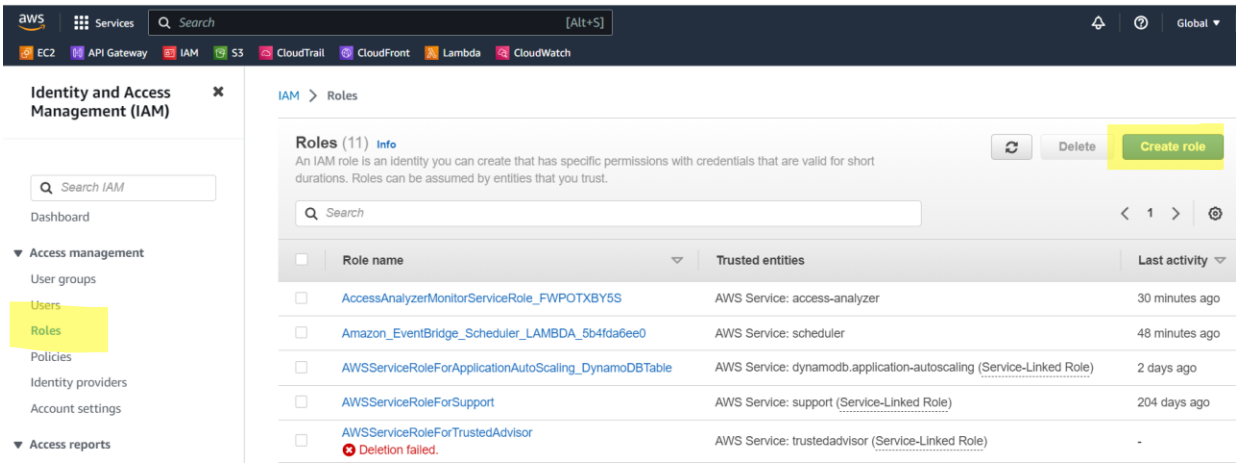
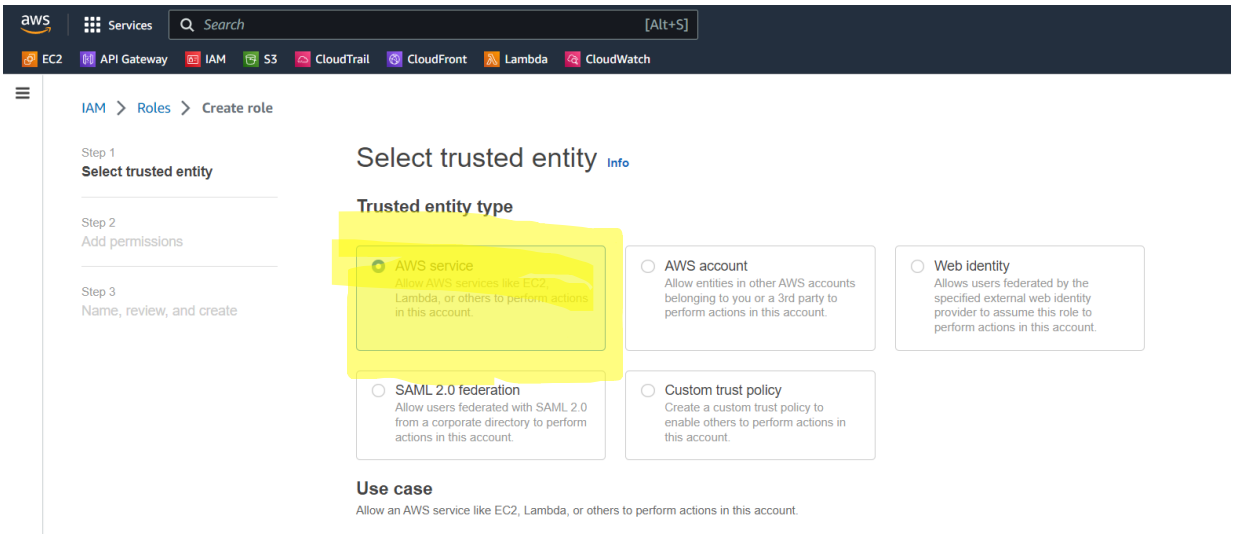
Create a lambda function to notify on slack channel with webhook of slack channel with Event Bridge and IAM Role

