

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Assignment 2

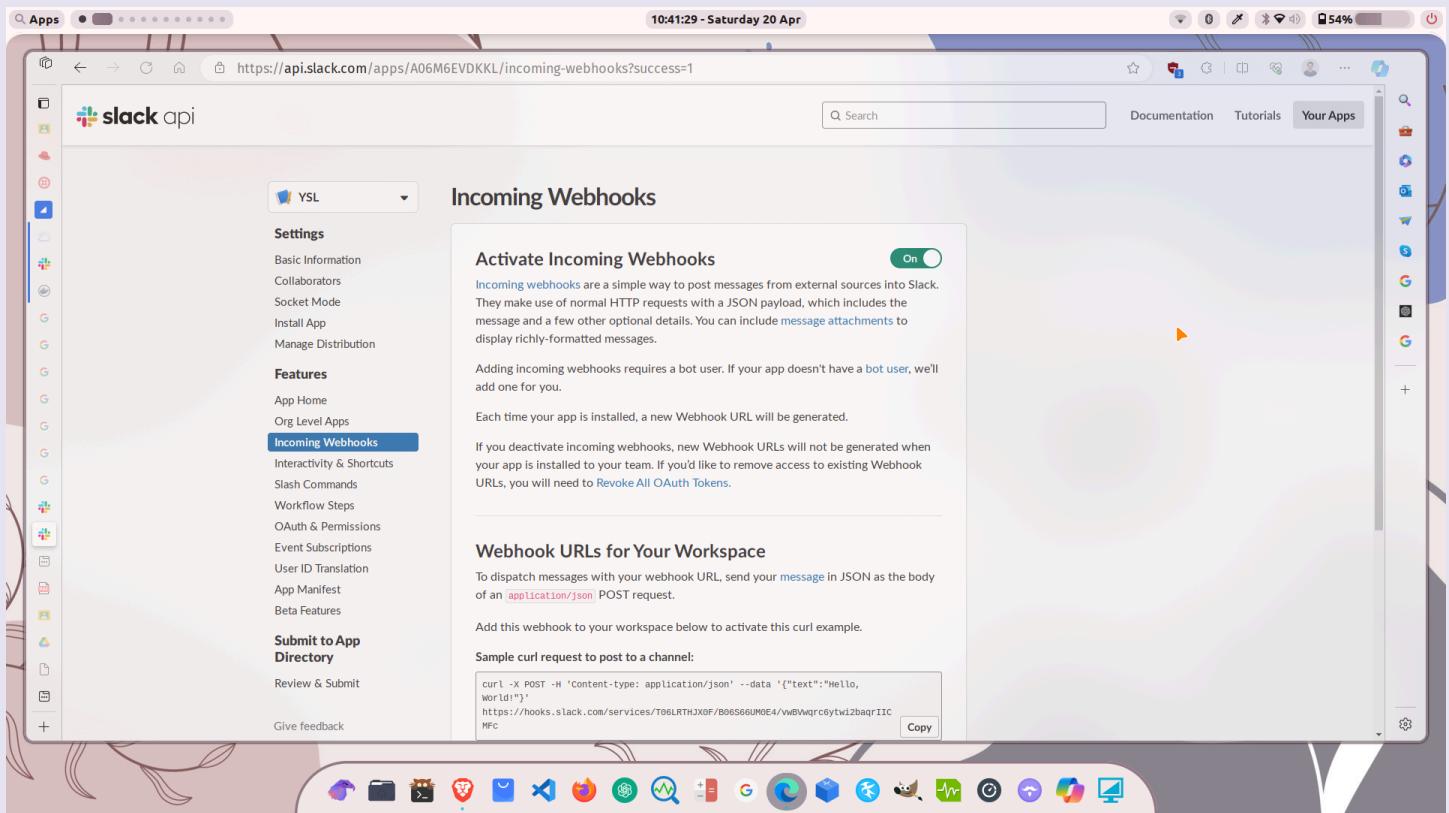
## **Aim : Cloud Integration and Graph Database Exploration**

### **Tasks, steps and screenshots :**

#### **1. Configure Issue Tracker Notifications on IBM Cloud :**

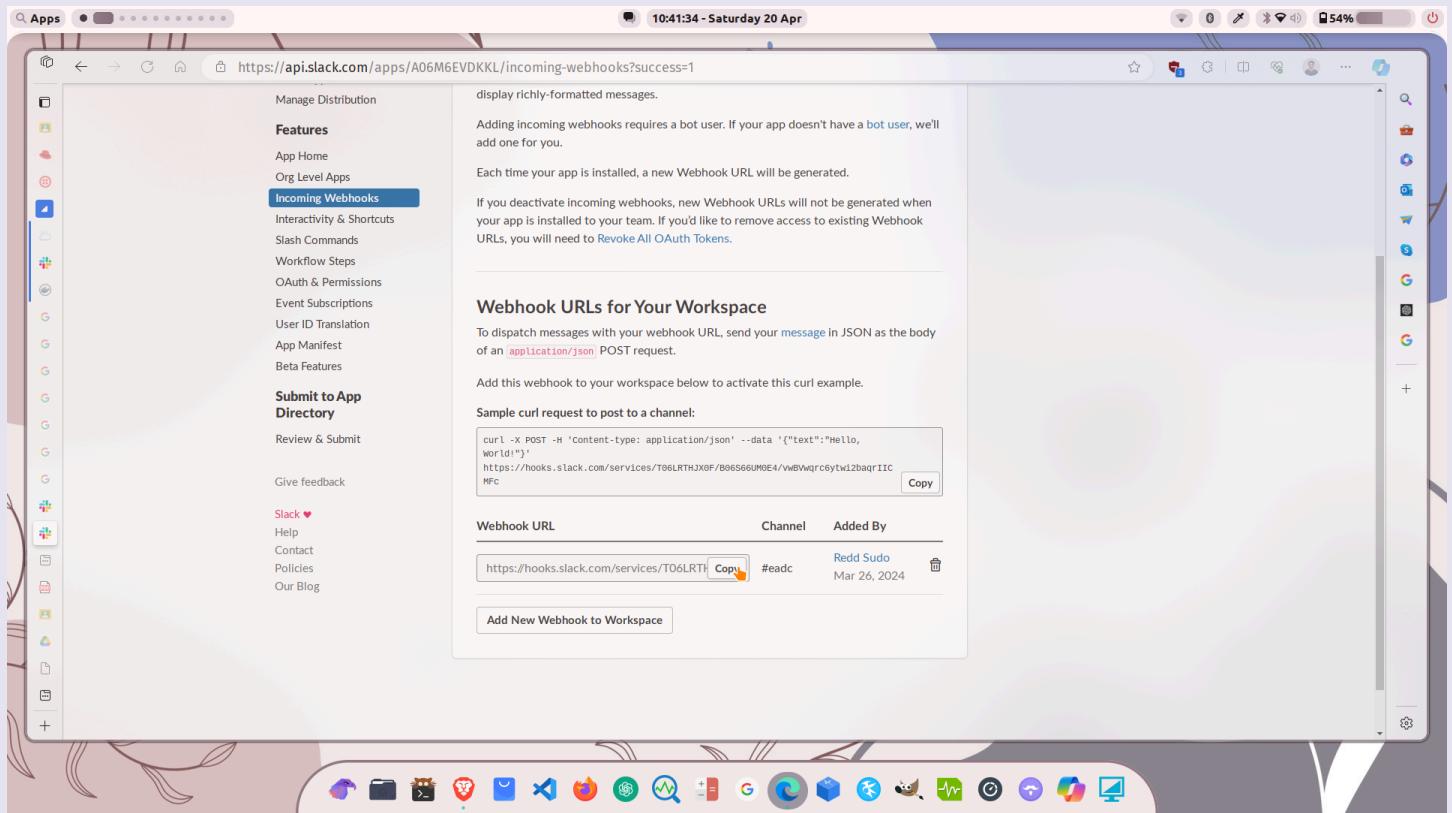
- Set up an issue tracker service on IBM Cloud using a toolchain template.**
- Configure notifications to send alerts to Slack channels for issues published, closed, or reopened.**

- Visit [api.slack.com](https://api.slack.com) and login, activate incoming webhook and create new one if not exists



**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Assignment 2**

**b. Copy the webhook URL for using it later in IBM Cloud**



The screenshot shows a web browser window with the URL <https://api.slack.com/apps/A06M6EVVKKL/incoming-webhooks?success=1>. The page displays the configuration for incoming webhooks. On the left, there's a sidebar with various app features like Manage Distribution, Features (with Incoming Webhooks selected), App Home, Org Level Apps, and more. The main content area shows instructions for adding a bot user and generating a webhook URL. A sample curl command is provided:

```
curl -X POST -H 'Content-type: application/json' --data '{"text":"Hello, world!"}' https://hooks.slack.com/services/T06LRTJX0F/B06566UM0E4/VwBwqrc6ytwizBaqrIIC
```

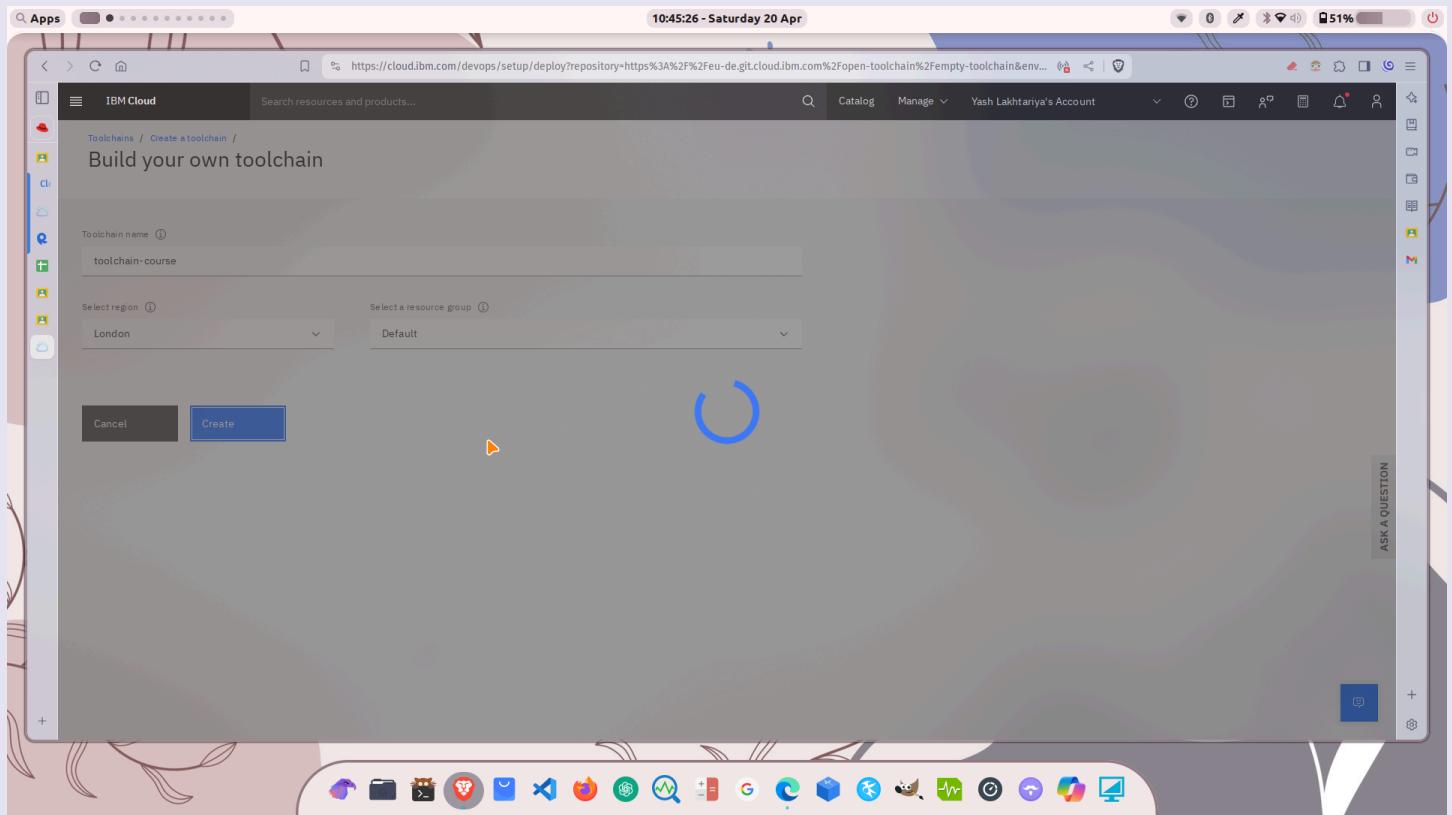
A 'Copy' button is visible next to the command. Below this, a table lists the current webhook entry:

Webhook URL	Channel	Added By
<a href="https://hooks.slack.com/services/T06LRTJX0F/B06566UM0E4/VwBwqrc6ytwizBaqrIIC">https://hooks.slack.com/services/T06LRTJX0F/B06566UM0E4/VwBwqrc6ytwizBaqrIIC</a>	#eadc	Redd Sudo Mar 26, 2024

At the bottom, there's a 'Copy' button for the webhook URL.

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Assignment 2**

**c. Create new empty toolchain in IBM Cloud**



Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

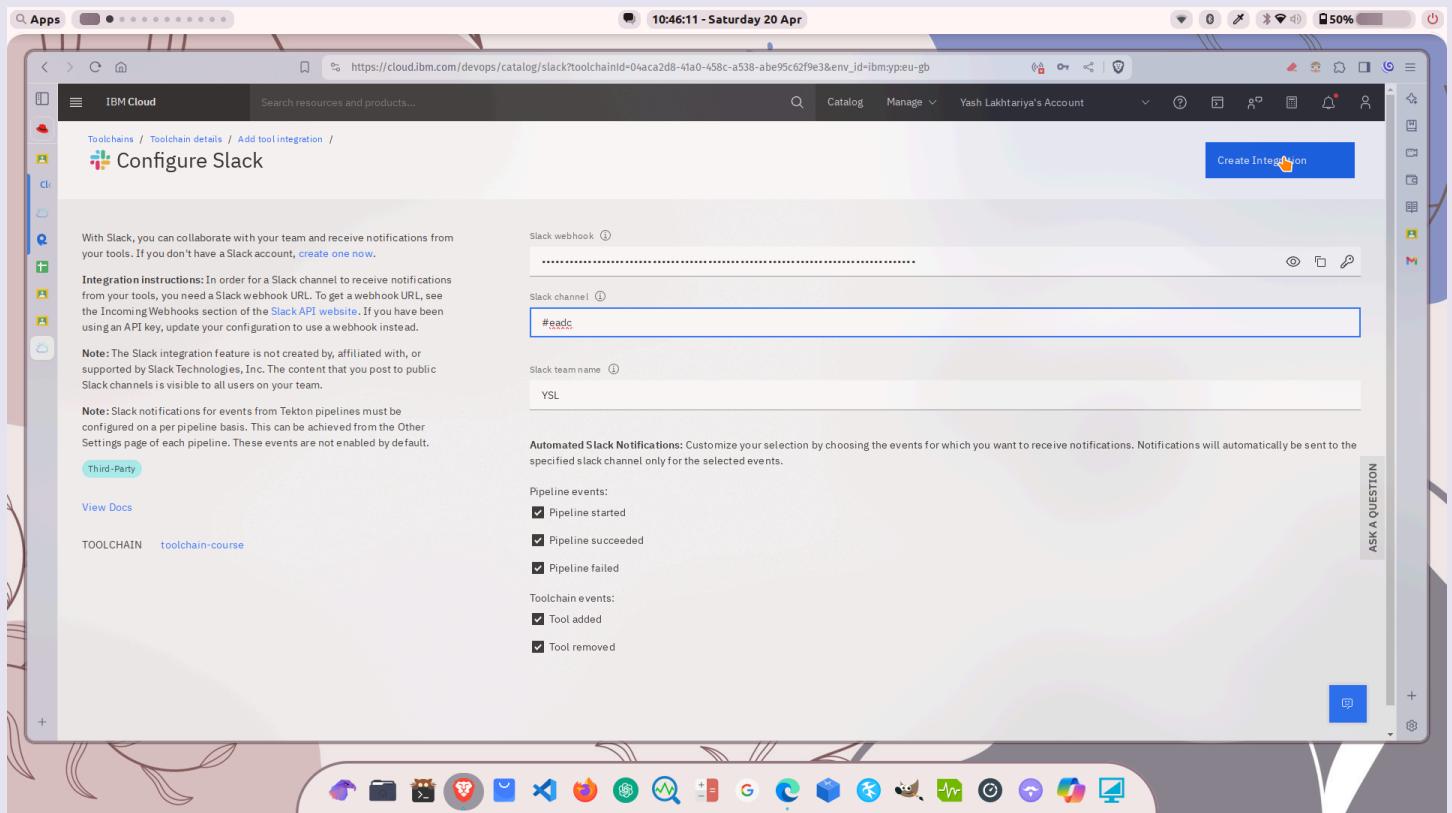
EADC Assignment 2

d. Click Add to add integrations, where Slack can be added

The screenshot shows the IBM Cloud DevOps toolchain interface. At the top, there's a navigation bar with 'Catalog' and 'Manage' tabs, and a user account dropdown for 'Yash Lakhtariya'. Below the navigation is a search bar and a 'Details' button. The main area is titled 'Toolchains / toolchain-course' and contains sections for 'Overview', 'Connections', and 'Manage'. A prominent message box says, 'Your toolchain is ready! Quick start: You can now add tool integrations. For more details, view the documentation.' Below this are three cards: 'Repositories 0 total', 'Delivery pipelines 0 total', and 'IBM Cloud tools 0 total'. At the bottom right of the main area, there's a blue 'Add' button with a '+' icon. The status bar at the bottom shows various application icons.

