

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

**Aim :** Imagine you're developing a distributed application spanning across IBM Cloud and AWS. You want to ensure seamless communication between components deployed on both platforms, particularly for notifying about events or triggering actions based on certain conditions.

The Event Notification Service (ENS) on IBM Cloud and the Simple Notification Service (SNS) on AWS are both messaging services designed to facilitate real-time communication and event-driven workflows within their respective cloud ecosystems.

### **Exercises :**

#### **Setting up IBM Cloud ENS:**

Create an IBM Cloud account if you haven't already.

Access the IBM Cloud dashboard and navigate to the Event Notification Service.

Define your event sources and configure event rules to specify the conditions triggering notifications.

Generate necessary credentials or access tokens for authentication.

Subscribe your application endpoints (e.g., HTTP endpoints, messaging queues) to receive notifications.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

Steps :

## 1. Search for Event Notification Service

The screenshot shows the IBM Cloud dashboard with a search bar at the top containing the text "event notification". Below the search bar, the "Catalog Results" section displays a list of services. The "Event Notifications" service is highlighted with a cursor, indicating it is selected. Other visible services include "IBM Cloud Activity Tracker", "Event Streams", "Managed Network Security Services", and "Functions". To the right of the catalog, there are cards for "Watson Studio", "Build a virtual machine", and "Choose a Database". At the bottom of the dashboard, there are sections for "News", "Recent support cases", "Planned maintenance", and "IBM Cloud status". The status map shows "Clear skies!"

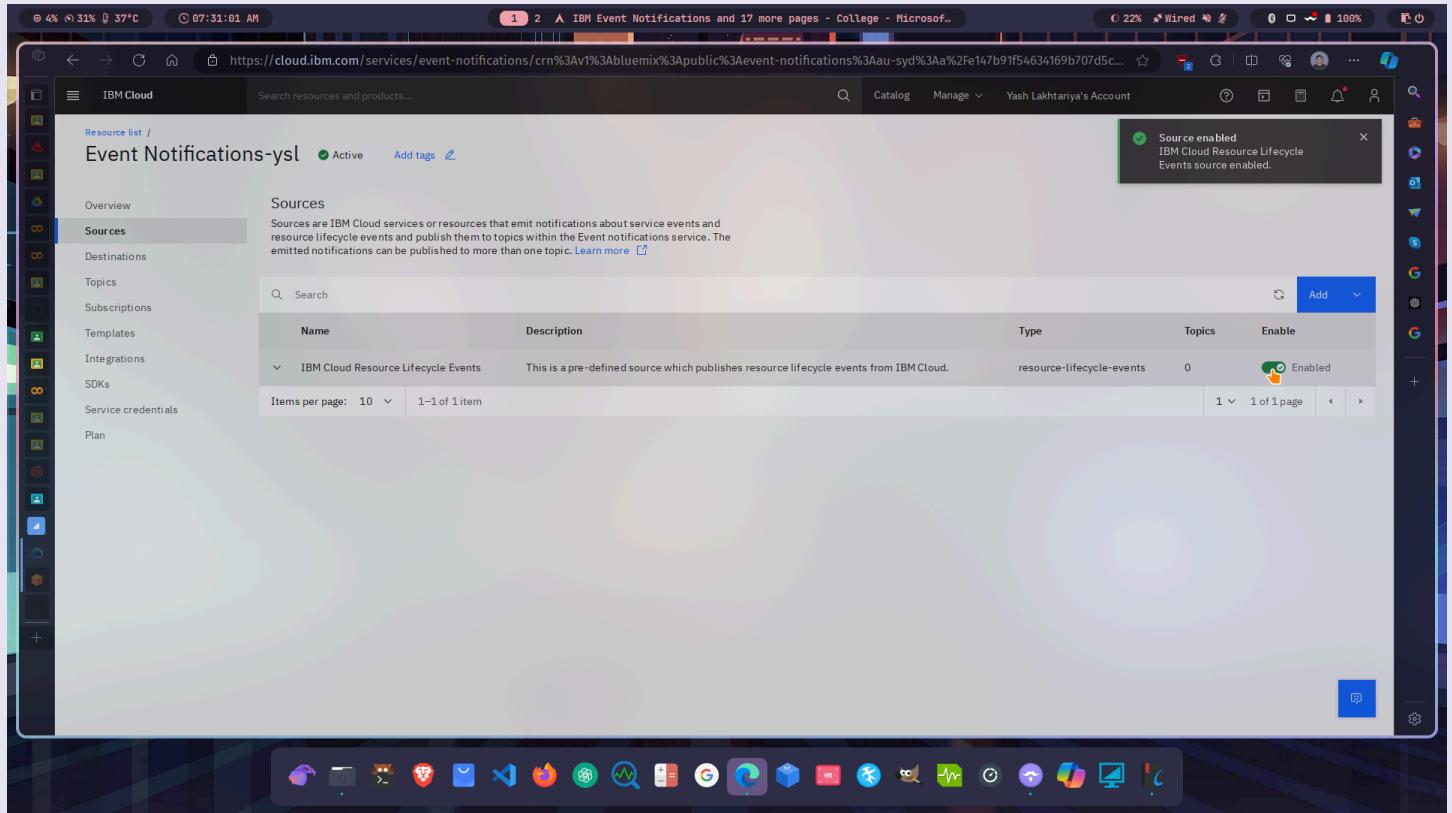
Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

## 2. Enter name, region and other details and create the service.

The screenshot shows the IBM Cloud Catalog interface for creating a new service. The URL in the address bar is <https://cloud.ibm.com/catalog/services/event-notifications>. The left sidebar lists 'IBM Cloud' services, including 'Event Notifications'. The main content area displays two plan options: 'Lite' and 'Standard'. The 'Lite' plan is selected, described as 'Free' and 'Financial Services Validated'. It includes a summary of capabilities: unlimited ingested events, 10 topics, 2 filters per topic, 5 destinations, 20 outbound emails, 20 outbound SMSes, 20 outbound webhooks, and 1000 notifications per push destination. Subscriptions are limited to 10, and each can have up to 3 email recipients. A note states that 'Lite plan services are deleted after 30 days of inactivity'. The 'Standard' plan is also listed, described as 'Click to view tiers and pricing detail' and 'Financial Services Validated'. On the right side, a 'Summary' panel shows the service details: Location: Sydney, Plan: Lite, Service name: Event Notifications-ysl, and Resource group: Default. Below the summary, there's a checkbox for accepting license agreements and a large blue 'Create' button. The bottom of the screen shows a dock with various application icons.

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 7**

### 3. Enable the default source



The screenshot shows the IBM Cloud Event Notifications service interface. The left sidebar includes options like Overview, Sources (which is selected), Destinations, Topics, Subscriptions, Templates, Integrations, SDKs, Service credentials, and Plan. The main content area displays a table of sources. One row is highlighted, showing 'IBM Cloud Resource Lifecycle Events' as the name, 'This is a pre-defined source which publishes resource lifecycle events from IBM Cloud.' as the description, 'resource-lifecycle-events' as the type, 0 topics, and an 'Enabled' status with a green checkmark icon. A modal window at the top right confirms the source is enabled: 'Source enabled IBM Cloud Resource Lifecycle Events source enabled.' Below the table, there are pagination controls for 'Items per page: 10' and '1-1 of 1 item'.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

#### 4. Search for Object Storage Resource

The screenshot shows the IBM Cloud dashboard interface. At the top, there is a search bar with the query "cloud object storage". Below the search bar, the "Catalog Results" section displays several service offerings:

- Cloud Object Storage module** (Software)
- Cloud Object Storage on VPC for SAP HANA Backup** (Software)
- Object Storage** (Service) - This item is highlighted with a cursor, indicating it is selected.
- Custom Migrations and Disaster Recovery as a Service** (Service)
- Komprise Elastic Data Migration** (Service)

On the right side of the dashboard, there are promotional cards for Watson Studio and a "View services with free Lite plans" section. The bottom of the screen shows the Microsoft Edge taskbar with various pinned icons.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

## 5. Create the Cloud Object Storage

The screenshot shows the IBM Cloud Object Storage creation page. At the top, there's a navigation bar with various icons and a search bar. Below it, a sidebar on the left lists different service categories like Compute, Data, and Security. The main content area is titled "Cloud Object Storage - IBM Cloud and 17 more pages - College -". It starts with a section for selecting a pricing plan, showing three options: Lite, Standard, and One Rate. The Lite plan is highlighted as free. To the right of the plans is a "Summary" panel displaying details: Region: Global, Plan: Lite, Service name: Cloud Object Storage-ysl, and Resource group: Default. Below the plans, there's a "Configure your resource" section where the service name is set to "Cloud Object Storage-ysl" and the resource group is set to "Default". There are also fields for Tags and Examples. A progress bar at the bottom indicates the process is "Creating...". The status bar at the bottom shows system information like battery level, signal strength, and temperature.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

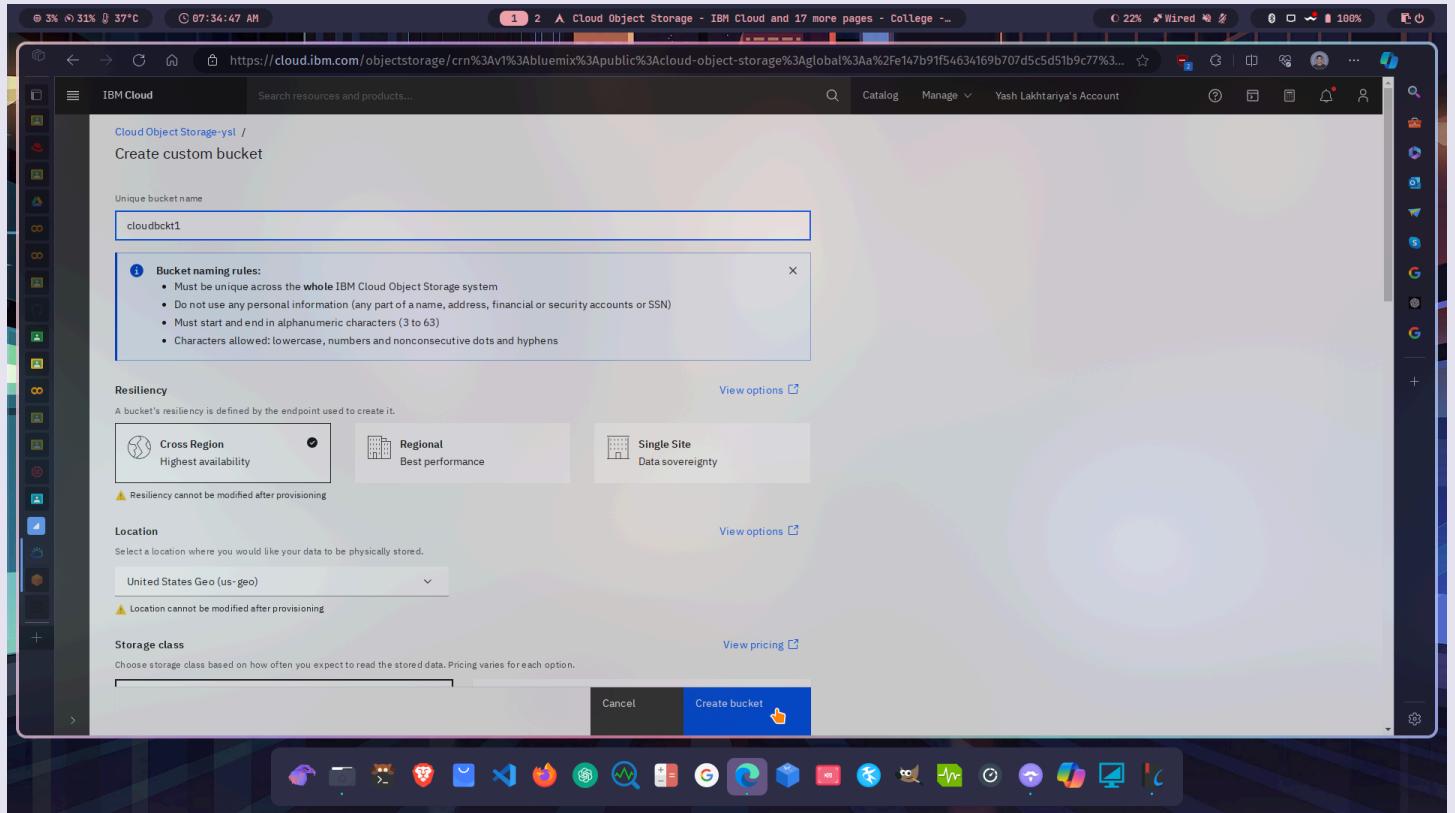
## 6. Now, create the custom bucket

The screenshot shows the 'Create bucket' page in the IBM Cloud Object Storage interface. The left sidebar lists 'IBM Cloud' services: Instances, Integrations, Endpoints, Documentation, and Billing. The main content area has a heading 'Create bucket' and a sub-heading: 'Get started by creating a bucket to store unstructured data. A bucket is a storage resource available in IBM Cloud Object Storage service. The bucket can be used to organize objects (storage data) along with their metadata. Create a custom bucket of your own, or choose from our pre-defined configurations.' Below this are four cards:

- Create a Custom Bucket**: 'Create a bucket by selecting bucket configurations that meet your object storage needs.' Includes a large blue circular icon and a 'Create' button.
- Quickly get started**: 'Create a Smart Tier storage class bucket in a region close to you and a service credential to connect your application.' Includes a 'Template' button and a 'Create' button.
- Archive your data**: 'Create a Smart Tier storage class bucket in a region close to you with an archive rule and a service credential to connect your application.' Includes a 'Template' button and a 'Create' button.
- Host a static website**: 'Create a Smart Tier storage class bucket in a region close to you with static web hosting configuration and a service credential to connect your application.' Includes a 'Template' button and a 'Create' button.

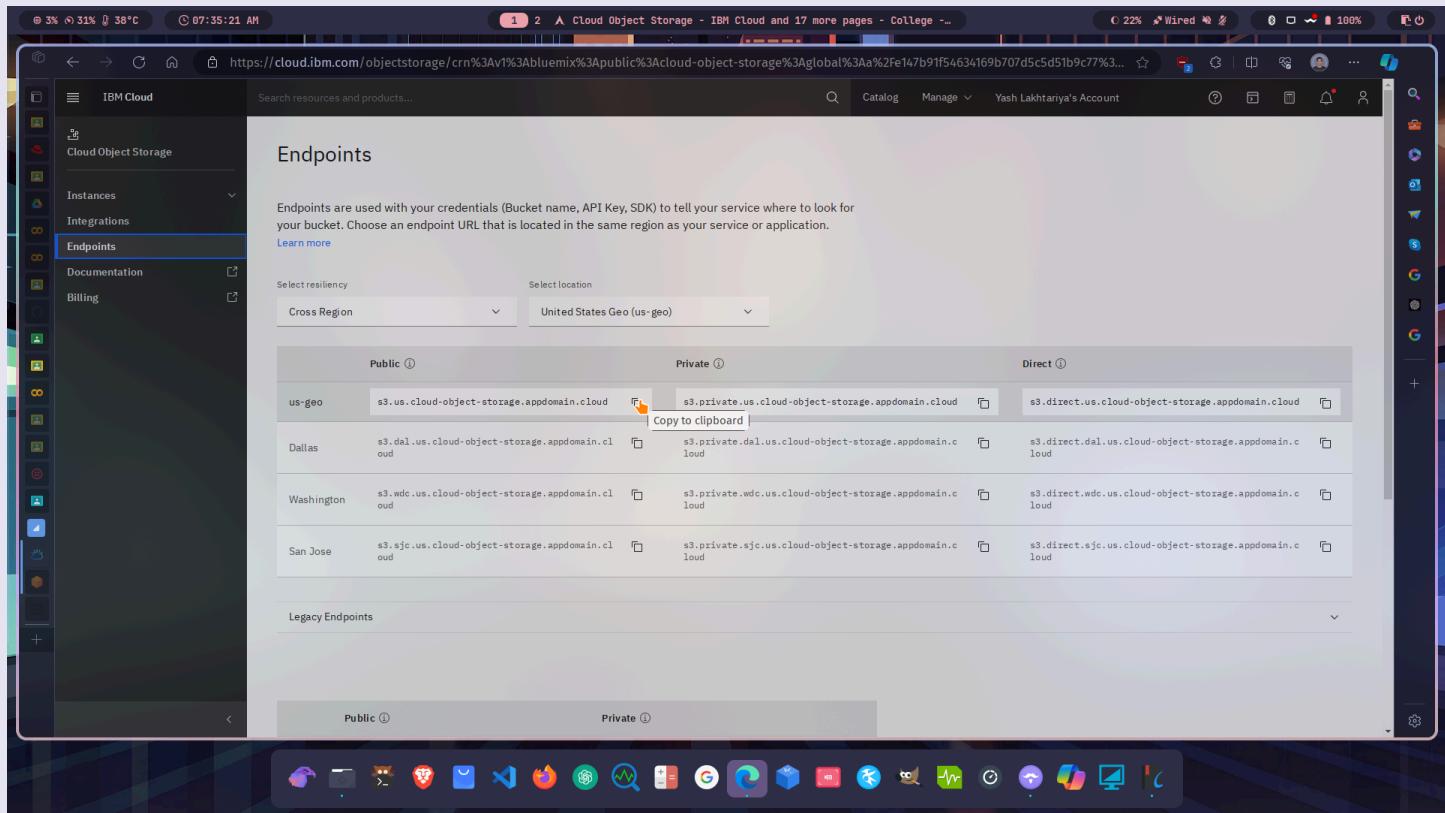
Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

## 7. Assign the name, select cross region and rest default options



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

## 8. Copy the public endpoint of the selected region of bucket



The screenshot shows the IBM Cloud Object Storage Endpoints page. The left sidebar has 'Cloud Object Storage' selected under 'Instances'. The main area is titled 'Endpoints' and contains instructions about using endpoints with credentials. It shows three categories: 'Public', 'Private', and 'Direct'. Under 'Public', there are four rows for 'us-geo':

- us-geo: s3.us.cloud-object-storage.appdomain.cloud (selected, highlighted with a yellow box)
- Dallas: s3.dal.us.cloud-object-storage.appdomain.cloud
- Washington: s3.wdc.us.cloud-object-storage.appdomain.cloud
- San Jose: s3.sjc.us.cloud-object-storage.appdomain.cloud

Each row has three columns: 'Public', 'Private', and 'Direct'. A 'Copy to clipboard' button is located between the Public and Private columns for each row. The 'Private' column for the us-geo row is currently active, showing 's3.private.us.cloud-object-storage.appdomain.cloud'. The 'Direct' column for the us-geo row is also visible.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

## 9. Turn on the IBM SMS service and configure it by adding the bucket details

The screenshot shows the IBM Cloud Event Notifications interface. On the left, there's a sidebar with options like Overview, Sources, Destinations (which is selected), Topics, Subscriptions, Templates, Integrations, SDKs, Service credentials, and Plan. The main area is titled "Event Notifications-ysl" and shows a "Destinations" section. It lists two items: "IBM Cloud SMS service" and "IBM Cloud Email service". Both entries have a "Type" column indicating "IBM SMS" or "IBM Email", a "Subscriptions" column with a value of 0, and a "Collect failed events" toggle switch which is off for both. A search bar and a pagination indicator (1 of 1 page) are also present.

This screenshot shows the same IBM Cloud Event Notifications interface, but with a modal dialog open on the right side titled "Collect Failed Events". The dialog is used to find a Cloud Object Storage (COS) bucket to store failed delivery events. It includes fields for "Instance name" (set to "Cloud Object Storage-ysl"), "Bucket name" (set to "cloudbkt2"), and "Endpoint" (set to "https://s3.us.cloud-object-storage.appdomain.cloud/"). At the bottom of the dialog are "Cancel" and "Save" buttons, with "Save" being highlighted with a cursor icon.

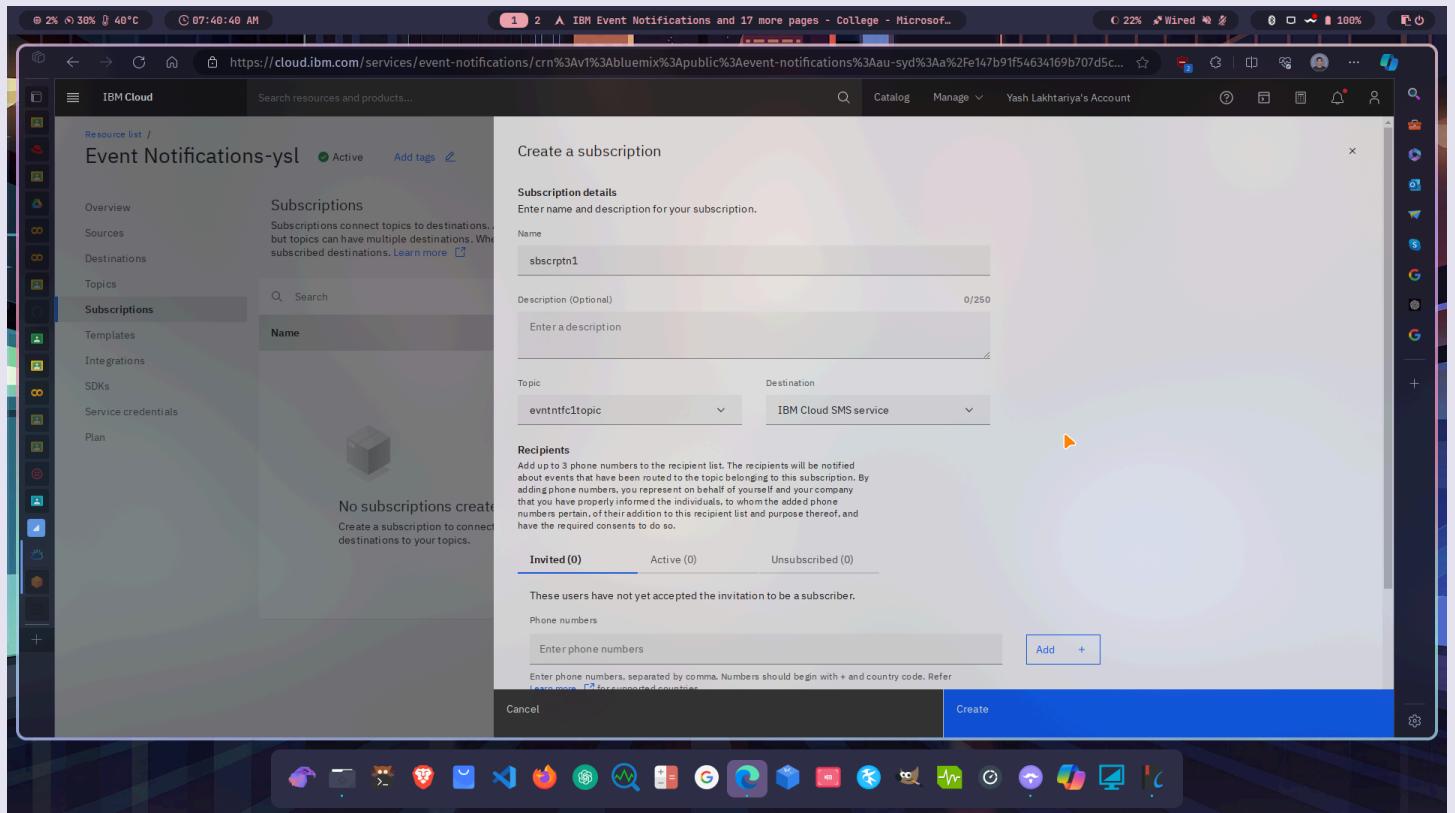
Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

## 10. Now, create a topic with the default source

The screenshot shows the IBM Event Notifications service within the IBM Cloud interface. The left sidebar lists various resources like Overview, Sources, Destinations, Topics, Subscriptions, Templates, Integrations, SDKs, Service credentials, and Plan. The 'Topics' section is selected, showing a message 'No topics created' and a placeholder 'Create a topic to apply condition sources and route events.' The main panel is titled 'Event Notifications-ysl' and shows 'Topic details'. It prompts to 'Provide a name and description for your topic'. The 'Name' field contains 'evntnfc1topic'. Below it is a 'Description (Optional)' field with a placeholder 'Enter a description'. Under the 'Sources' section, 'Sources for your topic' are listed, with 'IBM Cloud Resource Lifecycle Events' selected. There are dropdowns for 'Event type (Optional)' and 'Event subtype (Optional)', both currently set to 'Select a type' and 'Select a subtype' respectively. An 'Advanced conditions-JSON path (Optional)' field contains the JSON path '\$.data.severity == 'LOW' && \$.\* == 'any''. At the bottom right of the panel is a blue 'Create' button with a hand cursor icon.

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 7**

## 11. Create the subscription of the topic we created for SMS service



The screenshot shows the IBM Cloud Event Notifications service interface. On the left, there's a sidebar with options like Overview, Sources, Destinations, Topics, and Subscriptions. The Subscriptions section is active, showing a message: "No subscriptions created. Create a subscription to connect destinations to your topics." In the center, a modal window titled "Create a subscription" is open. It has a "Subscription details" section where "Name" is set to "sbscrptn1". Below it is a "Description (Optional)" field with placeholder text "Enter a description". Under "Topic", the dropdown is set to "evntntfc1topic". Under "Destination", it is set to "IBM Cloud SMS service". At the bottom of the modal, there are tabs for "Invited (0)", "Active (0)", and "Unsubscribed (0)". A note about recipients is present: "Add one or more numbers to the recipient list. The recipients will be notified about events that have been routed to the topic belonging to this subscription. By adding phone numbers, you represent on behalf of yourself and your company that you have properly informed the individuals, to whom the added phone numbers pertain, of their addition to this recipient list and purpose thereof, and have the required consents to do so." Below this, there's a "Phone numbers" section with a text input field "Enter phone numbers" and a "Create" button. The overall interface is modern with dark-themed UI elements.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

## 12. Invite (add) phone number to use SMS service

The screenshot shows the IBM Cloud Event Notifications interface. On the left, there's a sidebar with options like Overview, Sources, Destinations, Topics, Subscriptions (which is selected), Templates, Integrations, SDKs, Service credentials, and Plan. The main area is titled "Event Notifications-ysl" and shows a "Subscriptions" section. It says "Subscriptions connect topics to destinations, but topics can have multiple destinations. When subscribed destinations, Learn more". Below this, there's a search bar and a table header with columns "Name", "Topic", "Destination", and "Recipients". A single row is shown: "evntrnfc1topic" is connected to "IBM Cloud SMS service". Under "Recipients", it says "Add up to 3 phone numbers to the recipient list. The recipients will be notified about events that have been routed to the topic belonging to this subscription. By adding phone numbers, you represent on behalf of yourself and your company that you have properly informed the individuals, to whom the added phone numbers pertain, of their addition to this recipient list and purpose thereof, and have the required consents to do so." There are tabs for "Invited (1)", "Active (0)", and "Unsubscribed (0)". Below this, it says "These users have not yet accepted the invitation to be a subscriber." and has a "Phone numbers" input field with "Enter phone numbers" placeholder text and an "Add +" button. A table below shows one item: "+916351573711". At the bottom, there are buttons for "Cancel" and "Create". The top of the browser window shows the URL "https://cloud.ibm.com/services/event-notifications/crn%3Av1%3Abuemix%3Apublic%3Aevent-notifications%3Aau-syd%3Aa%2Fe147b91f54634169b707d5c..." and the status bar shows battery at 2%, signal strength, and other system info.

