CLOUD & SERVERLESS COMPUTING (20CS3281AA)

Project Report

Name: Thatishetty Aakash Chandra

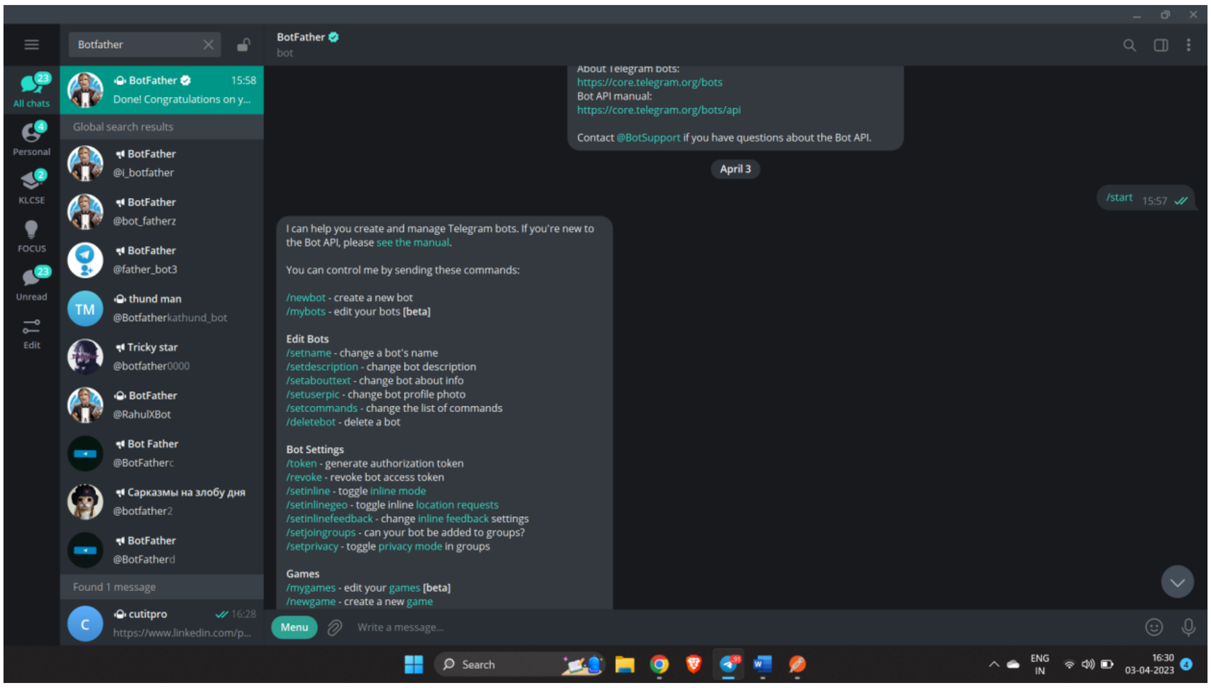
ID NO: 2000031377
Section: 15
Faculty: Mr. K V Ravi Teja Sir

CutitPro

The Bot reduces the length of the URL you provide.

The successful execution of this project necessitates the utilization of two critical components: Telegram Botfather and Amazon Serverless Services.

Telegram Botfather:



BotFather 15:58
Done! Congratulations on y...

BotFather
@i_botfather

BotFather
@bot_fatherz

BotFather
@father_bot3

thund man
@Botfatherkathund_bot

Tricky star
@botfather0000

BotFather
@RahulXBot

Bot Father
@BotFatherc

Сарказмы на злобу дня
@botfather2

BotFather
@BotFatherd

cutitpro 16:28
https://www.linkedin.com/p...

BotFather
bot

About Telegram bots:
https://core.telegram.org/bots
Bot API manual:
https://core.telegram.org/bots/api

Contact @BotSupport if you have questions about the Bot API.

April 3

/start 15:57

I can help you create and manage Telegram bots. If you're new to the Bot API, please see the manual.

You can control me by sending these commands:

/newbot - create a new bot
/mybots - edit your bots [beta]

Edit Bots
/setname - change a bot's name
/setdescription - change bot description
/setabouttext - change bot about info
/setuserpic - change bot profile photo
/setcommands - change the list of commands
/deletebot - delete a bot

Bot Settings
/token - generate authorization token
/revoke - revoke bot access token
/setinline - toggle inline mode
/setinlinegeo - toggle inline location requests
/setinlinefeedback - change inline feedback settings
/setjoingroups - can your bot be added to groups?
/setprivacy - toggle privacy mode in groups

Games
/mygames - edit your games [beta]
/newgame - create a new game

ENG IN 16:30 03-04-2023

Commands used:

/start: It will provide the menu of the services of the bot provide.

/newbot: Create a new bot.

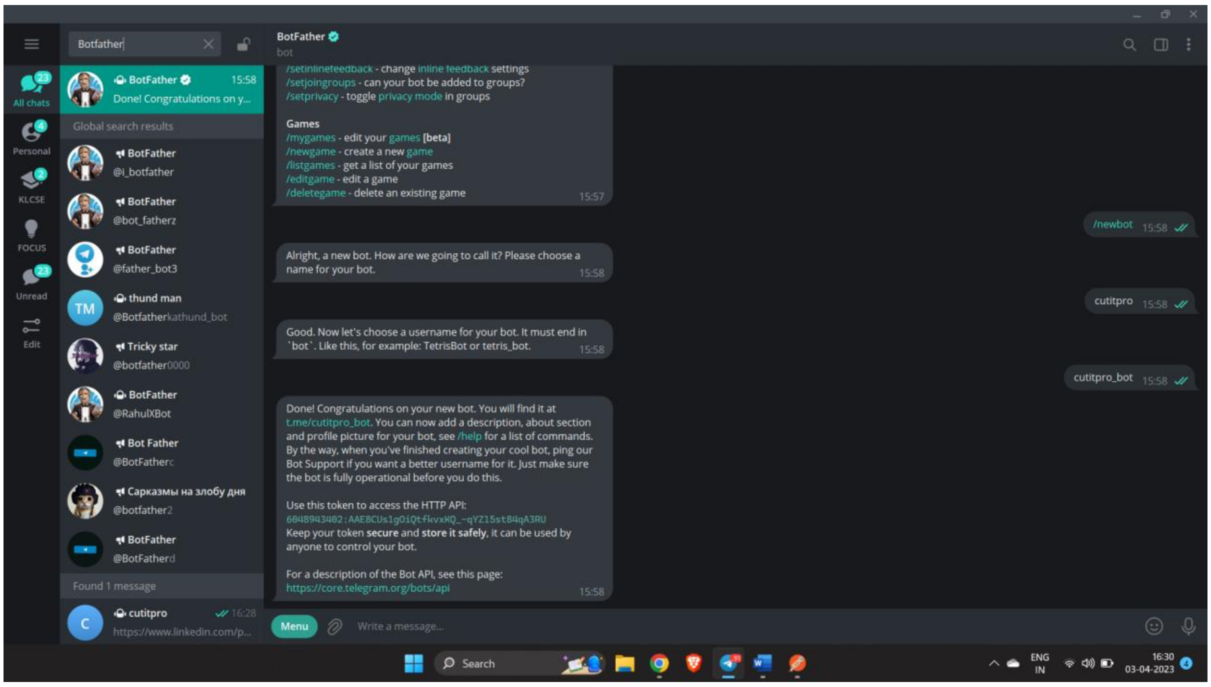
You will be asked:

Alright, a new bot. How are we going to call it? Please choose a name for your bot.