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Assignment 2**

e. Add Slack Integration and paste the webhook URL, channel and team name there



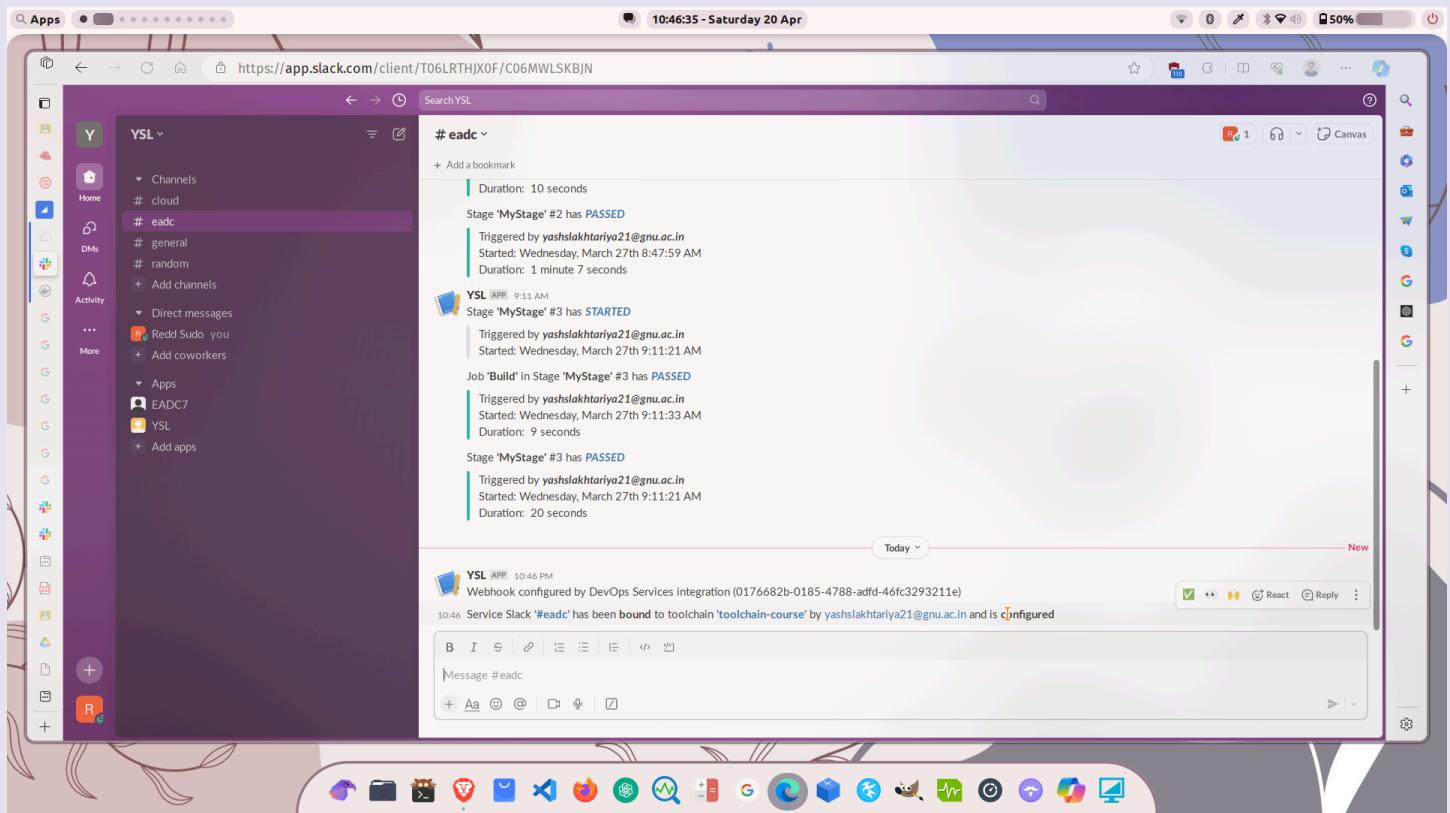
Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

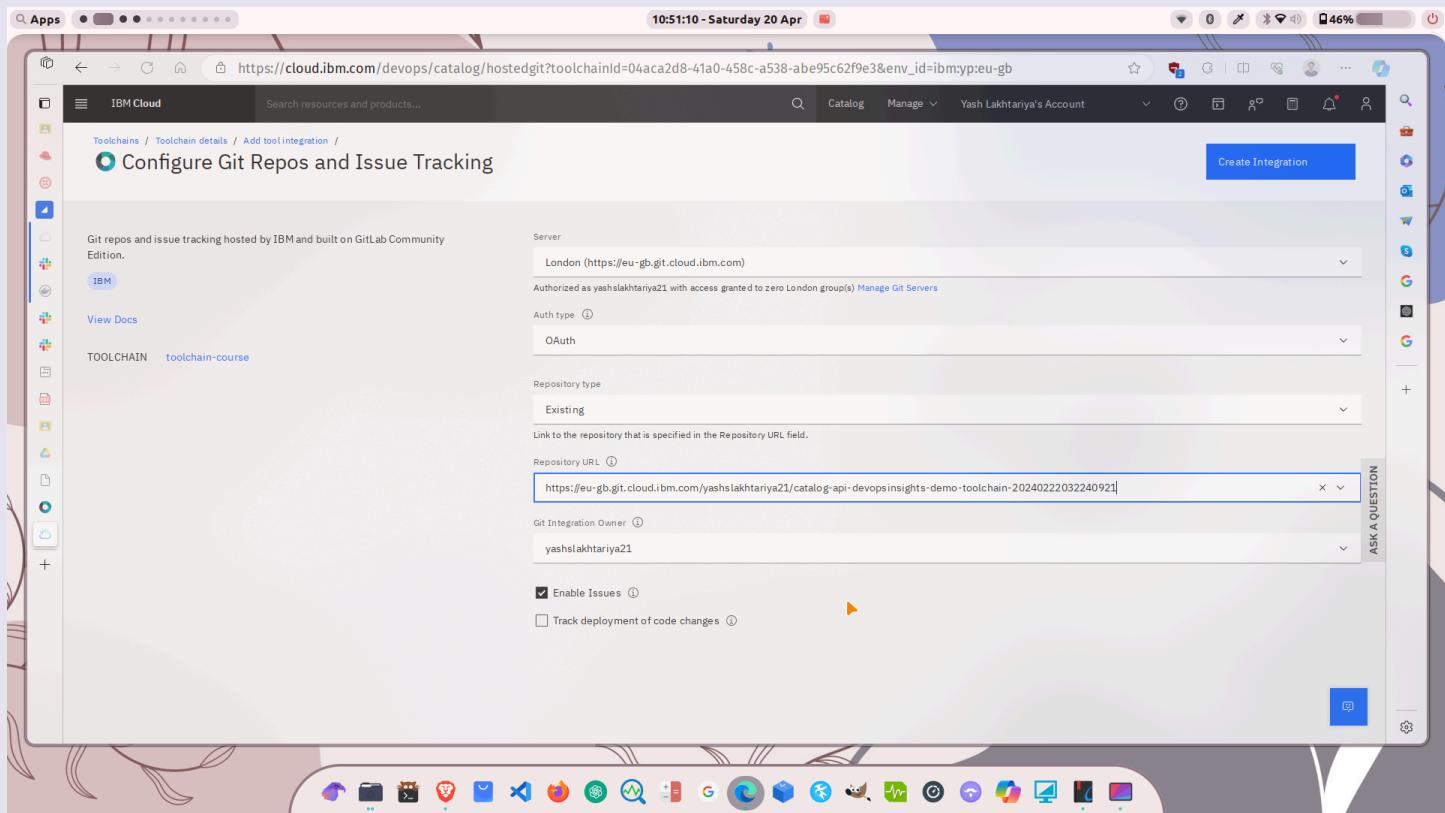
EADC Assignment 2

- f. Visit [app.slack.com](https://app.slack.com) and the App which exists, where the notification is received of new integration



**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Assignment 2**

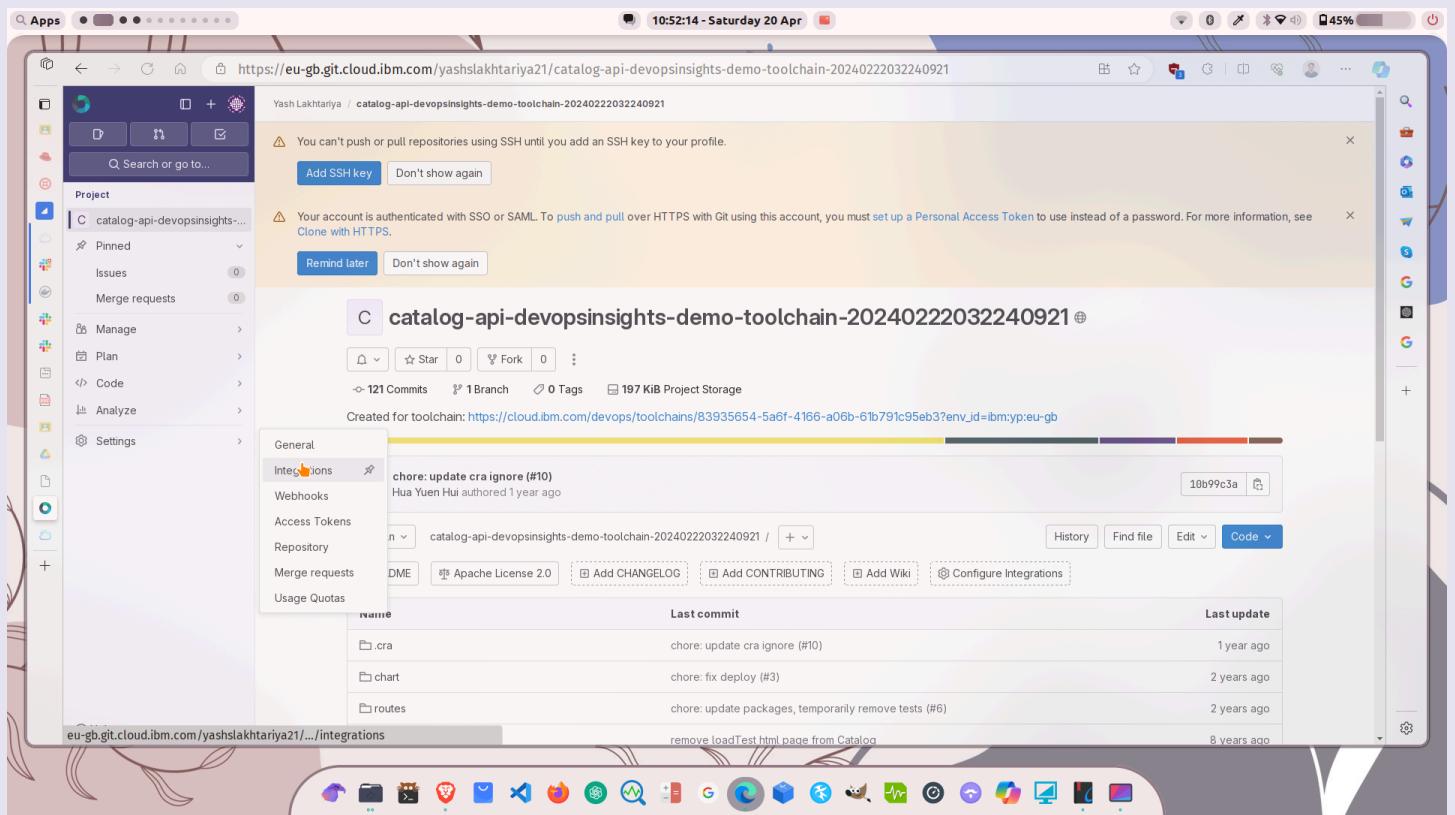
**g. Now, add integration of Git Repos and Issue tracking and paste the credentials of existing repository if there or new one if not**



The screenshot shows a web browser window on a Mac OS X desktop. The URL is [https://cloud.ibm.com/devops/catalog/hostedgit?toolchainId=04aca2d8-41a0-458c-a538-abe95c62f9e3&env\\_id=ibmyp:eu-gb](https://cloud.ibm.com/devops/catalog/hostedgit?toolchainId=04aca2d8-41a0-458c-a538-abe95c62f9e3&env_id=ibmyp:eu-gb). The title bar indicates it's Saturday, April 20 at 10:51:10. The page is titled "Configure Git Repos and Issue Tracking". It shows a configuration form for a Git repository. The "Server" dropdown is set to "London (<https://eu-gb.git.cloud.ibm.com>)". The "Auth type" dropdown is set to "OAuth". The "Repository type" dropdown is set to "Existing". The "Repository URL" input field contains the URL <https://eu-gb.git.cloud.ibm.com/yashslakhtariya21/catalog-api-devopsinsights-demo-toolchain-20240222032240921>. The "Git Integration Owner" dropdown is set to "yashslakhtariya21". There are two checkboxes at the bottom: "Enable Issues" (checked) and "Track deployment of code changes" (unchecked). A blue "Create Integration" button is visible on the right. The left sidebar shows "TOOLCHAIN" and "toolchain-course". The right sidebar has an "ASK A QUESTION" button.

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Assignment 2**

**h. In Git Repos and Issue Tracking, visit Integrations through Settings option**

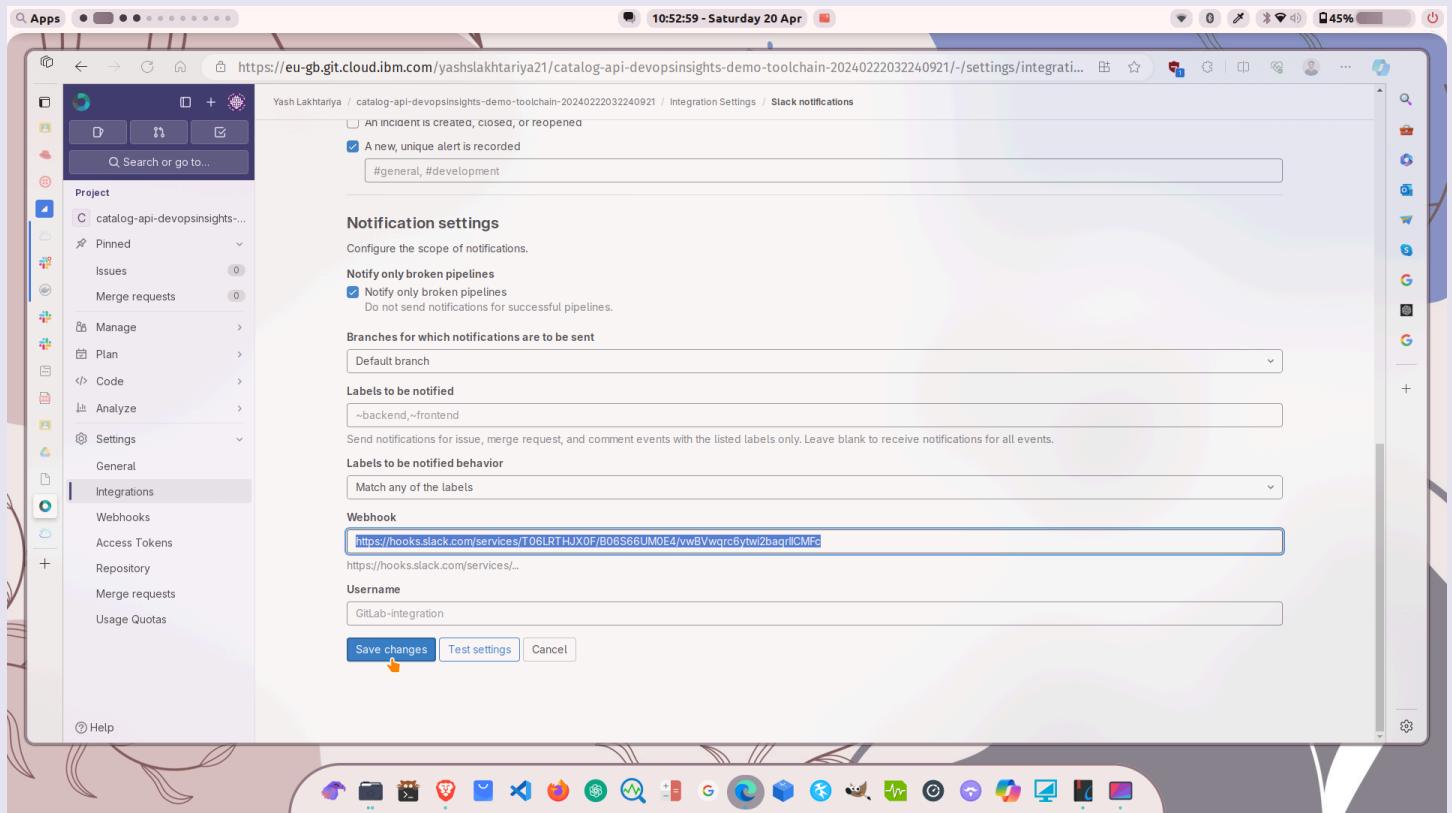


The screenshot shows a web browser window with the URL <https://eu-gb.git.cloud.ibm.com/yashslakhtariya21/catalog-api-devopsinsights-demo-toolchain-20240222032240921>. The page title is "catalog-api-devopsinsights-demo-toolchain-20240222032240921". The left sidebar includes links for Project, Issues, Merge requests, Manage, Plan, Code, Analyze, and Settings. The main content area displays project statistics: 121 Commits, 1 Branch, 0 Tags, and 197 KiB Project Storage. Below this, a "General" tab is selected in the settings menu, and the "Integrations" tab is active. A message at the top states: "You can't push or pull repositories using SSH until you add an SSH key to your profile." and "Your account is authenticated with SSO or SAML. To push and pull over HTTPS with Git using this account, you must set up a Personal Access Token to use instead of a password. For more information, see Clone with HTTPS." A "Remind later" button is present. The integrations table lists three entries:

Name	Last commit	Last update
:cra	chore: update cra ignore (#10)	1 year ago
chart	chore: fix deploy (#3)	2 years ago
routes	chore: update packages, temporarily remove tests (#6) remove loadTest html page from Catalog	2 years ago