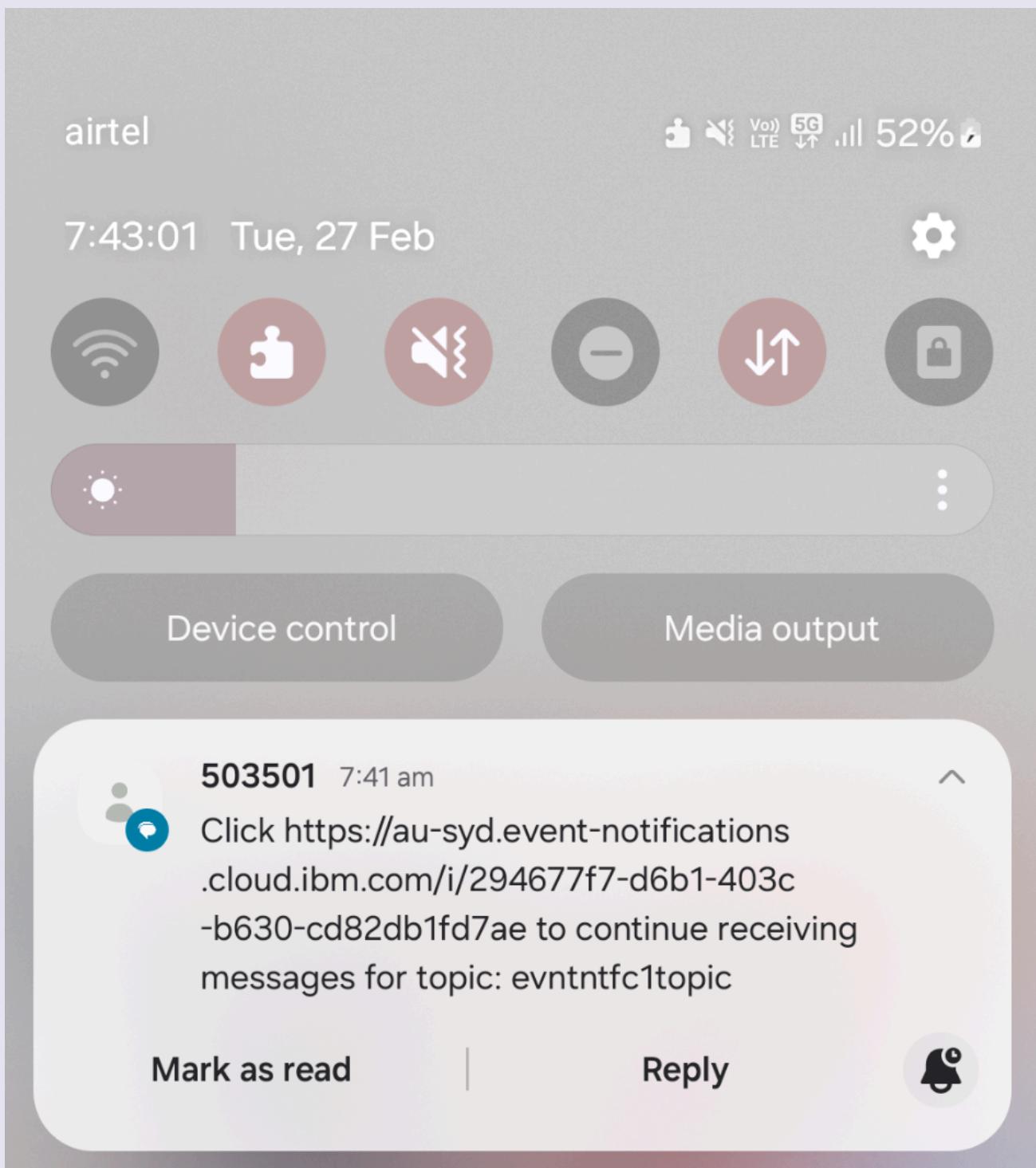
Name - Yash Lakhtariya

Enrollment number - 21162101012

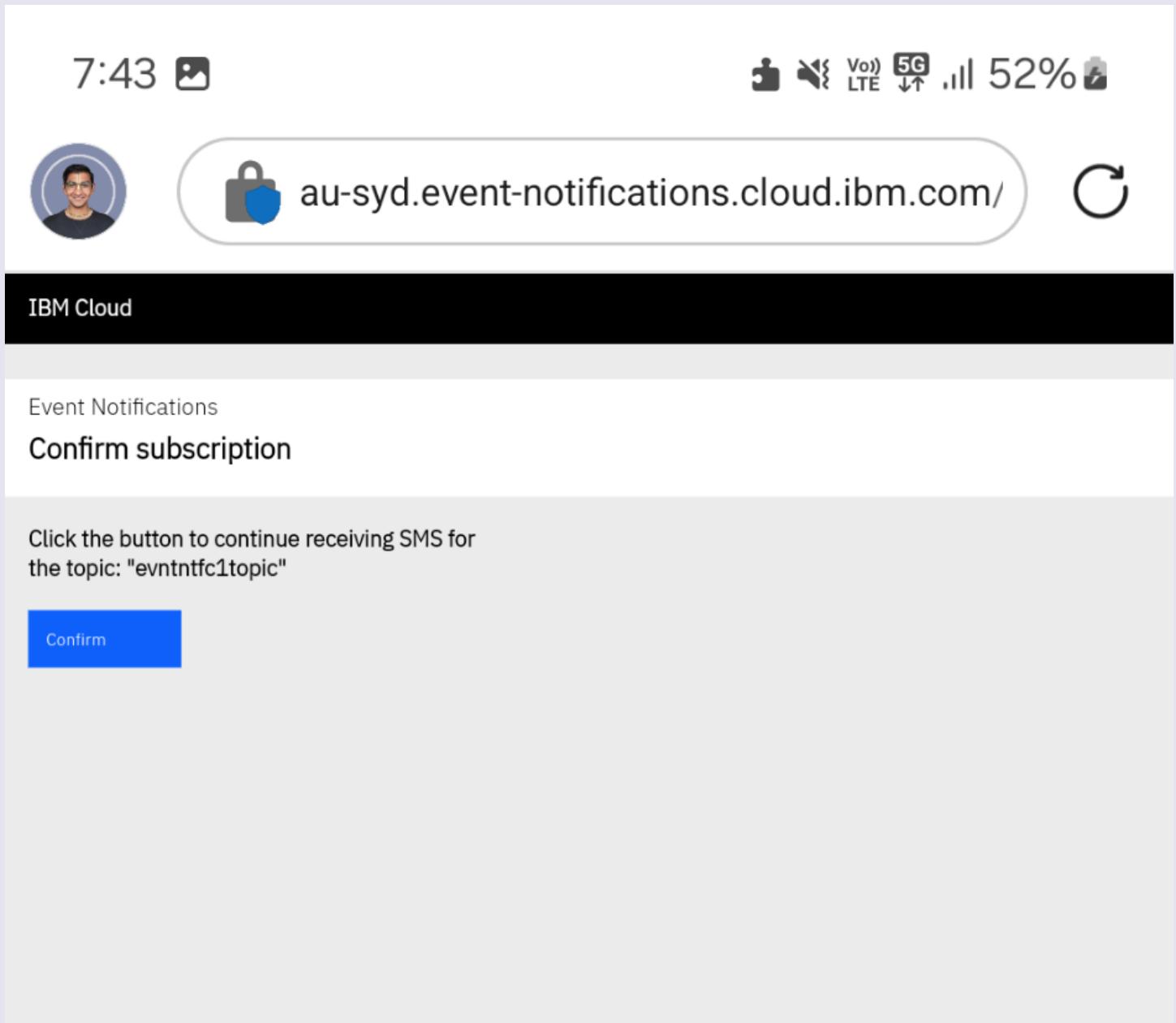
Branch - CBA      Batch - 61

EADC Practical 7

13. Confirm the subscription by URL received as SMS



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

The screenshot shows a mobile phone interface with the following details:

- Top bar: 7:43, camera icon, signal strength, VoLTE, 5G, battery level (53%).
- Header: A circular profile picture of a person with glasses, followed by a lock icon and the URL "au-syd.event-notifications.cloud.ibm.com/".
- Header bar: IBM Cloud.
- Main content area:
  - Section title: Event Notifications
  - Message: Successfully Subscribed!
  - Text: Since you've accepted the invitation, you are successfully subscribed to topic: "evntntfc1topic"
  - Text: Your phone number +916351573711 has been subscribed and will start receiving SMS sent by admin via IBM Cloud Event Notifications.
  - Link: If you wish to not receive sms, click or visit [link](#)

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

## 14. Now, create Speech to Text service to check if subscription is successful

The screenshot shows the IBM Cloud Catalog interface. The user is creating a new service named "Speech to Text". The service is listed as "Free" with the location set to "Sydney". The "Lite" plan is selected, offering 500 Minutes per Month at no cost. The "Create" button is highlighted with a cursor, indicating the action being performed.

**Catalog / Speech to Text**  
Low-latency, streaming transcription

**Create**      **About**

Type: Service  
Provider: IBM  
Last updated: 01/17/2024  
Category: AI / Machine Learning  
Compliance: EU Supported, HIPAA Enabled, IAM-enabled  
Location: Sydney, Frankfurt, London, Tokyo, Washington DC, Dallas  
Related links: API docs, Docs, Terms

Select a location: Sydney (au-syd)

Select a pricing plan  
Displayed prices do not include tax. Monthly prices shown are for country or location: [United States](#)

Plan	Features and capabilities	Pricing
Lite	500 Minutes per Month	Free
Plus - NEW!	Minutes Per Month Simple Volume Tiers	Click to view tiers and pricing details
Premium - NEW!	Everything in Plus Plan, plus... The Premium Plan provides the same features and benefits of using the Plus Plan, but with significantly greater capacity for concurrent transcriptions, enhanced security features to ensure that your data is isolated and encrypted end-to-end while in transit and at rest, and HIPAA readiness. Up to 500 concurrent transcriptions with the option to add more, and 150,000 free minutes to start. HIPAA Enabled in Washington DC (US-East) & Dallas (US-South) EU Supported available in Frankfurt (EU-DE)	

I have read and agree to the following license agreements:  
[Terms](#)

**Create**      **Add to estimate**

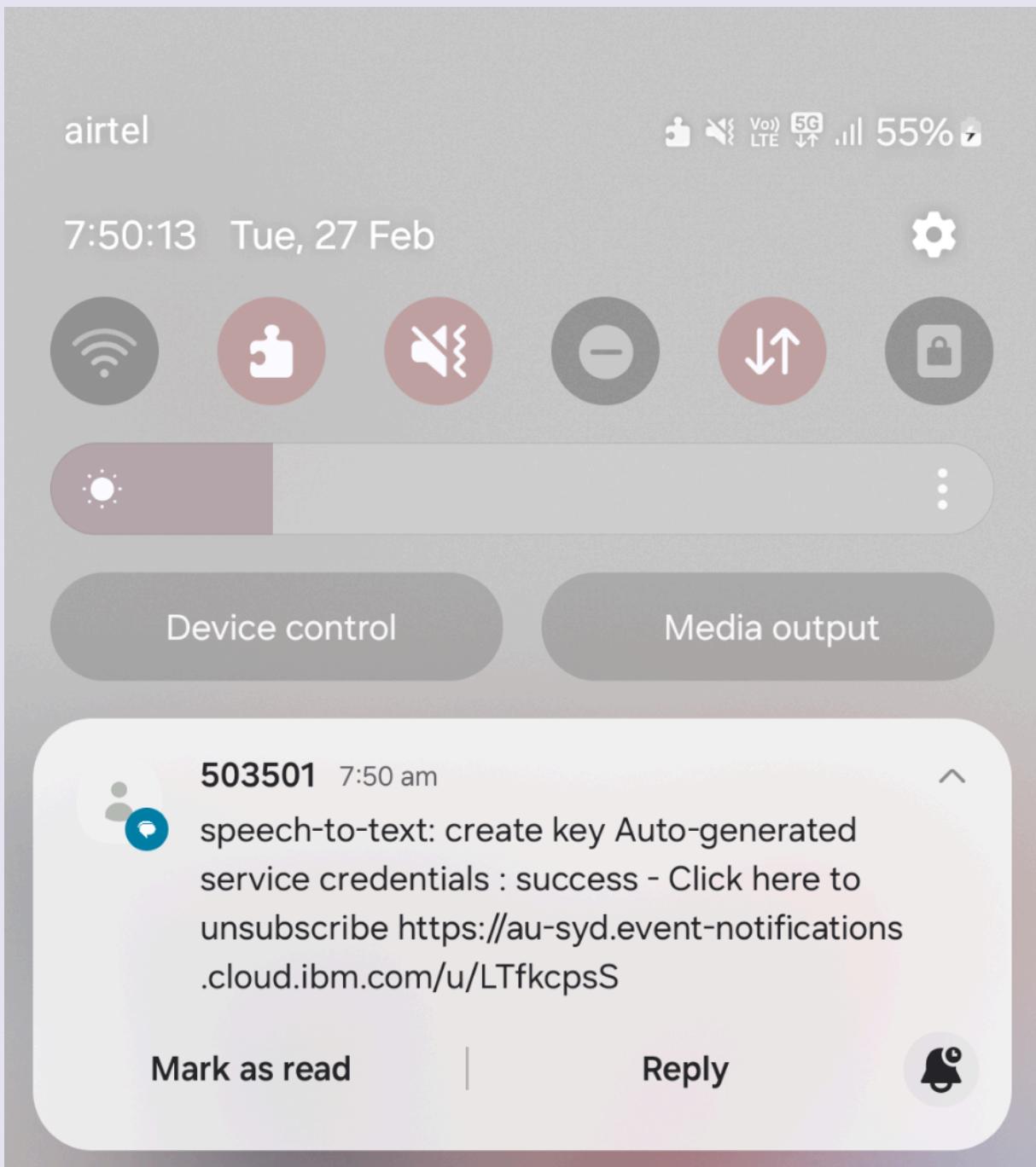
Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

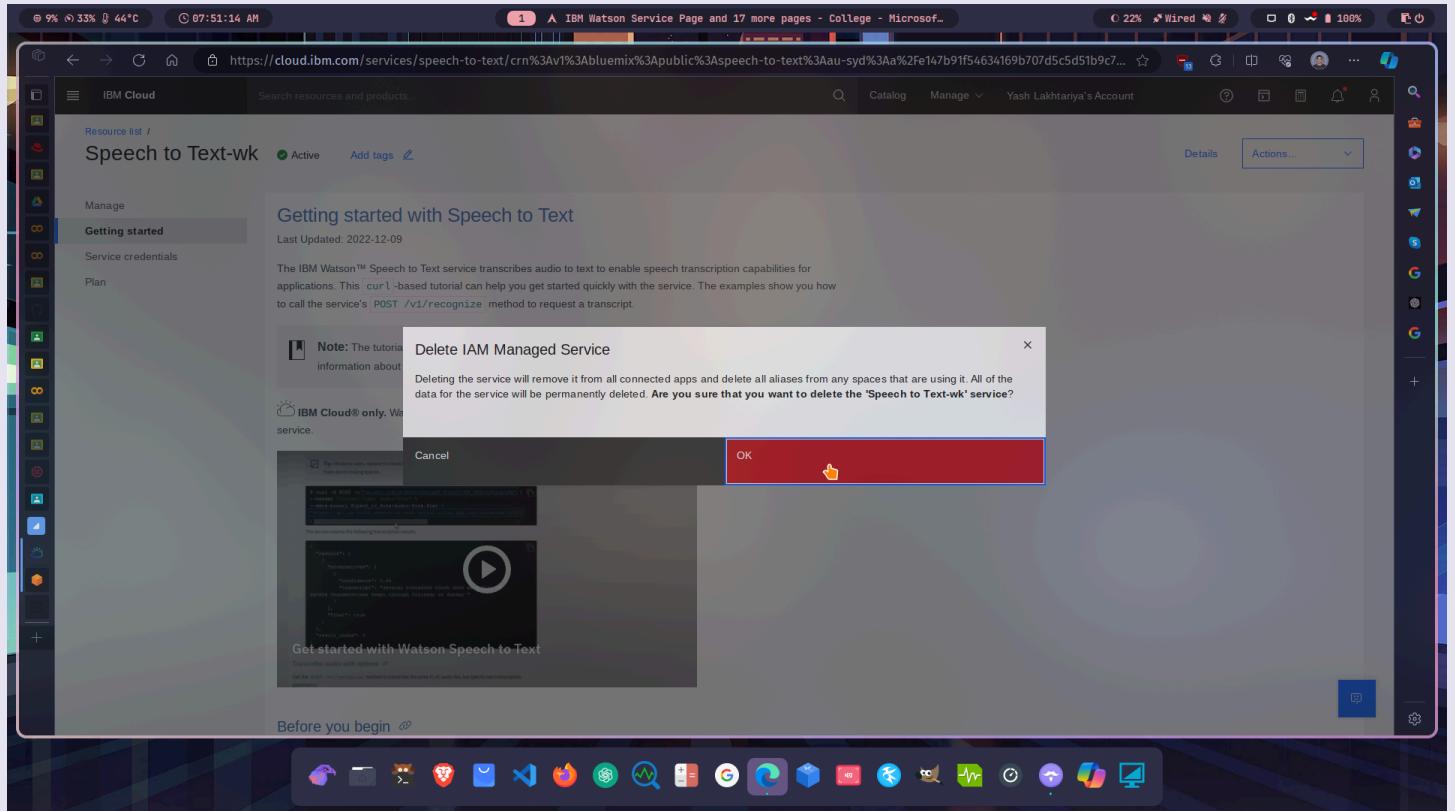
EADC Practical 7

15. Check if SMS is received of creation of service



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

## 16. Similarly check if SMS arrives on service deletion also



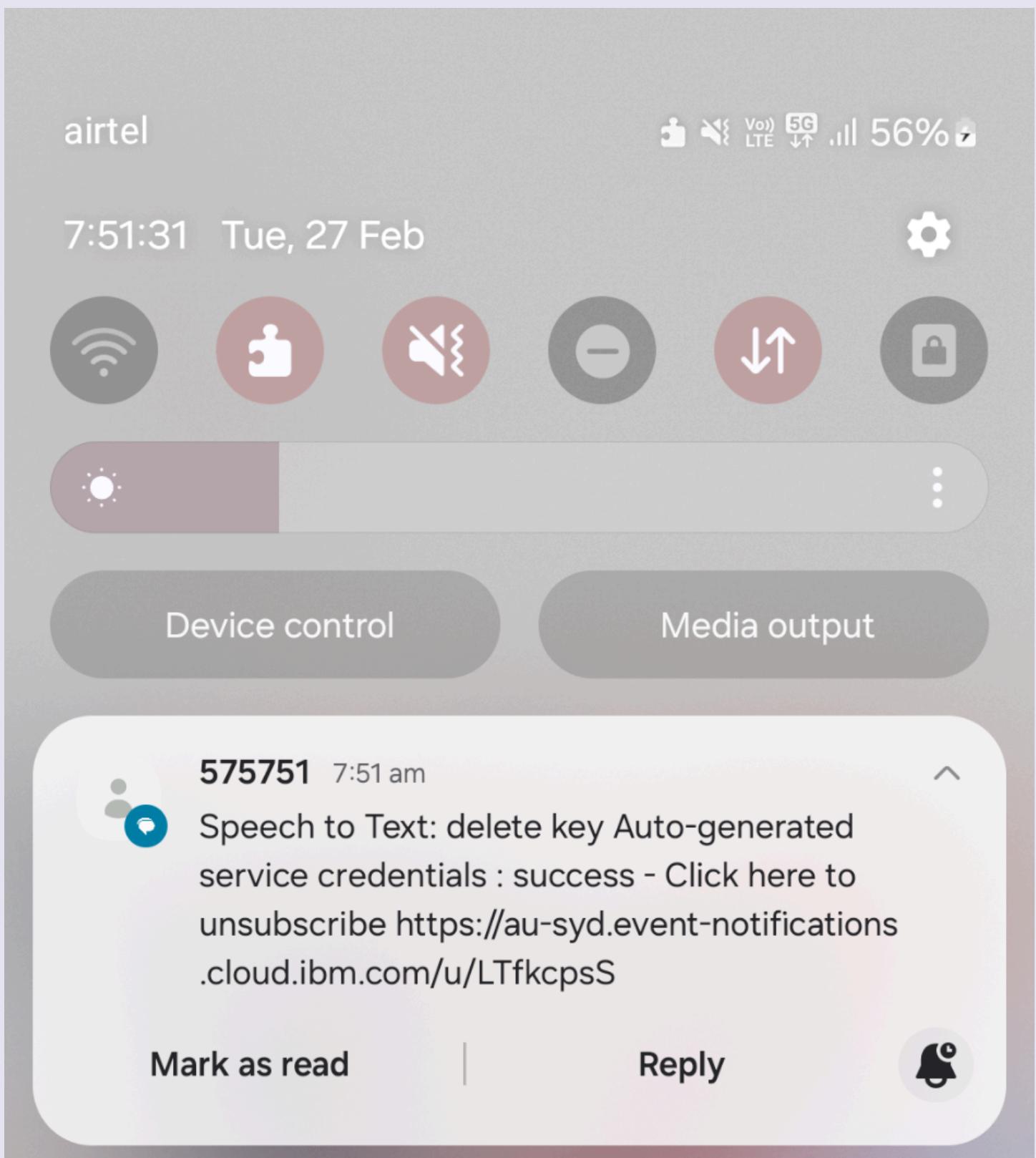
**Name - Yash Lakhtariya**

**Enrollment number - 21162101012**

**Branch - CBA      Batch - 61**

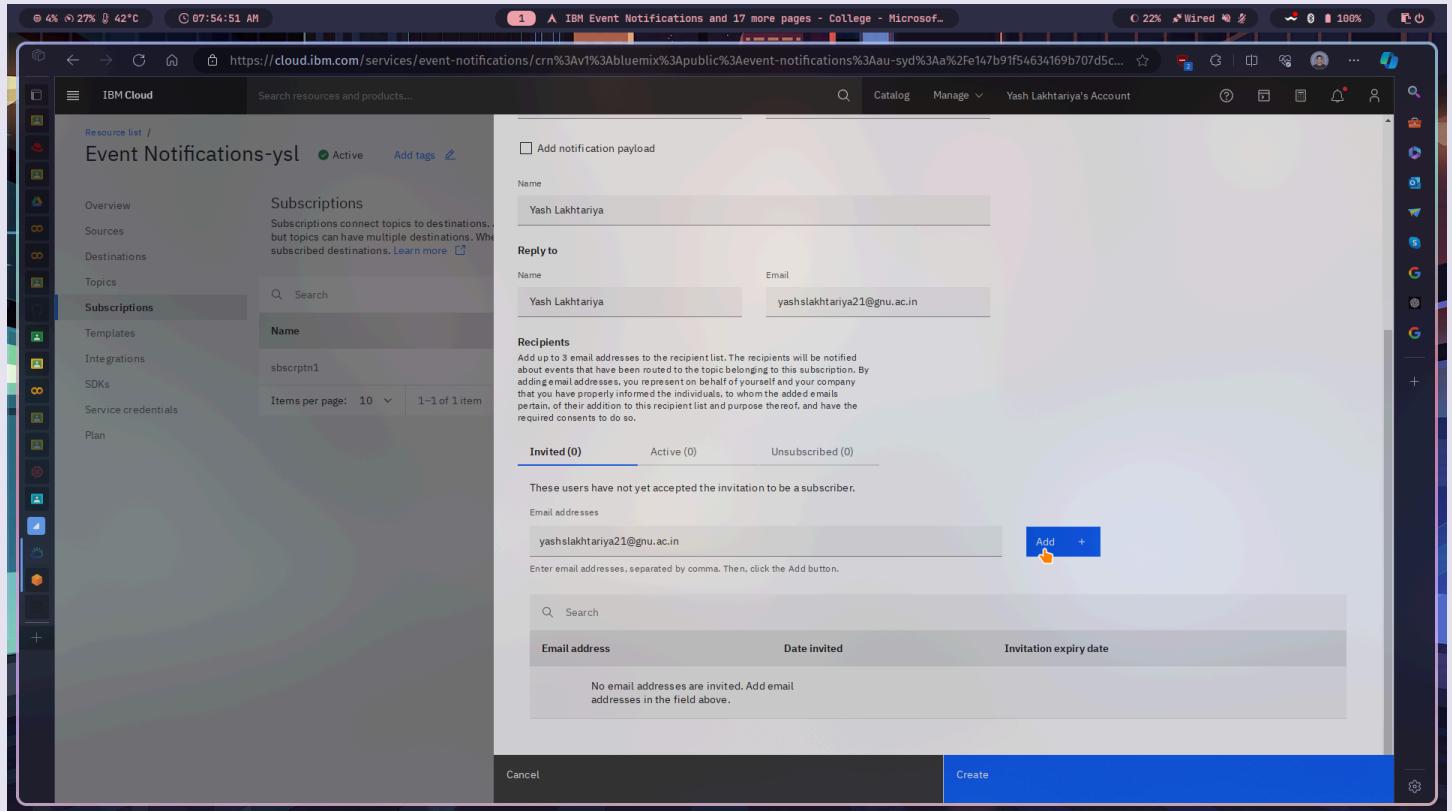
**EADC Practical 7**

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7



**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 7**

## 17. Now, configure the IBM Email Service and add the email to use



The screenshot shows the IBM Cloud Event Notifications interface. On the left, there's a sidebar with options like Overview, Sources, Destinations, Topics, Subscriptions (which is selected), Templates, Integrations, SDKs, Service credentials, and Plan. The main area is titled "Event Notifications-ysl" and shows a "Subscriptions" section. A table lists one item: "sbscrptn1". The details for this subscription include:

- Name:** Yash Lakhtariya
- Reply to:** Yash Lakhtariya, yashslakhtariya21@gnu.ac.in
- Recipients:** A note states: "Add up to 3 email addresses to the recipient list. The recipients will be notified about events that have been routed to the topic belonging to this subscription. By adding email addresses, you represent on behalf of yourself and your company that you have properly informed the individuals, to whom the added emails pertain, of their addition to this recipient list and purpose thereof, and have the required consents to do so." Below this, there are three tabs: "Invited (0)" (underlined), "Active (0)", and "Unsubscribed (0)".
- Email addresses:** A text input field contains "yashslakhtariya21@gnu.ac.in". To its right is a blue "Add" button with a plus sign, which has a red circle and a hand cursor icon pointing at it.
- Table:** A table with columns "Email address", "Date invited", and "Invitation expiry date". It displays the single entry: "yashslakhtariya21@gnu.ac.in" (Email address), "2024-05-15" (Date invited), and "2024-05-15" (Invitation expiry date).

