pair:
  - Create a new key pair (e.g., asg) or use an existing one.
8. Security Group: Select an existing security group or use the default one.
9. Click Create Launch Template.
10. After creation, click View Template.

### Step 2: Create an Auto Scaling Group

1. Go to Auto Scaling Groups.
2. Click Create Auto Scaling Group.
3. Provide Name: (e.g., ASEDemo).
4. Select Launch Template: Choose the previously created launch template (ASETemp).
5. Choose Availability Zones:
  - Select at least two availability zones for better fault tolerance.
6. Click Next.

### Step 3: Configure Load Balancer & Health Checks

1. Attach a Load Balancer (Optional):
  - Choose Default Routing.
  - Create a Target Group (for default routing).
2. Enable Health Checks:
  - Turn on EC2 status checks.
  - Enable ELB (Elastic Load Balancing) health checks if using a load balancer.
3. Click Next.

### Step 4: Configure Scaling Policies

1. Desired Capacity: Set the number of instances to launch initially (e.g., 1).
2. Scaling Limits:
  - Minimum Capacity: 1
  - Maximum Capacity: 6
3. Target Tracking Scaling Policy:
  - Enable automatic scaling based on CPU utilization or other metrics.
4. Click Next.

### Step 5: Add Tags and Review

1. (Optional) Add tags to manage instances efficiently.
2. Click Next → Create Auto Scaling Group.

#### Step 6: Verify Instance Creation

1. The instance will automatically launch as per the defined scaling rules.
2. Navigate to EC2 Instances to verify the running instances.
3. Check Load Balancers → Auto Scaling Groups to ensure proper configuration.

The screenshot shows the AWS CloudShell interface. The browser tab is titled "Create launch template | EC2". The URL is "us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateTemplate:autoScalingGuidance=true". The page shows a success message: "Successfully created AsgTemp1(lt-0274266c46ccbb77).". Below this, there's an "Actions log" section and a "Next Steps" section. The "Next Steps" section includes links for "Launch an instance", "Launch instance from this template", "Create an Auto Scaling group from your template", "Create a Auto Scaling group", and "Create Spot Fleet". The status bar at the bottom shows "CloudShell Feedback", "Type here to search", "Create launch tem...", "Spotify Free", "29°C", "ENG 17:55 23-03-2025".

The screenshot shows the AWS CloudShell interface. The browser tab is titled "Create Auto Scaling group | EC2". The URL is "us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateAutoScalingGroup:launchTemplateId=lt-0274266c46ccbb77&launchTemplateVersion=1". The page shows a "Scaling Info" section with "Min desired capacity" set to 1 and "Max desired capacity" set to 6. It also shows options for "Automatic scaling - optional" (No scaling policies vs Target tracking scaling policy), and a "Scaling policy name" field containing "Target Tracking Policy". The status bar at the bottom shows "CloudShell Feedback", "Type here to search", "Create Auto Scal...", "Spotify Free", "Rain...", "ENG 18:04 23-03-2025".

The screenshot shows the AWS CloudShell interface. The browser tab is titled "Auto Scaling groups | EC2". The URL is "us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#AutoScalingGroups". The page displays a table for "Auto Scaling groups (1)". The single entry is "ASGDemo", which has a "Launch template/configuration" of "AsgTemp1 | Version Default", "Instances" count of 0, and "Desired capacity" of 1. The status is "Updating capacity...". The status bar at the bottom shows "CloudShell Feedback", "Type here to search", "Auto Scaling group...", "Spotify Free", "28°C", "ENG 18:07 23-03-2025".