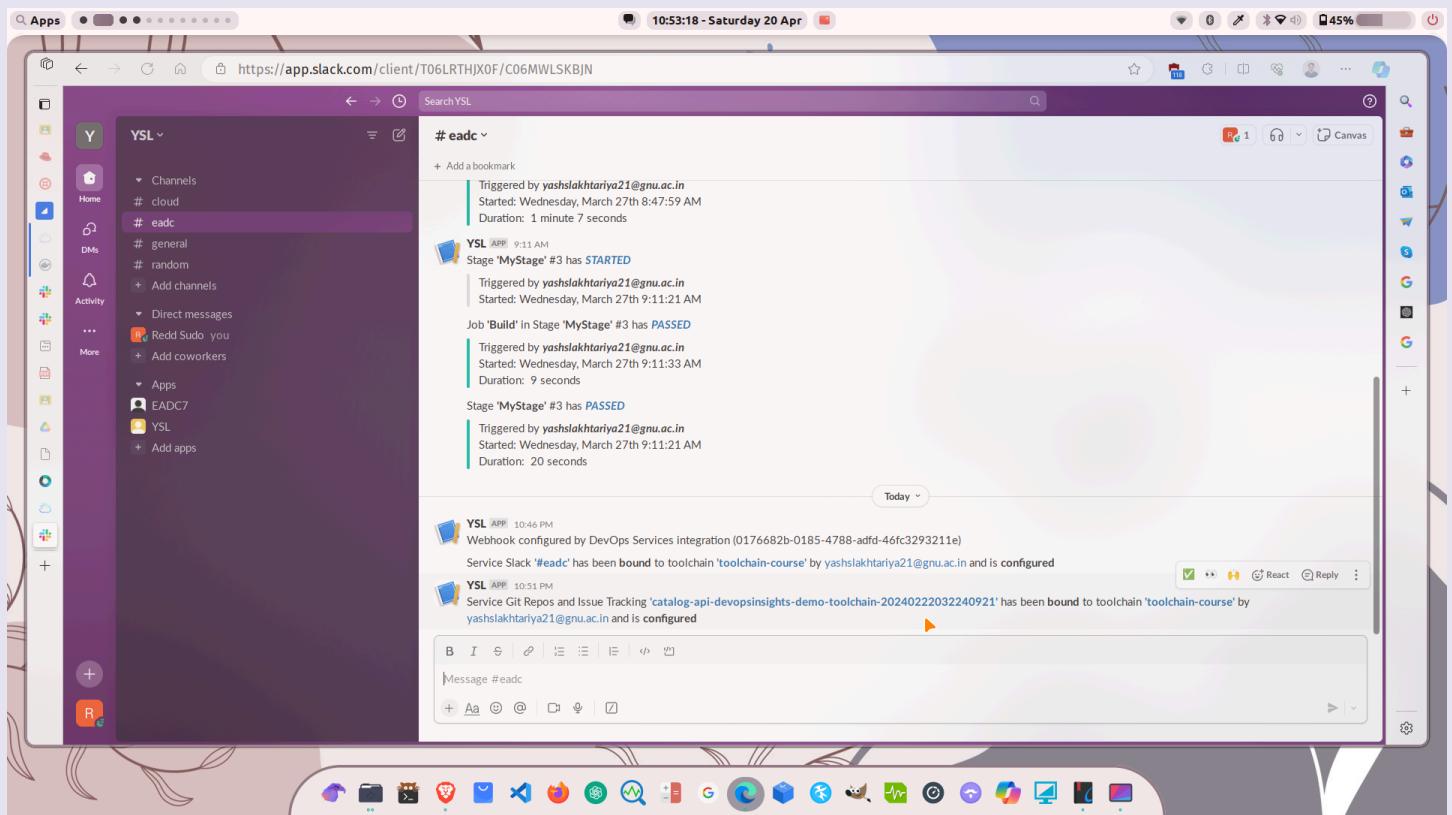
**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Assignment 2**

- i. Add Slack notification integration and paste the webhook URL there



**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Assignment 2**

j. The notification will be thus received on Slack Application



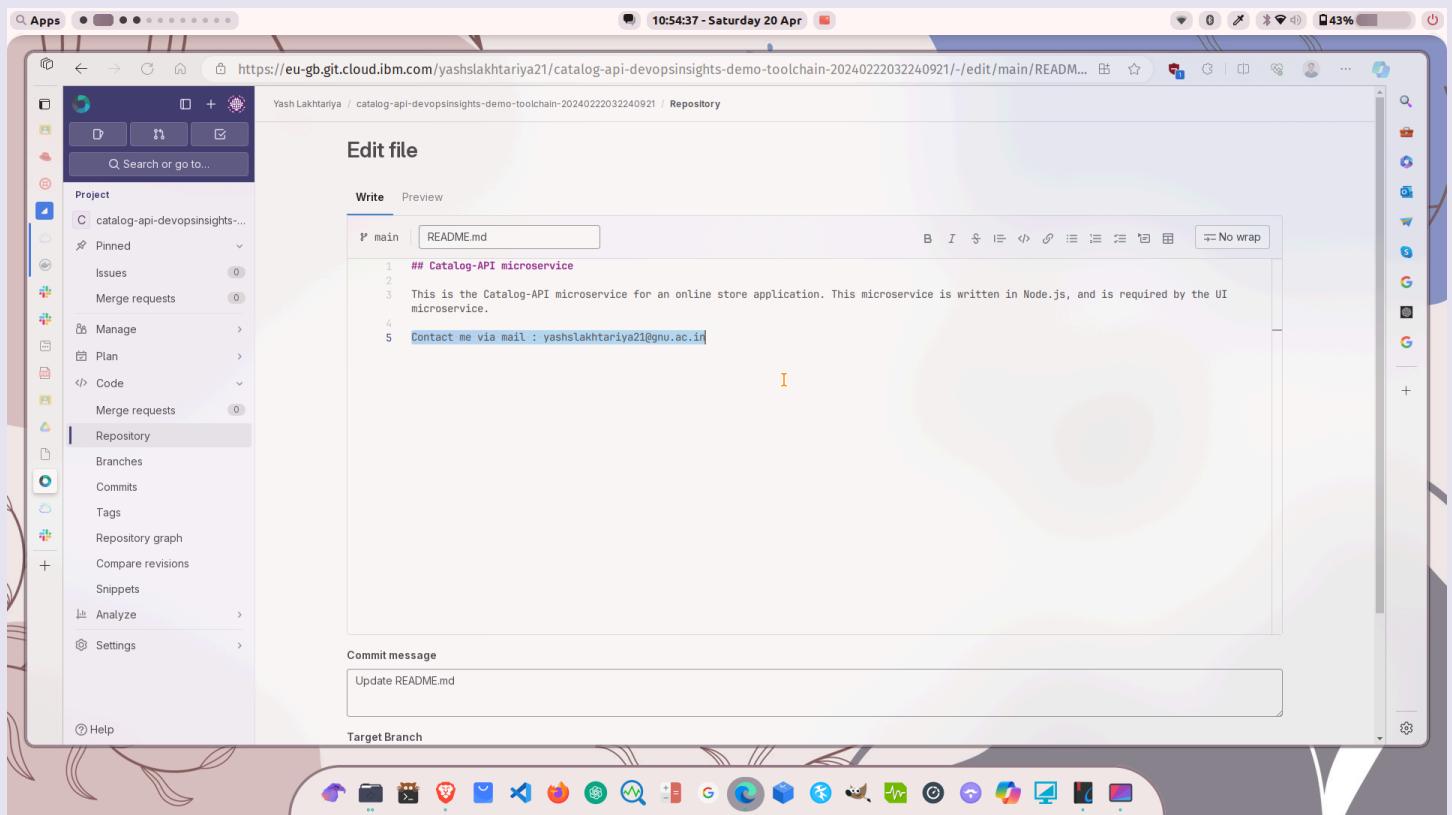
Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Assignment 2

k. Modify any code, say Readme here and commit changes

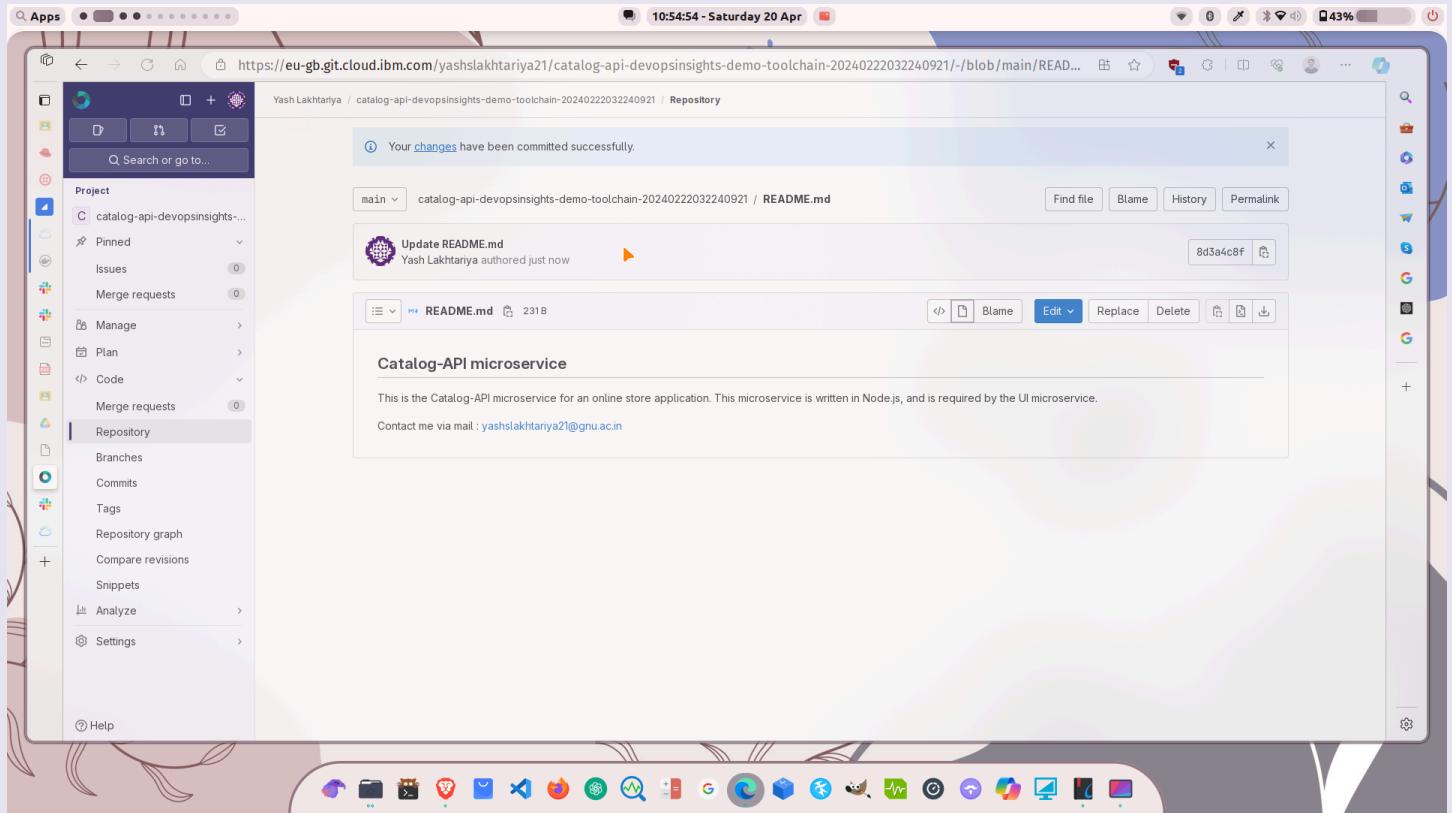


Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Assignment 2



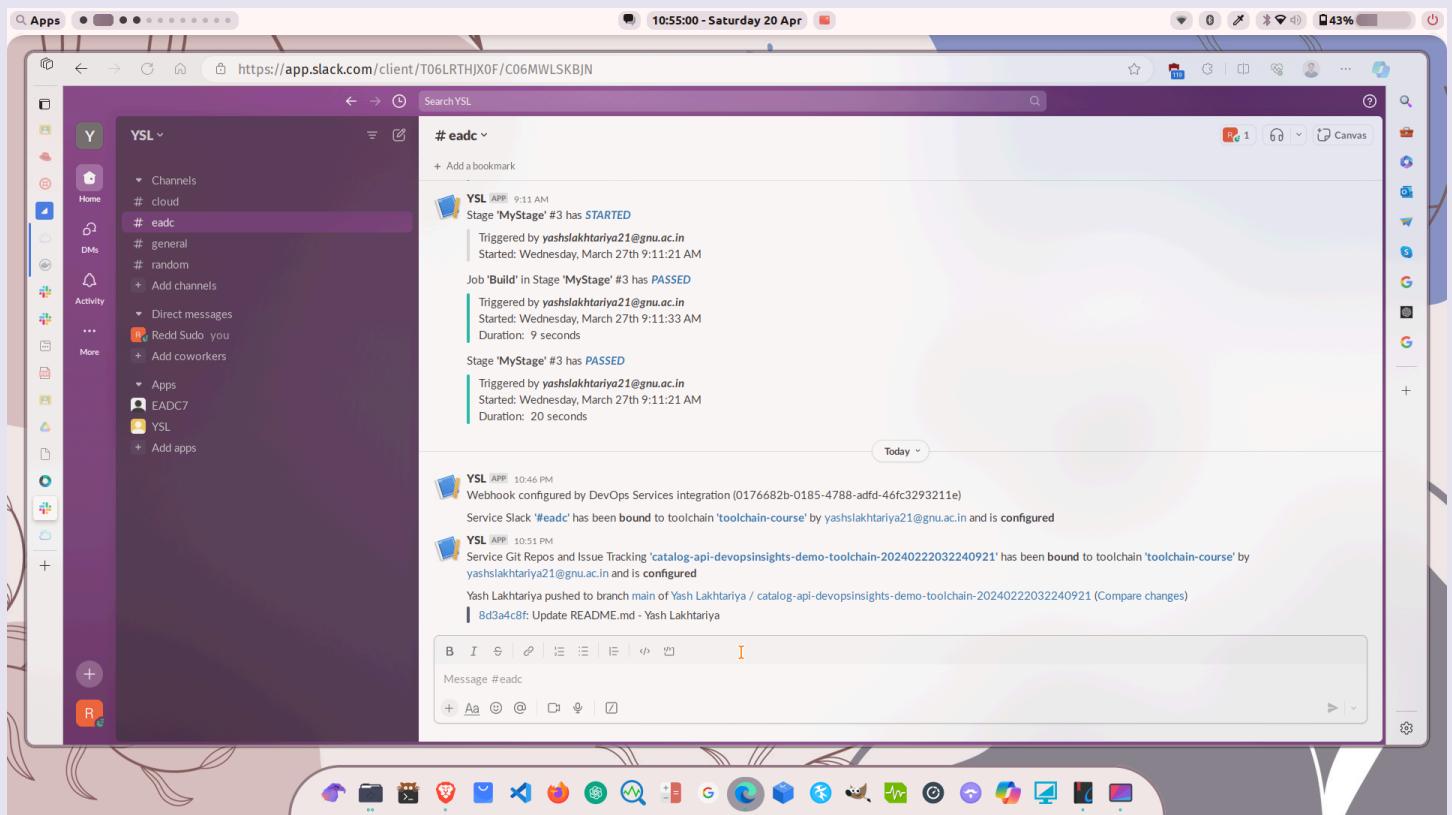
Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Assignment 2