At the bottom, there are "Cancel" and "Create" buttons, with "Create" being the active button.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

## 18. Add the new subscription with 'Add notification payload' option ON

The screenshot shows the IBM Cloud Event Notifications interface. On the left, there's a sidebar with options like Overview, Sources, Destinations, Topics, Subscriptions (which is selected), Templates, Integrations, SDKs, Service credentials, and Plan. The main area shows a list of existing subscriptions: 'sbscrptn1' and 'emailIntfc'. A modal window titled 'Edit emailIntfc' is open, showing the configuration for this subscription. The 'Subscription details' section has the name 'emailIntfc'. The 'Description (Optional)' field is empty. Under 'Topic', it is set to 'evntntfc1topic'. Under 'Destination', it is set to 'IBM Cloud Email service'. The 'Add notification payload' checkbox is checked. In the 'Recipients' section, there is one recipient listed: 'Yash Lakhtariya' with the email 'yashslakhtariya21@gnu.ac.in'. At the bottom of the modal, there are 'Cancel' and 'Save' buttons.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

## 19. Now, try creating Speech to Text service and check if email arrives

The screenshot shows the IBM Cloud Catalog interface. On the left, there's a sidebar with various service categories like Type, Provider, Category, and Compliance. The main area is titled 'Speech to Text' and shows a 'Create' button. Below it, there's a dropdown for 'Select a location' set to 'Sydney (au-syd)'. A section for 'Select a pricing plan' follows, with 'Lite' being the selected option. To the right, a summary panel displays the service details: 'Speech to Text', 'Location: Sydney', 'Plan: Lite', 'Service name: Speech to Text-o9', and 'Resource group: Default'. At the bottom right, there's a 'Create' button with a checked checkbox for accepting license agreements.

The screenshot shows the Gmail inbox. It displays 41 unread emails. One specific email from 'yashlakhtariya21@gnu.ac.in' is highlighted, with the subject 'Speech to Text: create instance Speech to Text-o9 : success'. The email body contains a link to the IBM Cloud event notifications page. To the right of the inbox, a detailed view of the email is shown, including the 'Details' tab which provides a JSON representation of the notification details. The JSON includes information such as the account ID, context, action (speech-to-text.instance.create), outcome (success), target name (Speech to Text-o9), and subject ID (IBMID-6940008YBM).

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 7**

## 20. Also, check the same on deleting the service if email arrives

The screenshot shows the IBM Cloud interface with the 'Speech to Text-o9' service selected for deletion. A modal dialog box is displayed, asking for confirmation to delete the service. The 'OK' button in the dialog is highlighted with a yellow arrow.

The screenshot shows an email inbox with several unread messages. One message is from 'Yash Lakhtariya' with the subject 'Speech to Text: delete key Auto-generated service credentials : success'. The message body contains JSON data about the deleted service key.

```

{
  "account_id": "e147b91f54634169b707d5c5d51b9c77",
  "context": {
    "activity": {
      "action": "speech-to-text-key-delete",
      "message": "Speech to Text: delete key Auto-generated service credentials"
    },
    "outcome": "success",
    "reason_reasonCode": 204,
    "target_name": "Auto-generated service credentials"
  },
  "subject_id": "IBMId-6940008YBM",
  "subject_name": "yashlakhtariya21@gnu.ac.in",
  "event_id": "0f3ae0a2-c20a-4d88-a07f-eac091ade2b3",
  "event_properties": {
    "created_at": "2024-02-27T02:29:56.410366619Z",
    "created_by": "IBMId-6940008YBM"
  },
  "resource_controller": "speech-to-text-autogen-key",
  "resource_id": "c973e070db0894b1e8d5dfaf583ae70a2",
  "state": "removed",
  "updated_at": "2024-02-27T02:30:33.722859418Z"
}

```

Note: This is not a marketing communication by IBM. You received this email because you have been subscribed by an IBM Cloud

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

21. Now, we can also check these services for toolchain notifications. Create a from scratch toolchain to do so

The screenshot shows the IBM Cloud DevOps Catalog interface. On the left, there's a sidebar with icons for Slack, DevOps Practices, DevSecOps, and Infrastructure as Code. The main area displays a grid of toolchain options:

- practices** by IBM: Continuously scan your deployed code based on DevSecOps best practices and Continuous Compliance(CC). Tools: CircleCI, Jenkins, SonarQube, GitLab CI.
- Deploy an application for VPC with multiple deployment strategies** by IBM: Deploy an application to virtual server instances in a VPC using deployment strategies such as Rolling, Blue-Green and Canary. Tools: CircleCI, Jenkins, SonarQube, GitLab CI.
- CI - Develop secure infrastructure as code with DevSecOps** by IBM: Deliver infrastructure as code using DevSecOps best practices and Continuous Integration(CI). Tools: CircleCI, Jenkins, SonarQube, GitLab CI.
- DevOps Insights Quick Start Demo** by IBM: Get to know DevOps Insights with this fully functional demo with populated Quality Dashboard, Team Dynamics, and more. Tools: Jenkins, SonarQube.
- Develop a Kubernetes app with Helm** by IBM: Continuously deliver a secure Docker app to a Kubernetes Cluster using a Helm Chart. Tools: CircleCI, Jenkins, SonarQube, GitLab CI.
- Develop and test microservices on Kubernetes with Helm** by IBM: Continuously deliver a microservices app on Kubernetes using quality gates and Helm release coordination. Tools: Jenkins, SonarQube, GitLab CI.
- Build your own toolchain** by IBM: For advanced users, create your toolchain from scratch. This option has a small orange arrow pointing to it.

The URL in the browser bar is: [https://cloud.ibm.com/devops/setup/deploy?repository=https%3A%2F%2Feu-de.git.cloud.ibm.com%2fopen-toolchain%2fempty-toolchain&env\\_id=ibm:yp:eu-de](https://cloud.ibm.com/devops/setup/deploy?repository=https%3A%2F%2Feu-de.git.cloud.ibm.com%2fopen-toolchain%2fempty-toolchain&env_id=ibm:yp:eu-de)

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

## 22. Remember to choose the same region (London here) where Continuous Delivery service is created

The screenshot shows the 'Continuous Delivery-ysl' resource details in the IBM Cloud interface. The 'Access' tab is selected, showing options to manage toolchains and assign access. A 'Details' panel on the right provides an overview of the service, including its offering (Continuous Delivery), creation date (2/22/2024), resource group (Default), user (yashlakhtariya), location (London), and status (Active). There is also a 'Add tags' button.

The screenshot shows the 'Create a Toolchain' page. The 'Toolchain name' field contains 'empty-toolchain-20240227025148594'. The 'Select region' dropdown is set to 'London', and the 'Select a resource group' dropdown is set to 'Default'. At the bottom, there are 'Cancel' and a blue 'Create' button.

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Practical 7

23. The toolchain is successfully created, to add the integration click 'Add' button.

The screenshot shows the IBM Cloud DevOps Toolchains interface. The URL in the address bar is [https://cloud.ibm.com/devops/toolchains/bb827e32-4aca-445a-bd35-29268f087536?env\\_id=ibm:yp:eu-gb](https://cloud.ibm.com/devops/toolchains/bb827e32-4aca-445a-bd35-29268f087536?env_id=ibm:yp:eu-gb). The page title is "empty-toolchain-20240227025148594 - IBM Cloud". The main header includes "Catalog", "Manage", and "Yash Lakhtariya's Account". On the left, there's a sidebar with "Toolchains / empty-toolchain-20240227025148594" and "Add tags". Below the sidebar are "Overview", "Connections", and "Manage" buttons. A search bar says "Search resources and products...". A message box at the top right says "Your toolchain is ready! Quick start: You can now add tool integrations. For more details, view the documentation." Below the message are three sections: "Repositories 0 total", "Delivery pipelines 0 total", and "IBM Cloud tools 0 total". At the bottom right, there's a blue "Ask a question" button and a "Feedback" icon. The overall theme is light blue and white.

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

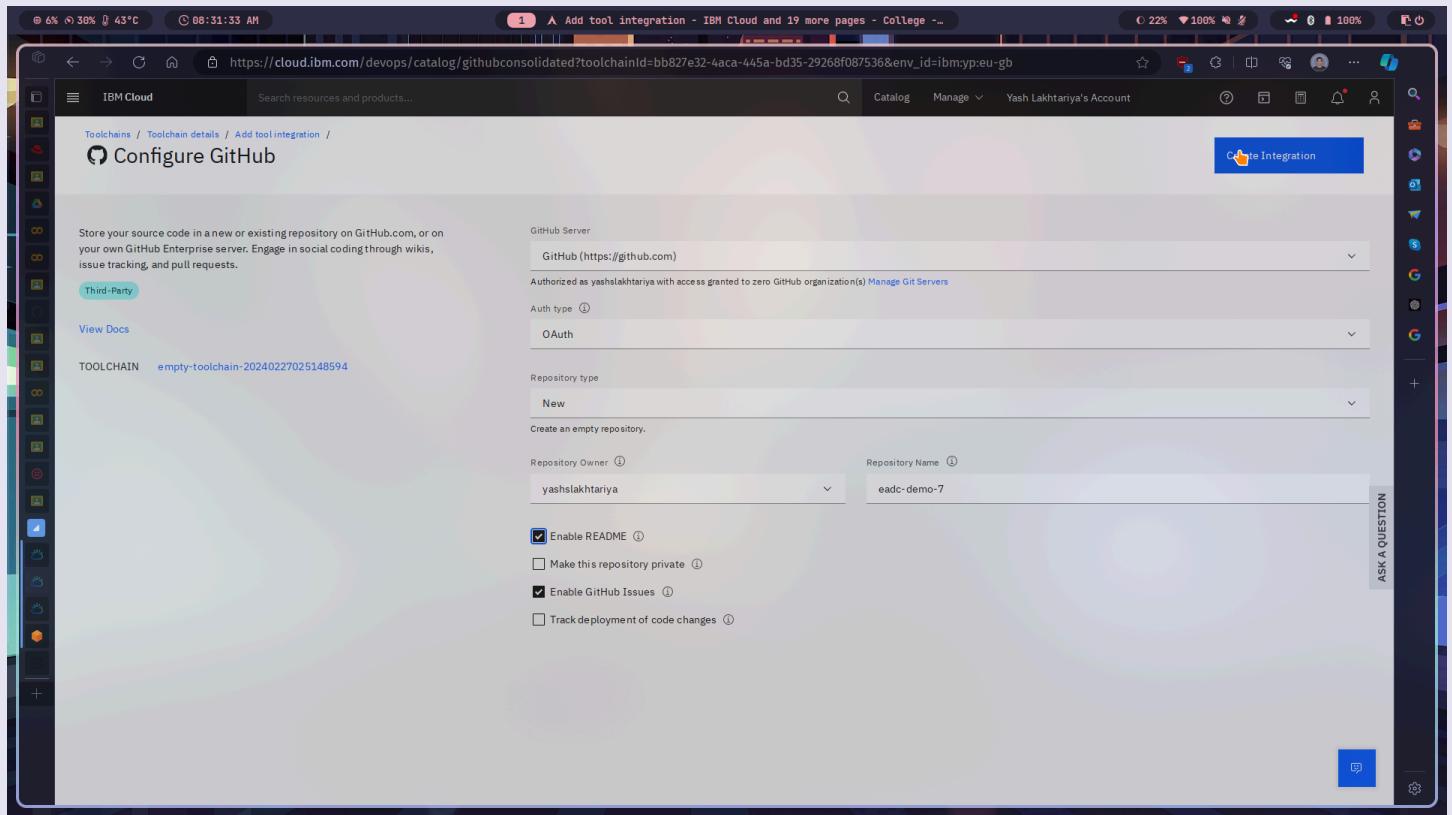
EADC Practical 7

## 24. Here, add the Github integration and authorize the same

The screenshot shows a web browser window for the IBM Cloud DevOps Catalog. The URL is [https://cloud.ibm.com/devops/catalog/githubconsolidated?toolchainId=bb827e32-4aca-445a-bd35-29268f087536&env\\_id=ibm:yp:eu-gb](https://cloud.ibm.com/devops/catalog/githubconsolidated?toolchainId=bb827e32-4aca-445a-bd35-29268f087536&env_id=ibm:yp:eu-gb). The page title is "Add tool integration - IBM Cloud and 19 more pages - College - ...". The main content area is titled "Configure GitHub". It includes a description of GitHub's features, a "GitHub Server" dropdown set to "GitHub (https://github.com)", an "Auth type" dropdown set to "OAuth", and a note: "You must authorize before you can configure this tool integration." with an "Authorize" button. The left sidebar shows various toolchains and a "TOOLCHAIN" section with "empty-toolchain-20240227025148594". The right sidebar has "ASK A QUESTION" and other navigation links.

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 7**

**25. The new repo can be created and used here for the same**



The screenshot shows the IBM Cloud DevOps Catalog interface. A modal window titled "Configure GitHub" is open, allowing users to set up a GitHub integration. The "GitHub Server" dropdown is set to "GitHub (https://github.com)". The "Auth type" dropdown is set to "OAuth". Under "Repository type", "New" is selected. The "Repository Owner" dropdown is set to "yashlakhtariya" and the "Repository Name" is "eadc-demo-7". Several checkboxes are checked: "Enable README", "Enable GitHub Issues", and "Track deployment of code changes". A large blue button labeled "Create Integration" is visible at the top right of the modal.

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Practical 7

26. After this, we can add delivery pipeline

The screenshot shows the IBM Cloud DevOps toolchain interface. At the top, there's a navigation bar with various icons and a search bar. Below it, the main dashboard displays a message: "Your toolchain is ready! Quick start: You can now add tool integrations. For more details, view the documentation." On the left, a sidebar has sections for Overview, Connections, and Manage, with a "Search..." input field. The main content area has three tabs: "Repositories" (1 total), "Delivery pipelines" (0 total), and "IBM Cloud tools" (0 total). The "Repositories" tab shows a single entry: "eadc-demo-7" (New, GitHub, https://github.com/yashlakhtariya/eadc-demo-7...). A blue arrow points from the text "After this, we can add delivery pipeline" in the question above to the "Delivery pipelines" tab.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

## 27. Select classic as type

The screenshot shows a browser window for the IBM Cloud DevOps Catalog. The URL is [https://cloud.ibm.com/devops/catalog/pipeline?toolchainId=bb827e32-4aca-445a-bd35-29268f087536&env\\_id=ibm:yp:eu-gb](https://cloud.ibm.com/devops/catalog/pipeline?toolchainId=bb827e32-4aca-445a-bd35-29268f087536&env_id=ibm:yp:eu-gb). The page title is "Add tool integration - IBM Cloud and 20 more pages - College ...". The main content is titled "Configure Delivery Pipeline". It includes a pipeline name input field containing "dlvrypln" and a dropdown menu for "Pipeline type" set to "Classic". A note states: "Classic pipelines provide an easy to use graphical UI for defining stages and jobs that run on public shared infrastructure, with support for running individual stages on Private Workers." There is also a checkbox for "Show apps in the View app menu". The sidebar on the left shows various toolchains and a toolchain named "empty-toolchain-20240227025148594". The top and bottom navigation bars include links like Catalog, Manage, and a user account section.

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

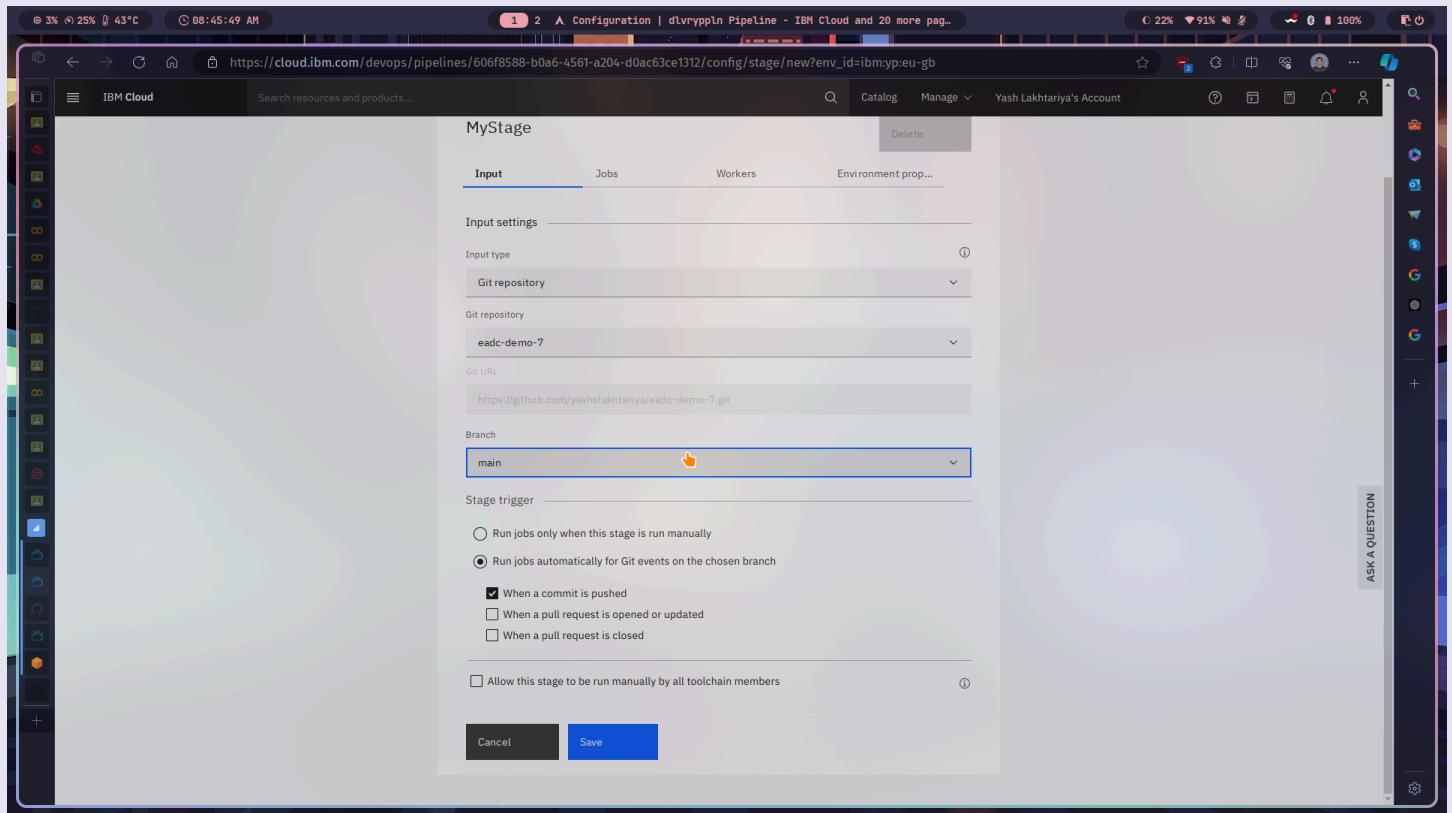
EADC Practical 7

28. Now, the pipeline is successfully created and needs to be configured for adding stages

The screenshot shows the IBM Cloud DevOps toolchain interface. At the top, a banner says "empty-toolchain-20240227025148594 - IBM Cloud and 20 more page..." with a status of "22% 88%". The main title is "empty-toolchain-20240227025148594". A message box says "Your toolchain is ready! Quick start: You can now add tool integrations. For more details, view the documentation." Below this, there are three sections: "Repositories" (1 total, "eadc-demo-7" GitHub), "Delivery pipelines" (1 total, "dlvryppin" Delivery Pipeline), and "IBM Cloud tools" (0 total). A sidebar on the left shows various icons for different services like databases, storage, and monitoring. A "Catalog" tab is visible at the top right.

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 7**

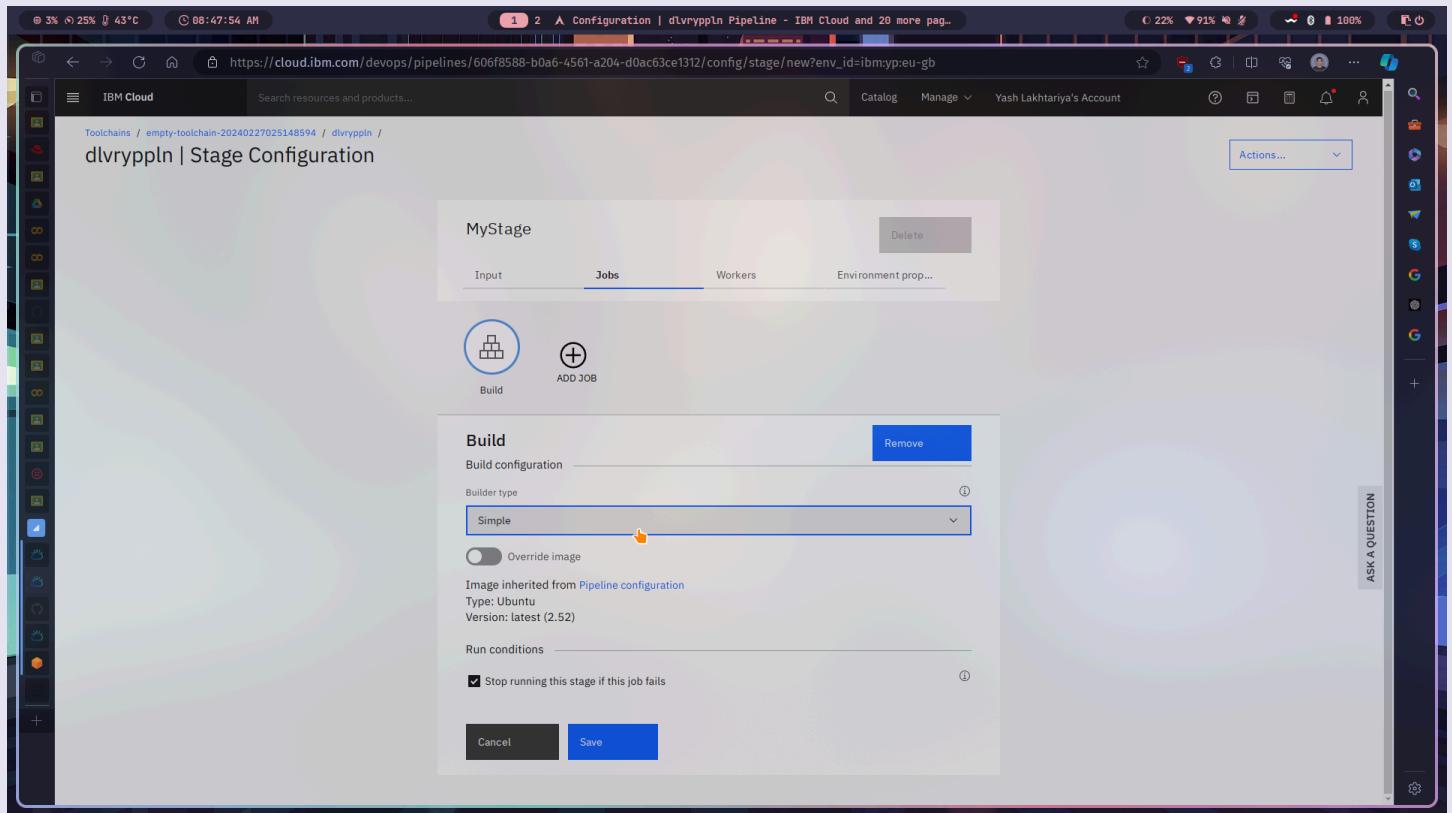
## 29. First, add a stage and take input from Git repository



The screenshot shows the IBM Cloud DevOps Pipeline configuration interface. A new stage named "MyStage" is being created. The "Input" tab is selected. Under "Input settings", the "Input type" is set to "Git repository", with the repository "eadc-demo-'7" selected. The "Git URL" is listed as "https://github.com/yashslakhtariya/eadc-demo-7.git". The "Branch" dropdown is set to "main". In the "Stage trigger" section, the option "Run jobs automatically for Git events on the chosen branch" is selected, with "When a commit is pushed" checked. There is also an unchecked checkbox for "Allow this stage to be run manually by all toolchain members". At the bottom, there are "Cancel" and "Save" buttons.