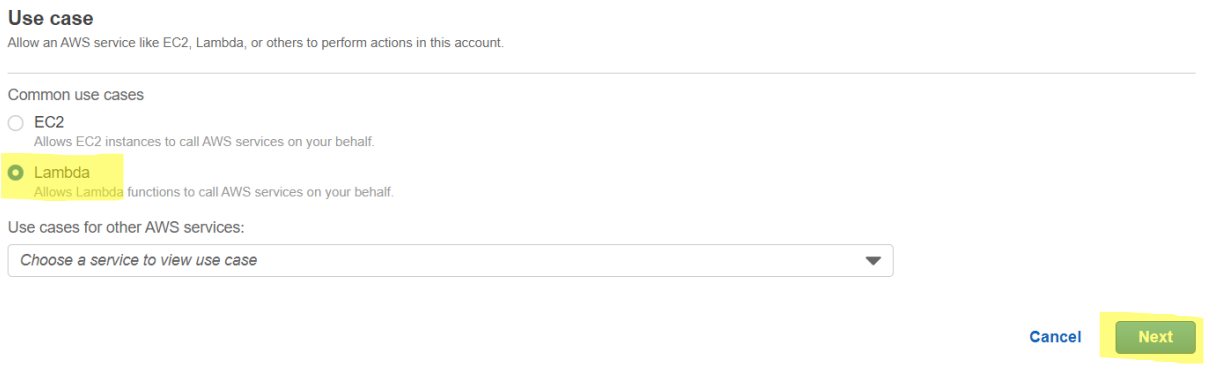
- Just create a IAM role first



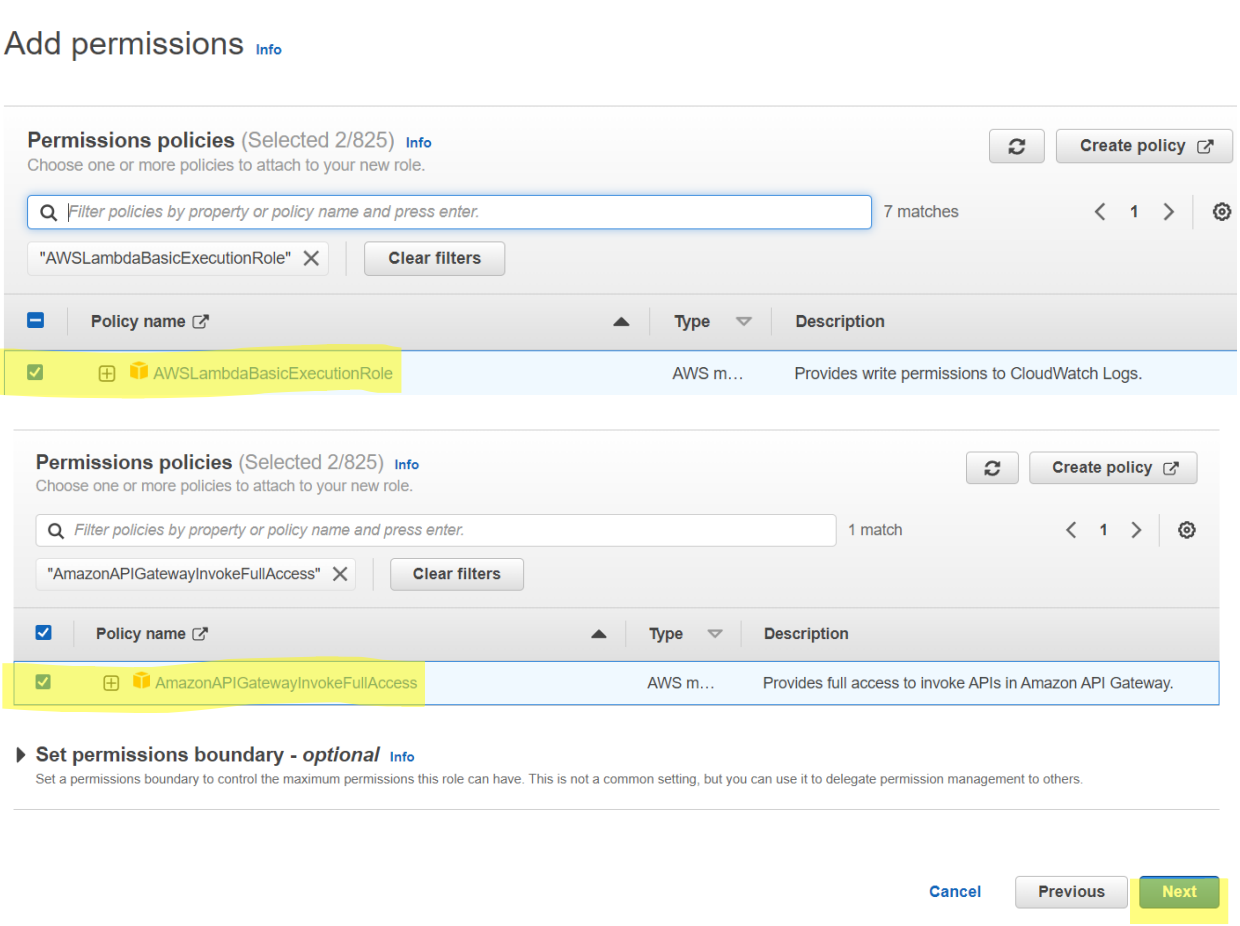
- Just open the IAM role interface and then click on the **create role**