So, choose a name. Let me name my bot as cutitpro.

Next, Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot.

So, choose a username, let me name my username as cutitpro_bot.



/setinlinefeedback - change inline feedback settings
/setjoingroups - can your bot be added to groups?
/setprivacy - toggle privacy mode in groups

Games
/mygames - edit your games [beta]
/newgame - create a new game
/listgames - get a list of your games
/editgame - edit a game
/deletegame - delete an existing game 15:57

/newbot 15:58

Alright, a new bot. How are we going to call it? Please choose a name for your bot. 15:58

cutitpro 15:58

Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot. 15:58

cutitpro_bot 15:58

Done! Congratulations on your new bot. You will find it at t.me/cutitpro_bot. You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API:
6048943402:AAE8CUs1gOiQtfkvxKQ_-qYZ15st84qA3RU
Keep your token **secure** and **store it safely**, it can be used by anyone to control your bot.

For a description of the Bot API, see this page:
https://core.telegram.org/bots/api 15:58

Copy the Message you receive once you set the username and store it,

Done! Congratulations on your new bot. You will find it at t.me/jainpixie_bot. You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

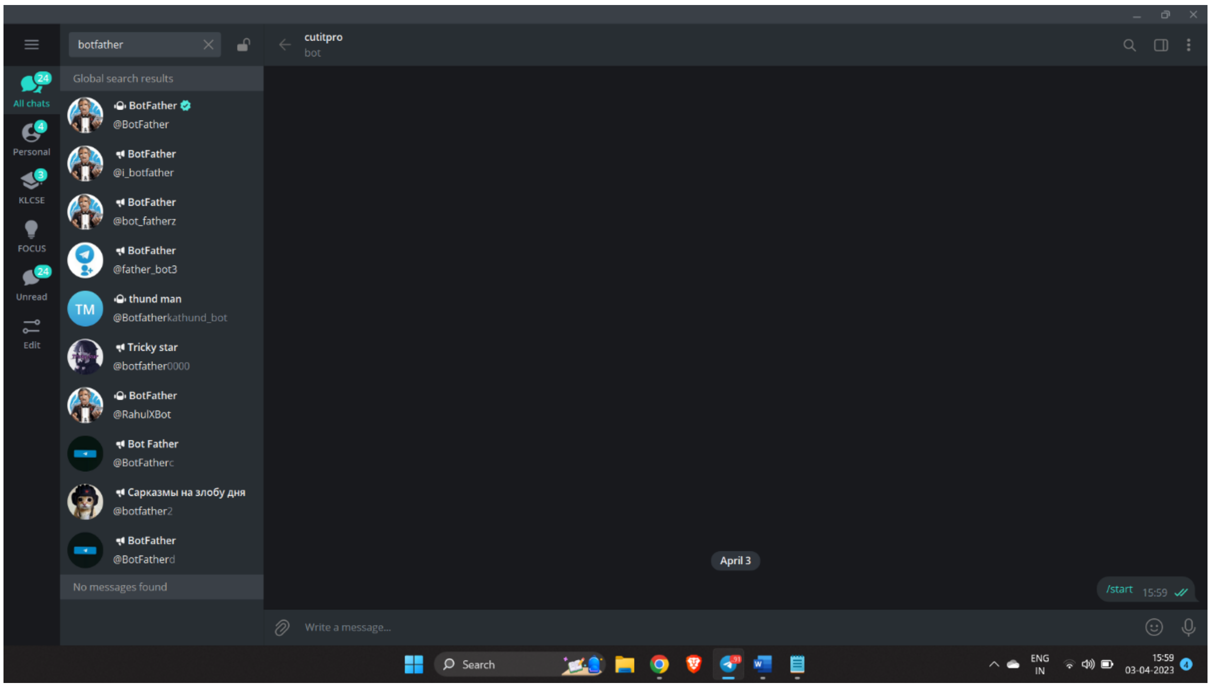
Use this token to access the HTTP API:

5903681678:AAGtPzYVcyQhp4_6dLYVSTc3GT2lkfegnSQ

Keep your token secure and store it safely, it can be used by anyone to control your bot.

For a description of the Bot API, see this page: https://core.telegram.org/bots/api

Your initial bot would look like:



Amazon Serverless Services:

1. API GATEWAY

2. LAMBDA FUNCTION

3. CLOUD WATCH

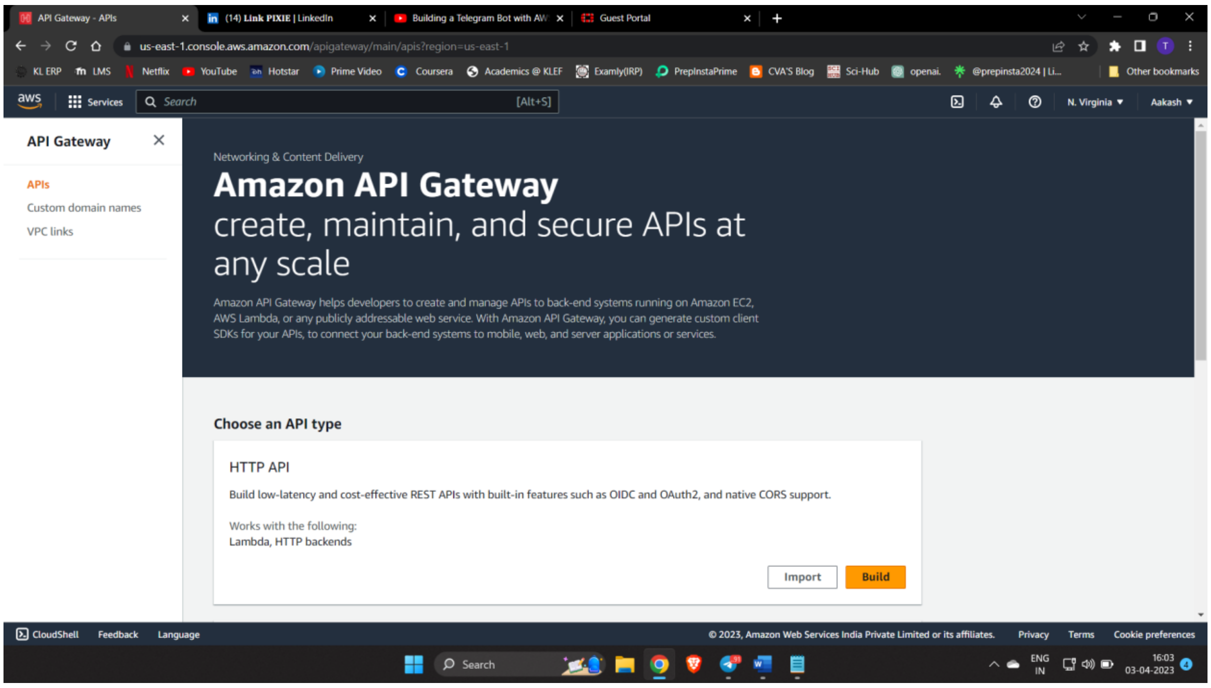
API GATEWAY

Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. APIs act as the "front door" for applications to access data, business logic, or functionality from your backend services.