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 7**

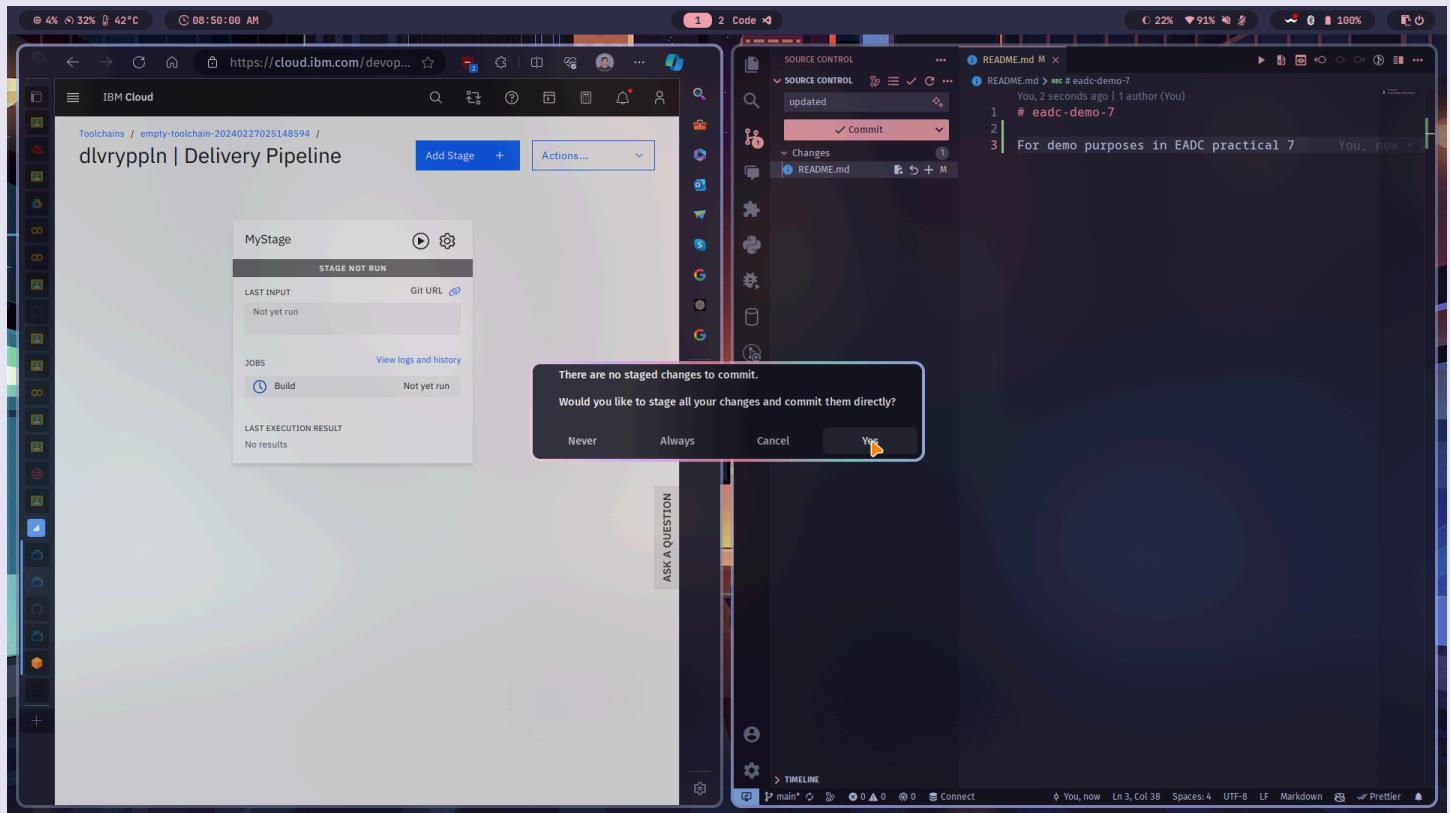
**30. For jobs, choose 'Build' and 'Simple' as type**



The screenshot shows the IBM Cloud Pipeline configuration interface. The URL in the browser is [https://cloud.ibm.com/devops/pipelines/606f8588-b0a6-4561-a204-d0ac63ce1312/config/stage/new?env\\_id=ibm:yp:eu-gb](https://cloud.ibm.com/devops/pipelines/606f8588-b0a6-4561-a204-d0ac63ce1312/config/stage/new?env_id=ibm:yp:eu-gb). The page title is "dlvryppln | Stage Configuration". The main section is titled "MyStage" and has tabs for "Input", "Jobs" (which is selected), "Workers", and "Environment prop...". Under the "Jobs" tab, there is a "Build" configuration section. The "Builder type" dropdown is set to "Simple". A yellow arrow points to this dropdown. Below it, there is an "Override image" toggle switch, which is off. The "Image inherited from Pipeline configuration" is listed as Type: Ubuntu Version: latest (2.52). There is also a "Run conditions" section with a checkbox for "Stop running this stage if this job fails" (which is checked). At the bottom of the "Build" section are "Cancel" and "Save" buttons.

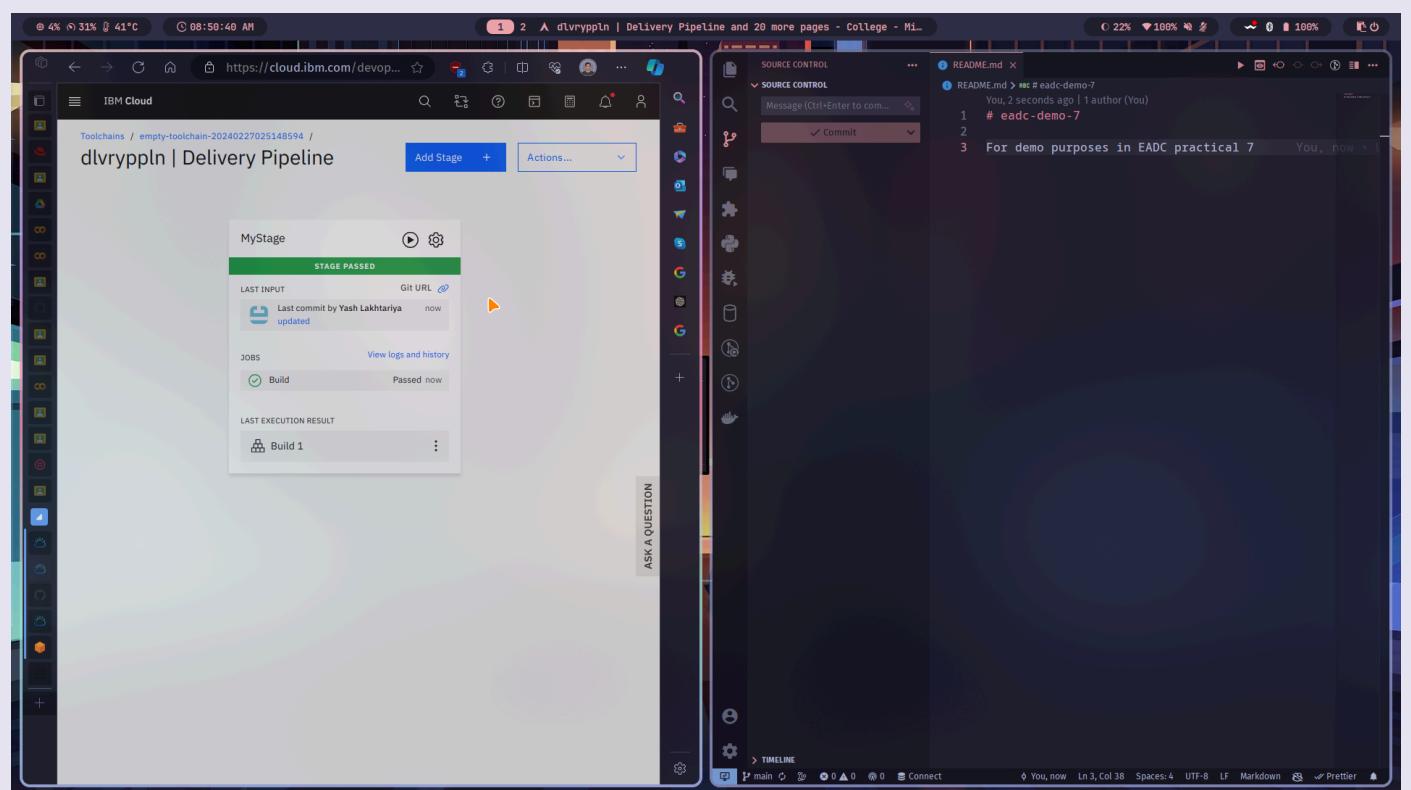
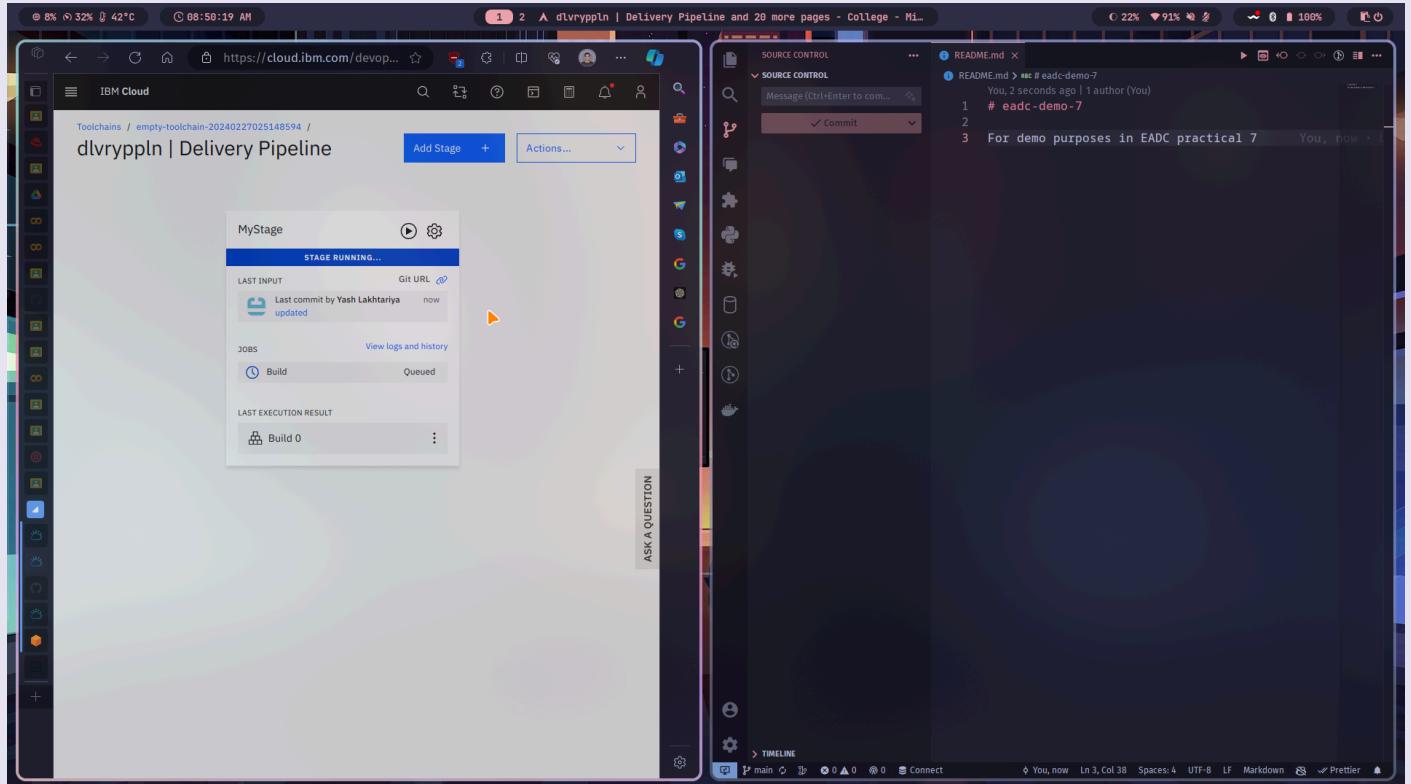
Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

31. Thereafter, make changes to the code to trigger automatic run of build stage



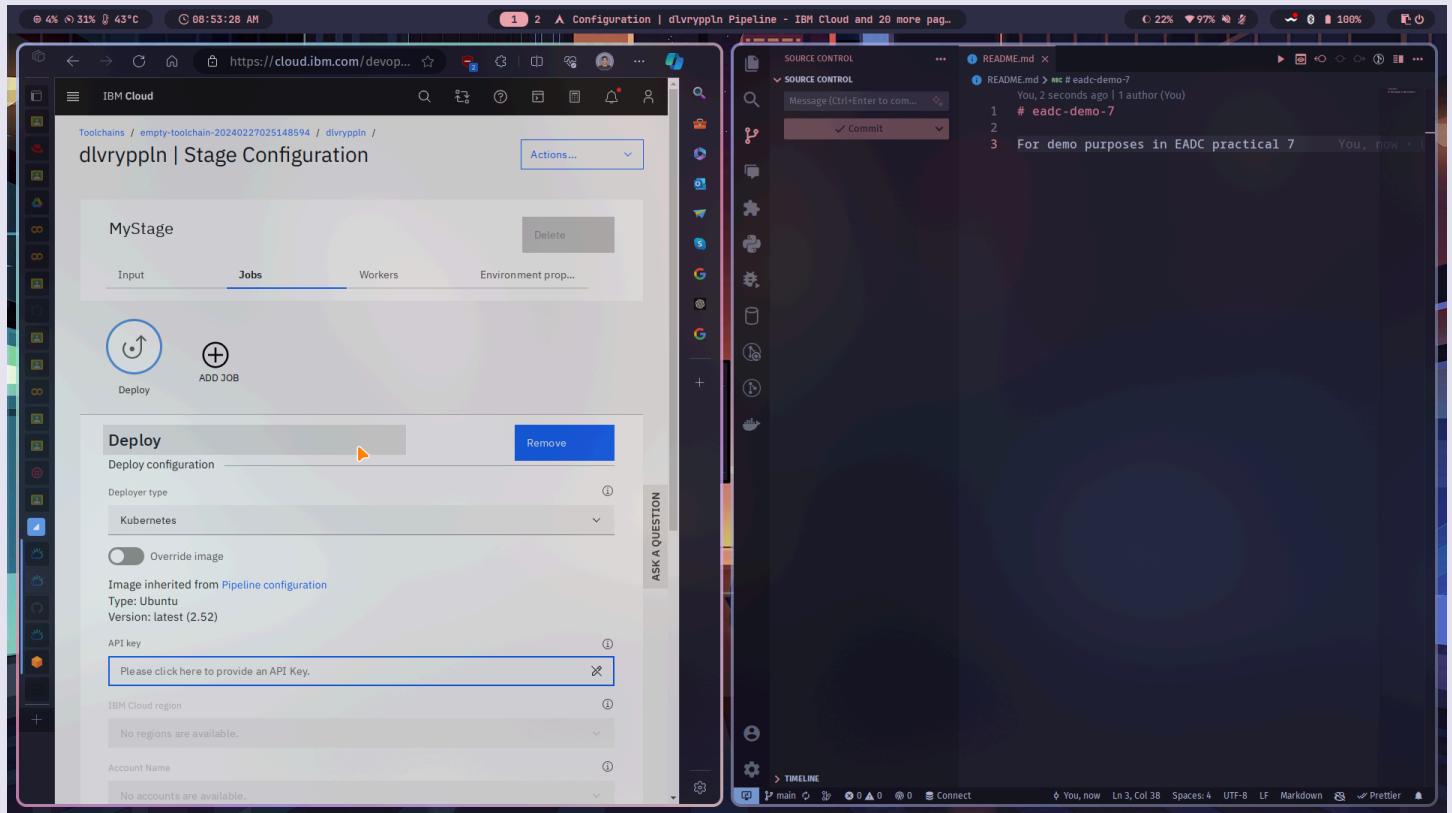
Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

32. After committing the changes on Github, the stage will run automatically



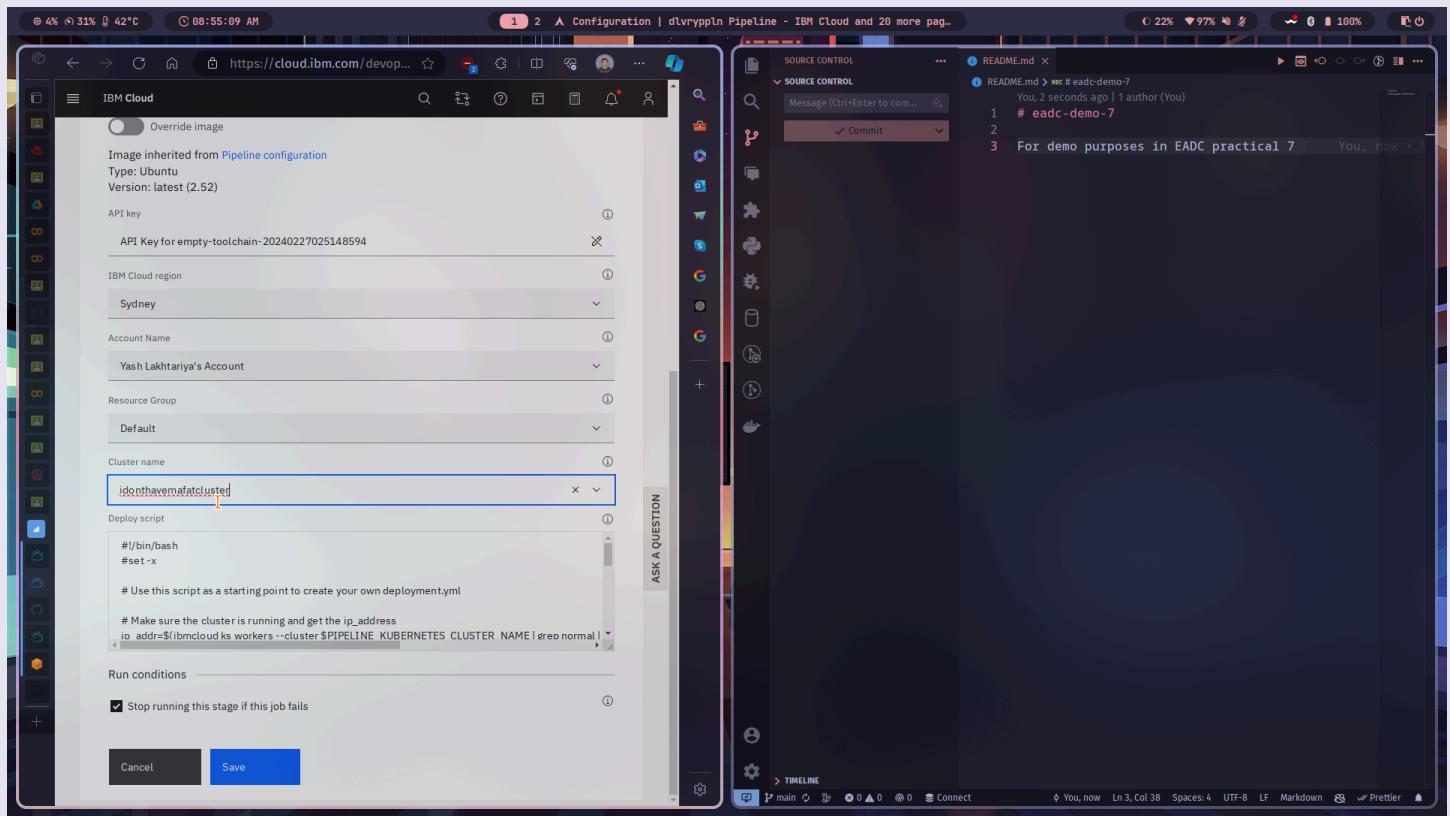
Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

33. For educational purposes, add deploy stage to see the available options, say Kubernetes can be selected. This stage will run after build stage, so select the option to run after previous stage



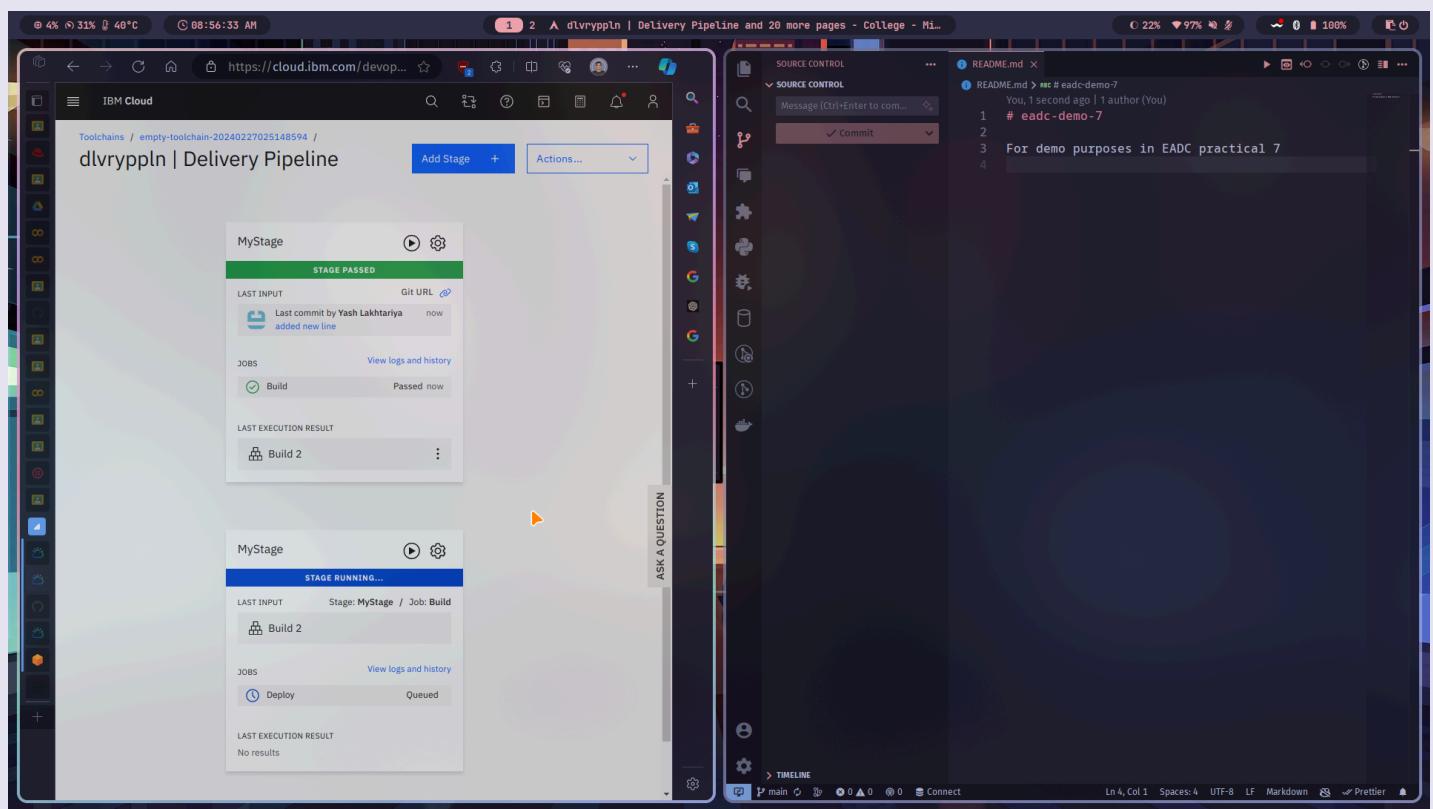
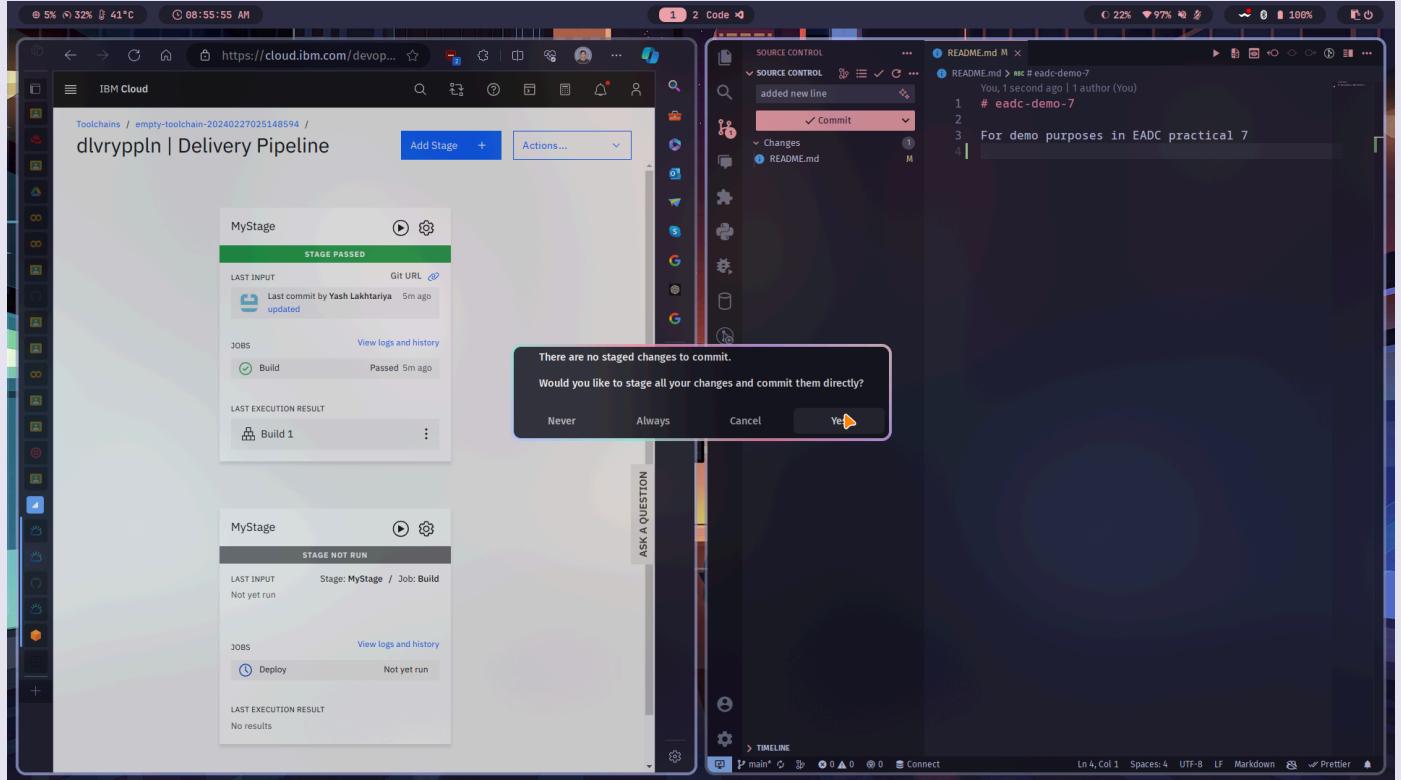
Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

34. Generate API key for the same and as we don't have paid cluster of K8s, it will fail the job, but keep any random name to proceed