- l. The change notification will be received in Slack app

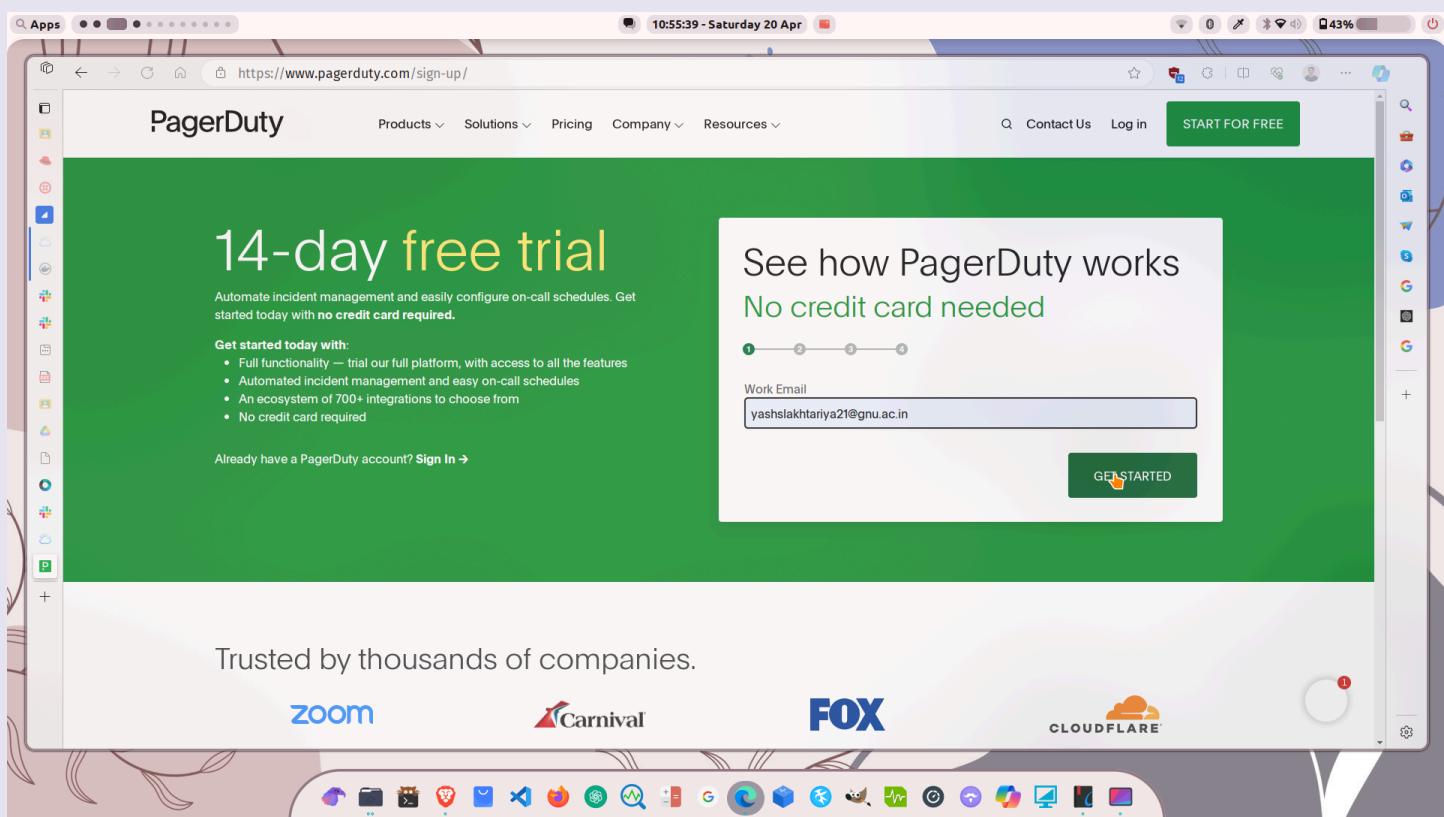


Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Assignment 2

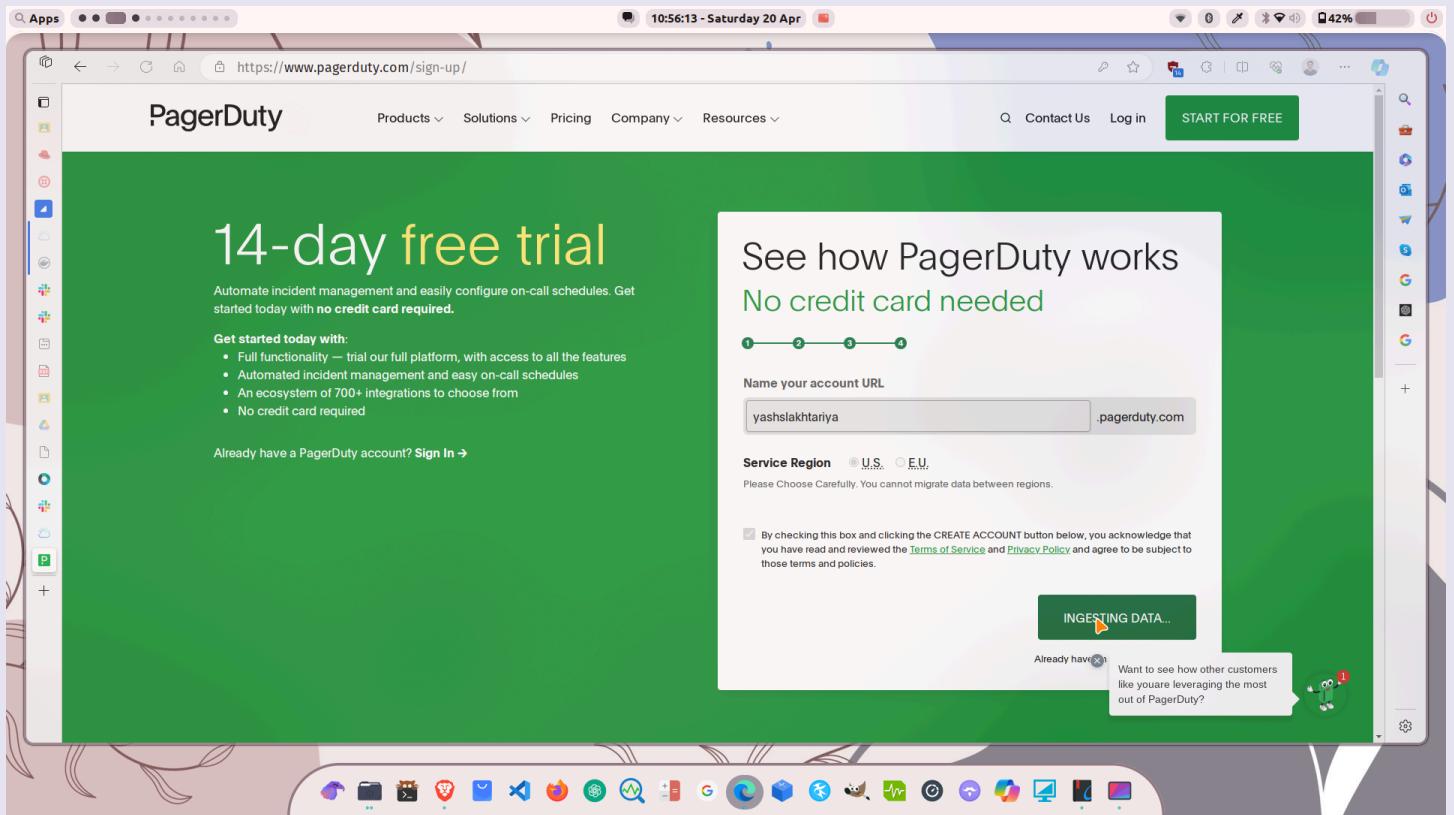
## 2. Set Up Toolchain for PagerDuty Notifications :

- Create a toolchain on IBM Cloud.
- Integrate PagerDuty notification tool into the toolchain.
- Configure the tool to send notifications for specific events.

- a. Create account on PagerDuty if not exists

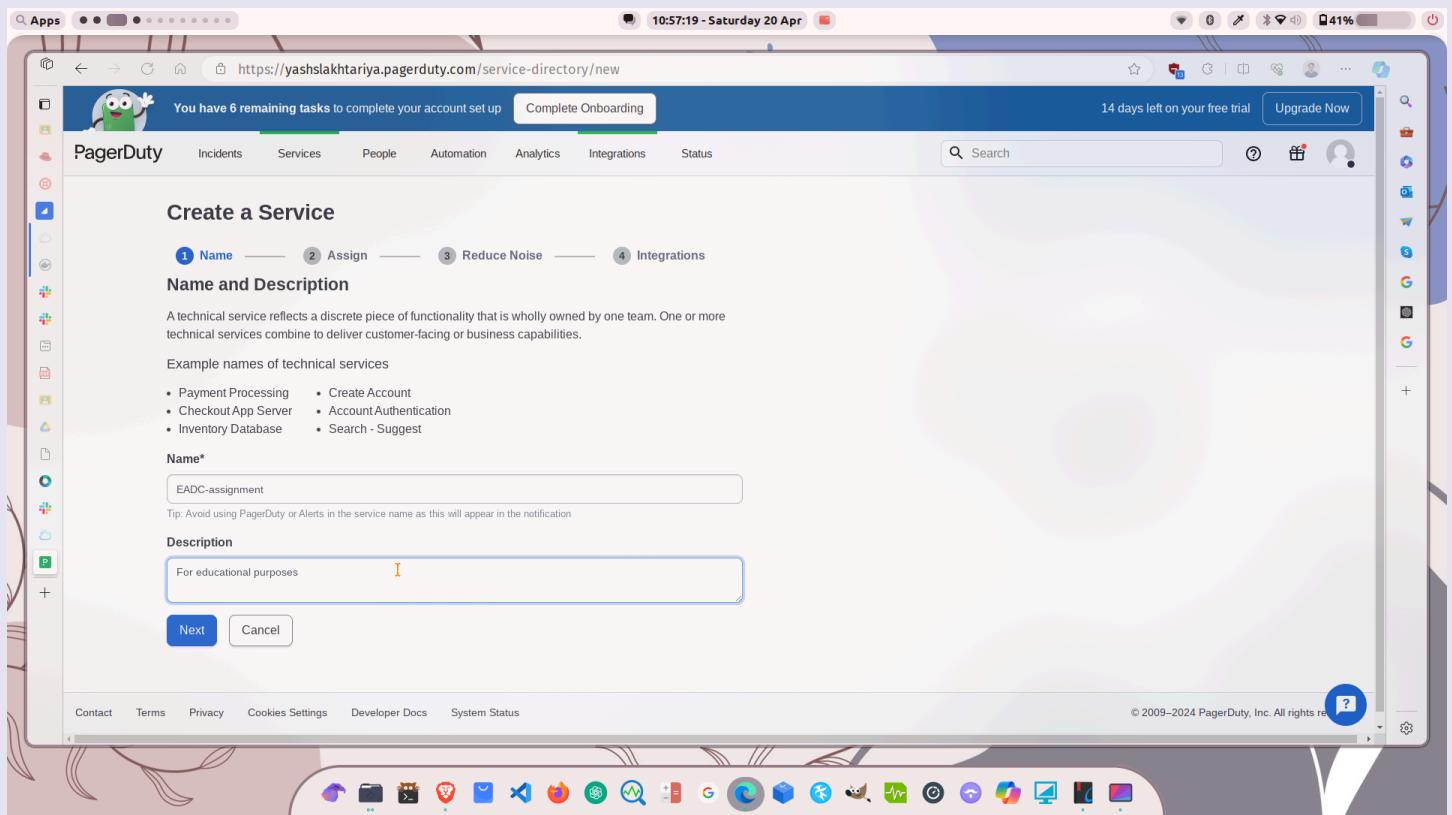


Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Assignment 2



**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Assignment 2**

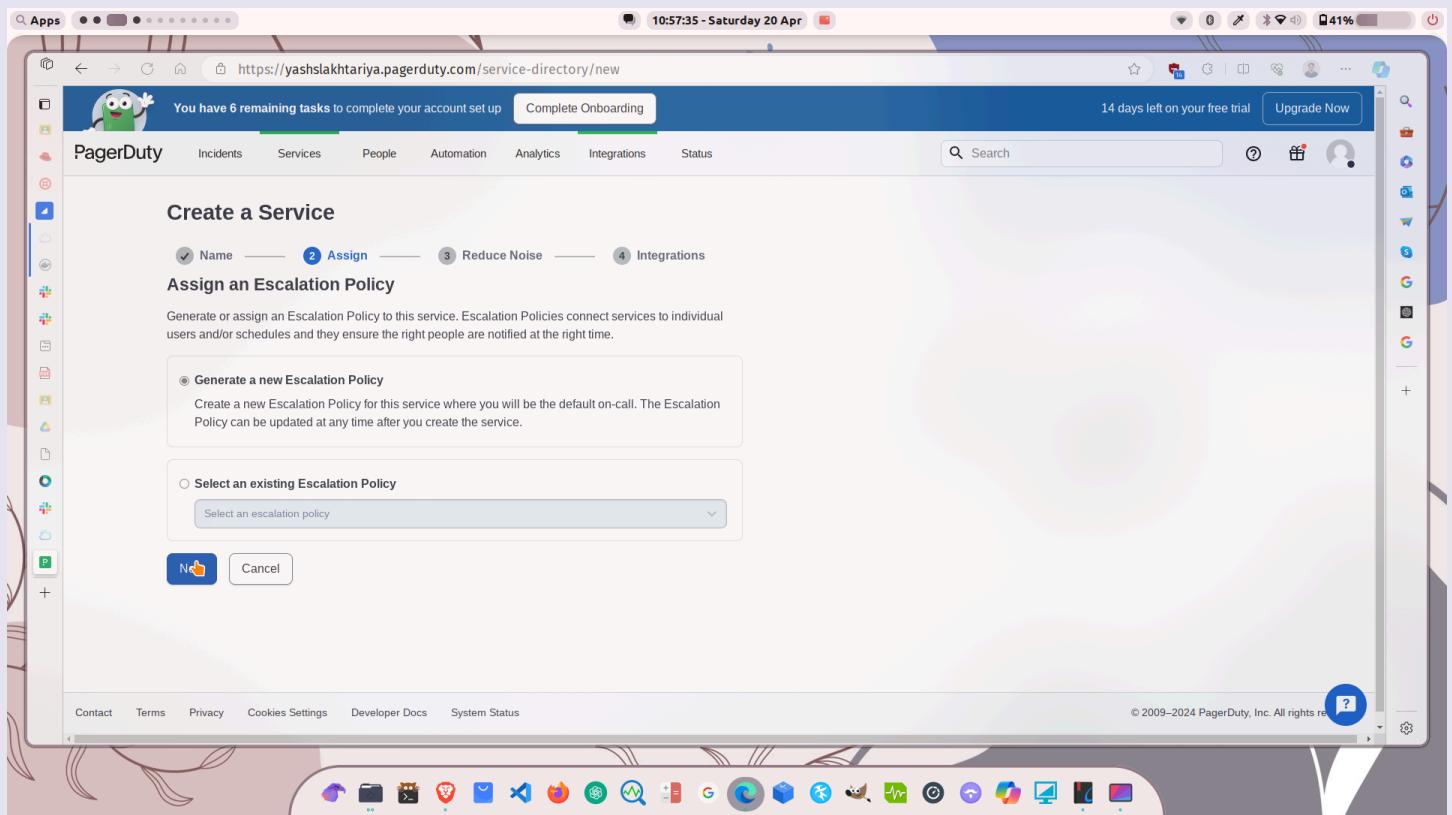
**b. Create a new service there via button at right side**



The screenshot shows a web browser window for the PagerDuty Service Directory. The URL is <https://yashslakhtariya.pagerduty.com/service-directory/new>. The top navigation bar includes links for Incidents, Services, People, Automation, Analytics, Integrations, and Status. A search bar and a sidebar with various icons are also visible. The main content area is titled "Create a Service" and is divided into four steps: 1. Name, 2. Assign, 3. Reduce Noise, and 4. Integrations. The "Name" step is active, showing a form with a "Name\*" field containing "EADC-assignment" and a "Description" field containing "For educational purposes". Below the form are "Next" and "Cancel" buttons. The status bar at the bottom of the browser shows various application icons.

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Assignment 2**

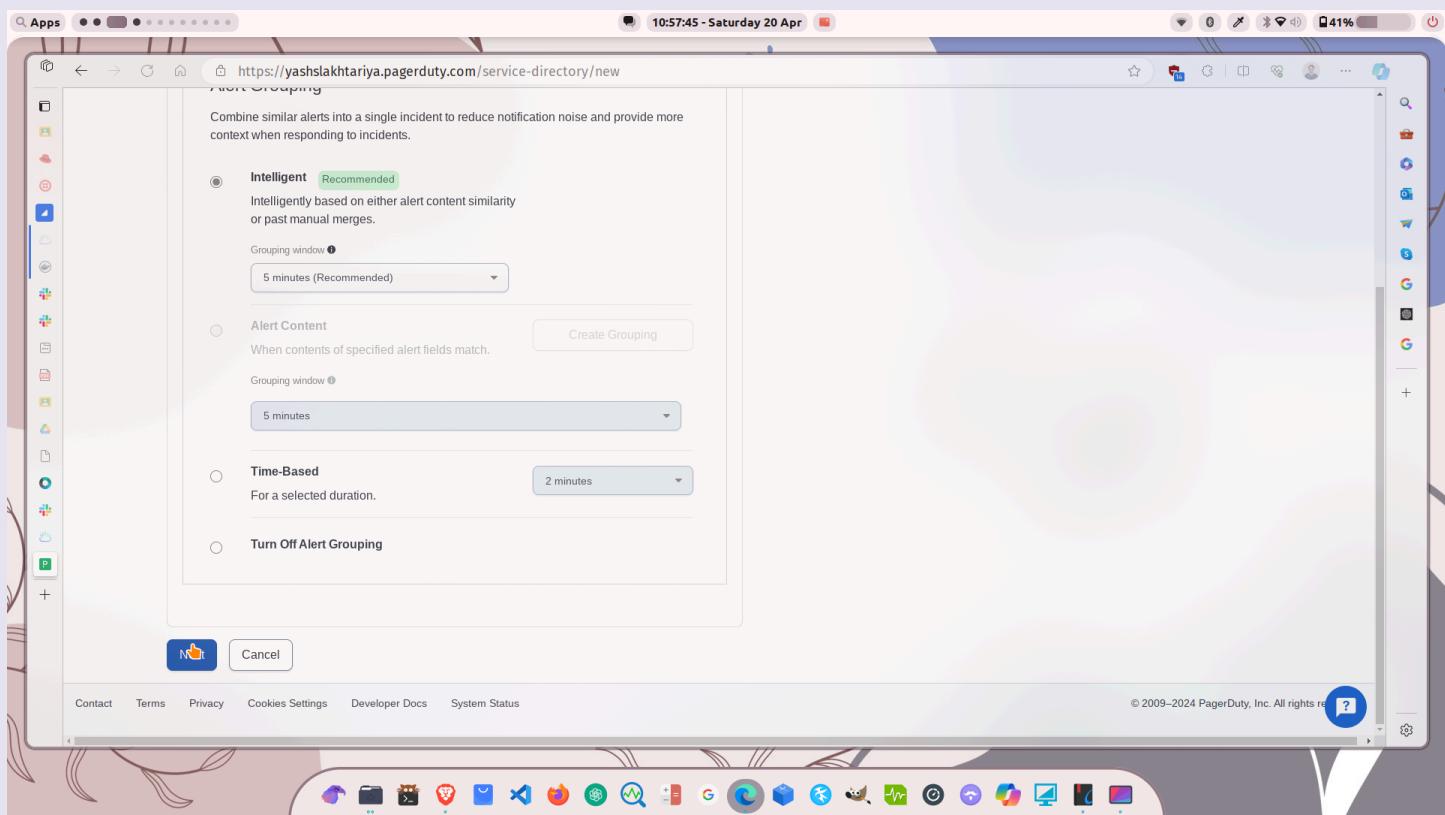
**c. Keep Policy options defaults to create new policy**



The screenshot shows a web browser window for the PagerDuty service directory creation page at <https://yashlakhtariya.pagerduty.com/service-directory/new>. The top navigation bar includes links for Incidents, Services, People, Automation, Analytics, Integrations, and Status. A banner at the top indicates "You have 6 remaining tasks to complete your account set up" and "Complete Onboarding". A message "14 days left on your free trial" and a "Upgrade Now" button are also present. The main content area is titled "Create a Service" and displays a progress bar with four steps: 1. Name (completed), 2. Assign (selected), 3. Reduce Noise, and 4. Integrations. Below the progress bar, a section titled "Assign an Escalation Policy" asks "Generate or assign an Escalation Policy to this service. Escalation Policies connect services to individual users and/or schedules and they ensure the right people are notified at the right time." It offers two options: "Generate a new Escalation Policy" (radio button selected) and "Select an existing Escalation Policy" (radio button unselected). A dropdown menu labeled "Select an escalation policy" is shown. At the bottom of the form are "Next" and "Cancel" buttons. The footer of the page includes links for Contact, Terms, Privacy, Cookies Settings, Developer Docs, and System Status, along with a copyright notice: "© 2009–2024 PagerDuty, Inc. All rights reserved". The browser's toolbar and status bar are visible at the top and bottom of the window.