Using API Gateway, user can create RESTful APIs and WebSocket APIs that enable real-time two-way communication applications. API Gateway supports containerized and serverless workloads, as well as web applications.

API Gateway handles all the tasks involved in accepting and processing up to hundreds of thousands of concurrent API calls, including traffic management, CORS support, authorization and access control, throttling, monitoring, and API version management.

API Gateway has no minimum fees or startup costs. You pay for the API calls you receive, and the amount of data transferred out and, with the API Gateway tiered pricing model, you can reduce your cost as your API usage scales



API Gateway - APIs | (14) Link PIXIE | LinkedIn | Building a Telegram Bot with AW | Guest Portal

us-east-1.console.aws.amazon.com/apigateway/main/apis?region=us-east-1

KL ERP | LMS | Netflix | YouTube | Hotstar | Prime Video | Coursera | Academics @ KLEF | Examly(IRP) | PrepInstaPrime | CVA'S Blog | Sci-Hub | openai. | @prepinsta2024 | Li... | Other bookmarks

aws | Services | Search [Alt+S] | N. Virginia ▾ | Aakash ▾

API Gateway

APIs

Custom domain names

VPC links

Networking & Content Delivery

Amazon API Gateway

create, maintain, and secure APIs at any scale

Amazon API Gateway helps developers to create and manage APIs to back-end systems running on Amazon EC2, AWS Lambda, or any publicly addressable web service. With Amazon API Gateway, you can generate custom client SDKs for your APIs, to connect your back-end systems to mobile, web, and server applications or services.

Choose an API type

HTTP API

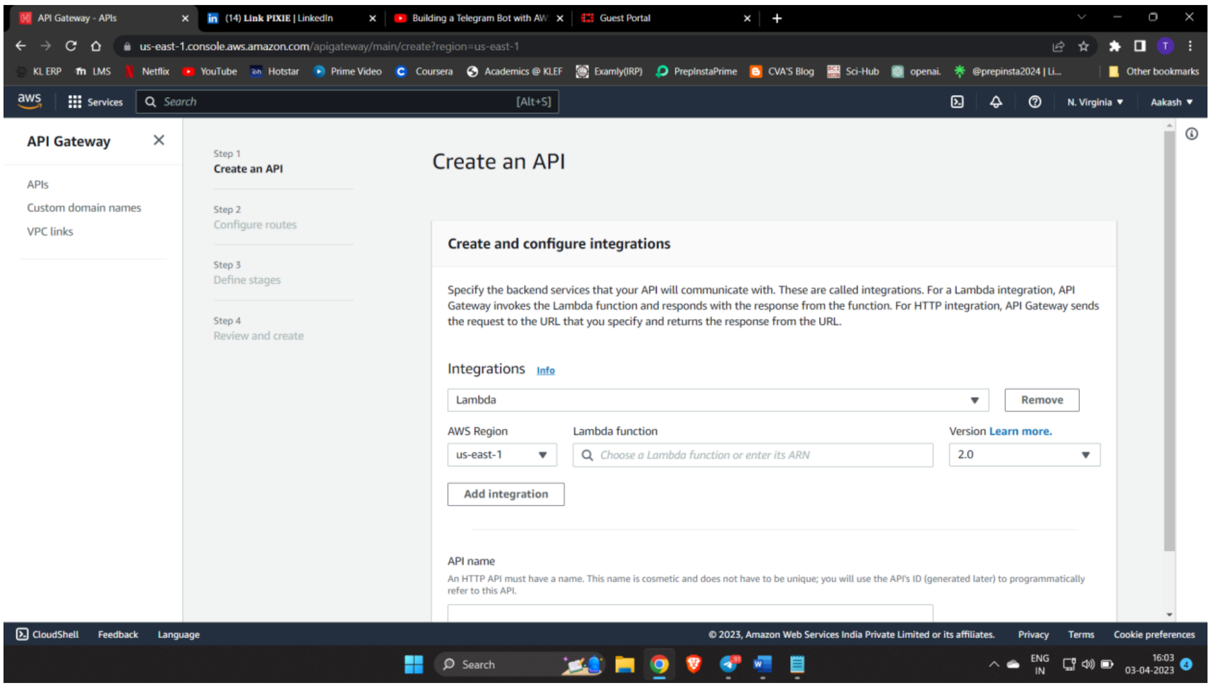
Build low-latency and cost-effective REST APIs with built-in features such as OIDC and OAuth2, and native CORS support.

Works with the following:
Lambda, HTTP backends

Import | Build

CloudShell | Feedback | Language | © 2023, Amazon Web Services India Private Limited or its affiliates. | Privacy | Terms | Cookie preferences

Search | ENG IN | 16:03 03-04-2023



API Gateway - APIs | (14) Link PIXIE | LinkedIn | Building a Telegram Bot with AW | Guest Portal

us-east-1.console.aws.amazon.com/apigateway/main/create?region=us-east-1

KL ERP | LMS | Netflix | YouTube | Hotstar | Prime Video | Coursera | Academics @ KLEF | Examly(IRP) | PrepInstaPrime | CVA'S Blog | Sci-Hub | openai. | @prepinsta2024 | Li... | Other bookmarks

aws | Services | Search [Alt+S] | N. Virginia ▾ | Aakash ▾

API Gateway

APIs

Custom domain names

VPC links

Step 1
Create an API

Step 2
Configure routes

Step 3
Define stages

Step 4
Review and create

Create an API

Create and configure integrations

Specify the backend services that your API will communicate with. These are called integrations. For a Lambda integration, API Gateway invokes the Lambda function and responds with the response from the function. For HTTP integration, API Gateway sends the request to the URL that you specify and returns the response from the URL.

Integrations Info

Lambda ▾ | Remove

AWS Region	Lambda function	Version Learn more.
us-east-1 ▾	Choose a Lambda function or enter its ARN	2.0 ▾

Add integration

API name

An HTTP API must have a name. This name is cosmetic and does not have to be unique; you will use the API's ID (generated later) to programmatically refer to this API.