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

35. Now, make changes to the code again to check it



Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

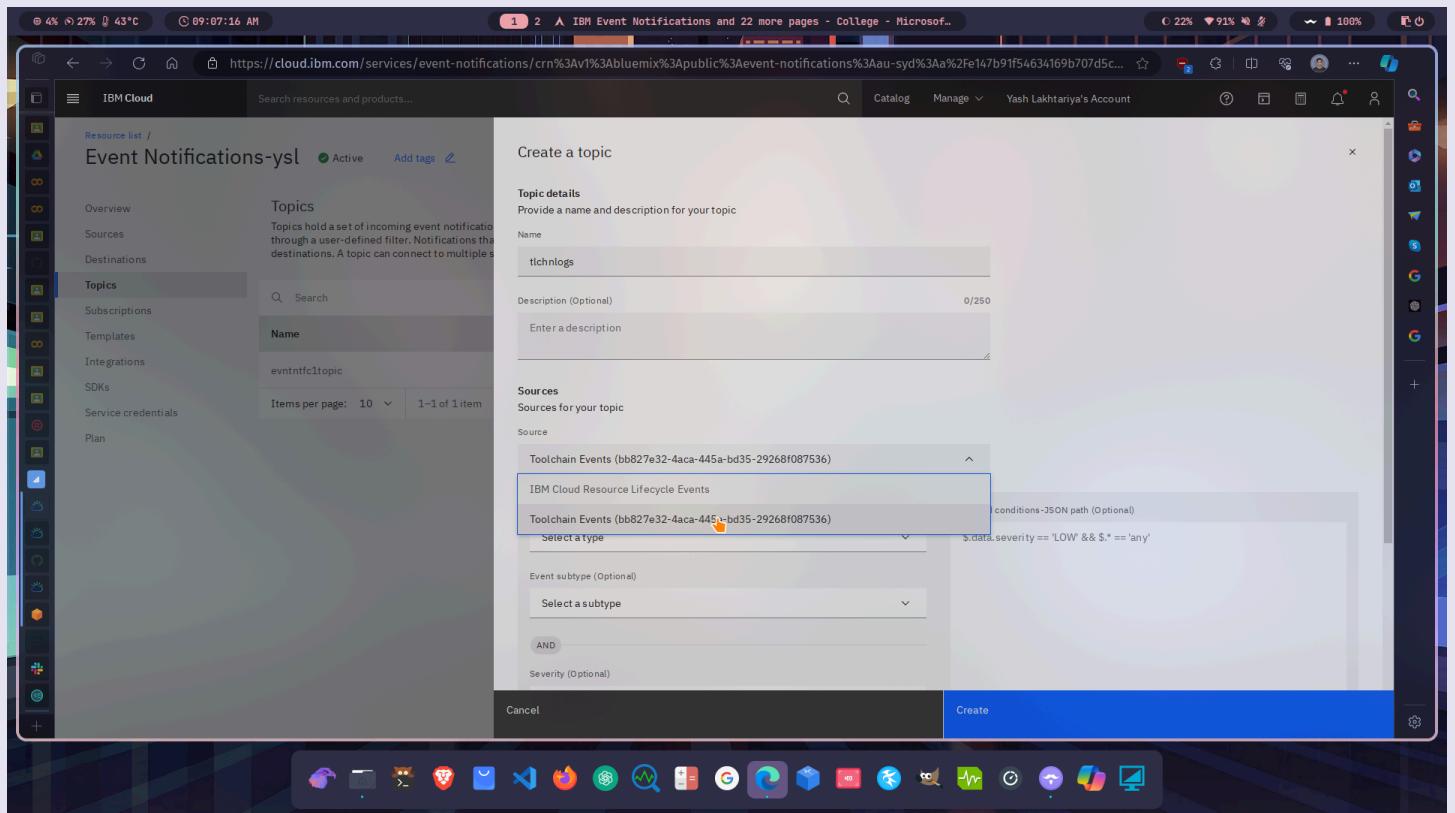
EADC Practical 7

36. Now, add event notification service as integration to our toolchain, after authorizing, the option will be available in Event Notification service for events of toolchain

The screenshot shows a web browser window for the IBM Cloud DevOps Catalog. The URL is [https://cloud.ibm.com/devops/catalog/eventnotifications?toolchainId=bb827e32-4aca-445a-bd35-29268f087536&env\\_id=ibm:yp:eu-gb](https://cloud.ibm.com/devops/catalog/eventnotifications?toolchainId=bb827e32-4aca-445a-bd35-29268f087536&env_id=ibm:yp:eu-gb). The page title is "Add tool integration - IBM Cloud and 20 more pages - College -". The main content area is titled "Configure Event Notifications". It includes fields for "Name" (set to "evntntfctchn"), "Event Notifications instance" (set to "Event Notifications-ysl (f19f2a25-d2ac-43a8-9e94-cc935f7f40f2)", and "Authorization" (with a "Create Authorization" button highlighted by a cursor). The left sidebar shows "Toolchains / Toolchain details / Add tool integration /". The bottom navigation bar contains various icons for different tools and services.

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 7**

### 37. Create a new topic with Toolchain events as source instead of IBM Resource Lifecycle events as done previously



The screenshot shows the IBM Cloud Event Notifications interface. On the left, there's a sidebar with options like Overview, Sources, Destinations, Topics (which is selected), Subscriptions, Templates, Integrations, SDKs, Service credentials, and Plan. The main area shows a list of topics under 'Topics'. One topic is listed: 'evntntfc1topic'. Below this, there are filters for 'Items per page: 10' and '1-1 of 1 item'. On the right, a modal window titled 'Create a topic' is open. In the 'Topic details' section, the 'Name' field contains 'tchnlogs'. The 'Sources' section lists 'Toolchain Events (bb827e32-4aca-445a-bd35-29268f087536)' and 'IBM Cloud Resource Lifecycle Events'. The 'Toolchain Events' option is highlighted with a blue border. At the bottom of the modal, there are 'Cancel' and 'Create' buttons.

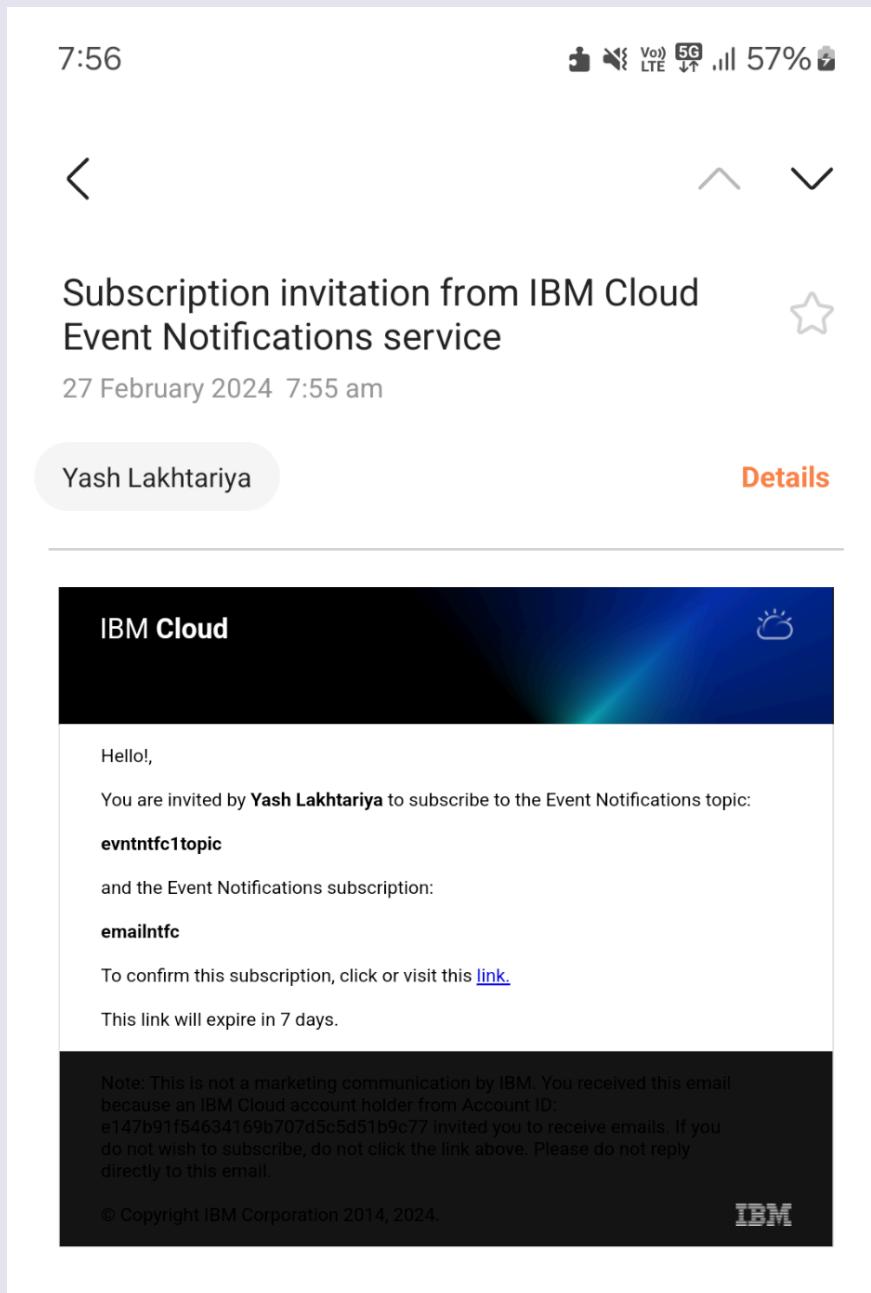
Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Practical 7

38. Like previous steps, add subscription or configure the existing one and check if the SMS arrives



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

The screenshot shows a mobile browser interface. At the top, the time is 7:56, and the status bar includes icons for signal strength, battery level at 57%, and connectivity (5G LTE). Below the status bar is a header bar with a user profile picture on the left, a lock icon indicating security, and the URL "au-syd.event-notifications.cloud.ibm.com/" in the center. To the right of the URL is a refresh icon. A black navigation bar below the header contains the text "IBM Cloud". The main content area has a light gray background. It displays the text "Event Notifications" and "Confirm subscription". Below this, a message reads "Click the button to continue receiving Emails for the topic: \"evntntfc1topic\"". At the bottom left of this area is a blue rectangular button with the word "Confirm" in white text.

7:56

5G LTE 57%

au-syd.event-notifications.cloud.ibm.com/

IBM Cloud

Event Notifications

Confirm subscription

Click the button to continue receiving Emails for the topic: "evntntfc1topic"

Confirm

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Practical 7

The screenshot shows a mobile browser interface. At the top, the status bar displays the time (7:56), signal strength, battery level (57%), and connectivity (5G). Below the status bar is a header bar containing a user profile icon, a lock icon indicating security, the URL "au-syd.event-notifications.cloud.ibm.com/", and a refresh button. A black navigation bar below the header bar contains the text "IBM Cloud". The main content area of the browser displays the following text:

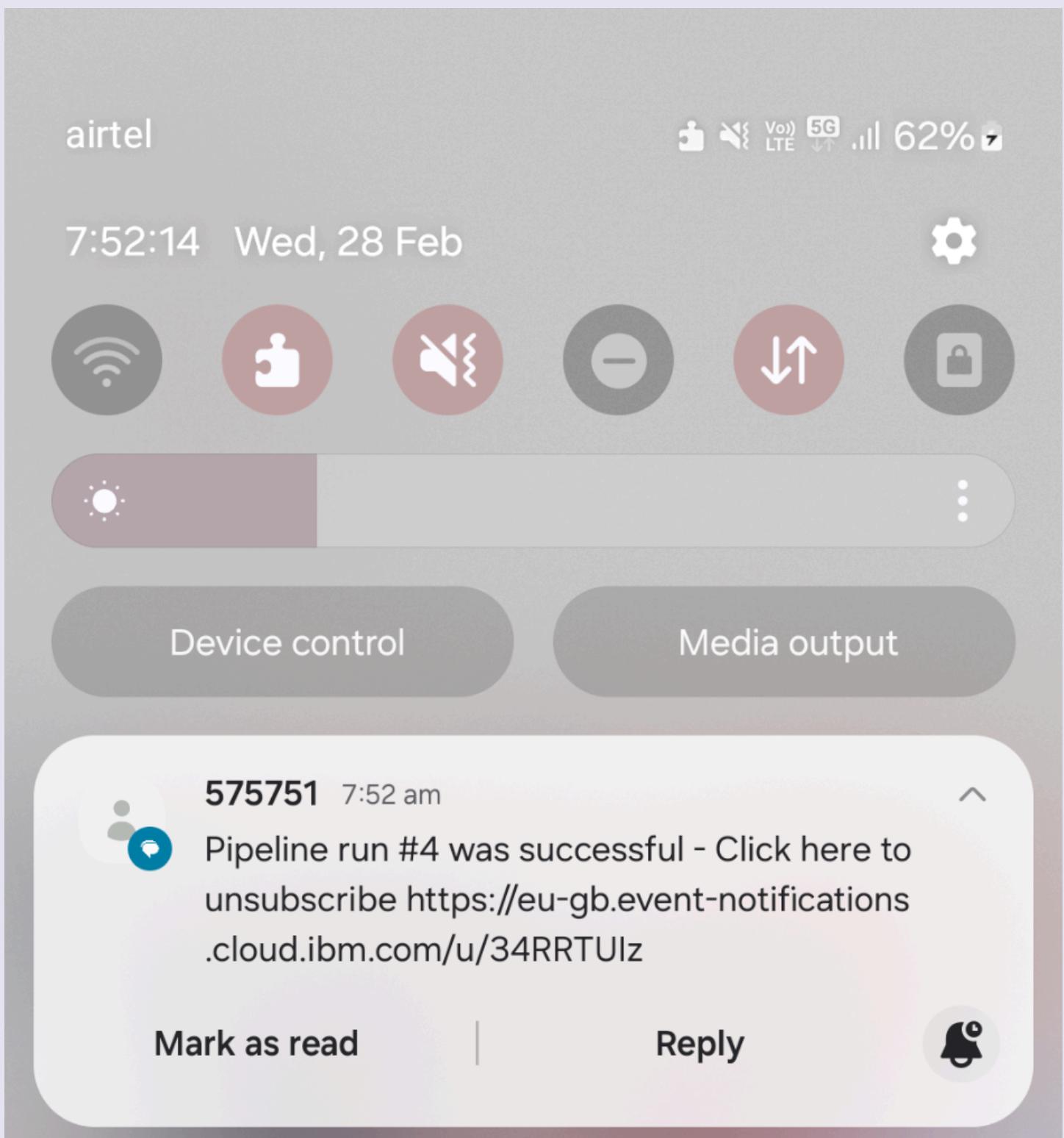
Event Notifications  
**Successfully Subscribed!**

Since you've accepted the invitation, you are successfully subscribed to topic: "evntntfc1topic"

Your email address yashslakhtariya21@gnu.ac.in has been subscribed and will start receiving email notifications sent by admin via IBM Cloud Event Notifications.

If you wish to not receive email, click or visit [link](#)

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

### **Configuring AWS SNS :**

**Sign in to the AWS Management Console.**

**Navigate to the SNS service.**

**Create a new topic or select an existing one to which notifications will be published.**

**Define subscriptions, specifying endpoints (e.g., email addresses, HTTP/HTTPS endpoints, Lambda functions) that should receive notifications.**

**Configure access policies to ensure proper authentication and authorization.**

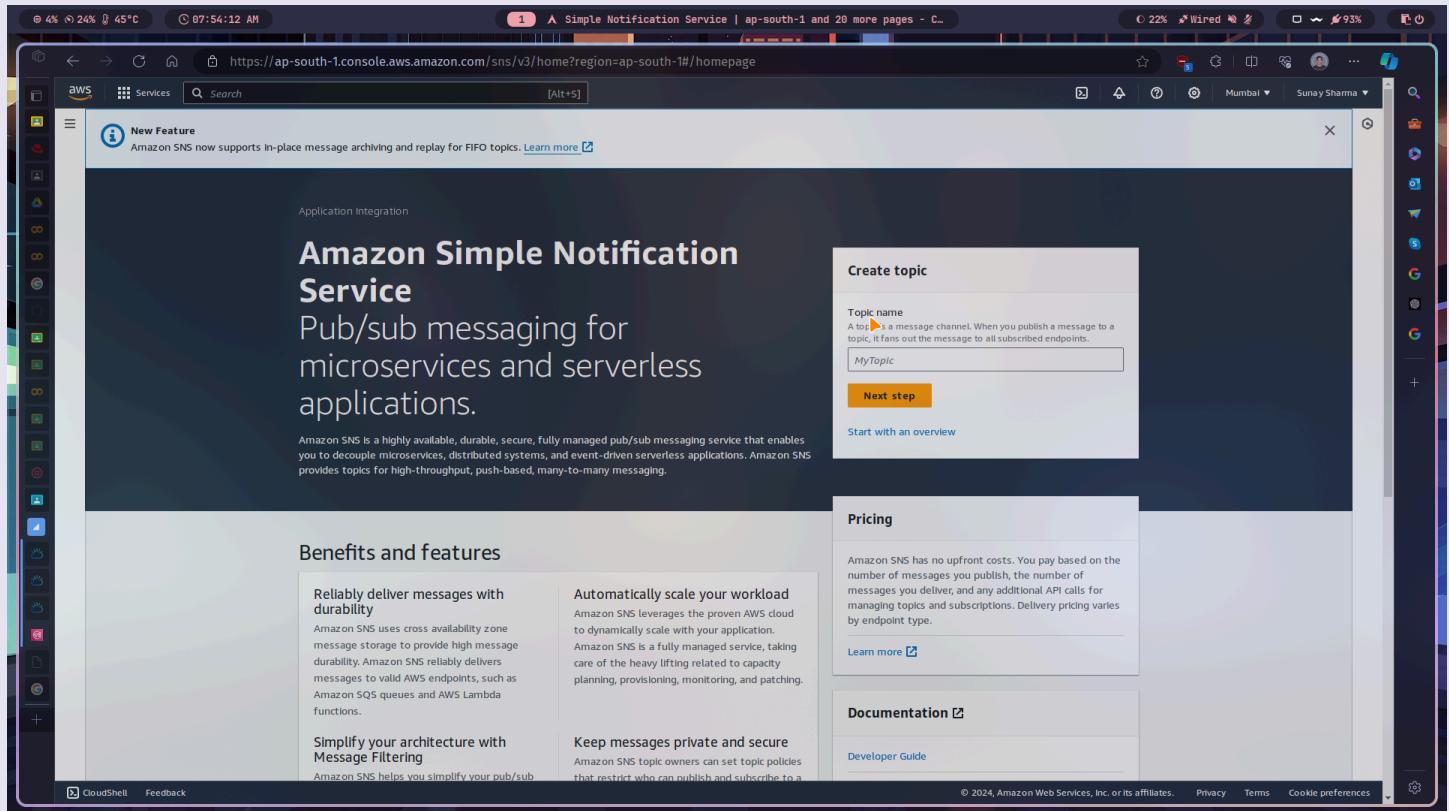
**Obtain necessary credentials (e.g., access keys, IAM roles) for authentication when accessing SNS APIs.**

**Integrate your application with SNS using AWS SDKs or RESTful APIs.**

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 7**

**Steps :**

**1. Login to AWS console and search for Amazon SNS service**



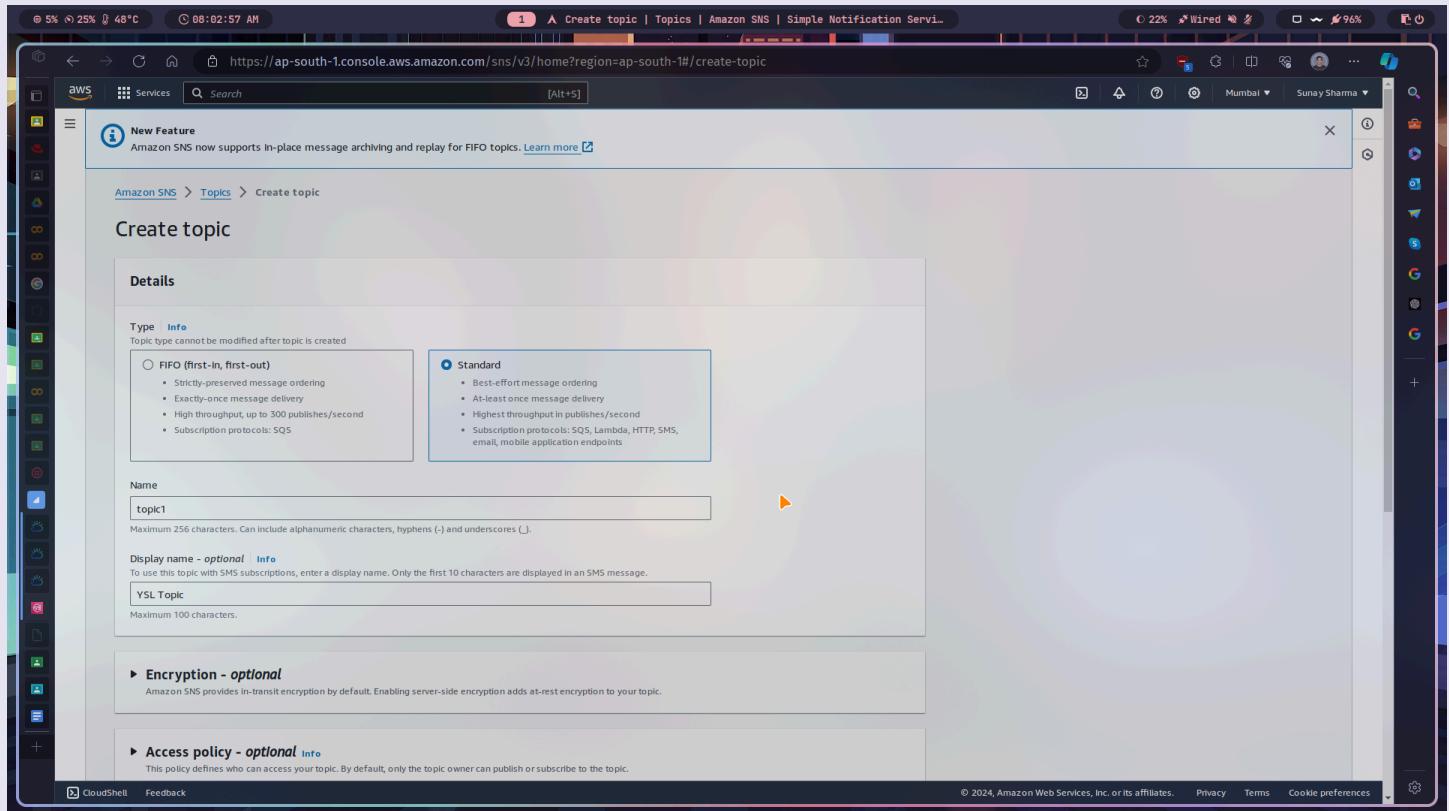
Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

## 2. Add the new topic like IBM Cloud here also

The screenshot shows the AWS Simple Notification Service (SNS) homepage. A modal dialog box titled "Create topic" is open in the center. Inside the dialog, there is a "Topic name" input field containing "topic1". Below the input field is a "Next step" button. To the right of the input field, there is a link "Start with an overview". The background of the page displays the main SNS landing page with sections for "Application integration", "Amazon Simple Notification Service", "Pub/sub messaging for microservices and serverless applications", and "Benefits and features". The "Benefits and features" section includes four items: "Reliably deliver messages with durability", "Automatically scale your workload", "Simplify your architecture with Message Filtering", and "Keep messages private and secure".