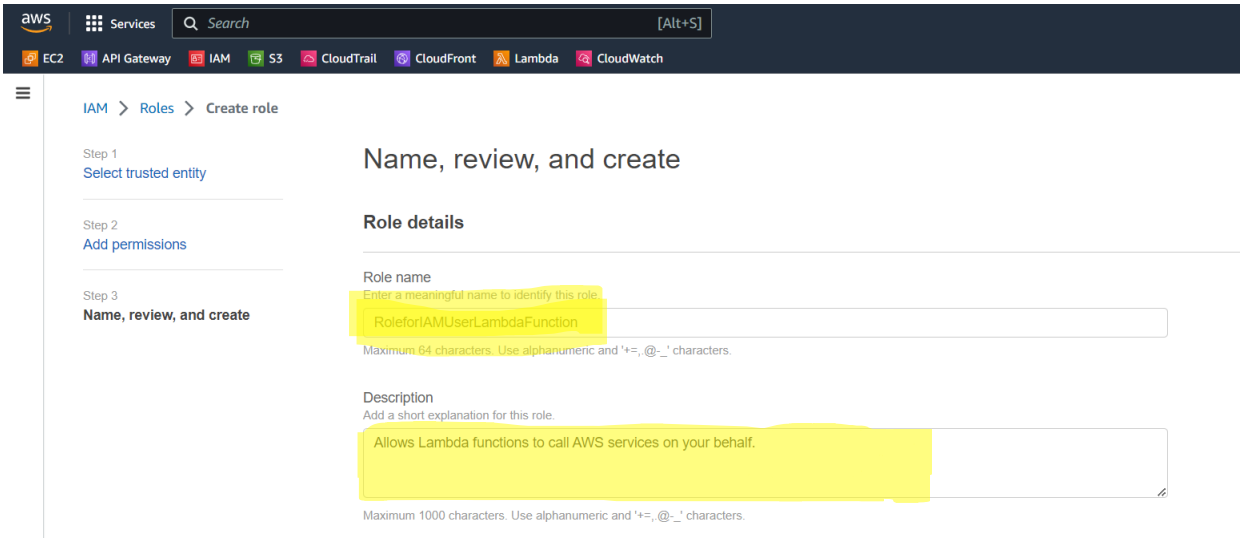
Make sure you have selected the **AWS service**