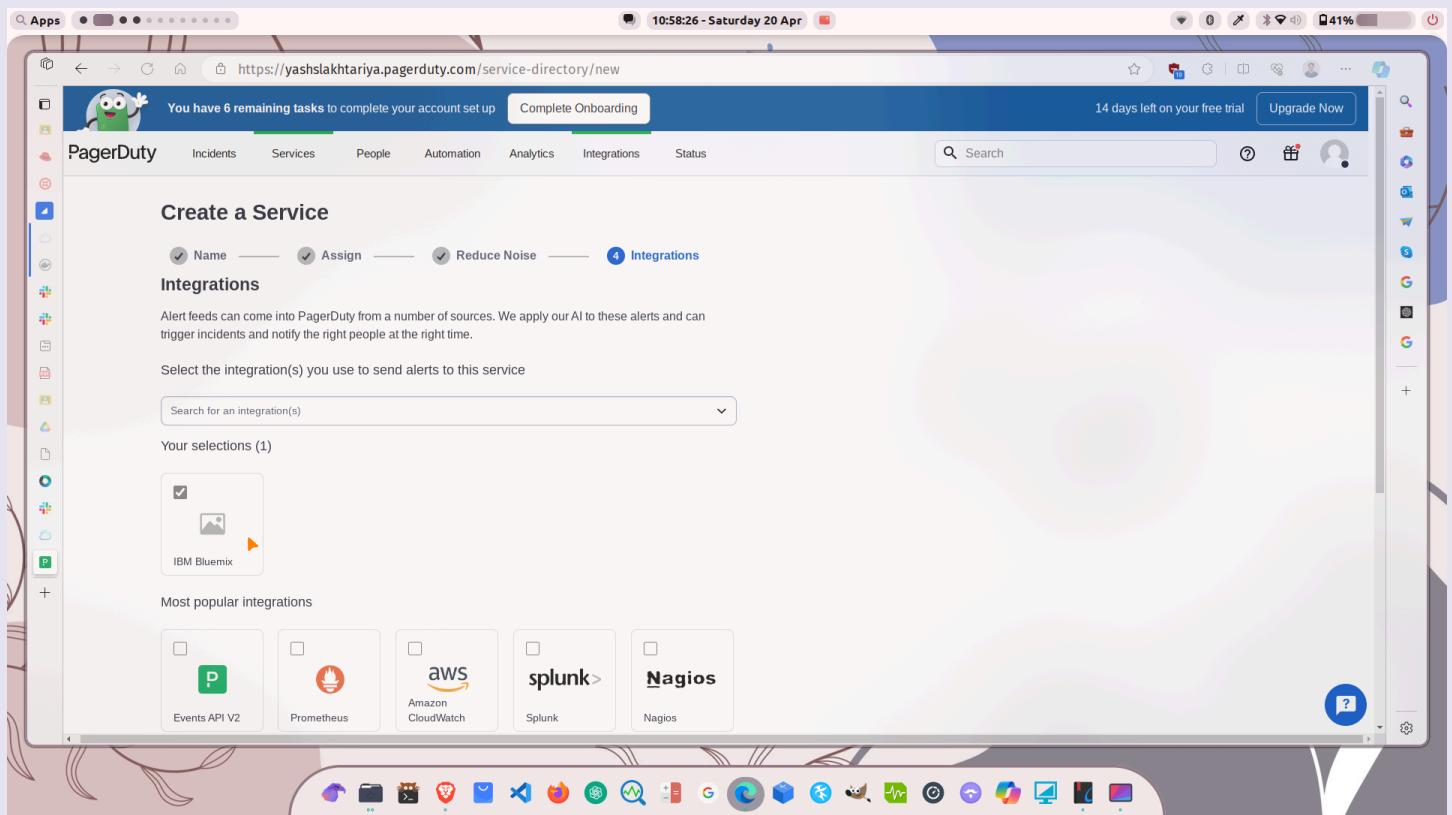
Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Assignment 2

d. Similarly, default options for reduce noise tab



**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Assignment 2**

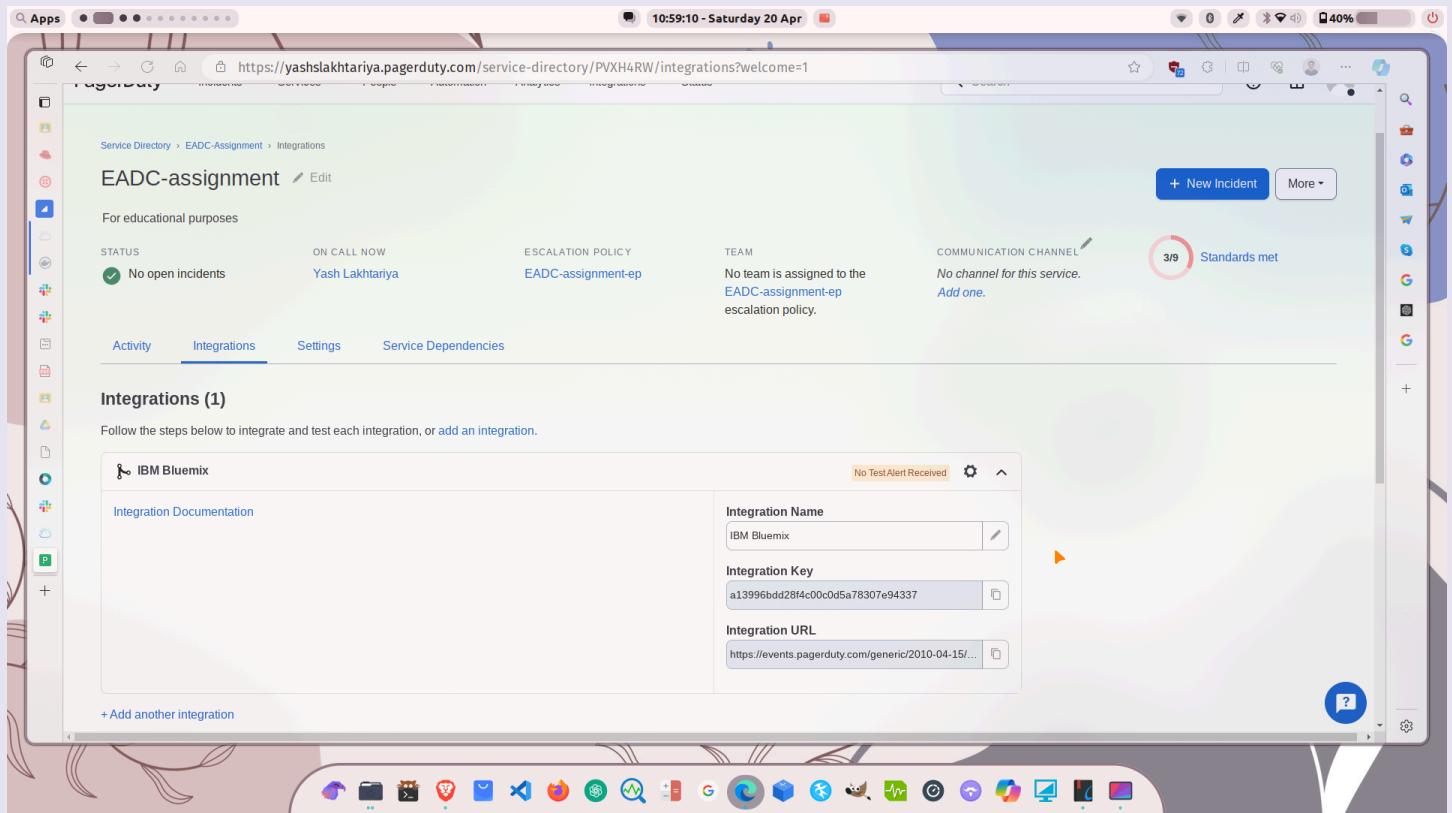
**e. In integrations, add IBM Bluemix**



The screenshot shows a web browser window for the PagerDuty service directory. The URL is <https://yashlakhtariya.pagerduty.com/service-directory/new>. The top navigation bar includes links for Incidents, Services, People, Automation, Analytics, Integrations, and Status. A search bar and a sidebar with various icons are also visible. The main content area is titled "Create a Service" and has tabs for Name, Assign, Reduce Noise, and Integrations. The "Integrations" tab is selected, indicated by a blue border and the number 4. Below this, a section titled "Integrations" explains that alert feeds can come from various sources and trigger incidents. A search bar allows selecting integration(s). Under "Your selections (1)", the "IBM Bluemix" integration is listed with a checked checkbox. A "Most popular integrations" section shows icons for Events API V2, Prometheus, Amazon CloudWatch, Splunk, and Nagios.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
**EADC Assignment 2**

f. Here, the key and URL are there as credentials



The screenshot shows a web browser window for the PagerDuty Service Directory. The URL is <https://yashslakhtariya.pagerduty.com/service-directory/PVXH4RW/integrations?welcome=1>. The page displays a service named "EADC-assignment". Key details shown include:

- STATUS:** No open incidents.
- ON CALL NOW:** Yash Lakhtariya.
- ESCALATION POLICY:** EADC-assignment-ep.
- TEAM:** No team is assigned to the EADC-assignment-ep escalation policy.
- COMMUNICATION CHANNEL:** No channel for this service. Add one.
- Standards met:** 3/9 (Standards met).

The "Integrations" tab is selected, showing one integration entry for "IBM Bluemix". The integration details are:

- Integration Name:** IBM Bluemix.
- Integration Key:** a13996bdd28f4c00c0d5a78307e94337.
- Integration URL:** <https://events.pagerduty.com/generic/2010-04-15/>.

At the bottom of the integration card, there is a link to "+ Add another integration".

Name - Yash Lakhtariya

Enrollment number - 21162101012

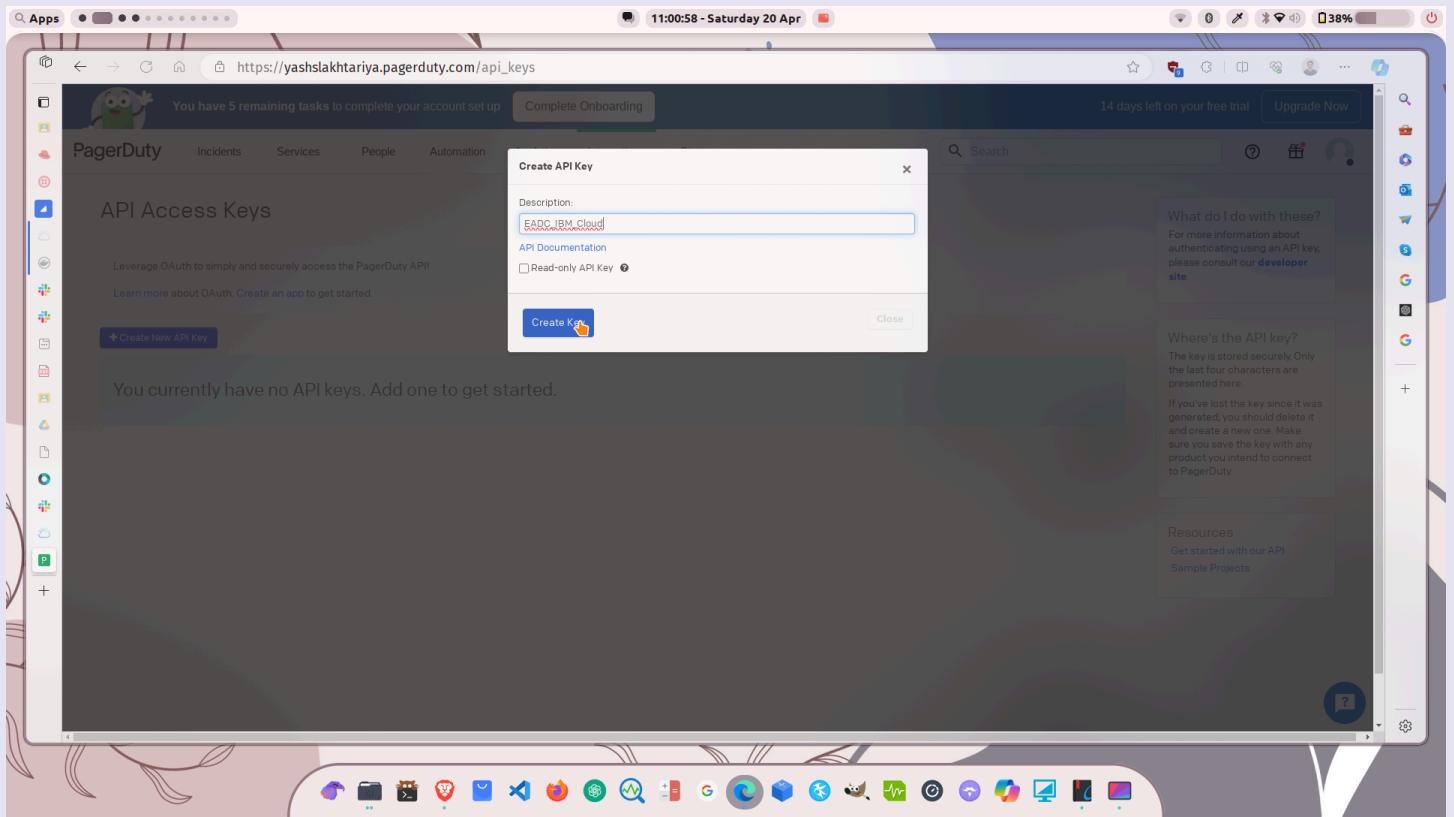
Branch - CBA      Batch - 61

EADC Assignment 2

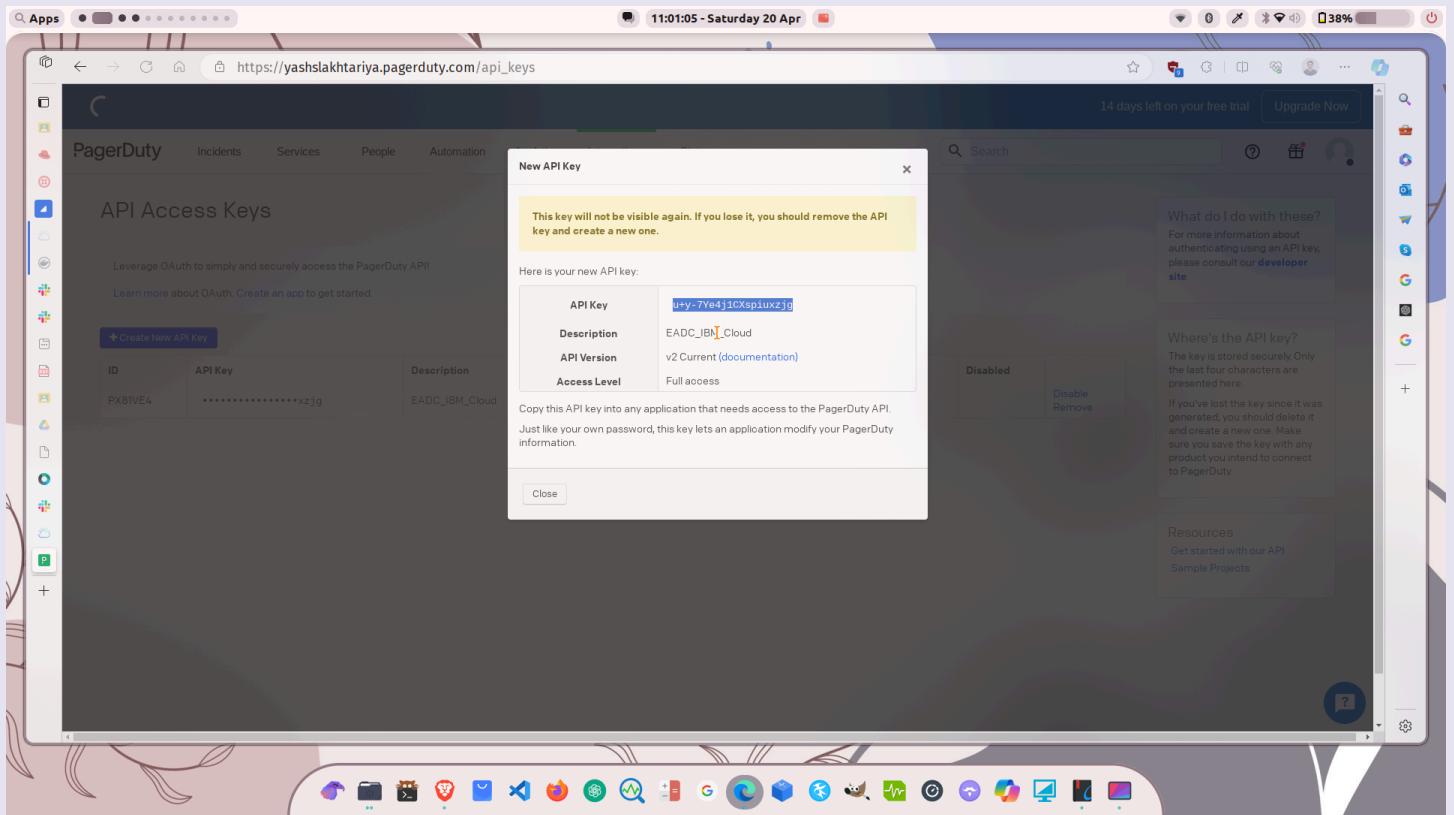
g. Visit API Keys in integrations tab, and create new API Key

The screenshot shows a web browser window for the PagerDuty API Access Keys page. The URL in the address bar is [https://yashslakhtariya.pagerduty.com/api\\_keys](https://yashslakhtariya.pagerduty.com/api_keys). The page header includes a navigation bar with links for Incidents, Services, People, Automation, Analytics, Integrations, and Status. A banner at the top indicates "You have 5 remaining tasks to complete your account set up" and "Complete Onboarding". A message box says "14 days left on your free trial" and "Upgrade Now". The main content area is titled "API Access Keys" and contains a sub-instruction: "Leverage OAuth to simply and securely access the PagerDuty API!". It also provides a link to "Learn more about OAuth: Create an app to get started.". A prominent blue button labeled "+Create New API Key" is centered on the page. To the right, there are two informational boxes: "What do I do with these?" and "Where's the API key?". Below these are sections for "Resources" and "Sample Projects". The bottom of the browser window shows the Mac OS Dock with various application icons.