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

3. Type can either be selected FIFO or Standard, here Standard will be more suitable as it guarantees message delivery even if late, rest defaults can be kept



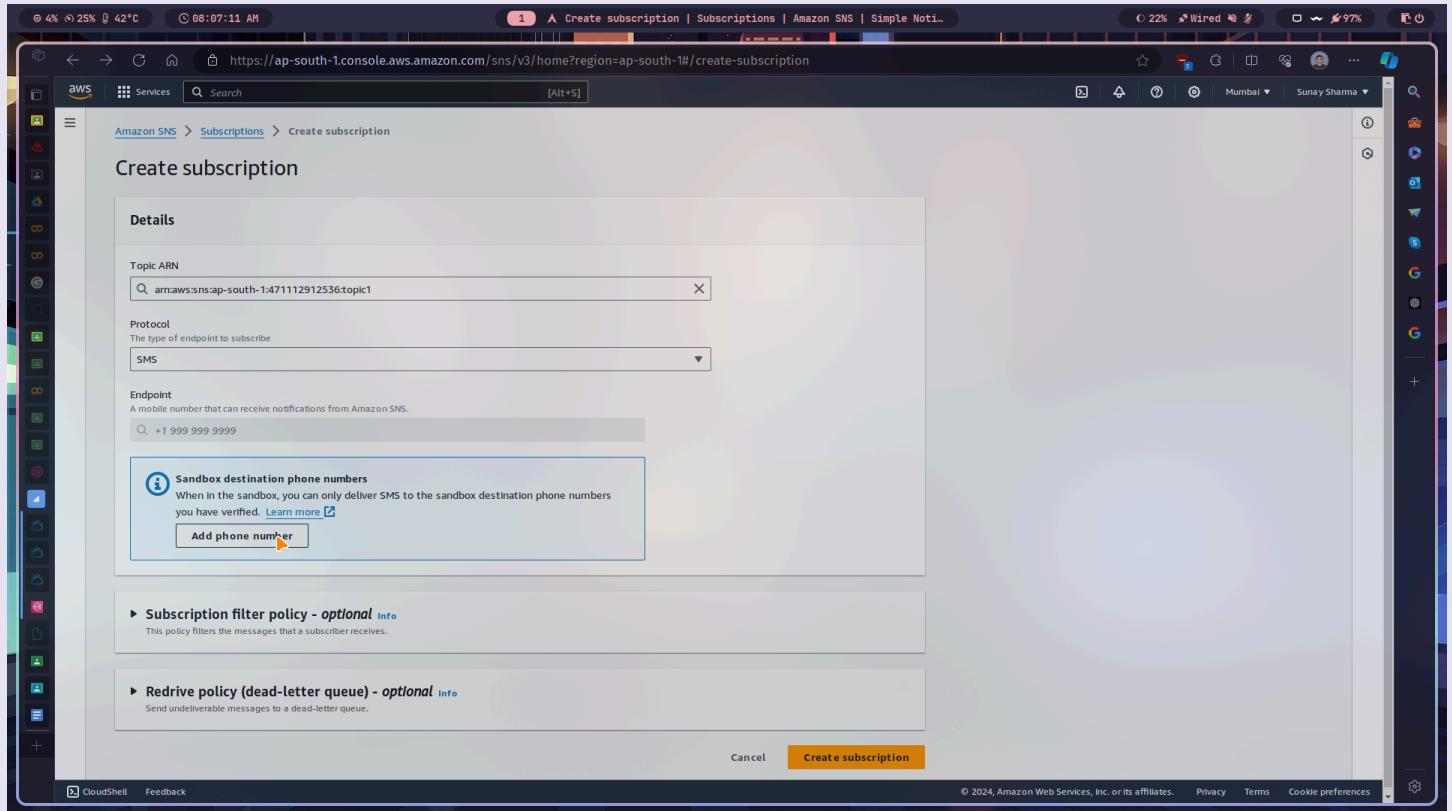
Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

#### 4. So, the topic should be created now

The screenshot shows the AWS SNS Topics page with a green success message banner at the top stating "Topic topic1 created successfully. You can create subscriptions and send messages to them from this topic." Below the banner, the topic details are displayed: Name: topic1, Display name: YSL Topic, ARN: arn:aws:sns:ap-south-1:471112912536:topic1, and Type: Standard. The Subscriptions section shows "Subscriptions (0)" with a "Create subscription" button. The page includes standard AWS navigation elements like CloudShell, Feedback, and footer links.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

## 5. Create subscription now with SMS as protocol and add phone number



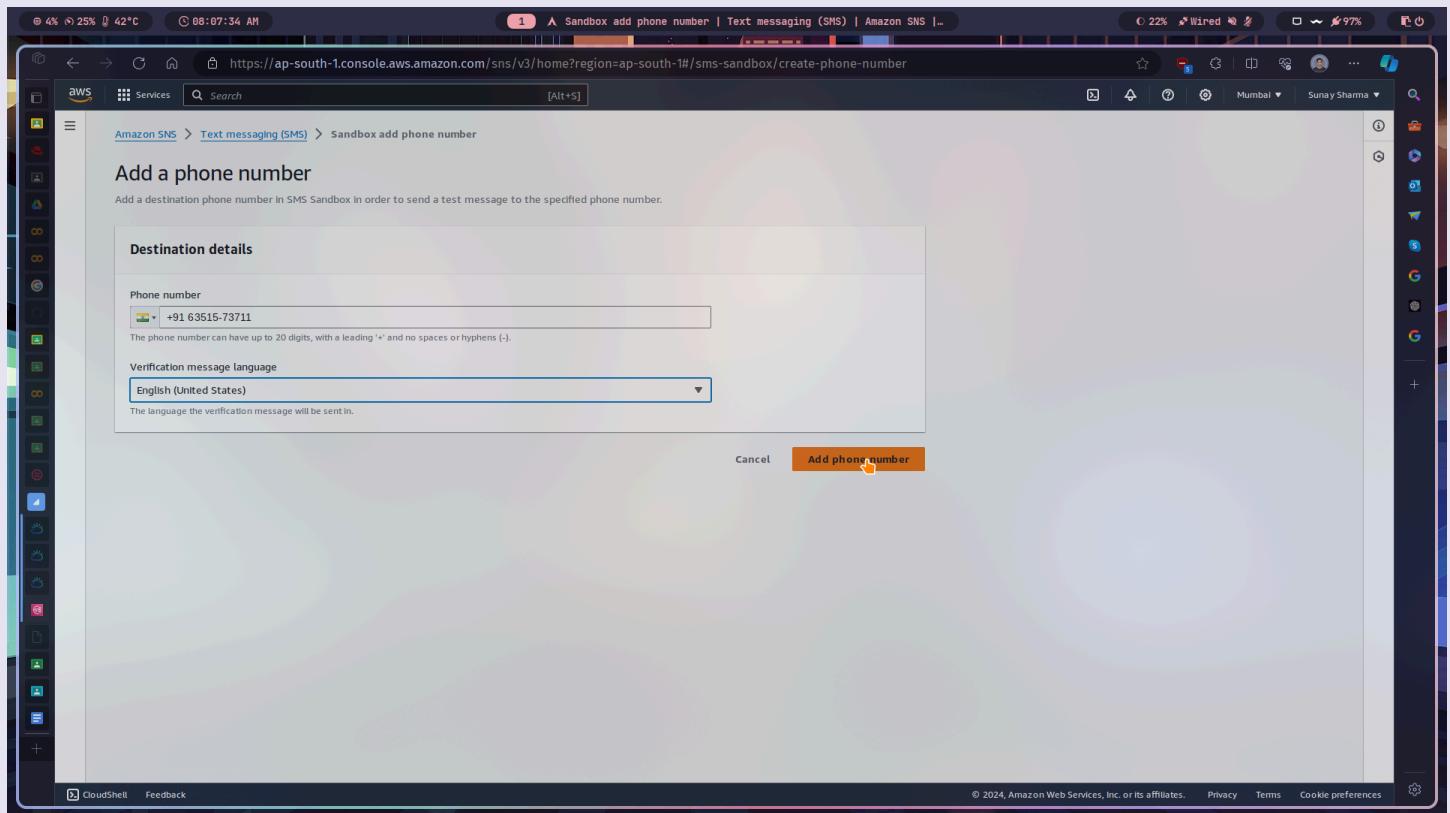
Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Practical 7

6. Add sandboxed phone number here and select the preferred message language



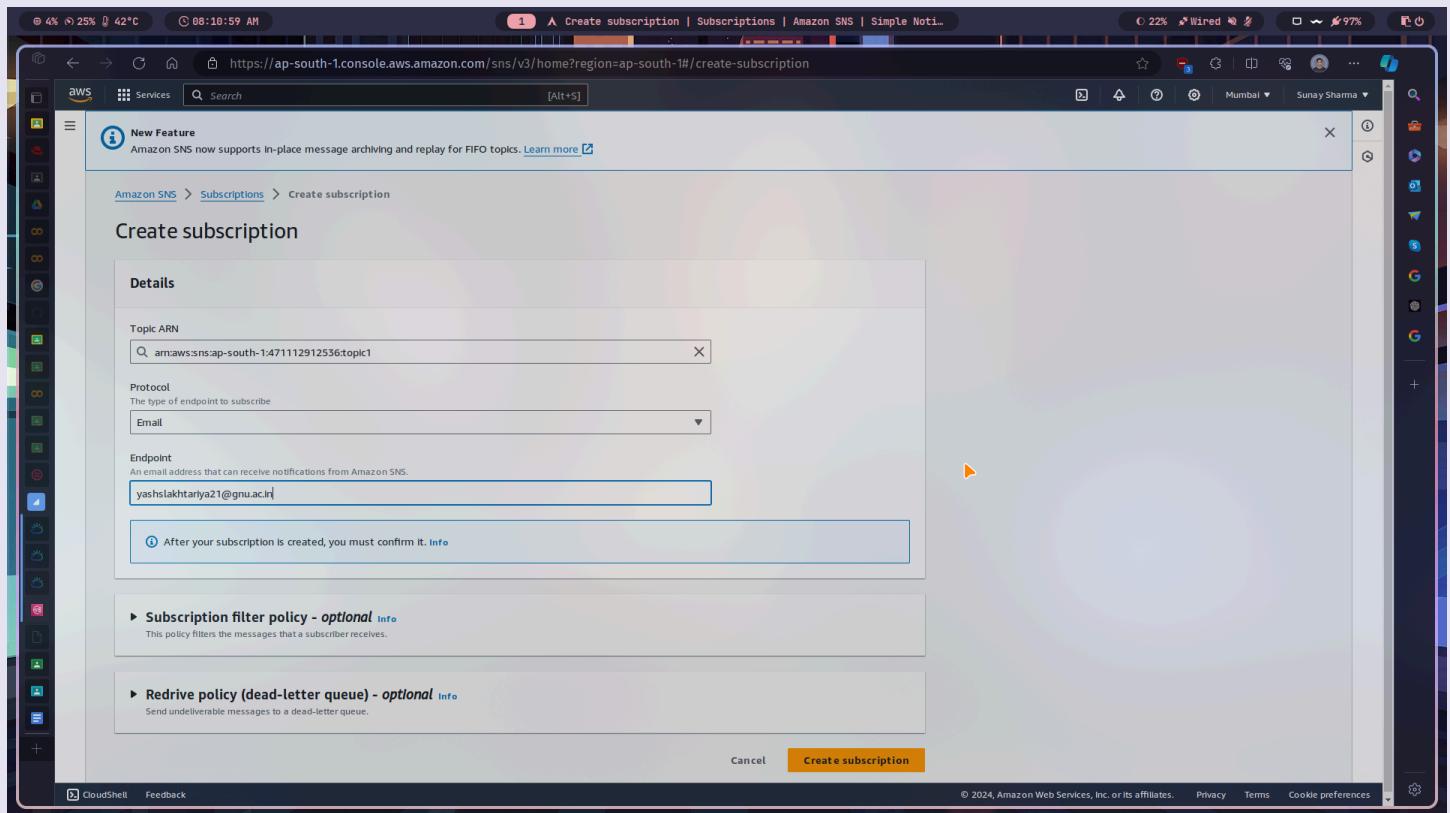
Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Practical 7

7. Email can also be taken as protocol and mail endpoint can be configured in case SMS doesn't work (like the case here)



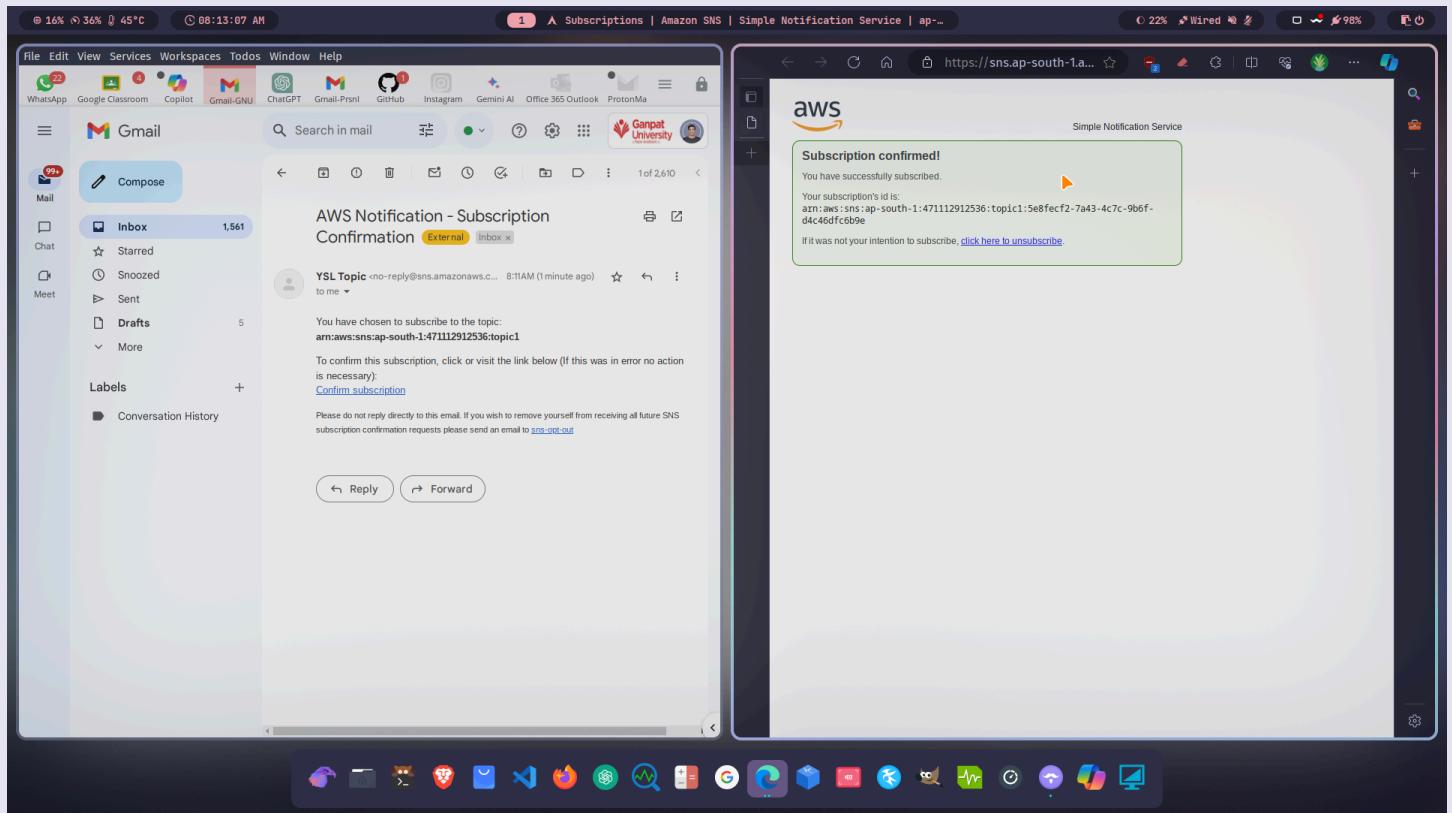
Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

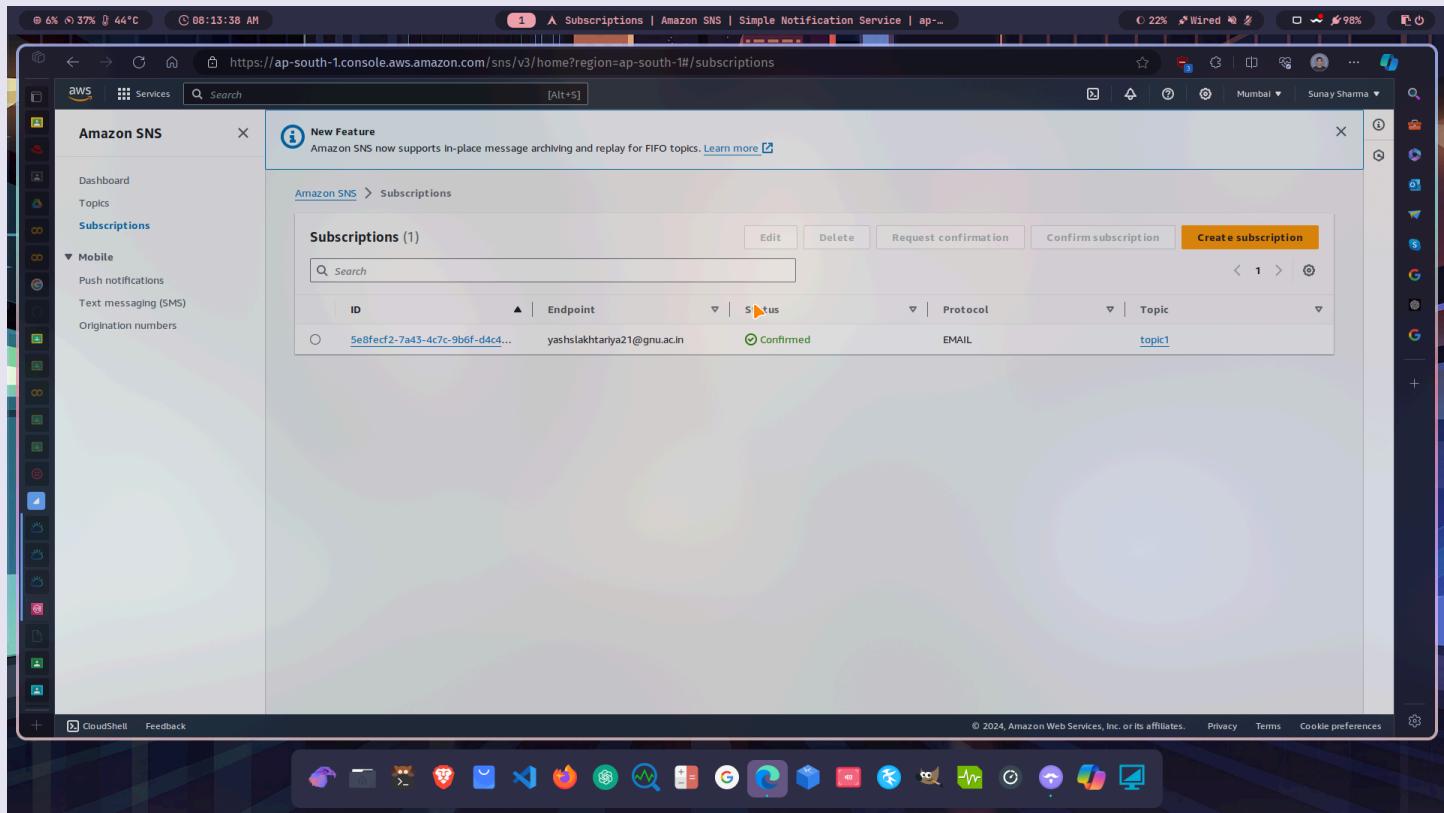
EADC Practical 7

## 8. Confirm the subscription using the confirmation email



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

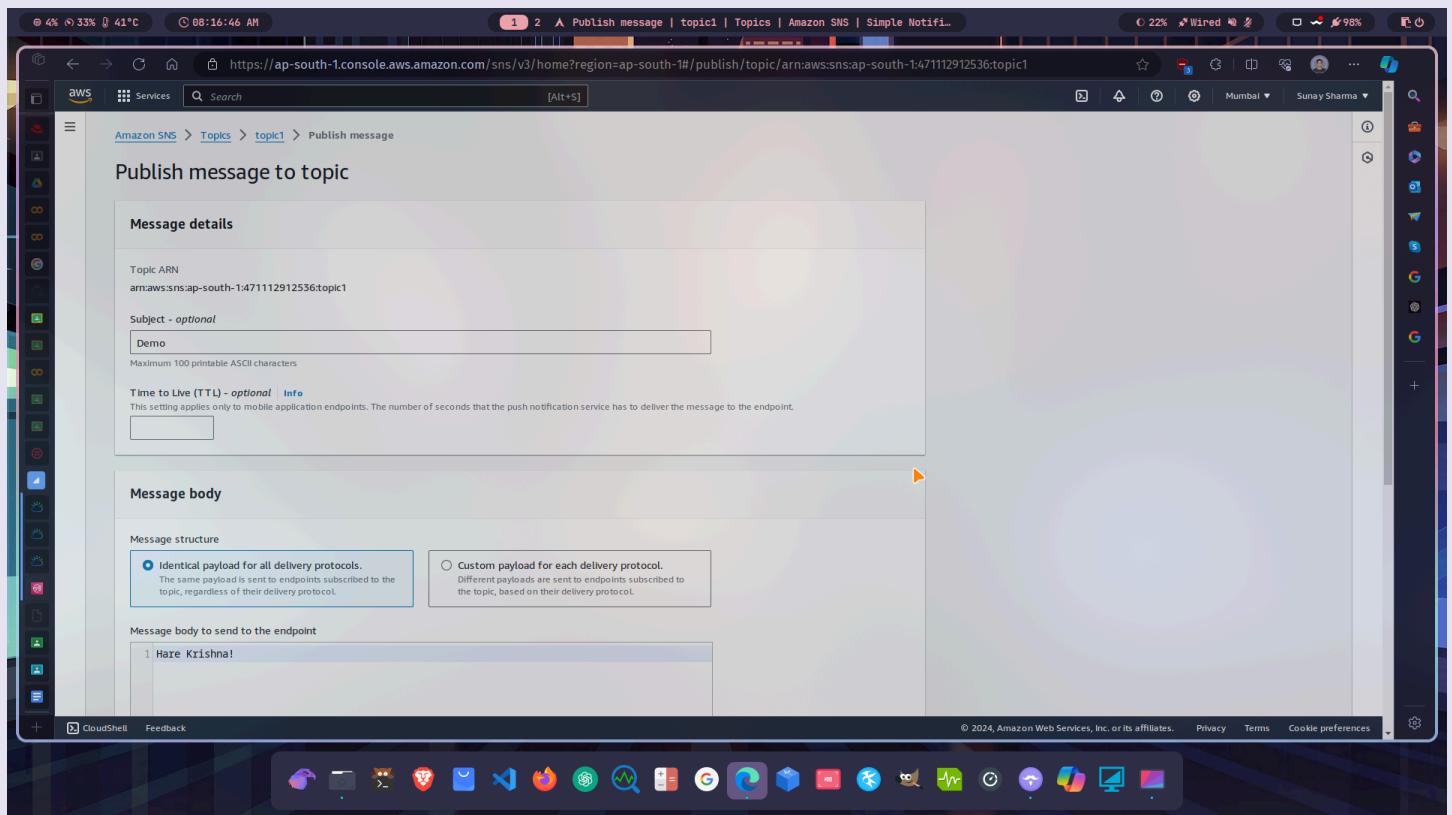
## 9. Now, the status is confirmed and it should work



The screenshot shows the AWS SNS Subscriptions page. The left sidebar is titled "Amazon SNS" and includes links for Dashboard, Topics, Subscriptions, and Mobile (Push notifications, Text messaging (SMS), Origination numbers). The main content area is titled "Subscriptions (1)" and displays a table with one row. The table columns are ID, Endpoint, Status, Protocol, and Topic. The single row shows an ID starting with "5e8fecf2-7a43-4c7c-9b6f-d4c4...", an endpoint of "yashslakhtariya21@gnu.ac.in", a status of "Confirmed" (with a green circular icon), a protocol of "EMAIL", and a topic of "topic1". Above the table, there are buttons for Edit, Delete, Request confirmation, Confirm subscription, and Create subscription. A search bar is also present above the table.

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 7**

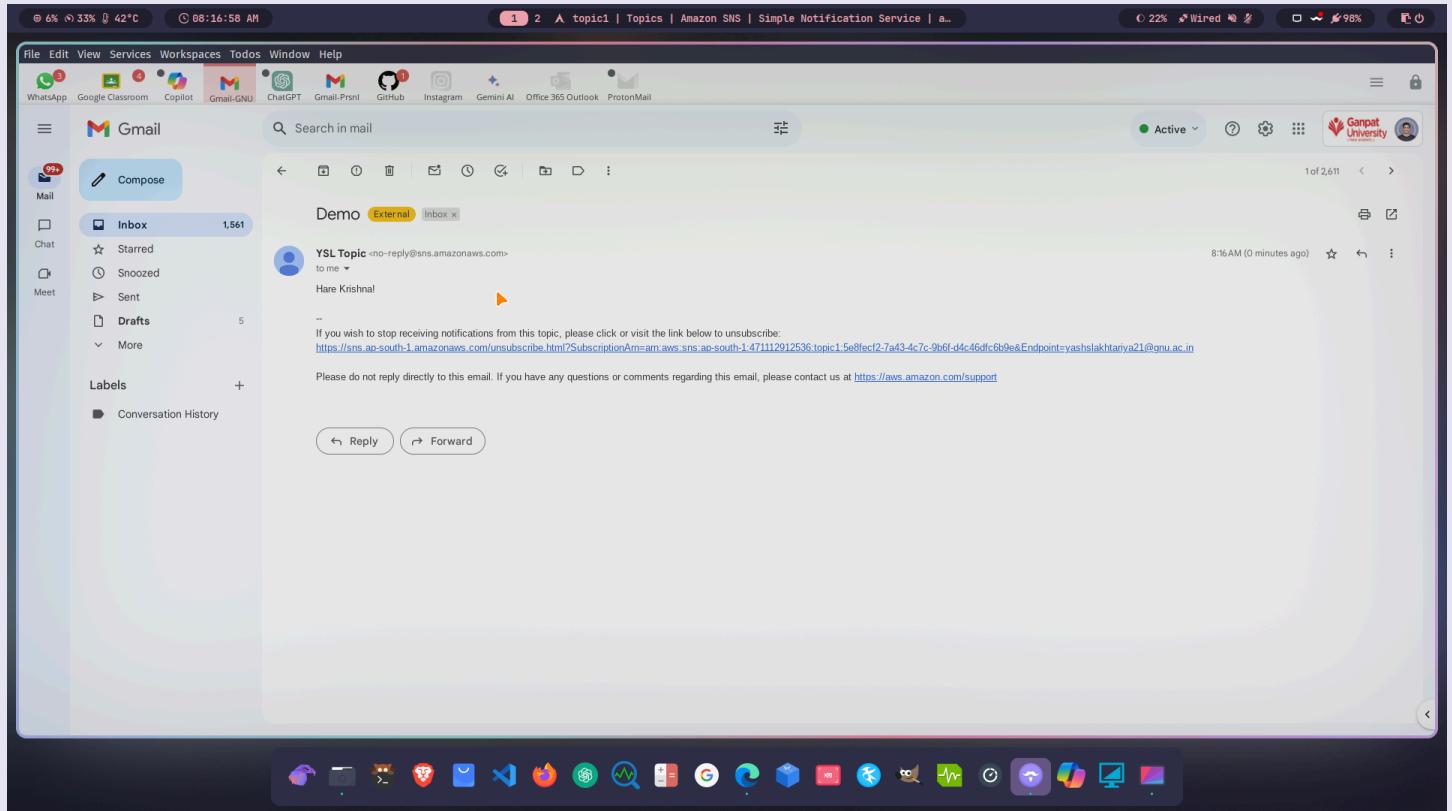
## 10. To check, publish a message in topic we created



The screenshot shows the AWS SNS 'Publish message to topic' interface. At the top, the URL is https://ap-south-1.console.aws.amazon.com/sns/v3/home?region=ap-south-1#/publish/topic/arn:aws:sns:ap-south-1:471112912536:topic1. The page title is 'Publish message | topic1 | Topics | Amazon SNS | Simple Notifi...'. The main section is titled 'Publish message to topic' and contains two tabs: 'Message details' and 'Message body'. In the 'Message details' tab, the 'Topic ARN' is arn:aws:sns:ap-south-1:471112912536:topic1. The 'Subject - optional' field contains 'Demo'. The 'Time to Live (TTL) - optional' field is empty. In the 'Message body' tab, under 'Message structure', the 'Identical payload for all delivery protocols.' option is selected. The message body text area contains 'Hare Krishna!'. The bottom of the screen shows the AWS CloudShell interface and a toolbar with various icons.