CloudShell | Feedback | Language | © 2023, Amazon Web Services India Private Limited or its affiliates. | Privacy | Terms | Cookie preferences

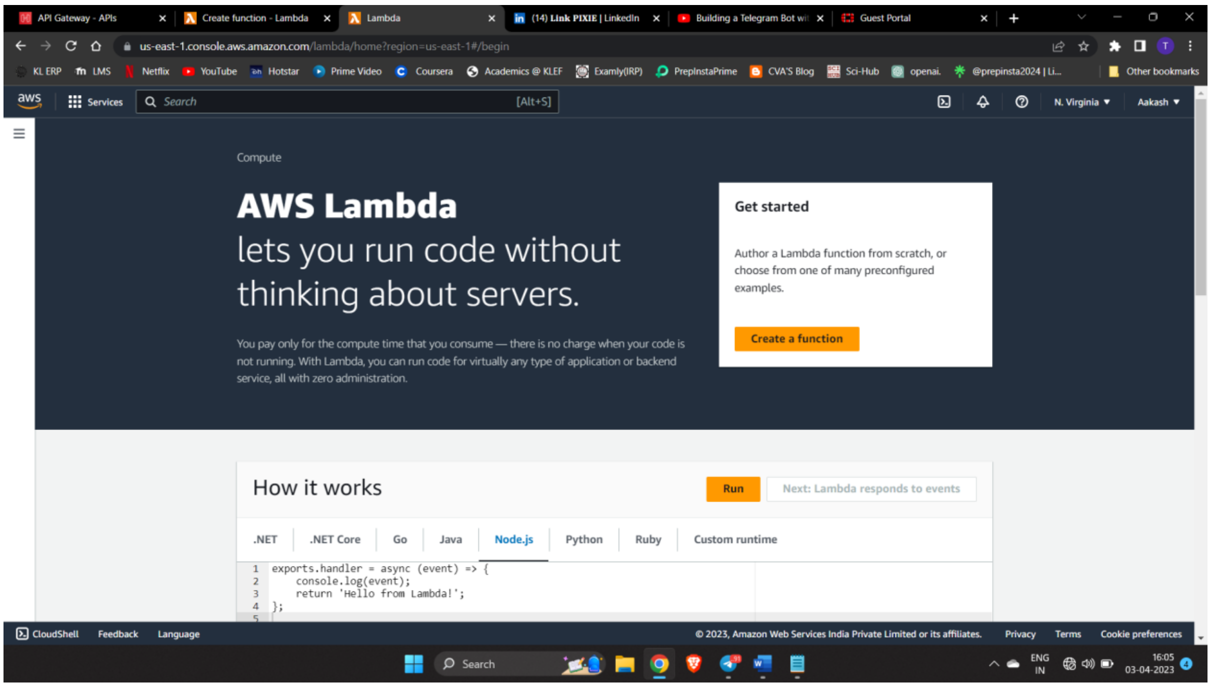
Search | ENG IN | 16:03 03-04-2023

To integrate API Gateway with a Lambda function, we need to create a Lambda function first. Therefore, we can pause the current step in API Gateway and focus on creating the Lambda function. Once we have created the function, we can then integrate it with API Gateway to complete the integration process.

LAMBDA FUNCTION

AWS Lambda is an event-driven, serverless computing platform provided by Amazon as a part of Amazon Web Services. Therefore, the user/customer doesn't need to worry about which AWS resources to launch, or how to manage them. Instead, the user needs to put the code on Lambda, and it runs.

In AWS Lambda the code is executed based on the response of events in AWS services such as add/delete files in S3 bucket, HTTP request from Amazon API gateway, etc. However, Amazon Lambda can only be used to execute background tasks. AWS Lambda function helps you to focus on your core product and business logic instead of managing operating system (OS) access control, OS patching, right-sizing, provisioning, scaling, etc.



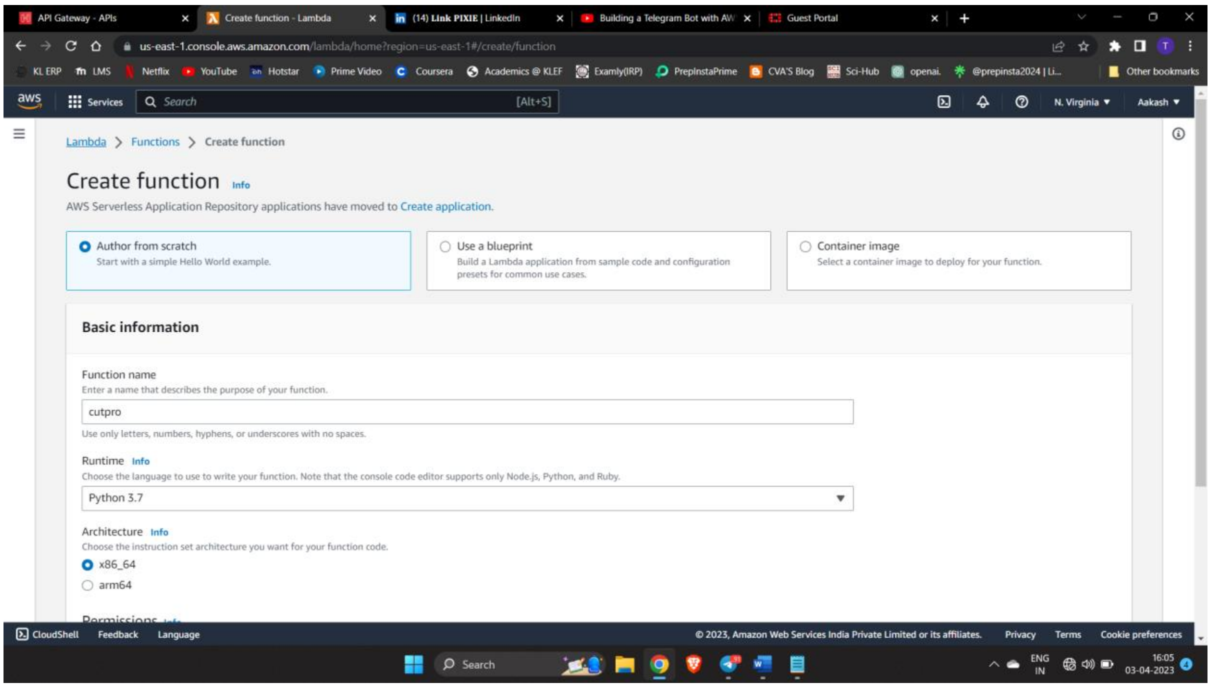
AWS Lambda
lets you run code without thinking about servers.

You pay only for the compute time that you consume — there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration.

Author a Lambda function from scratch, or choose from one of many preconfigured examples.

How it works

```
exports.handler = async (event) => {
    console.log(event);
    return 'Hello from Lambda!';
};
```



Lambda > Functions > Create function

Create function Info

AWS Serverless Application Repository applications have moved to Create application.