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Assignment 2**



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Assignment 2



Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA      Batch - 61

EADC Assignment 2

h. Add toolchain integration of PagerDuty and add credentials there

The screenshot shows the IBM Cloud interface for configuring a PagerDuty integration. The URL in the address bar is [https://cloud.ibm.com/devops/catalog/pagerduty?toolchainId=04aca2d8-41a0-458c-a538-abe95c62f9e3&env\\_id=ibm:yp:eu-gb](https://cloud.ibm.com/devops/catalog/pagerduty?toolchainId=04aca2d8-41a0-458c-a538-abe95c62f9e3&env_id=ibm:yp:eu-gb). The page title is "Configure PagerDuty". The "Create Integration" button is highlighted with a cursor. The integration type is set to "Account". The API access key field contains a redacted value. The PagerDuty service name is "EADC-assignment". The primary contact email is "yashslakhtariya21@gnu.ac.in" and the primary contact phone number is "06351573711". The left sidebar shows "Toolchains" and "TOOLCHAIN toolchain-course". The bottom dock contains various application icons.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Assignment 2

i. As seen, the service integration is visible

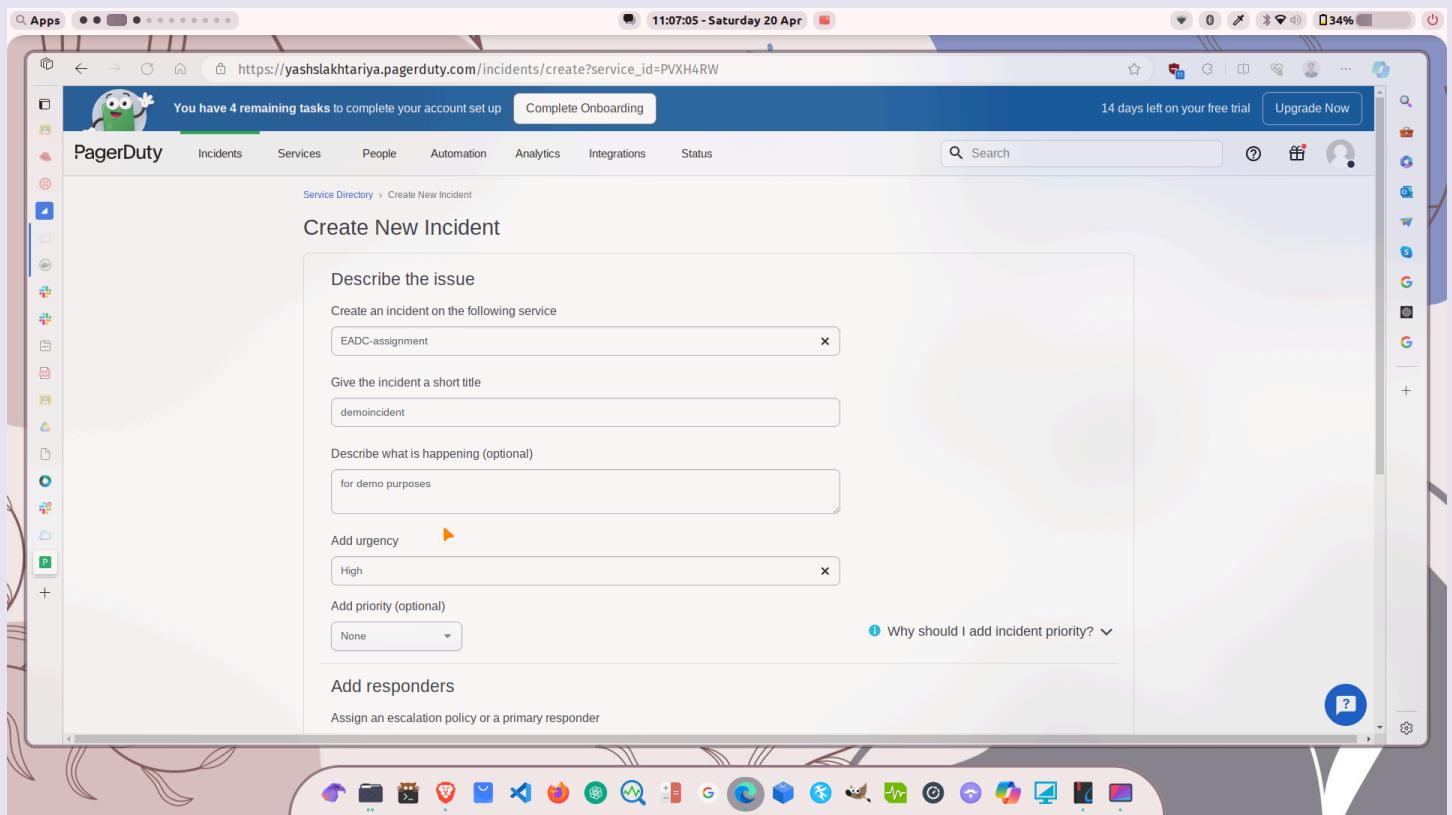
The screenshot shows the PagerDuty Service Directory interface. At the top, there are navigation links for Incidents, Services, People, Automation, Analytics, Integrations, and Status. A search bar and a 'New Service' button are also present. The main area is titled 'Service Directory' and contains a brief description of what a service represents. Below this, there are two tabs: 'Services' (selected) and 'Maintenance Windows'. A search bar and filter options for Team, Business Service, Last Incident, and Sort By are available. The 'Total services' section shows two entries:

Service	Team	On Call Now	Last Incident	Open Incidents	Standards met
Default Service	No team is assigned to the <a href="#">Default escalation policy</a> .	Yash Lakhtariya	Today at 5:34 PM	0 triggered 1 acknowledged	1/9 Standards met
EADC-assignment	No team is assigned to the <a href="#">EADC-assignment-ep escalation policy</a> .	Yash Lakhtariya	No incidents so far.	0 triggered 0 acknowledged	3/9 Standards met

At the bottom, there are links for Contact, Terms, Privacy, Cookies Settings, Developer Docs, and System Status. The URL in the address bar is https://yashlakhtariya.pagerduty.com/service-directory/PVXH4RW.

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Assignment 2**

j. Create new incident to check the availability of service



The screenshot shows a web browser window for PagerDuty at the URL [https://yashslakhtariya.pagerduty.com/incidents/create?service\\_id=PVXH4RW](https://yashslakhtariya.pagerduty.com/incidents/create?service_id=PVXH4RW). The browser's status bar indicates the time is 11:07:05 on Saturday, April 20. The main page header includes a 'Complete Onboarding' button and a '14 days left on your free trial' message. The 'Incidents' tab is selected in the navigation bar. The central 'Create New Incident' form has the following fields filled:

- Create an incident on the following service:** EADC-assignment
- Give the incident a short title:** demoincident
- Describe what is happening (optional):** for demo purposes
- Add urgency:** High
- Add priority (optional):** None
- Why should I add incident priority?** (A question mark icon with a tooltip)
- Add responders:** (Empty field)
- Assign an escalation policy or a primary responder:** (Empty field)

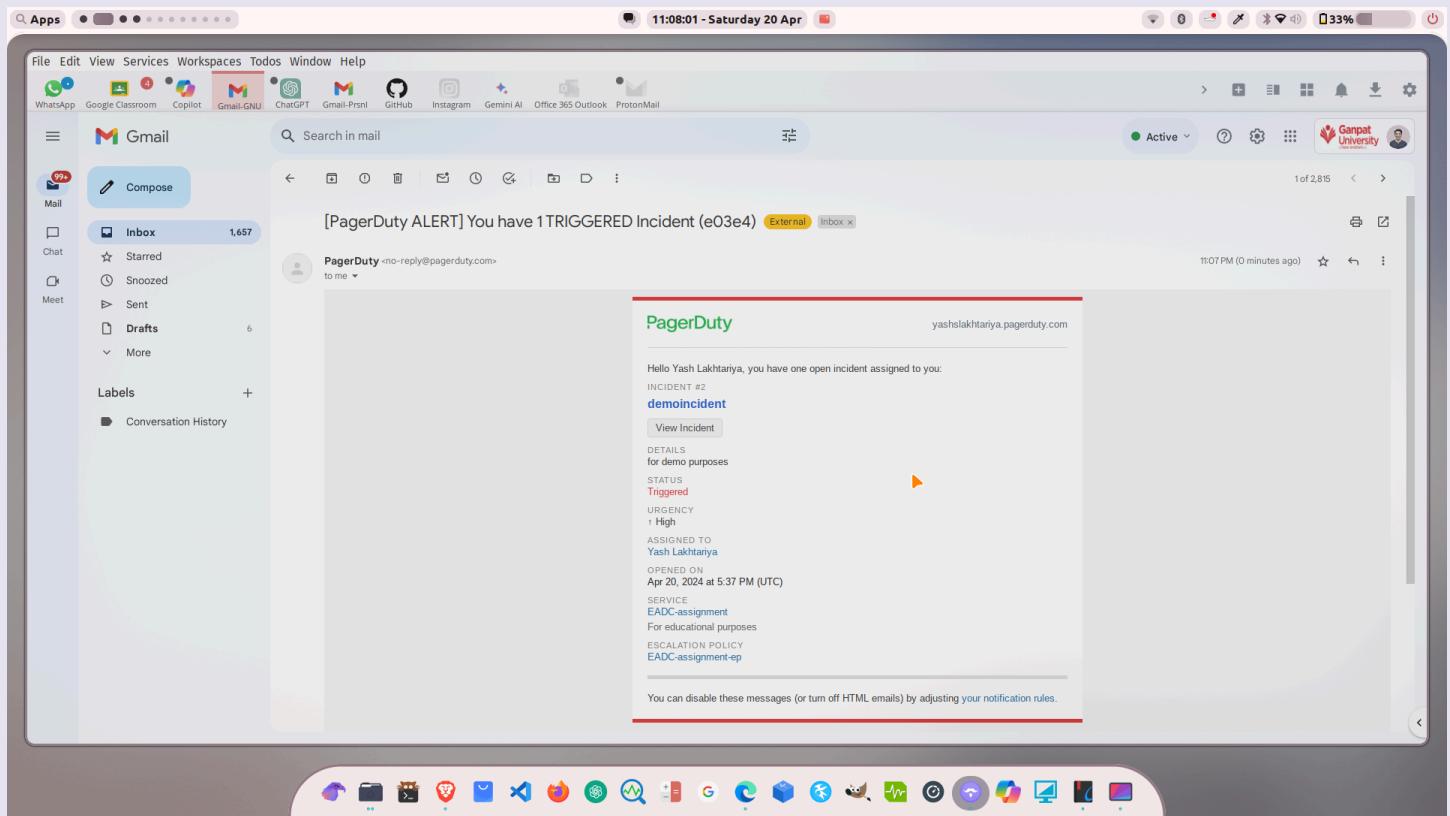
Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Assignment 2

k. First, it is in triggered stage

The screenshot shows the PagerDuty incident details page for an incident titled "demoincident". The status is "Triggered" and the duration is "00h 00m". The incident was opened at 5:37 PM (a few seconds ago). It is assigned to "Yash Lakhtariya" and impacted by the service "EADC-assignment". The service description is "For educational purposes". There are no responders yet. The custom fields section indicates no custom fields are configured. The notes section is empty. The timeline tab is selected.

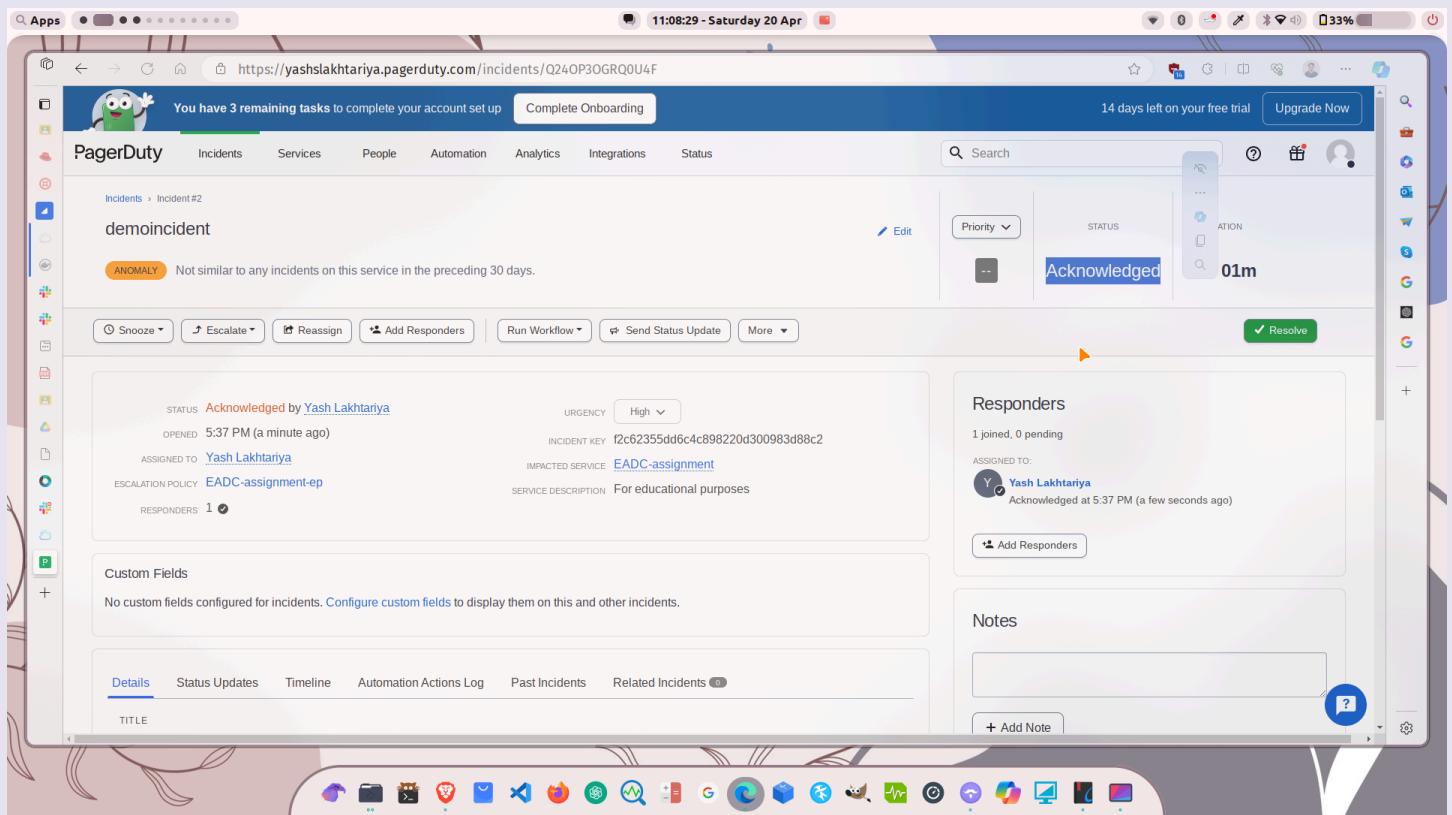
**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Assignment 2**

- l. The email is received and also a phone call if contact number is added, from which the acknowledgement can be done**



**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Assignment 2**

m. As seen here, from SMS or phone call, it can be acknowledged



The screenshot shows the PagerDuty web interface for an incident. The URL in the address bar is <https://yashslakhtariya.pagerduty.com/incidents/Q24OP30GRQ0U4F>. The top navigation bar includes links for Apps, Incidents, Services, People, Automation, Analytics, Integrations, and Status. A banner at the top says "You have 3 remaining tasks to complete your account set up" and "Complete Onboarding". It also indicates "14 days left on your free trial" and a "Upgrade Now" button.

The main content area displays an incident titled "demoincident". The status is "Acknowledged" with a timestamp of "01m". The incident details include:

- STATUS:** Acknowledged by [Yash Lakhtariya](#)
- OPENED:** 5:37 PM (a minute ago)
- URGENCY:** High
- INCIDENT KEY:** f2c62355dd6c4c898220d300983d88c2
- ASSIGNED TO:** [Yash Lakhtariya](#)
- IMPACTED SERVICE:** [EADC-assignment](#)
- ESCALATION POLICY:** [EADC-assignment-ep](#)
- SERVICE DESCRIPTION:** For educational purposes
- RESPONDERS:** 1

**Custom Fields:** No custom fields configured for incidents. [Configure custom fields](#) to display them on this and other incidents.

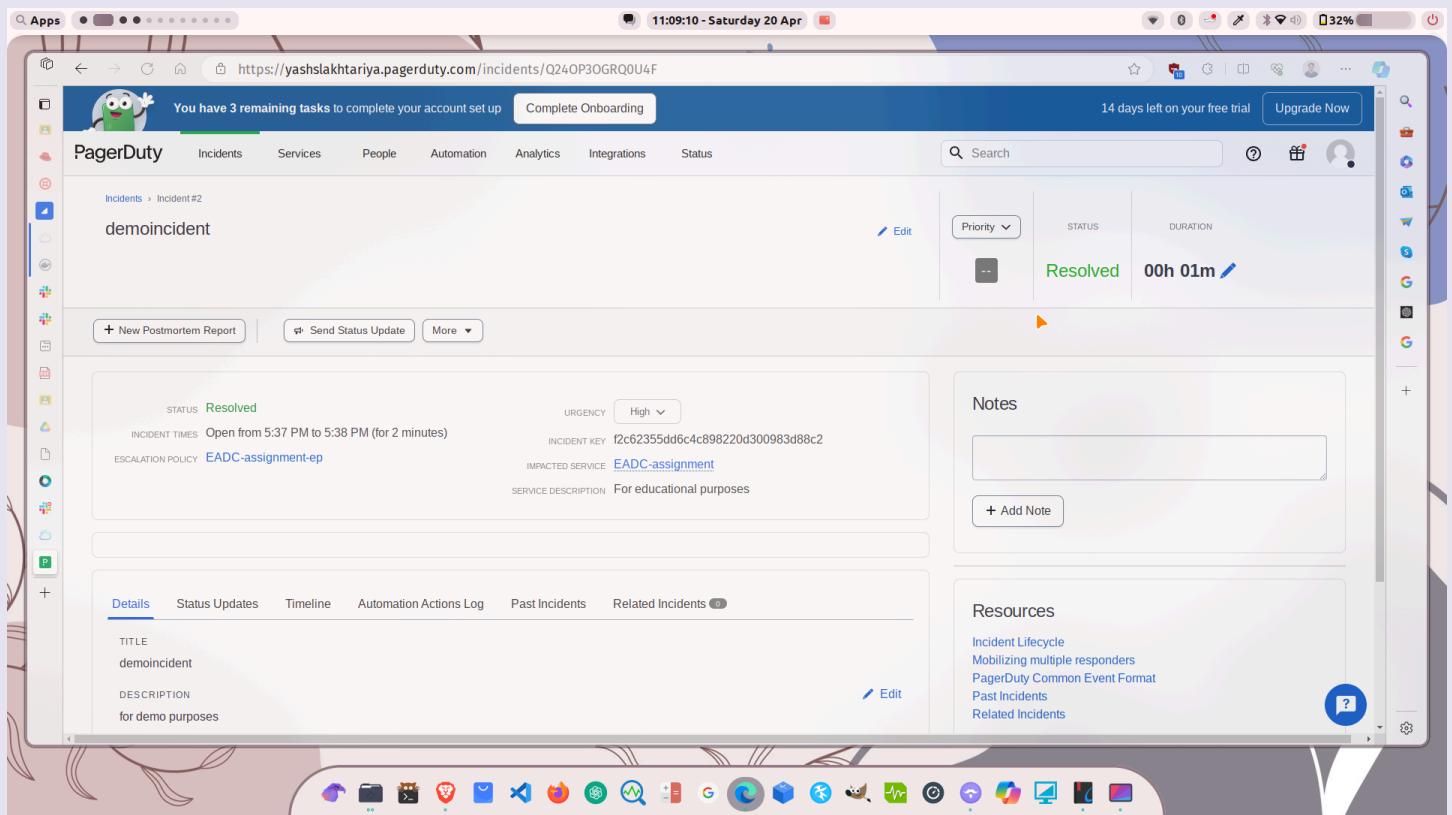
**Responder Details:** 1 joined, 0 pending. The responder is [Yash Lakhtariya](#), acknowledged at 5:37 PM (a few seconds ago). There is a button to "+ Add Responders".