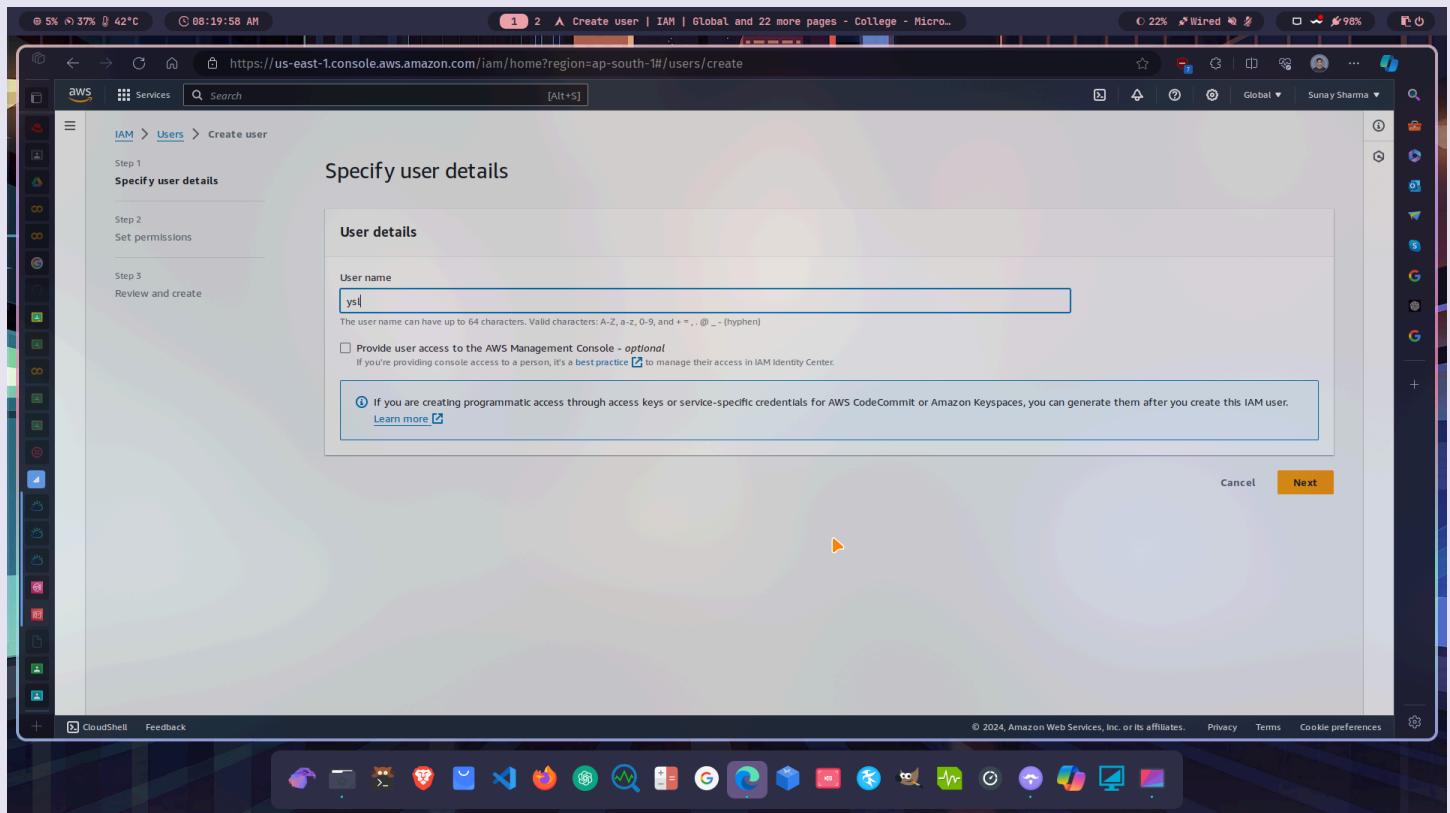
**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 7**

## 11. Check if the subscription works and message is published



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

12. Afterwards, to use this service from local code or application, we can configure IAM Users and add a user for the same



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

### 13. SNS permissions specified here can be set and given to the user

The screenshot shows the AWS IAM 'Create user' wizard at Step 2: Set permissions. The 'Attach policies directly' option is selected. In the 'Permissions policies' section, a search bar filters results for 'SNS'. The table lists five policies:

Policy name	Type	Attached entities
AmazonSNSFullAccess	AWS managed	0
AmazonSNSReadOnlyAccess	AWS managed	0
AmazonSNSSRole	AWS managed	0
AWSElasticBeanstalkRoleSNS	AWS managed	0
AWSIoTDeviceDefenderPublishFindingsToSNSM...	AWS managed	0

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

#### 14. Check if the user is successfully created

The screenshot shows a Microsoft Edge browser window with the AWS IAM console open at the URL <https://us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/users>. The browser's status bar indicates it's 08:22:23 AM, battery level is 6%, and the system temperature is 42°C.

The AWS IAM sidebar menu is visible on the left, showing options like Dashboard, Access management (Users, Roles, Policies), Access reports (Access Analyzer, External access, Unused access, Analyzer settings), and Related consoles (IAM Identity Center).

The main content area displays a green success message: "User created successfully" followed by the text "You can view and download the user's password and email instructions for signing in to the AWS Management Console." Below this message, the "Users (1) Info" section is shown, containing a table with one row:

User name	Path	Group	Last activity	MFA	Password age	Console last sign-in	Access key ID
ysl	/	0	-	-	-	-	-

At the top right of the main content area, there are "View user" and "Create user" buttons. The bottom right corner of the browser window shows the copyright notice "© 2024, Amazon Web Services, Inc. or its affiliates." and links for Privacy, Terms, and Cookie preferences.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

15. Click the user and select 'Security Credentials' tab

The screenshot shows the AWS Identity and Access Management (IAM) console in Microsoft Edge. The URL is https://us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/users/details/ysl?section=security\_credentials. The left sidebar shows the IAM navigation menu with 'Users' selected. The main content area displays the 'Summary' and 'Security credentials' tabs for the user 'ysl'. The 'Security credentials' tab is active, showing the following details:

- ARN:** arn:aws:iam:471112912536:user/ysl
- Console access:** Disabled
- Last console sign-in:** -
- Access key 1:** Create access key

Below this, the 'Console sign-in' section shows a 'Console sign-in link' (https://471112912536.signin.aws.amazon.com/console) and a note that 'Console password Not enabled'. There is a 'Enable console access' button.

The 'Multi-factor authentication (MFA)' section indicates '0' MFA devices assigned. It includes a 'Remove' button, a 'Resync' button, and an 'Assign MFA device' button. A note states: '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 'Learn more' link is provided.

At the bottom of the page, there are links for 'Assign MFA device', '© 2024, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

## 16. Create access key to use in local environment, select Local Code as Use case

The screenshot shows the AWS IAM 'Create access key' interface. On the left, there's a sidebar with navigation links: Step 1 (Access key best practices & alternatives), Step 2 (optional Set description tag), and Step 3 (Retrieve access keys). The main content area is titled 'Access key best practices & alternatives' with a note about avoiding long-term credentials. A large section titled 'Use case' contains several options, with 'Local code' being the selected one (indicated by a blue border). Other options include 'Command Line Interface (CLI)', 'Application running on an AWS compute service', 'Third-party service', 'Application running outside AWS', and 'Other'. At the bottom of this section, there's a yellow warning box titled 'Alternative recommended' suggesting the use of an IDE with the AWS Toolkit. The browser's address bar shows the URL: https://us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/users/details/ysl/create-access-key.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

## 17. Save or copy the access key to use in code

The screenshot shows the AWS IAM 'Create access key' page. At the top, a green banner says 'Access key created'. Below it, the URL is https://us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/users/details/ysl/create-access-key. The main content area has three steps: Step 1 (Access key best practices), Step 2 (optional Set description tag), and Step 3 (Retrieve access keys). Step 3 is active, showing the 'Access key' field containing AKIAW3MEER2MDTNVWRTW and the 'Secret access key' field containing a masked value. A 'Show' link is next to the secret key. Below this is a 'Access key best practices' section with a bulleted list: 'Never store your access key in plain text, in a code repository, or in code.', 'Disable or delete access key when no longer needed.', 'Enable least-privilege permissions.', and 'Rotate access keys regularly.' A note at the bottom of this section says 'For more details about managing access keys, see the [best practices for managing AWS access keys](#)'. At the bottom right are 'Download .csv file' and 'Done' buttons, with 'Done' being highlighted by a cursor.

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Practical 7

18. In code, install the @aws-sdk/client-sns node module and paste the phone number, credentials and other details like region, payload, etc.

```
// Import required AWS SDK modules for SNS (Simple Notification Service)
const { SNSClient, PublishCommand } = require('@aws-sdk/client-sns');

// Asynchronous function to send an SMS message using AWS SNS
async function sendSMSMessage(sns, params) {
    // Create a new PublishCommand with the specified parameters
    const command = new PublishCommand(params);

    // Send the SMS message using the SNS client and the created command
    const message = await sns.send(command);

    // Return the result of the message sending operation
    return message;
} // #6-15 async function sendSMSMessage(sns, params)

// Main asynchronous function (IIFE) to send an SMS message
(async () => {
    // Define parameters for the SMS message
    const params = {
        Message: `Your OTP code is: ${Math.random().toString().substring(2, 10)}`, // Generate a 6-digit OTP code
        PhoneNumber: '+919814893327', // Recipient's phone number from environment variables
        MessageAttributes: {
            'AWS.SNS.SMS.SenderID': {
                DataType: 'String'
            }
        }
    };
    const response = await sns.publish(params);
    console.log(`SMS sent to ${params.PhoneNumber}: ${response.MessageId}`);
})();

yash ~/Practical_7 main !?↑ @ v18.18.2 08:38 npm install @aws-sdk/client-sns
added 81 packages in 20s
1 package is looking for funding
  run 'npm fund' for details
yash ~/Practical_7 main !?↑ @ v18.18.2 08:39
```

Code :

```
// Import required AWS SDK modules for SNS (Simple Notification Service)
const { SNSClient, PublishCommand } = require('@aws-sdk/client-sns');

// Asynchronous function to send an SMS message using AWS SNS
async function sendSMSMessage(sns, params) {
    // Create a new PublishCommand with the specified parameters
    const command = new PublishCommand(params);

    // Send the SMS message using the SNS client and the created command
    const message = await sns.send(command);
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

```
// Return the result of the message sending operation
return message;
}

// Main asynchronous function (IIFE) to send an SMS message
(async () => {
    // Define parameters for the SMS message
    const params = {
        Message: `Your OTP code is: ${Math.random().toString().substring(2,
10)}`, // Generate a 6-digit OTP code
        PhoneNumber: '+918238680208', // Recipient's phone number from
environment variables
        MessageAttributes: {
            'AWS.SNS.SMS.SenderID': {
                'DataType': 'String',
                'StringValue': 'String'
            }
        }
    };
}

// Create an SNS client with the specified configuration
const sns = new SNSClient({
    region: "ap-south-1", // AWS region from environment variables
    credentials: {
        accessKeyId: "AKIAZI2LFDY46N3U3SFY", // AWS access key from
environment variables
        secretAccessKey: "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" // AWS secret key from environment variables
    }
});

// Send the SMS message using the defined SNS client and parameters
```

**Name - Yash Lakhtariya**

Enrollment number - 21162101012

**Branch - CBA**      **Batch - 61**

# EADC Practical 7