[✓ Author from scratch] Start with a simple Hello World example.

Use a blueprint — Build a Lambda application from sample code and configuration presets for common use cases.

Container image — Select a container image to deploy for your function.

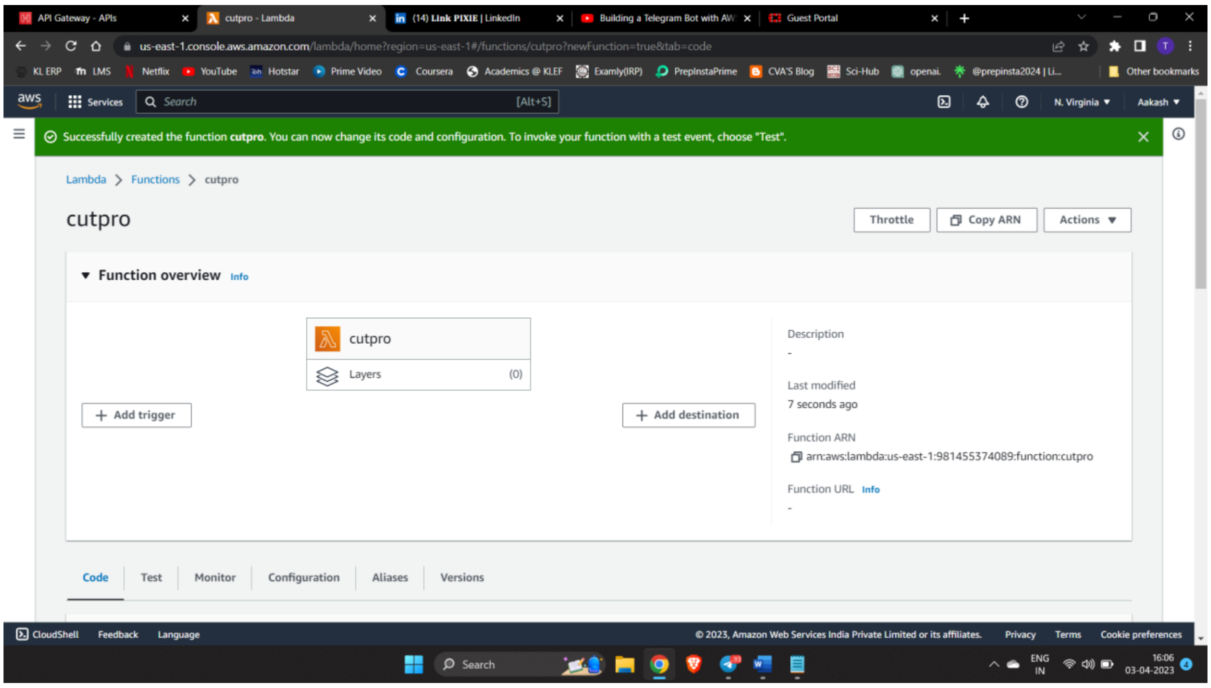
Basic information

Function name
Enter a name that describes the purpose of your function.
cutpro
Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.7

Architecture Info
Choose the instruction set architecture you want for your function code.
[✓ x86_64]
arm64

© 2023, Amazon Web Services India Private Limited or its affiliates.



Successfully created the function **cutpro**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > cutpro

cutpro

Function overview Info

cutpro
Layers (0)

Description
-

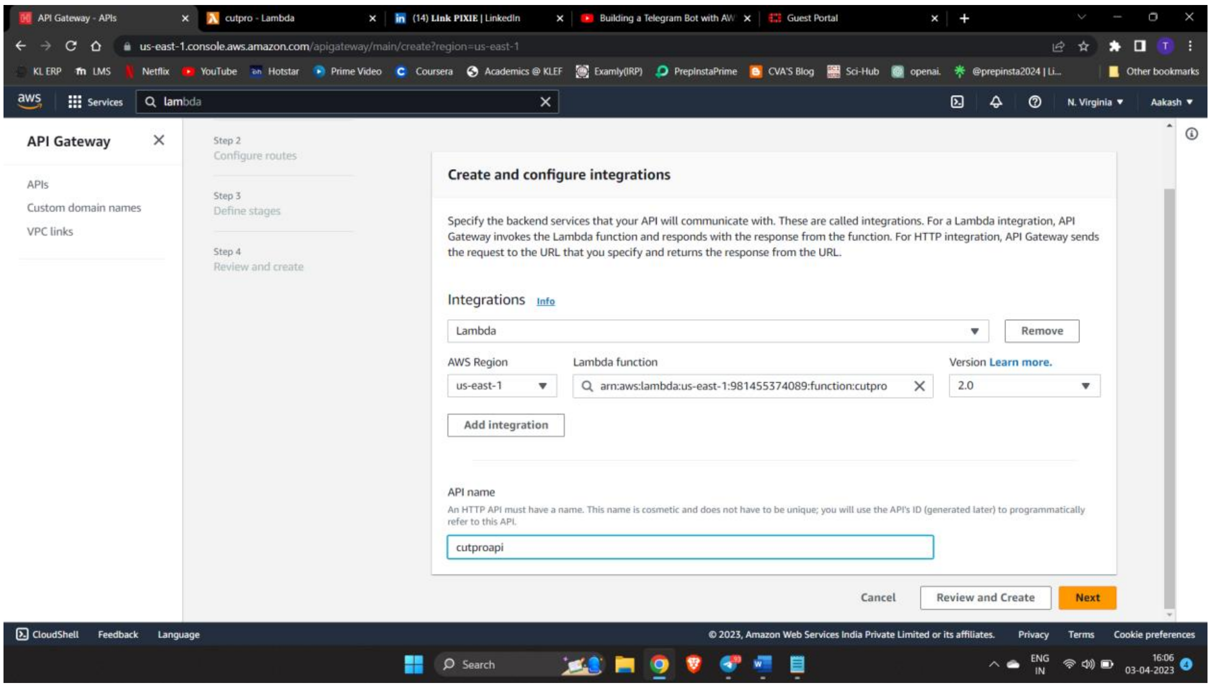
Last modified
7 seconds ago

Function ARN
arn:aws:lambda:us-east-1:981455374089:function:cutpro

Function URL Info
-

© 2023, Amazon Web Services India Private Limited or its affiliates.

Since we have created a Lambda function, we can continue with the integration process between the API Gateway and Lambda function. Therefore, we can move back to API Gateway and proceed with the next steps to complete the integration.


API Gateway

APIs
Custom domain names
VPC links

Step 2
Configure routes

Step 3
Define stages

Step 4
Review and create

Create and configure integrations

Specify the backend services that your API will communicate with. These are called integrations. For a Lambda integration, API Gateway invokes the Lambda function and responds with the response from the function. For HTTP integration, API Gateway sends the request to the URL that you specify and returns the response from the URL.