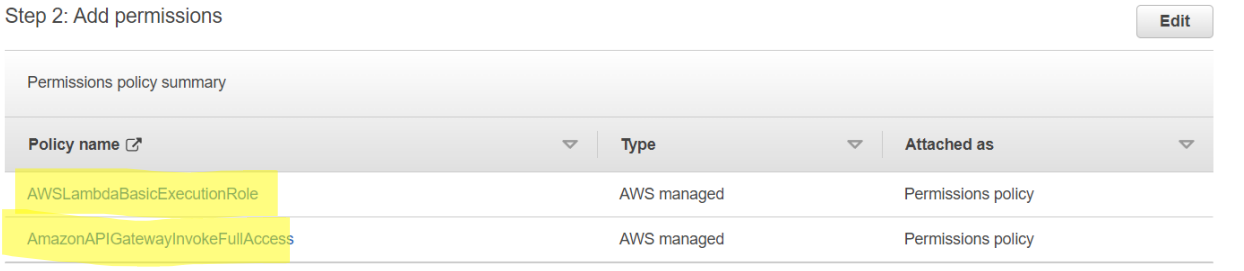
- Now you have to click on the lambda option and then click on the **NEXT** button for the next configuration
-



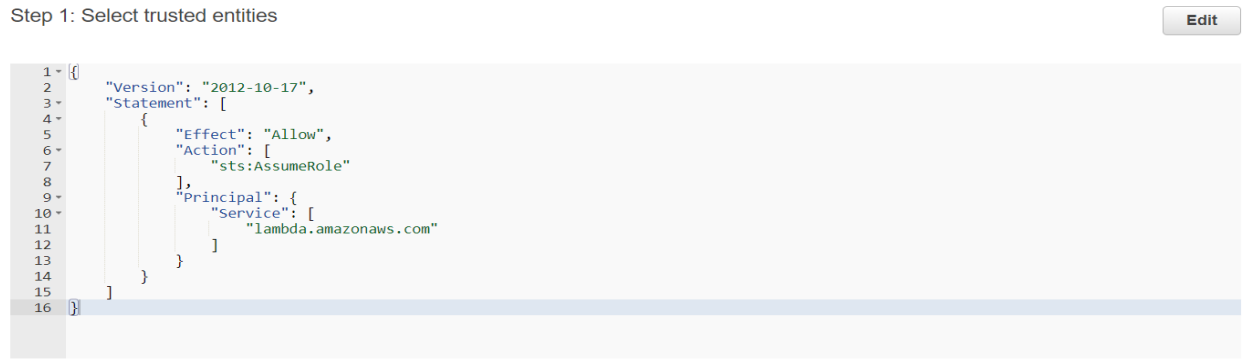
- Add the above permission and the click the next button for the further steps

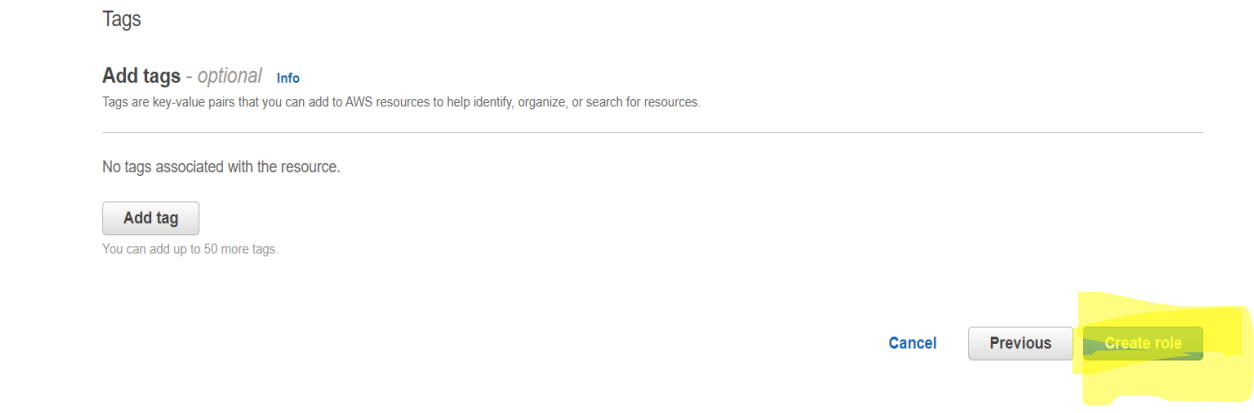


- Give the name and description of the role like the above.