```
    await sendSMSMessage(sns, params);  
})();
```

19. Run the code and check if the message arrives

The screenshot shows a VS Code interface with the following details:

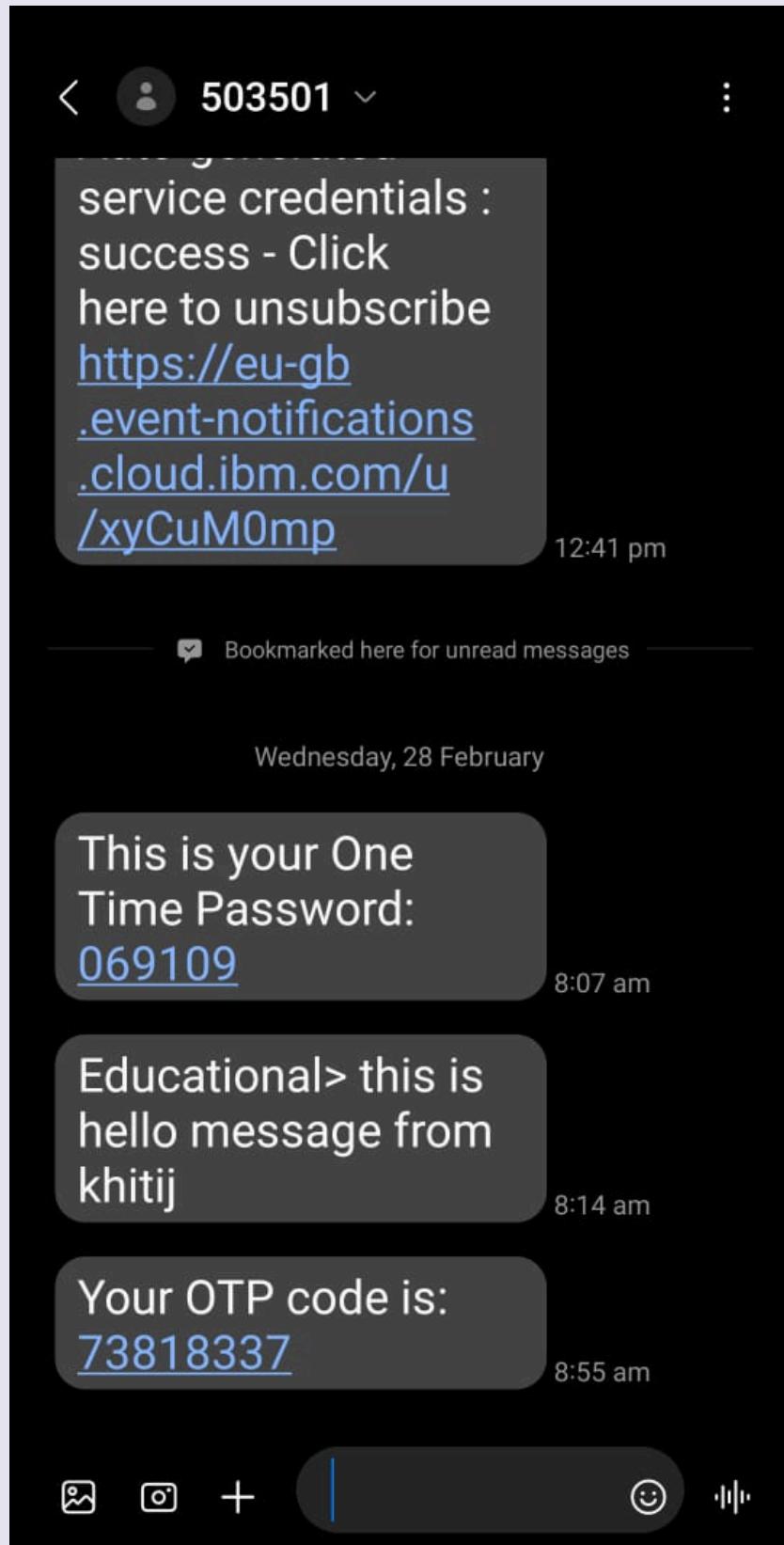
- Explorer View:** Shows a file tree with a folder named "PRACTICAL\_7" containing "node\_modules", "index.js", "package-lock.json", and "package.json".
- Editor View:** The main editor window displays "index.js" code. The code uses the AWS SDK for Node.js to send an SMS message via SNS. It includes environment variable placeholders for the OTP code, recipient phone number, and AWS credentials.
- Terminal View:** The terminal shows the command `node "/home/yash/Documents/sem6practicals/EADC/Practical\_7/index.js"`. The output indicates an error: `Error: InvalidClientTokenId`.
- Status Bar:** Shows the node version as v18.18.2 and the current date and time as 08:52.

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Practical 7



Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Practical 7

## 20. Visit the created toolchain to add Slack integration and add integration

The screenshot shows a web browser window for the IBM Cloud DevOps Catalog. The URL in the address bar is [https://cloud.ibm.com/devops/catalog?toolchainId=bb827e32-4aca-445a-bd35-29268f087536&env\\_id=ibmypseu-gb](https://cloud.ibm.com/devops/catalog?toolchainId=bb827e32-4aca-445a-bd35-29268f087536&env_id=ibmypseu-gb). The page title is "Add tool integration". On the left, there's a sidebar with categories like All integrations (23), Communication, Artifacts, Monitoring, Version Control, CI/CD, Quality, Planning, IBM, Secrets, and Financial Services Validated. A search bar at the top has "slack" typed into it. Below the search bar, the results show a card for "Slack" with the subtext "Third Party • Developer Tools" and the description "Coordinate and collaborate on your project." The "Communication" category is highlighted in blue. At the bottom of the page, the URL <https://cloud.ibm.com/devops/catalog/slack?toolchainId=bb827e32-4aca-445a-...> is visible.

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Practical 7

21. Visit Slack API website mentioned in IBM Cloud help text mentioned while creating the integration and now, create Slack app (if not exists)

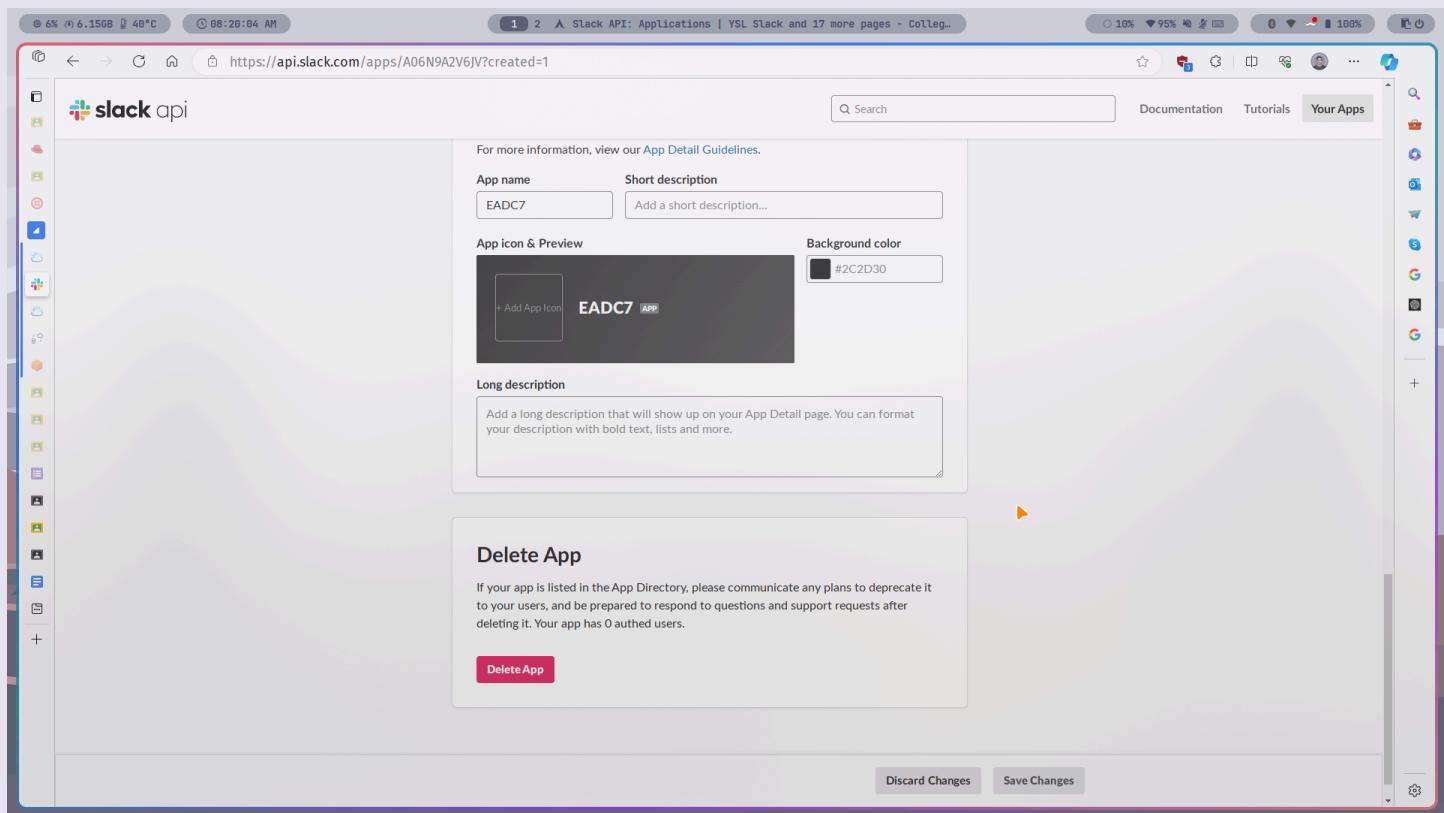
The screenshot shows a browser window displaying the Slack API documentation at <https://api.slack.com/messaging/webhooks>. The page is titled "Getting started with incoming webhooks". It provides a 4-step guide:

- 1. Create a Slack app (if you don't have one already)**: A button labeled "Create your Slack app". Below it, text says: "Pick a name, choose a workspace to associate your app with (bear in mind you'll probably be posting lots of test messages, so you may want to create a channel for sandbox use), then click **Create App**. If you've already created an app, you can use that one. Have a cookie for being prepared! 🍪".
- 2. Enable incoming webhooks**: Text says: "You'll be redirected to the settings page for your new app (if you're using an existing app, you can load its settings via your [app's management dashboard](#)). From here, select **Incoming Webhooks**, and toggle **Activate Incoming Webhooks** to on. If you already have this activated, well, you deserve another cookie! 🍪".
- 3. Create an incoming webhook**: This section is currently empty.

A purple banner at the bottom left encourages users to "Try something new!" and mentions the Slack API's modular automation platform. On the right, there's a sidebar with links to "Getting started with incoming webhooks" and other documentation sections like "Making it fancy with advanced formatting" and "Posting your message as a reply in a thread".

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 7**

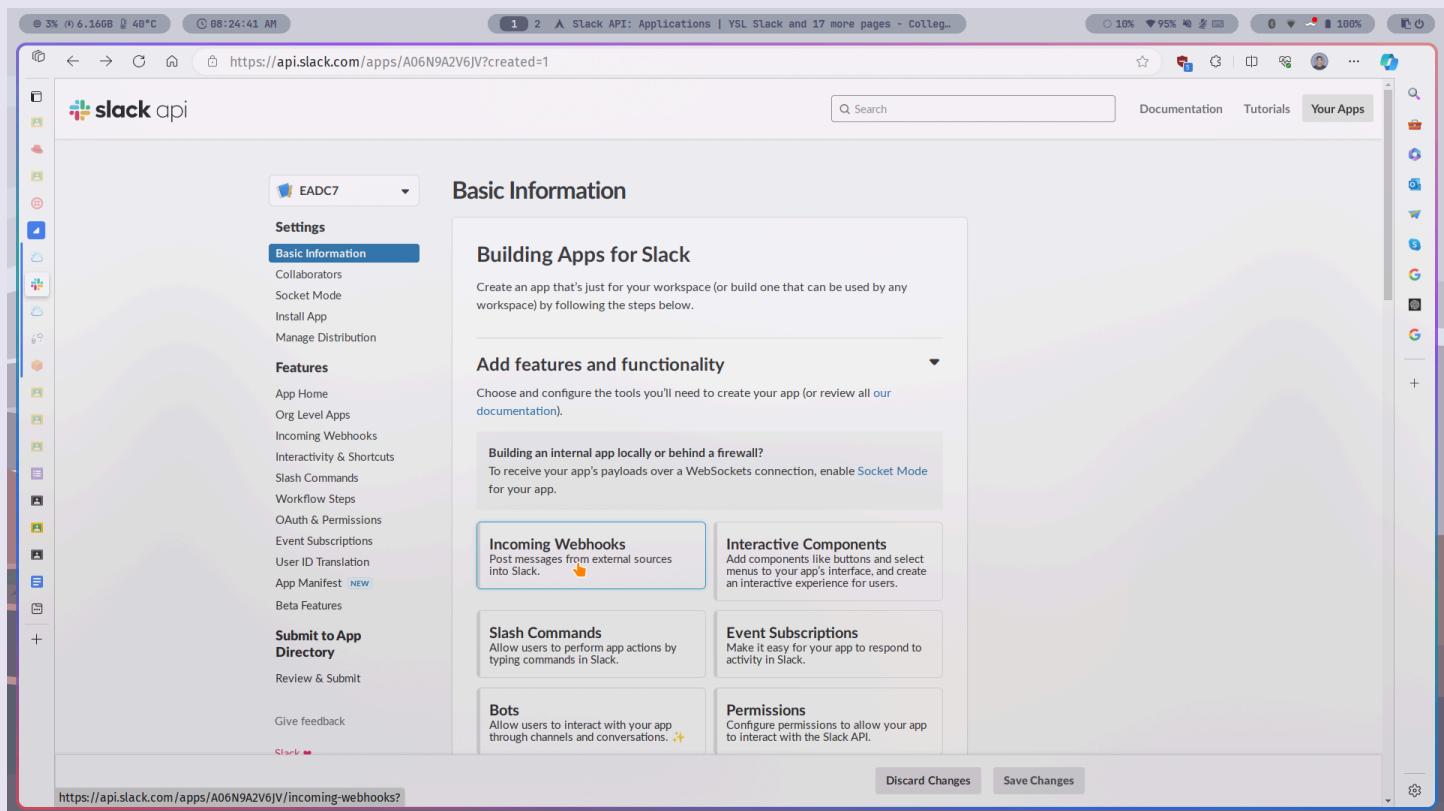
## 22. Specify the name and save the changes



The screenshot shows a web browser window with the URL <https://api.slack.com/apps/A06N9A2V6JV?created=1>. The page is titled "slack api". On the left, there's a sidebar with various icons. The main content area has sections for "App name" (EADC7), "Short description" (Add a short description...), "App Icon & Preview" (with a placeholder image and a "Add App Icon" button), "Background color" (#2C2D30), and "Long description" (Add a long description that will show up on your App Detail page. You can format your description with bold text, lists and more.). Below this is a "Delete App" section with a warning message about deprecating the app if it's listed in the App Directory, and a "Delete App" button. At the bottom are "Discard Changes" and "Save Changes" buttons.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

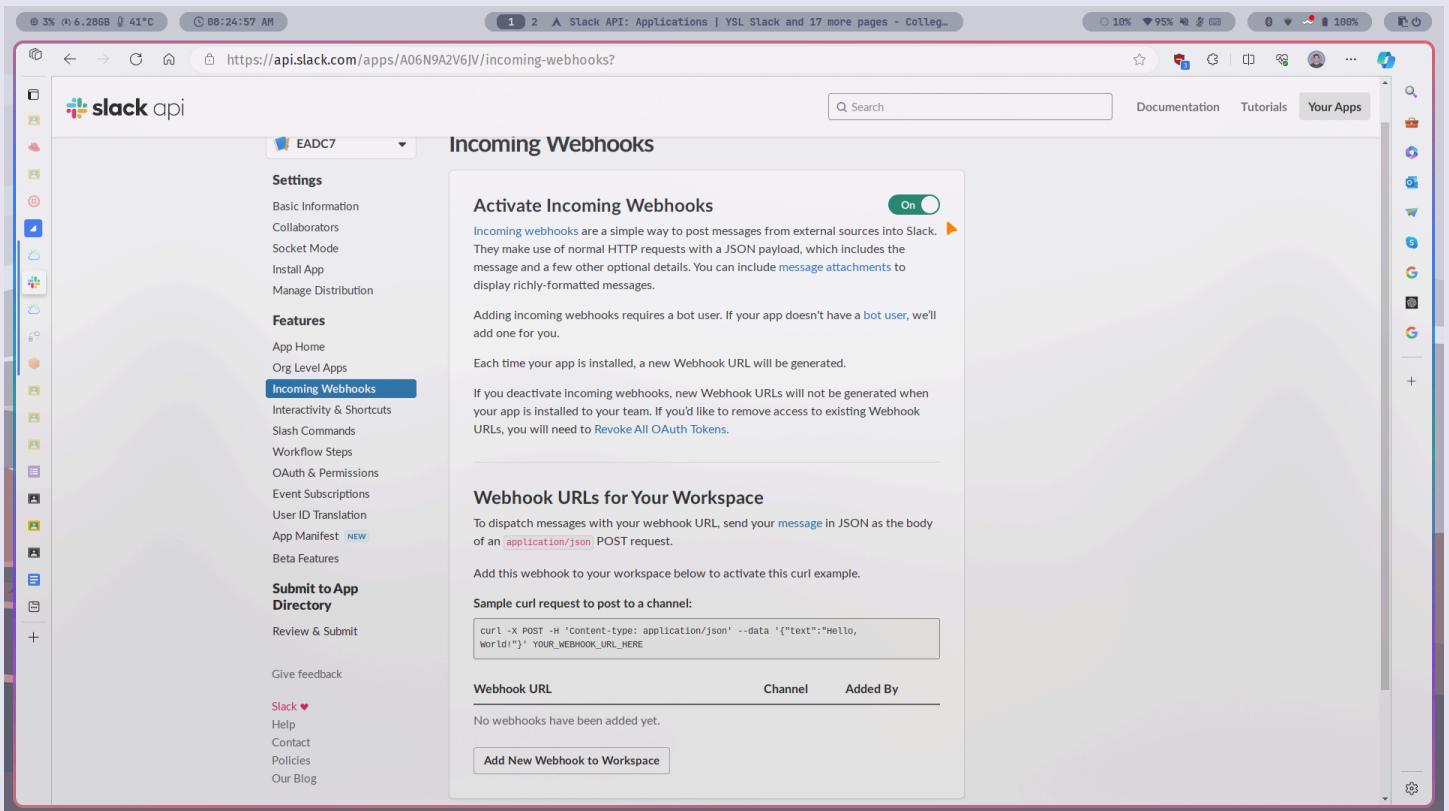
## 23. Now, add incoming webhooks feature in the app



The screenshot shows the Slack API Applications interface. The left sidebar lists various app settings like Collaborators, Socket Mode, and Manage Distribution. The main content area is titled 'Basic Information' under 'Building Apps for Slack'. It includes sections for 'Add features and functionality' (with options for Incoming Webhooks, Interactive Components, Slash Commands, Event Subscriptions, Bots, and Permissions), 'Building an internal app locally or behind a firewall?' (with instructions for Socket Mode), and a 'Documentation' and 'Tutorials' link. At the bottom, there are 'Discard Changes' and 'Save Changes' buttons. A yellow arrow points to the 'Incoming Webhooks' section.

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 7**

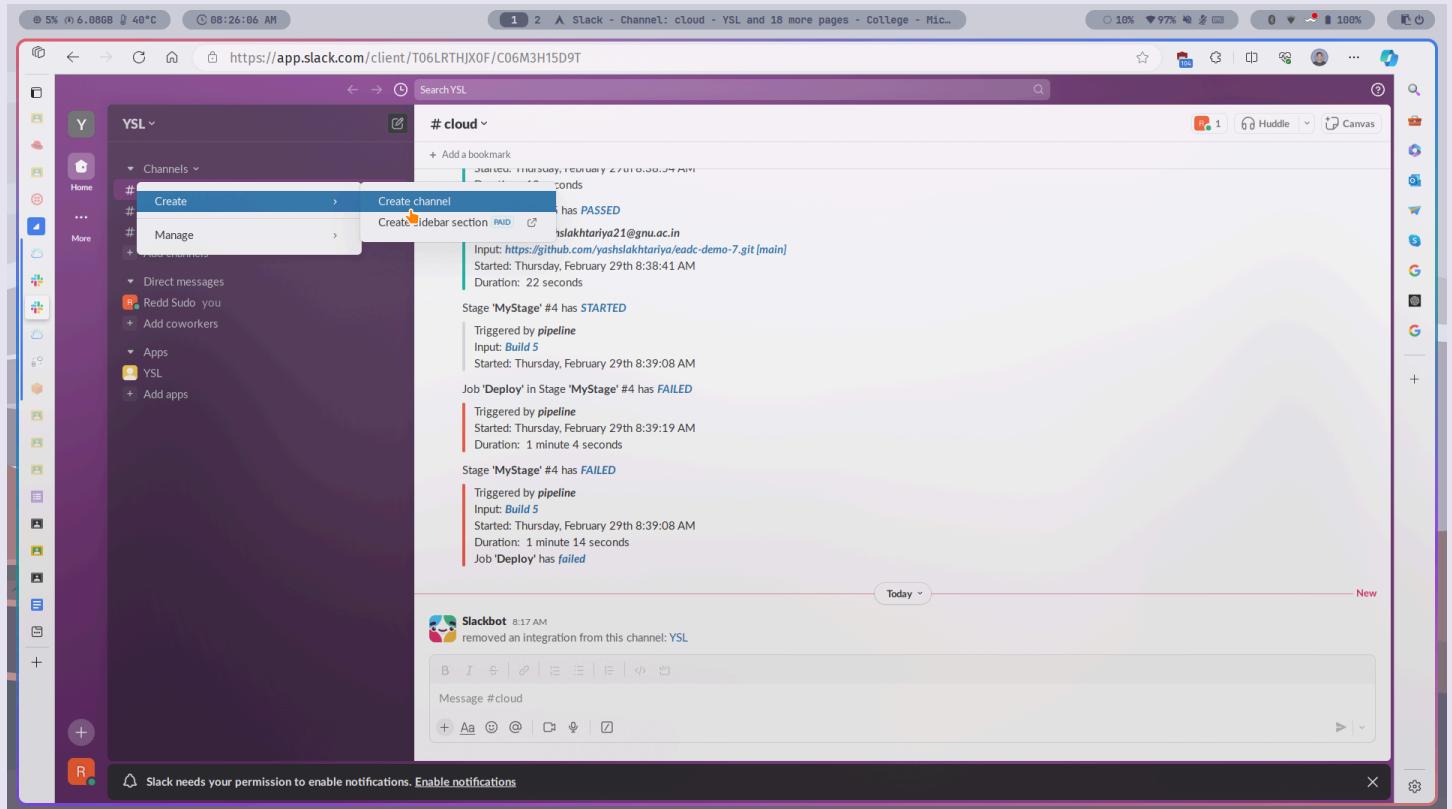
## 24. Activate the feature and add new webhook to Workspace



The screenshot shows the Slack API Applications interface. The URL in the address bar is <https://api.slack.com/apps/A06N9A2V6JV/incoming-webhooks>. The left sidebar shows the app 'EADC' with various settings and features listed. The 'Incoming Webhooks' section is currently selected. The main content area displays the 'Activate Incoming Webhooks' configuration, which is turned 'On'. It includes instructions for using incoming webhooks to post messages from external sources into Slack. Below this, there's a section for 'Webhook URLs for Your Workspace' with a sample curl command and a button to 'Add New Webhook to Workspace'. A note at the bottom states 'No webhooks have been added yet.'

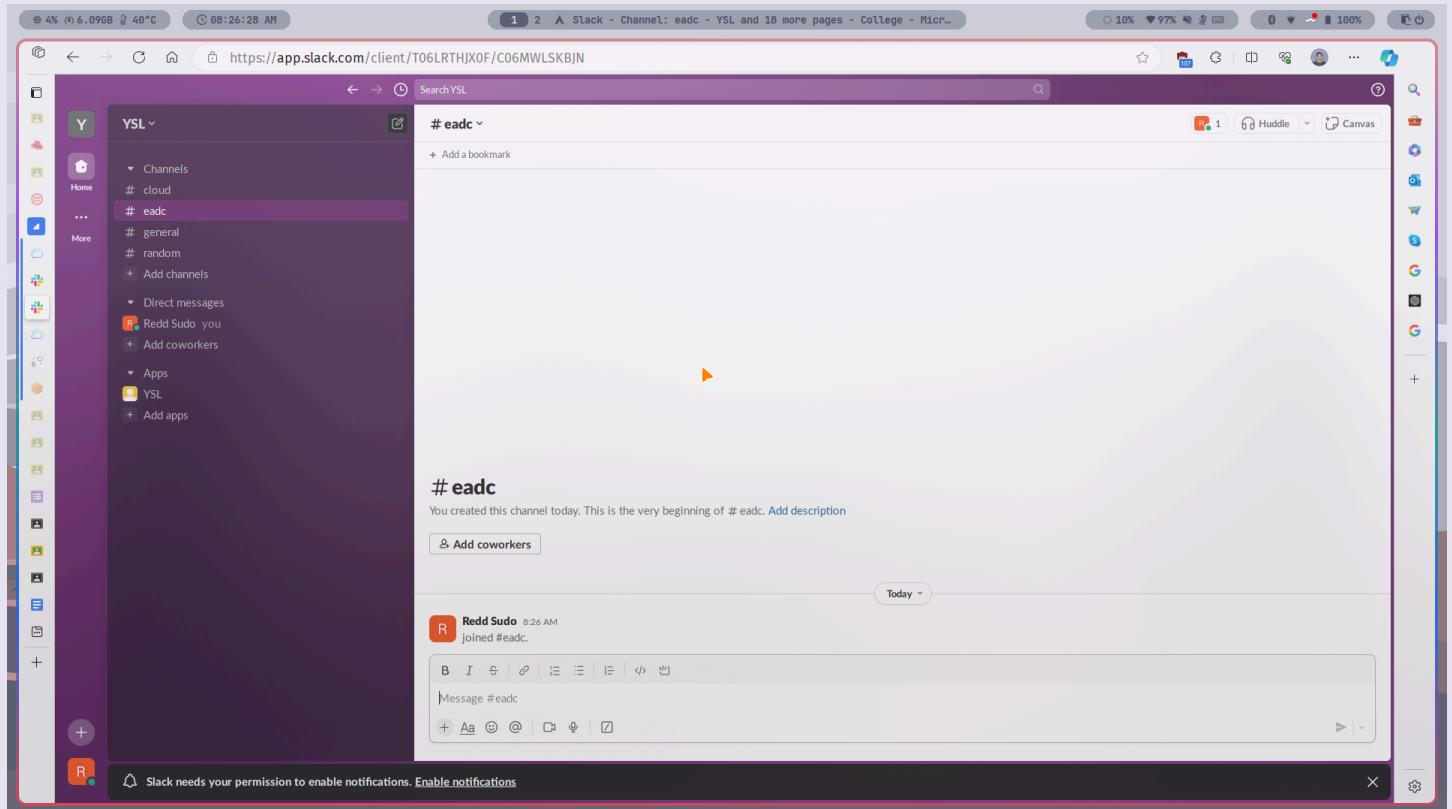
Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

## 25. Create the new channel if required in slack app



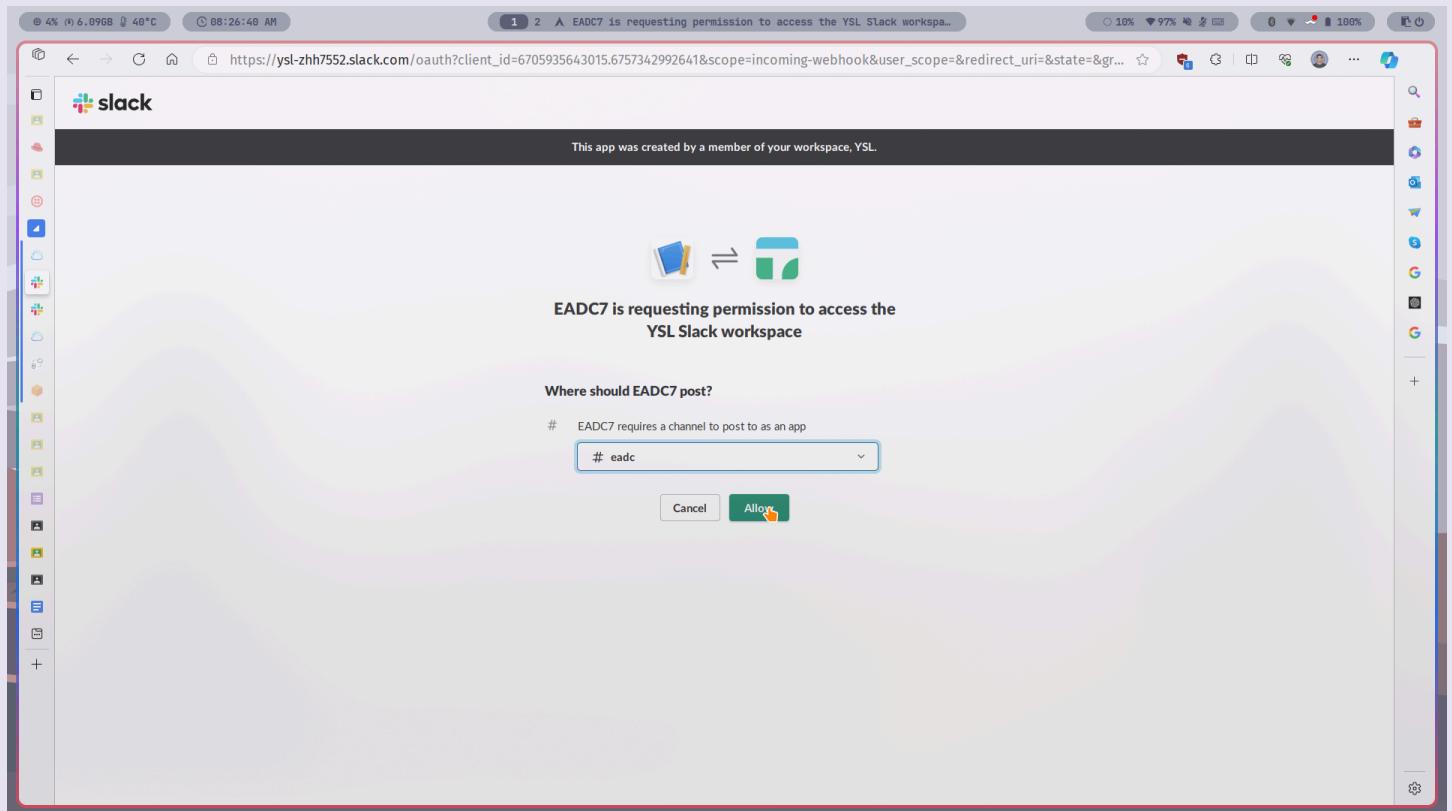
**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 7**

## 26. Now, the channel will be created



**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 7**

## 27. Allow access to the channel



Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

EADC Practical 7

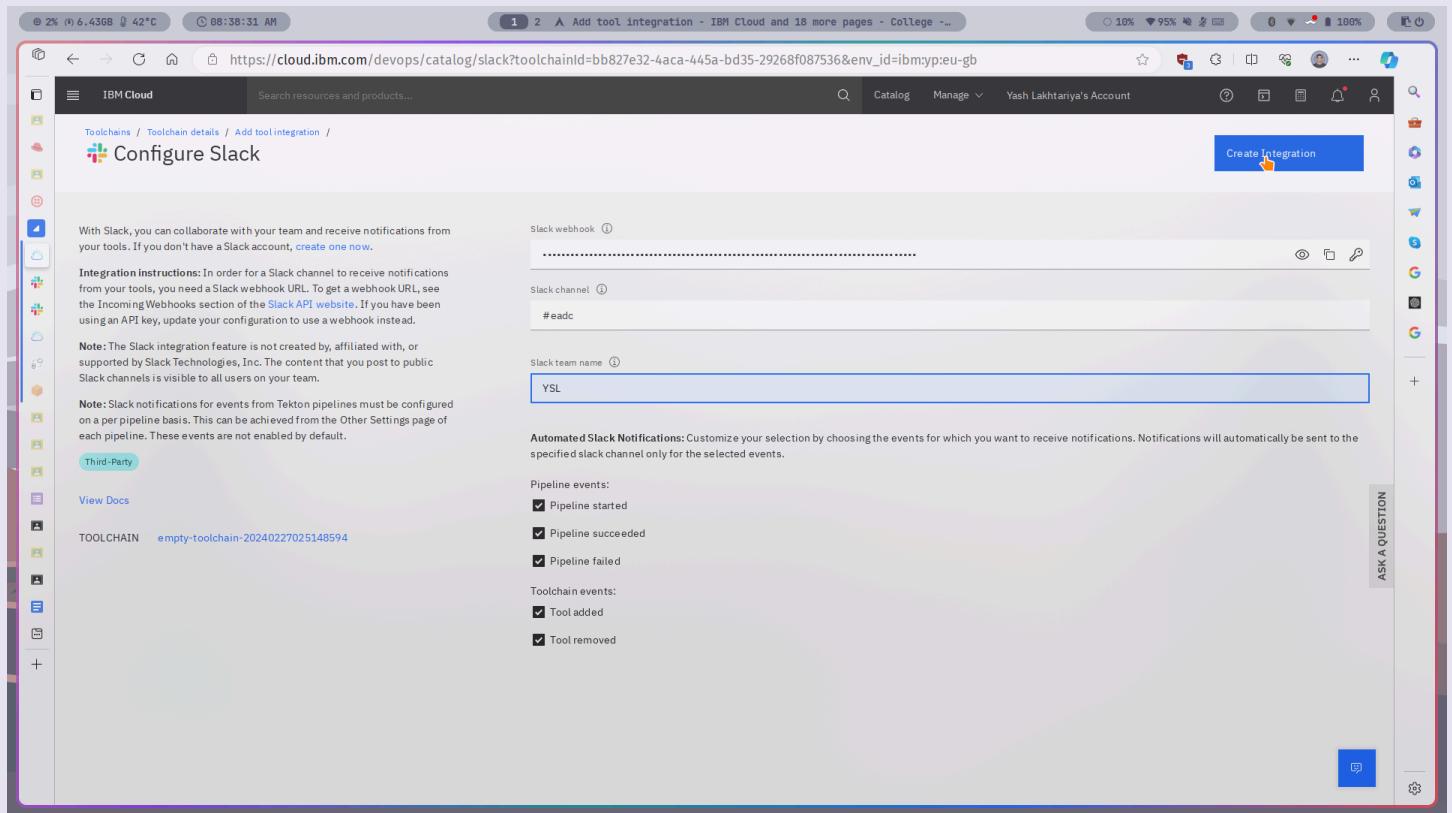
## 28. Now, webhook URL will be created and copy the URL

The screenshot shows a web browser window with the URL <https://api.slack.com/apps/A06N9A2V6JV/incoming-webhooks?success=1>. The page displays information about incoming webhooks, including a note that a bot user is required and a warning about deactivation. It then lists "Webhook URLs for Your Workspace" with a curl command example and a table of existing webhook entries. One entry is selected, and a "Copy" button is highlighted with a red box.

Webhook URL	Channel	Added By
<a href="https://hooks.slack.com/services/T06LRTHJX0F/B00MG489BF1/SfL1fKBLBm9uu1g3EYCPZ">https://hooks.slack.com/services/T06LRTHJX0F/B00MG489BF1/SfL1fKBLBm9uu1g3EYCPZ</a>	#eadc	Reddit Sudo Mar 4, 2024

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Practical 7**

## 29. Paste the credentials like webhook URL, channel name and team name



The screenshot shows the 'Configure Slack' page in the IBM Cloud interface. The URL in the address bar is [https://cloud.ibm.com/devops/catalog/slack?toolchainId=bb827e32-4aca-445a-bd35-29268f087536&env\\_id=ibm:yp:eu-gb](https://cloud.ibm.com/devops/catalog/slack?toolchainId=bb827e32-4aca-445a-bd35-29268f087536&env_id=ibm:yp:eu-gb). The page title is 'Add tool integration - IBM Cloud and 18 more pages - College ...'. On the left, there's a sidebar with various icons and a search bar. The main content area has sections for 'Slack webhook' (with a placeholder '.....'), 'Slack channel' (set to '#eadc'), and 'Slack team name' (set to 'YSL'). A large blue button labeled 'Create Integration' is at the top right. Below these fields, there's a section titled 'Automated Slack Notifications' with checkboxes for 'Pipeline events' (Pipeline started, Pipeline succeeded, Pipeline failed) and 'Toolchain events' (Tool added, Tool removed). A 'View Docs' link is also present.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

30. Now, the notification will be arrived for integration done

The screenshot shows a Slack interface with the following details:

- Channel:** #eadc
- Message:** "Service Slack '#eadc' has been bound to toolchain 'empty-toolchain-20240227025148594' by yashslakhtariya21@gnu.ac.in and is configured"
- Slack Status Bar:** Shows battery at 3%, signal strength, 6.43GB, 42°C, 88:39:03 AM, 10% battery, 95% volume, and 100% screen brightness.
- Bottom Bar:** "Slack needs your permission to enable notifications. Enable notifications"

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

### 31. Run the delivery pipeline one or more stages to check if notification arrives

The screenshot shows a web browser with two tabs open. The left tab is titled "dlvryppln | Delivery Pipeline" and displays a delivery pipeline interface. It shows two stages: "MyStage" (STAGE RUNNING) and "MyStage" (STAGE FAILED). The failed stage has a status message: "Build 5 Failed 5d ago". The right tab is titled "# eadc" and shows a Slack channel interface. It displays messages from users "Redd Sudo" and "EADC7" about the integration setup. A message from "EADC7" at 8:38 AM indicates that the Stage 'MyStage' #6 has started.

IBM Cloud  
dlvryppln | Delivery Pipeline

MyStage STAGE RUNNING...  
LAST INPUT Last commit by Yash Lakhtariya 6d ago updated  
JOBS Build Queued  
LAST EXECUTION RESULT Build 5

MyStage STAGE FAILED  
LAST INPUT Stage: MyStage / Job: Build  
Build 5  
JOBS View logs and history Deploy Failed 5d ago  
LAST EXECUTION RESULT No results

ASK A QUESTION

# eadc

You created this channel today. This is the very beginning of # eadc. Add description

Add coworkers

Today

Redd Sudo 8:26 AM joined #eadc.  
Redd Sudo 8:26 AM added an integration to this channel: EADC7

EADC7 8:38 AM Webhook configured by DevOps Services integration (51653a09-97ae-4a5c-9718-314bce6b35e)  
Service Slack '#eadc' has been bound to toolchain 'empty-toolchain-20240227025148594' by yashslakhtariya21@gnu.ac.in and is configured

EADC7 8:50 AM Stage 'MyStage' #6 has STARTED

Triggered by yashslakhtariya21@gnu.ac.in  
Input: <https://github.com/yashslakhtariya/eadc-demo-7.git> [main]  
Started: Tuesday, March 5th 8:50:54 AM

Message #eadc

Slack needs your permission to enable notifications. [Enable notifications](#)

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Practical 7

## 32. Hence, the integration is successful with Slack

The screenshot shows a dual-pane interface. On the left, the 'IBM Cloud' delivery pipeline interface displays two stages: 'MyStage' and 'MyStage'. The first stage is 'STAGE PASSED' with a green bar, showing 'Last commit by Yash Lakhtariya 6d ago updated' and a 'Build' job status of 'Passed now'. The second stage is 'STAGE RUNNING...' with a blue bar, showing 'Build 6' and a 'Deploy' job status of 'Queued'. On the right, a Slack window titled '# eadc' shows a message from 'EADC7' at 8:38 AM: 'added an integration to this channel: EADC7'. Below it, another message from 'EADC7' at 8:50 AM states: 'Webhook configured by DevOps Services integration (51653a09-97ae-4a5c-9718-314bceab35e) Service Slack '#adc' has been bound to toolchain 'empty-toolchain-20240227025148594' by yashlakhtariya21@gnu.ac.in and is configured'. A message from 'yashlakhtariya21@gnu.ac.in' at 8:50:54 AM indicates 'Stage 'MyStage' #6 has STARTED'. A message from 'yashlakhtariya21@gnu.ac.in' at 8:50:54 AM indicates 'Job 'Build' in Stage 'MyStage' #6 has PASSED'. A message from 'yashlakhtariya21@gnu.ac.in' at 8:50:54 AM indicates 'Triggered by yashlakhtariya21@gnu.ac.in Input: https://github.com/yashlakhtariya/eadc-demo-7.git [main] Started: Tuesday, March 5th 8:50:54 AM Duration: 10 seconds'. A message from 'yashlakhtariya21@gnu.ac.in' at 8:50:54 AM indicates 'Stage 'MyStage' #5 has PASSED'. A message from 'yashlakhtariya21@gnu.ac.in' at 8:50:54 AM indicates 'Triggered by yashlakhtariya21@gnu.ac.in Input: https://github.com/yashlakhtariya/eadc-demo-7.git [main] Started: Tuesday, March 5th 8:50:54 AM Duration: 22 seconds'. A message from 'yashlakhtariya21@gnu.ac.in' at 8:51:20 AM indicates 'Stage 'MyStage' #5 has STARTED'. A message from 'yashlakhtariya21@gnu.ac.in' at 8:51:20 AM indicates 'Triggered by pipeline Input: Build 6 Started: Tuesday, March 5th 8:51:20 AM'. At the bottom of the Slack window, a notification says 'Slack needs your permission to enable notifications. Enable notifications'.