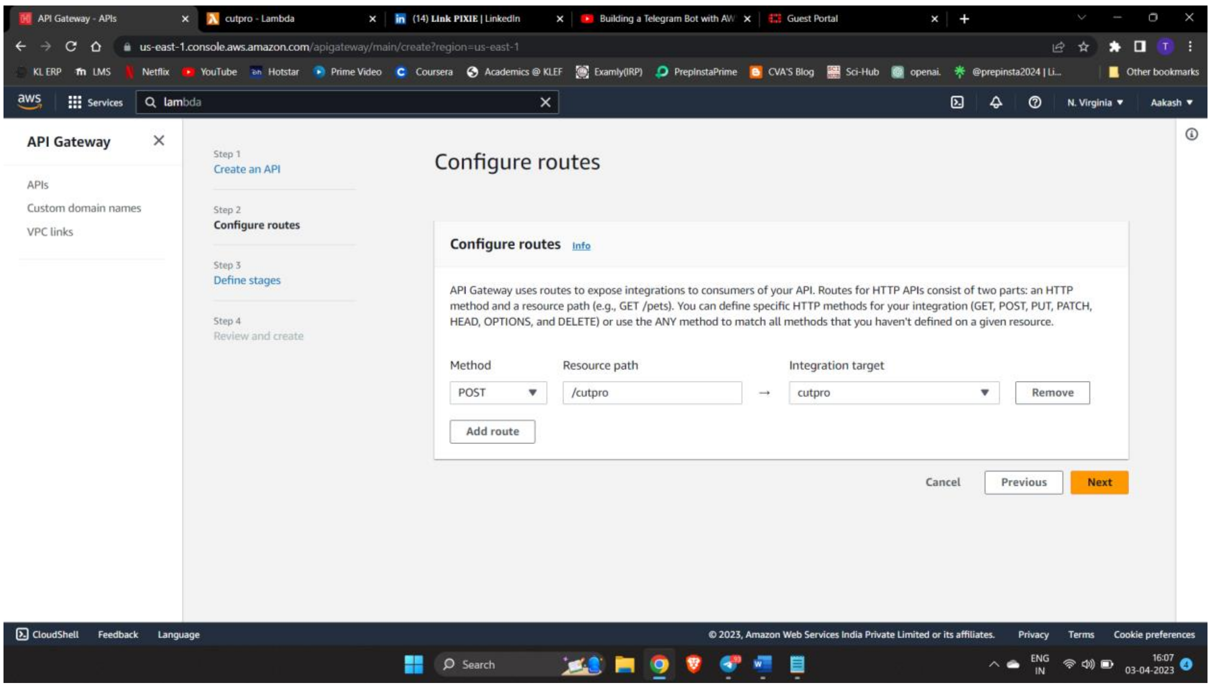
Integrations Info

Lambda

AWS Region: us-east-1
Lambda function: arn:aws:lambda:us-east-1:981455374089:function:cutpro
Version: 2.0

API name
An HTTP API must have a name. This name is cosmetic and does not have to be unique; you will use the API's ID (generated later) to programmatically refer to this API.

cutproapi


API Gateway

APIs
Custom domain names
VPC links

Step 1
Create an API

Step 2
Configure routes

Step 3
Define stages

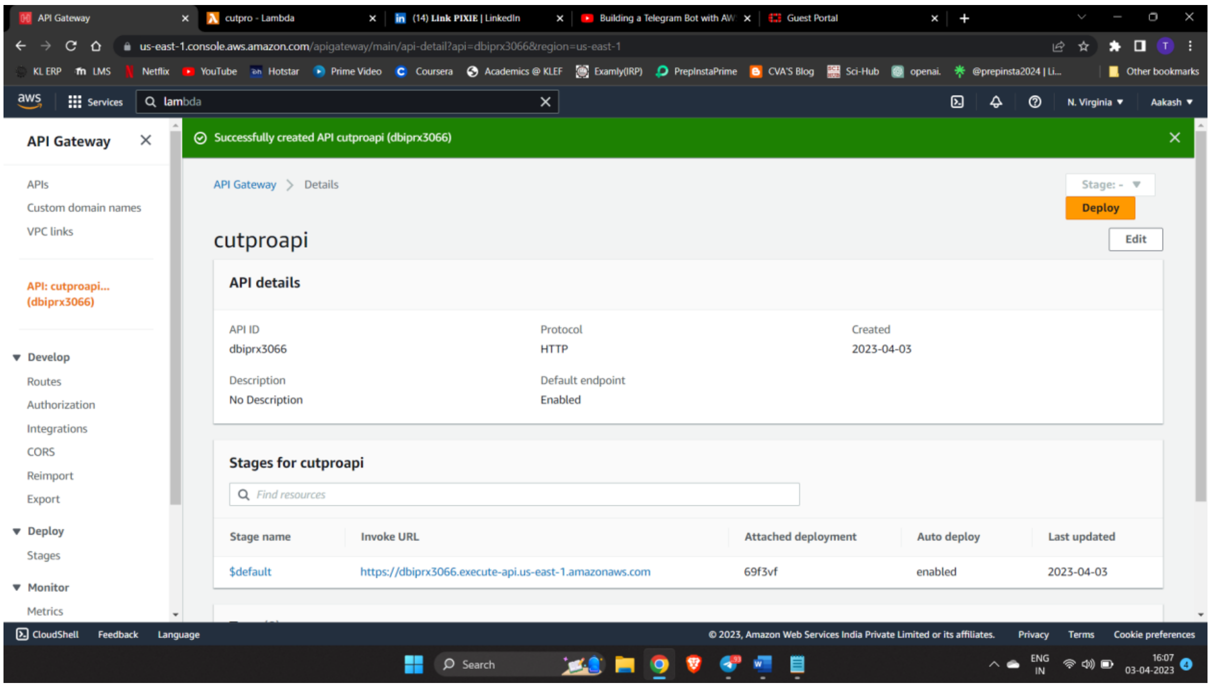
Step 4
Review and create

Configure routes

Configure routes Info

API Gateway uses routes to expose integrations to consumers of your API. Routes for HTTP APIs consist of two parts: an HTTP method and a resource path (e.g., GET /pets). You can define specific HTTP methods for your integration (GET, POST, PUT, PATCH, HEAD, OPTIONS, and DELETE) or use the ANY method to match all methods that you haven't defined on a given resource.

Method	Resource path	Integration target
POST	/cutpro	cutpro



Successfully created API cutproapi (dbiprx3066)

API Gateway > Details

cutproapi

API details

API ID	Protocol	Created
dbiprx3066	HTTP	2023-04-03
Description	Default endpoint	
No Description	Enabled	

Stages for cutproapi

Stage name	Invoke URL	Attached deployment	Auto deploy	Last updated
$default	https://dbiprx3066.execute-api.us-east-1.amazonaws.com	69f3vf	enabled	2023-04-03

INVOKE URL : https://dbiprx3066.execute-api.us-east-1.amazonaws.com

Resource path : /cutpro

API GATEWAY EndPoint: https://dbiprx3066.execute-api.us-east-1.amazonaws.com

/ cutpro

To set up a webhook for your bot, you will need an endpoint where the webhook can send requests, as well as a token that was generated when you created the bot using BotFather. The endpoint is the URL where your server is listening for incoming requests. The token serves as a secure identifier for your bot to authenticate with the messaging platform's servers.