**Notes:** There is a large text input field for notes and a "+ Add Note" button.

At the bottom of the screen, there is a dock with various application icons.

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Assignment 2**

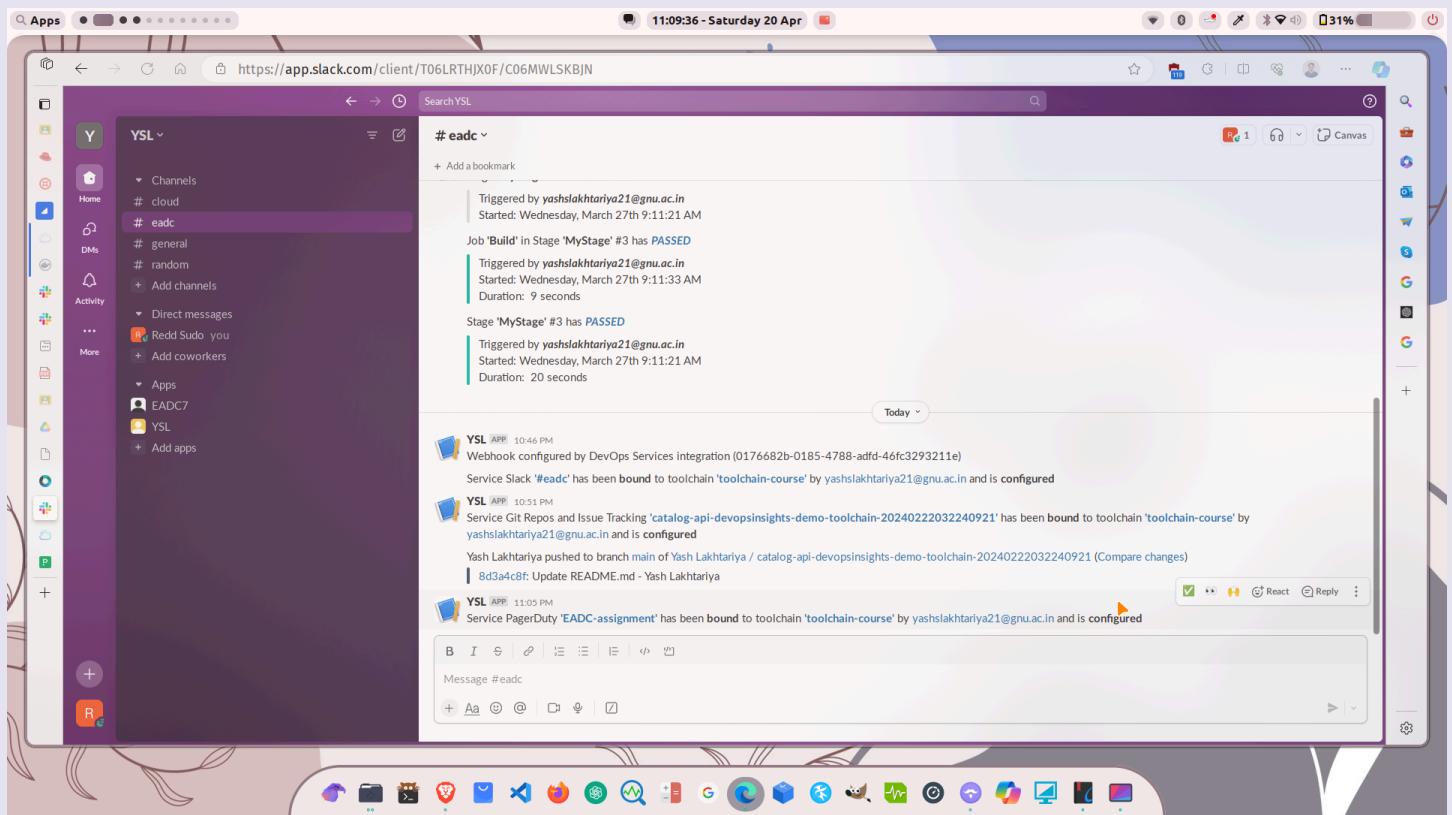
n. And also, to resolve it click option on right side



The screenshot shows the PagerDuty incident details page for incident #2. The incident is titled "demoincident". The status is set to "Resolved". The duration is listed as "00h 01m". The urgency is "High". The incident key is "f2c62355dd6c4c898220d300983d88c2". The impacted service is "EADC-assignment". The service description is "For educational purposes". The notes section contains a placeholder for adding a note. The resources section lists various links related to incident management. The bottom navigation bar includes tabs for Details, Status Updates, Timeline, Automation Actions Log, Past Incidents, and Related Incidents.

**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Assignment 2**

**o. If Slack integration is added, the notification will also be received in Slack app**

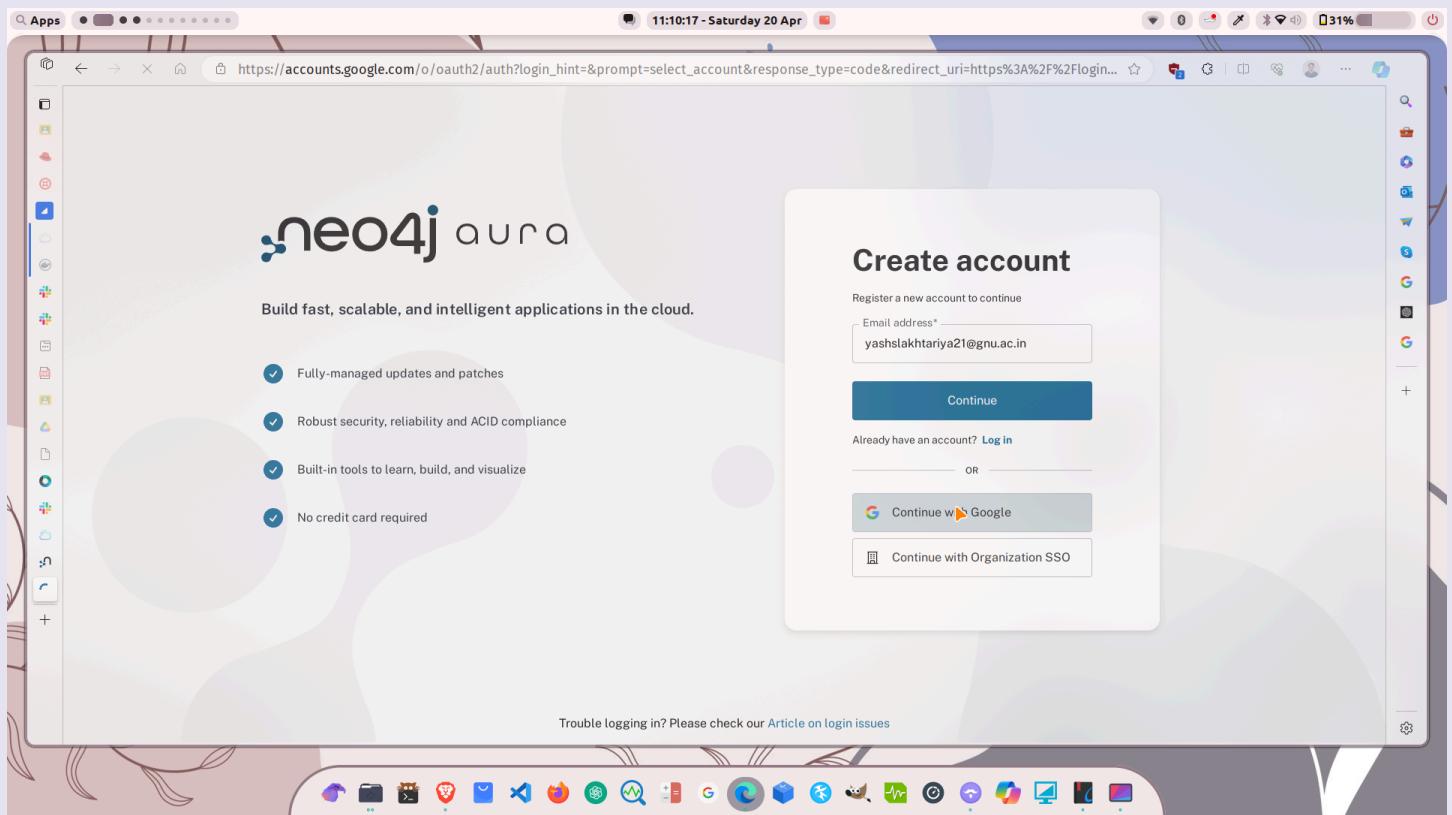


### **3. Explore Neo4j Graph Database:**

- Familiarize yourself with Neo4j graph database platform. Link to ne04j: <https://neo4j.com/>**
- Obtain a sample dataset related to social Network analysis.**
- Use Neo4j's Cypher query language to run queries on the dataset, exploring relationships and patterns within the graph.**

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Assignment 2

a. Create an account on neo4j if not exists

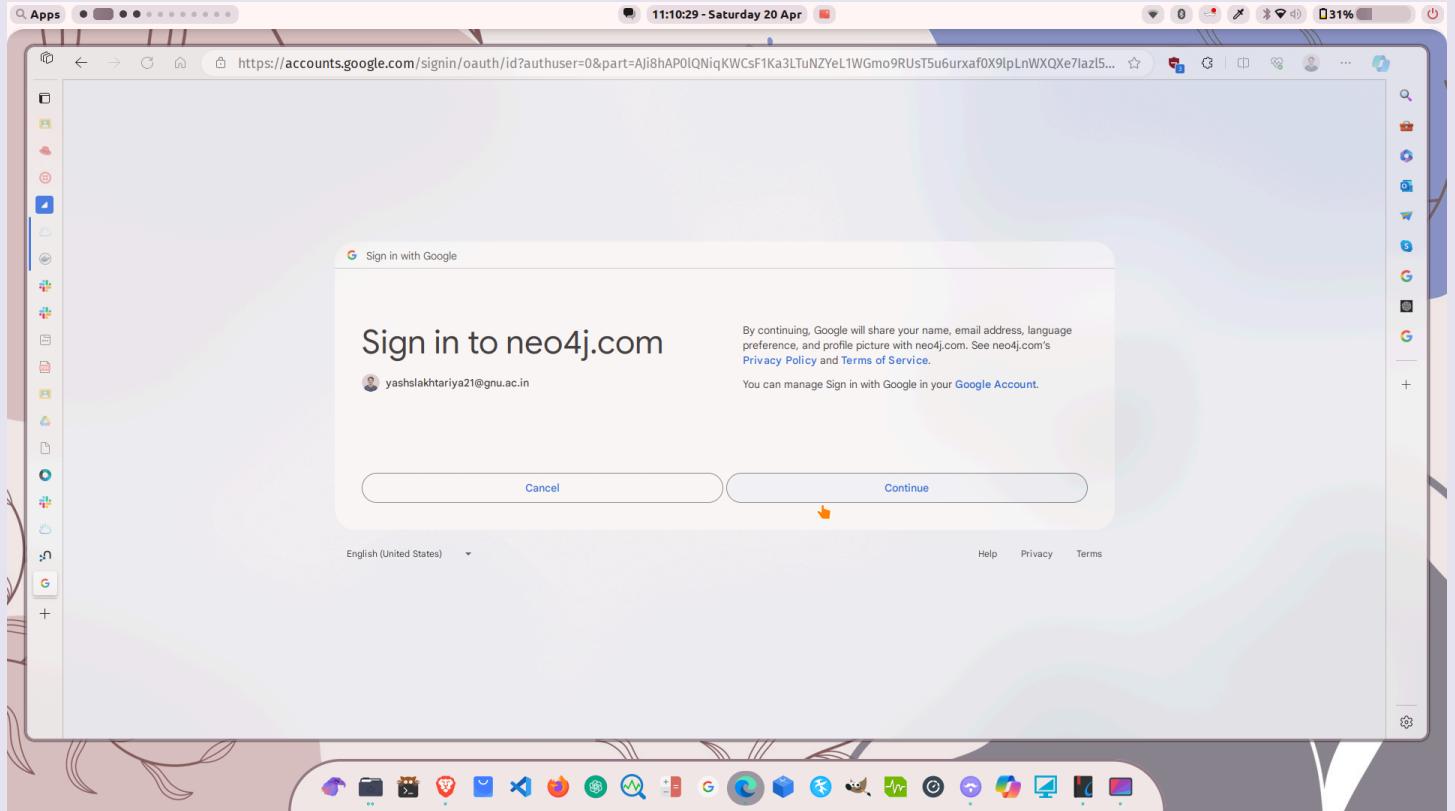


Name - Yash Lakhtariya

Enrollment number - 21162101012

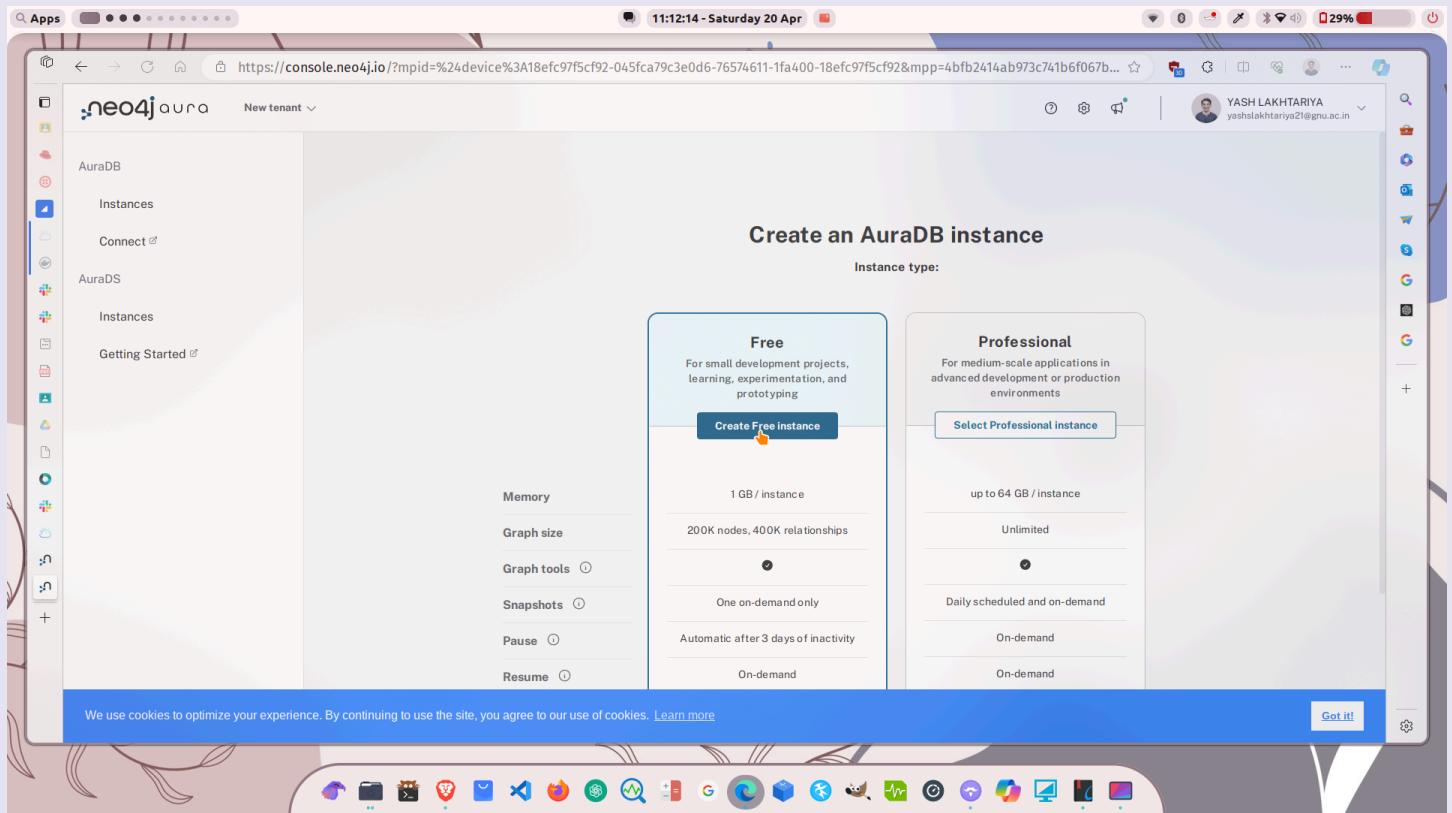
Branch - CBA      Batch - 61

EADC Assignment 2



**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Assignment 2**

**b. Create free instance for educational purposes**



The screenshot shows a web browser window for the neo4j aura platform at the URL <https://console.neo4j.io/?mpid=%24device%3A18efc97f5cf92-045fc79c3e0d-76574611-1fa400-18efc97f5cf92&mpp=4fbfb2414ab973c741b6f067b...>. The title bar indicates the time is 11:12:14 - Saturday 20 Apr. The user profile shows "YASH LAKHTARIYA" and the email "yash.lakhtariya21@gnu.ac.in".