EndPoint: https://api.telegram.org/bot{HTTP Token}/setWebhook

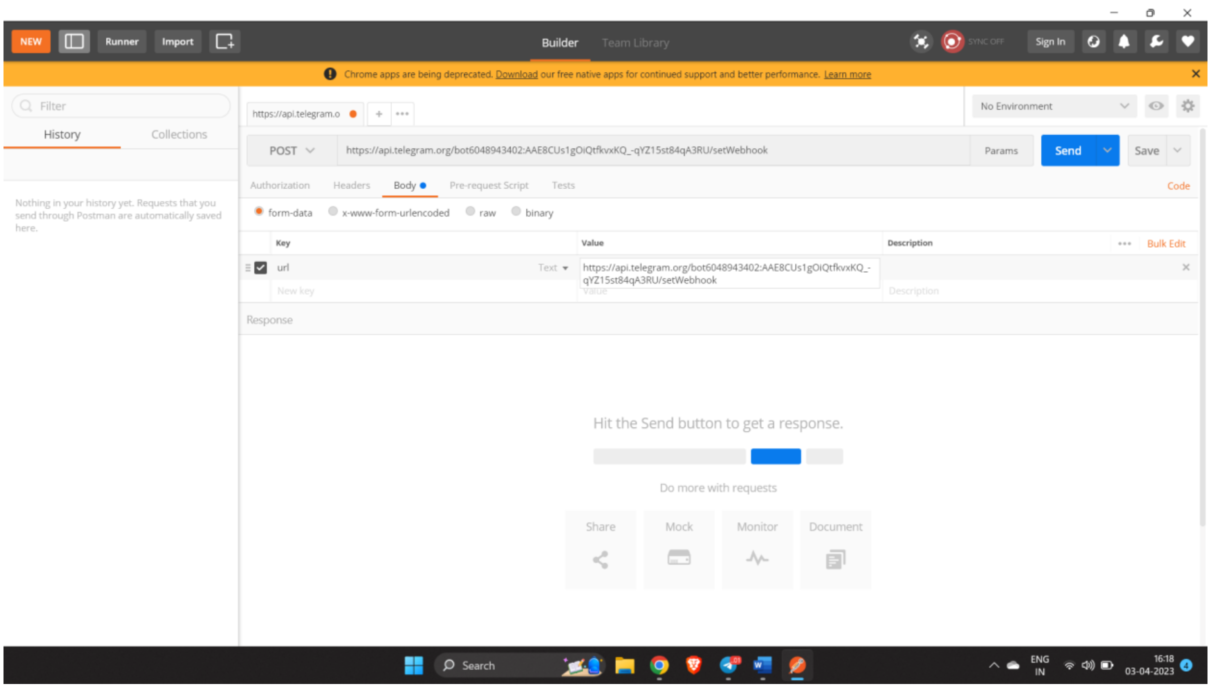
Token :

6048943402:AAE8CUs1gOiQtfkvxKQ_-qYZ15st84qA3RU

Perfect webhook endpoint url :

https://api.telegram.org/bot6048943402:AAE8CUs1gOiQtfkvxKQ_-qYZ15st84qA3RU/setWebhook

Now to setup,I will use Postman :



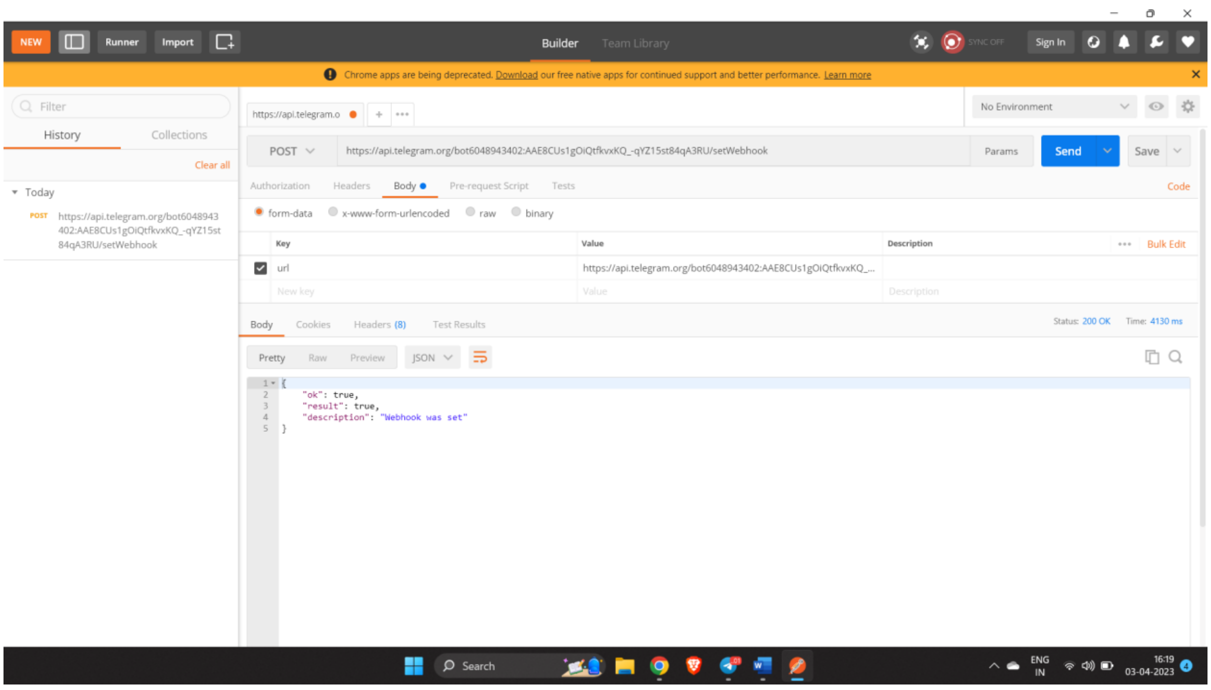
POST https://api.telegram.org/bot6048943402:AAE8CUs1gOiQtfkvxKQ_-qYZ15st84qA3RU/setWebhook

Body: form-data

Key	Value	Description
url	https://api.telegram.org/bot6048943402:AAE8CUs1gOiQtfkvxKQ_-qYZ15st84qA3RU/setWebhook	

Hit the Send button to get a response.

Click on Send your webhook will be successfully be setup:

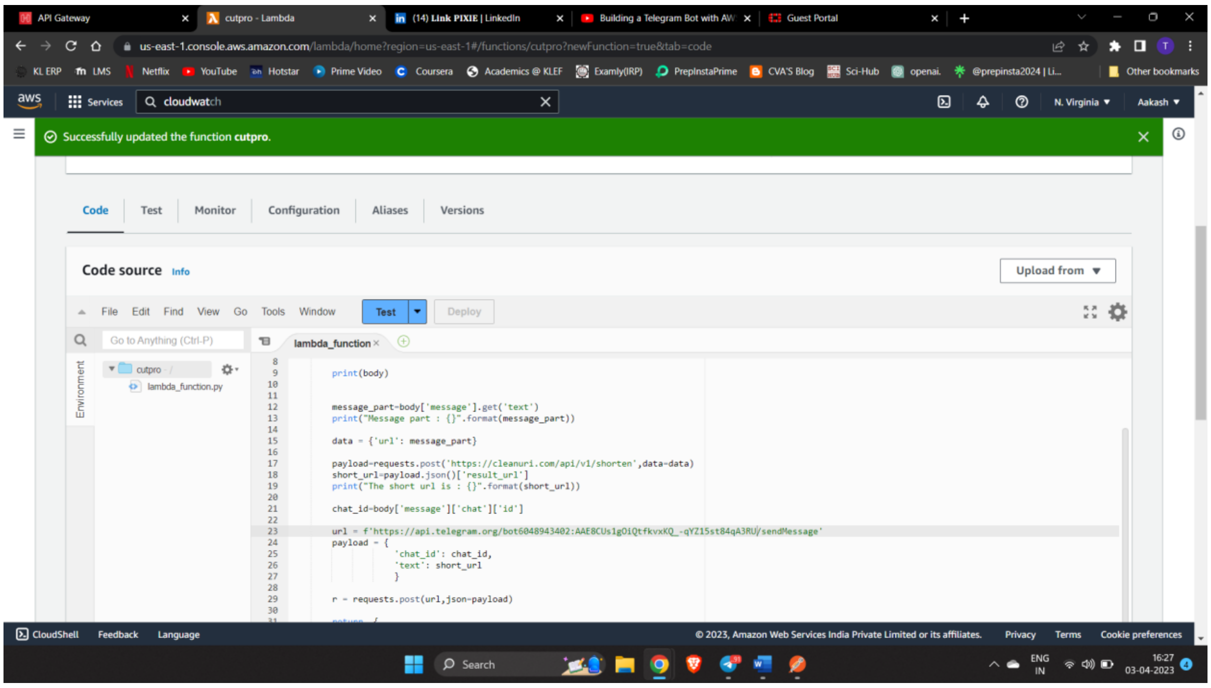


POST https://api.telegram.org/bot6048943402:AAE8CUs1gOiQtfkvxKQ_-qYZ15st84qA3RU/setWebhook

Status: 200 OK Time: 4130 ms

```
{
    "ok": true,
    "result": true,
    "description": "Webhook was set"
}
```

Now lets upload the code

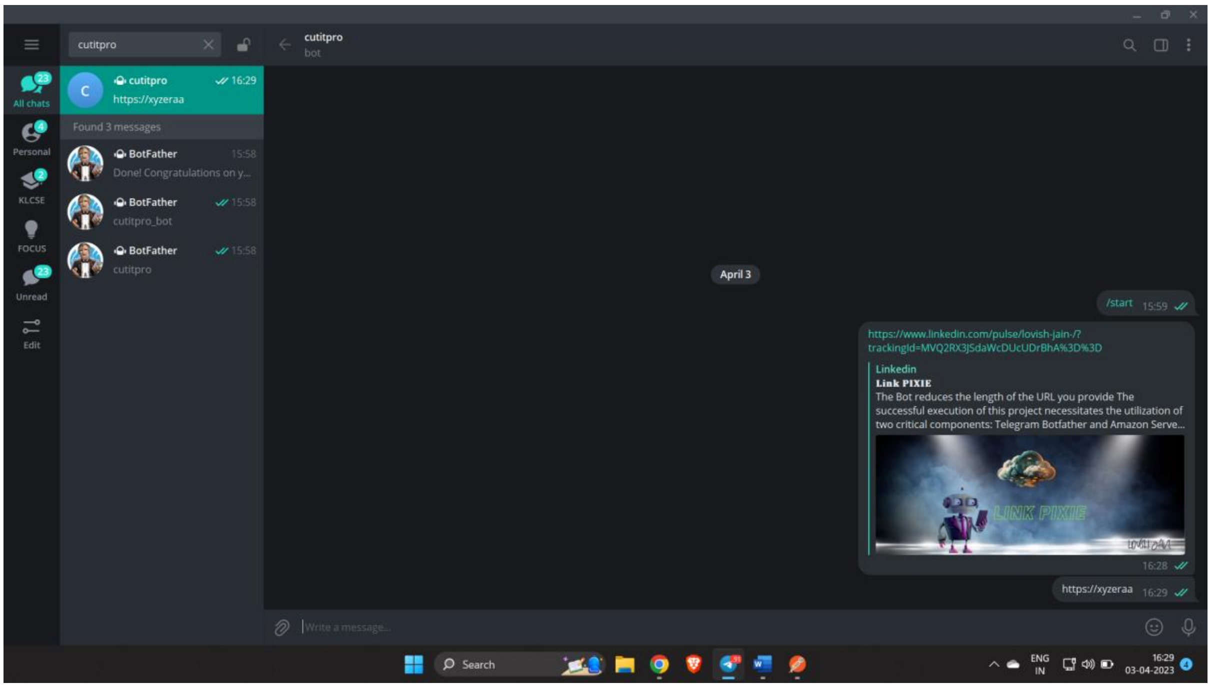


Successfully updated the function cutpro.

```
    print(body)


    message_part=body['message'].get('text')
    print("Message part : {}".format(message_part))

    data = {'url': message_part}

    payload=requests.post('https://cleanuri.com/api/v1/shorten',data=data)
    short_url=payload.json()['result_url']
    print("The short url is : {}".format(short_url))

    chat_id=body['message']['chat']['id']

    url = f'https://api.telegram.org/bot6048943402:AAE8CUs1gOiQtfkvxKQ_-qYZ15st84qA3RU/sendMessage'
    payload = {
                'chat_id': chat_id,
                'text': short_url
                }

    r = requests.post(url,json=payload)
```

By completing the previous steps, we have set up the backend of our bot. Now, when you type a message to the bot, it will be sent to the API gateway through the webhook. The API gateway will then forward the event to the Lambda function for processing.

Now let’s test the bot:



cutitpro
bot

April 3

/start 15:59

https://www.linkedin.com/pulse/lovish-jain-/?trackingId=MVQ2RX3JSdaWcDUcUDrBhA%3D%3D

Linkedin
Link PIXIE
The Bot reduces the length of the URL you provide The successful execution of this project necessitates the utilization of two critical components: Telegram Botfather and Amazon Serve...
16:28

https://xyzeraa 16:29