The left sidebar has sections for AuraDB (Instances, Connect), and AuraDS (Instances, Getting Started). The main content area is titled "Create an AuraDB instance" and includes a "Instance type:" dropdown. It offers two options: "Free" and "Professional".

**Free** (selected):  
For small development projects, learning, experimentation, and prototyping.  
Create Free instance (button highlighted with a cursor).

**Professional**:  
For medium-scale applications in advanced development or production environments.  
Select Professional instance (button).

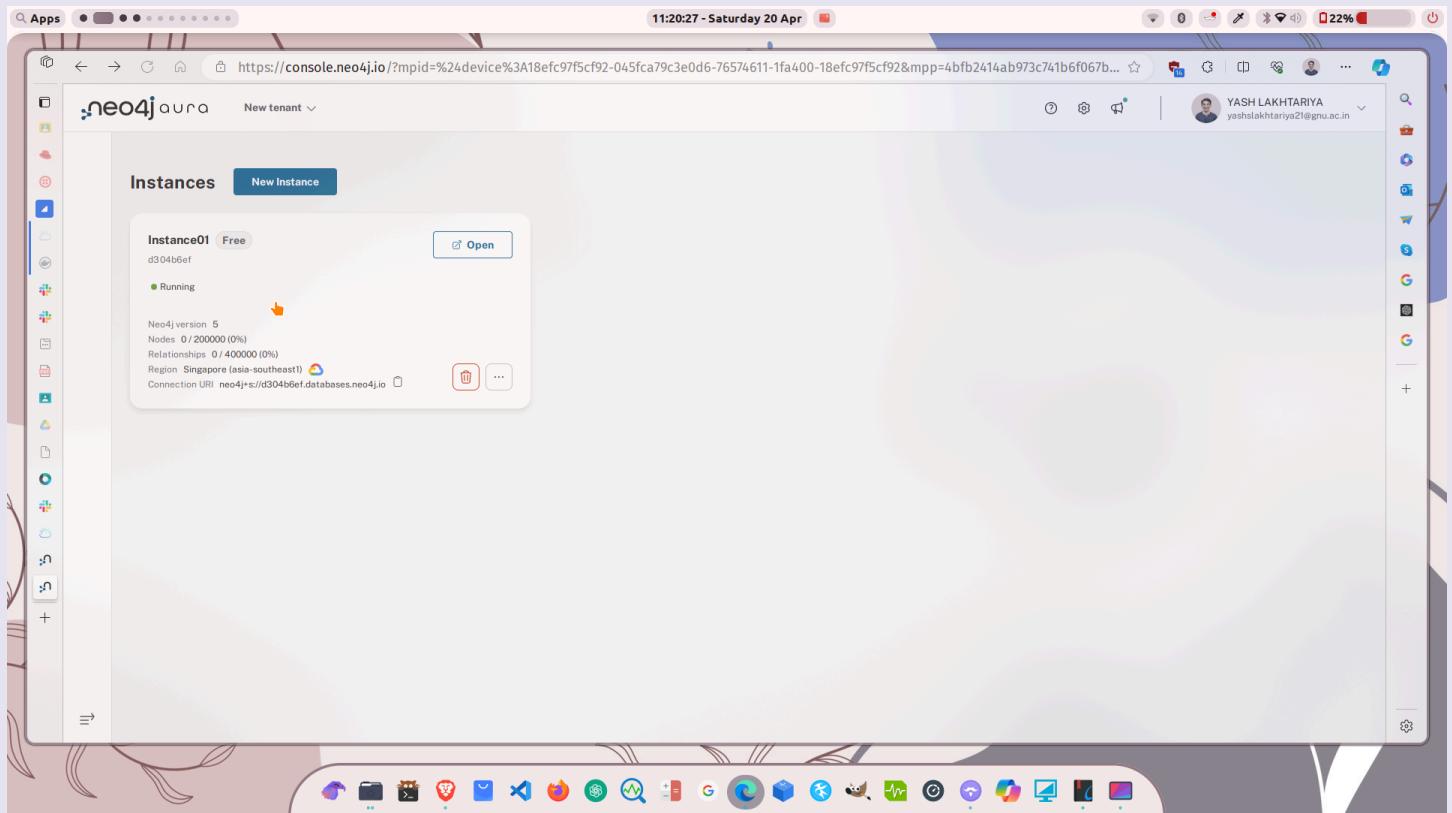
Below the instance type selection, there are detailed configuration tables:

Memory	Graph size	Graph tools	Snapshots	Pause	Resume
1 GB / instance	200K nodes, 400K relationships	•	One on-demand only	Automatic after 3 days of inactivity	On-demand
up to 64 GB / instance	Unlimited	•	Daily scheduled and on-demand	On-demand	On-demand

A blue banner at the bottom states: "We use cookies to optimize your experience. By continuing to use the site, you agree to our use of cookies. [Learn more](#)". A "Get it!" button is also present.

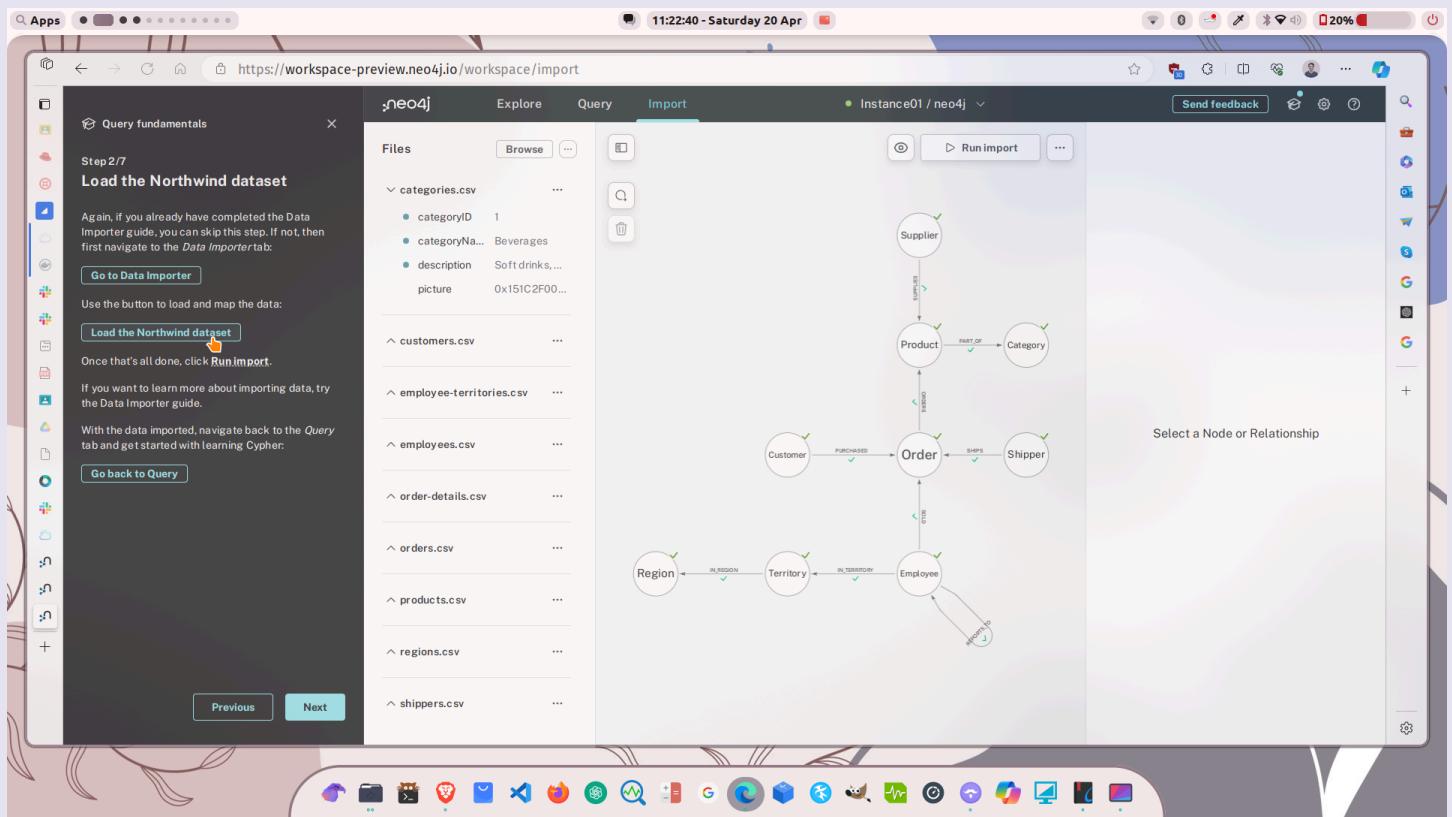
**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 61**  
**EADC Assignment 2**

c. Wait for few minutes and the instance will be ready



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Assignment 2

d. Load the demo database like Northwind database to run queries



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Assignment 2

e. The records, relationships and files are imported from database

The screenshot shows the Neo4j Browser interface on a Mac OS X desktop. The main window displays 'Import results' for the 'Northwind' dataset. It shows two completed imports: 'Territory' and 'Region'. The 'Territory' import summary is as follows:

Time taken	File size	File rows	Nodes created	Properties set	Labels added	Query time
00:00:00	965 B	53	53	106	53	00:00:00

The 'Region' import summary is as follows:

Time taken	File size	File rows	Nodes created	Properties set	Labels added	Query time
00:00:00	69 B	4	4	8	4	00:00:00

A tooltip suggests 'Start visually exploring your graph' with a button labeled 'Explore results'.

The Neo4j browser sidebar includes a 'Chat' section with a message from the system: 'In neo4j, I want to get example database of social ne...'. Below it are sections for 'Accessing Database', 'Neo4j Sandbox', 'Neo4j Browser Built-in Examples', 'Neo4j Graph Example Repository', 'Sample Queries', and 'Finding Friends'. A Cypher query example is shown: `MATCH (u:User {name: 'Alice'})-[:FRIEND]->-(friend)->-(friend)-<-[ :FRIEND]-(user1:User {name: 'Alice'})`.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Assignment 2

f. To check node data, run query

The screenshot shows the Neo4j web interface at <https://workspace-preview.neo4j.io/workspace/query>. The interface includes a sidebar with various icons, a top navigation bar with 'Explore', 'Query', and 'Import' tabs, and a main content area.

**Database information:**

- Nodes (1,104): Category, Customer, Employee, Order, Product, Region, Shipper, Supplier, Territory.
- Relationships (4,909): IN\_REGION, IN\_TERRITORY, ORDERS, PART\_OF, PURCHASED, REPORTS\_TO, SHIPS, SOLD, SUPPLIES.
- Property keys: address, birthDate, categoryID, categoryName, city, companyName, contactName, contactTitle, country, customerID, data, description, discontinued, discount, employeeID, extension, fax, firstName, freight, hireDate.

**Query results:**

```
$ MATCH (n:Product) RETURN n
```

The results show 25 records of products, each with properties like unitPrice, unitsInStock, reorderLevel, productID, discontinued, etc. The first few records are:

- (:Product {unitPrice: 18.0, unitsInStock: 39, reorderLevel: 10, productID: "1", discontinued: false})
- (:Product {unitPrice: 19.0, unitsInStock: 17, reorderLevel: 25, productID: "2", discontinued: false})
- (:Product {unitPrice: 10.0, unitsInStock: 13, reorderLevel: 25, productID: "3", discontinued: false})
- (:Product {unitPrice: 22.0, unitsInStock: 53, reorderLevel: 0, productID: "4", discontinued: false})
- (:Product {unitPrice: 21.35, unitsInStock: 0, reorderLevel: 0, productID: "5", discontinued: true})
- (:Product {unitPrice: 25.0, unitsInStock: 120, reorderLevel: 25, productID: "6", discontinued: false})
- (:Product {unitPrice: 30.0, unitsInStock: 15, reorderLevel: 10, productID: "7", discontinued: false})
- (:Product {unitPrice: 40.0, unitsInStock: 6, reorderLevel: 0, productID: "8", discontinued: false})
- (:Product {unitPrice: 97.0, unitsInStock: 29, reorderLevel: 0, productID: "9", discontinued: true})

Started streaming 25 records after 15ms and completed after 18ms.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Assignment 2

g. For specifying relationship in query, use [ ] and part of keyword, the graph results can also be seen

The screenshot shows the Neo4j browser interface on a Mac OS X desktop. The title bar indicates it's running on Instance01 / neo4j at 11:29:16 on Saturday, April 20. The main area displays a query results graph.

**Query fundamentals** sidebar:

- Step 4/7 Executing queries with graph results**
- Central to the Query UI is the Cypher editor on top where you write and run your queries and the list of result frames.
- When you run any query, a result frame shows the query and its results, pushing previous ones down.
- Try it out by typing in and executing the following Cypher query:

```
1 MATCH path=(-Product)-[:PART_OF]-> c WHERE c.categoryName = 'Produce'
2 RETURN path;
```

**Database information** sidebar:

- Nodes (1,104): Category, Customer, Employee, Order, Product, Region, Shipper, Supplier, Territory.
- Relationships (4,909): IN\_REGION, IN\_TERRITORY, ORDERS, PART\_OF, PURCHASED, REPORTS\_TO, SHIPS, SOLD, SUPPLIES.
- Property keys: address, birthDate, categoryID, categoryName, city, companyName, contactName, contactTitle, country, customerID, data, description, discontinued, discount, employeeID, extension, fax, firstName, freight, hireDate.

**Results overview** panel:

- Nodes (6): Category (1), Product (5).
- Relationships (5): PART\_OF (5).

The graph visualization shows a central node labeled "Produce" connected to five other nodes: "Longlife Tofu", "Manjim-up Drin..", "Onde Bala's D..", "Tofu", and "Rossle Sauerkrc.". Each connection is labeled with the relationship type: "PART\_OF".

Below the graph, a message states: "Started streaming 5 records after 47ms and completed after 54ms."

At the bottom, another query is shown:

```
neo4j$ MATCH (n:Product) RETURN n LIMIT 25;
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Assignment 2

h. While, the table tab displays the results in tabular format

The screenshot shows the Neo4j Query Editor interface. On the left, there is a sidebar with various icons and a main panel titled "Executing queries with graph results". It contains a Cypher query:1 MATCH path=(:Product)-[:PART\_OF]-(c:Category) WHERE c.categoryName = 'Pro...  
2 WHERE c.categoryName = 'Pro...  
3 RETURN path;A note below says: "You can edit the query within the frame and run again without creating a new result frame. The frame has two important views: Graph and Table. (Plans for showing query plans and RAW for the raw results). The default view is graph if your query returns graph elements like nodes, relationships or paths. Relationships are only shown if you return a path, or name and return the relationships as well as the nodes." At the bottom of this panel are "Previous" and "Next" buttons.

The central part of the interface is the "Database information" section, which lists nodes (1,104), relationships (4,909), and property keys. Below this is a search bar with the text "neo4j\$".

On the right, there are two result frames. The top one shows the output of the first query:\* neo4j\$ MATCH path=(:Product)-[:PART\_OF]-(c:Category) WHERE c.categoryName = 'Pro...  
+ (:Product {unitPrice: 30.0, unitsInStock: 15, reorderLevel: 10, productID: "7", discontinued: false}  
+ (:Product {unitPrice: 23.25, unitsInStock: 35, reorderLevel: 0, productID: "14", discontinued: false}  
+ (:Product {unitPrice: 45.6, unitsInStock: 26, reorderLevel: 0, productID: "28", discontinued: true}  
+ (:Product {unitPrice: 53.0, unitsInStock: 20, reorderLevel: 10, productID: "51", discontinued: false}  
+ (:Product {unitPrice: 10.0, unitsInStock: 4, reorderLevel: 5, productID: "74", discontinued: false}A progress bar at the bottom of this frame indicates "Started streaming 5 records after 47ms and completed after 54ms".

The bottom result frame shows the output of a second query:\* neo4j\$ MATCH (n:Product) RETURN n LIMIT 25;At the bottom of the interface is a toolbar with various application icons.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Assignment 2

i. Try some other queries with conditions for checking

The screenshot shows the Neo4j workspace interface. On the left, there's a sidebar with various icons and a 'Query fundamentals' panel. The main area has tabs for 'Explore', 'Query', and 'Import'. The 'Query' tab is active, showing a database information section with nodes (1,104) and relationships (4,909). Below this is a property keys section with many options like address, birthDate, categoryID, etc. A query editor window displays the following Cypher query:

```
1 MATCH (p:Product)-[:PART_OF]-(c:Category)
2 WHERE c.categoryName = 'Produce'
3 RETURN p.productName, c.categoryName;
```

The results table shows five rows of data:

p.productName	c.categoryName
"Uncle Bob's Organic Dried Pears"	"Produce"
"Tofu"	"Produce"
"Rössle Sauerkraut"	"Produce"
"Manjimup Dried Apples"	"Produce"
"Longlife Tofu"	"Produce"

At the bottom of the results window, it says "Started streaming 5 records after 33ms and completed after 34ms".

### Analysis of Neo4j platform and databases :

- Graph Database:** Neo4j is a graph database that stores data in a graph structure. It represents data as nodes (entities) and relationships (edges) between them.
- Property Graph Model:** Neo4j uses the property graph model, where nodes and relationships can have properties (key-value pairs).
- Schema-Free:** Unlike relational databases, Neo4j is schema-free, allowing flexible data modeling.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 61  
EADC Assignment 2

### Features :

- **Graph-Based:** Neo4j leverages the power of graphs for efficient querying and traversing complex relationships.
- **ACID Transactions:** Neo4j supports ACID transactions for data consistency.
- **Highly Scalable:** It can handle large-scale graphs and is horizontally scalable.
- **Cypher Query Language:** Neo4j uses Cypher, a declarative query language, to retrieve data from the graph.
- **Graph Algorithms:** Neo4j provides built-in graph algorithms for tasks like shortest path, community detection, etc.

### Working with Cypher :

- **Cypher Syntax:** Cypher is similar to SQL but designed for graphs. It uses ASCII-art syntax to represent patterns in the graph.
- **Pattern Matching:** When writing a Cypher query, you draw a graph pattern through your data. It's intuitive and expressive.
- **CRUD Operations:** You can use Cypher for create, read, update, and delete operations on the graph.
- **Primary Interface:** Cypher is the primary interface for Neo4j, making it easy to interact with the graph.

→ Hence, Neo4j is a powerful graph database that uses Cypher for querying and managing graph data. It's ideal for scenarios involving complex relationships and pattern-based queries