-



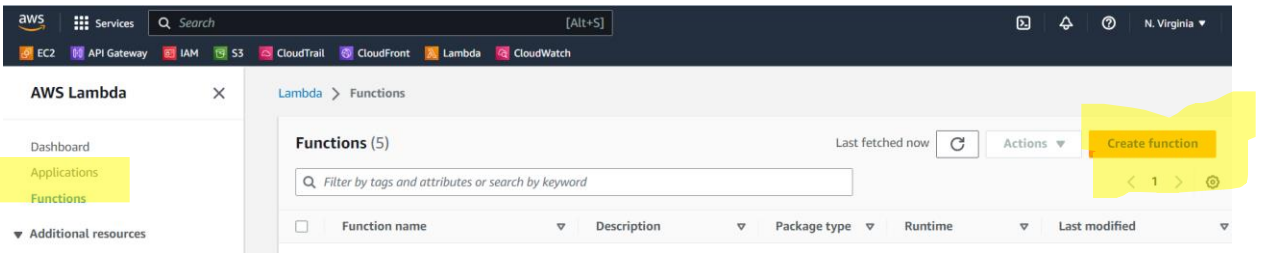


- Check the above steps and also after checking all the steps do click on the Create role button.

ALL SET ABOUT THE IAM ROLE

Now We have to create a Lambda with the follow steps:

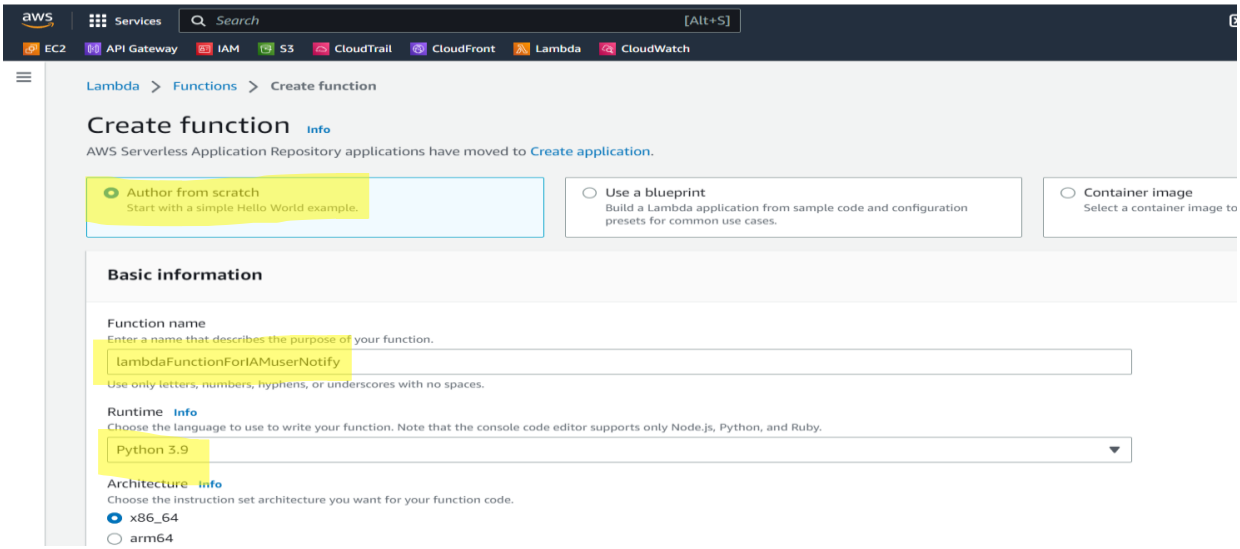
- First step is that we have to open the lambda service on aws platform.
- Just click on the create function.



- After click on the **create function** and then you should select **the Author from Scracth**

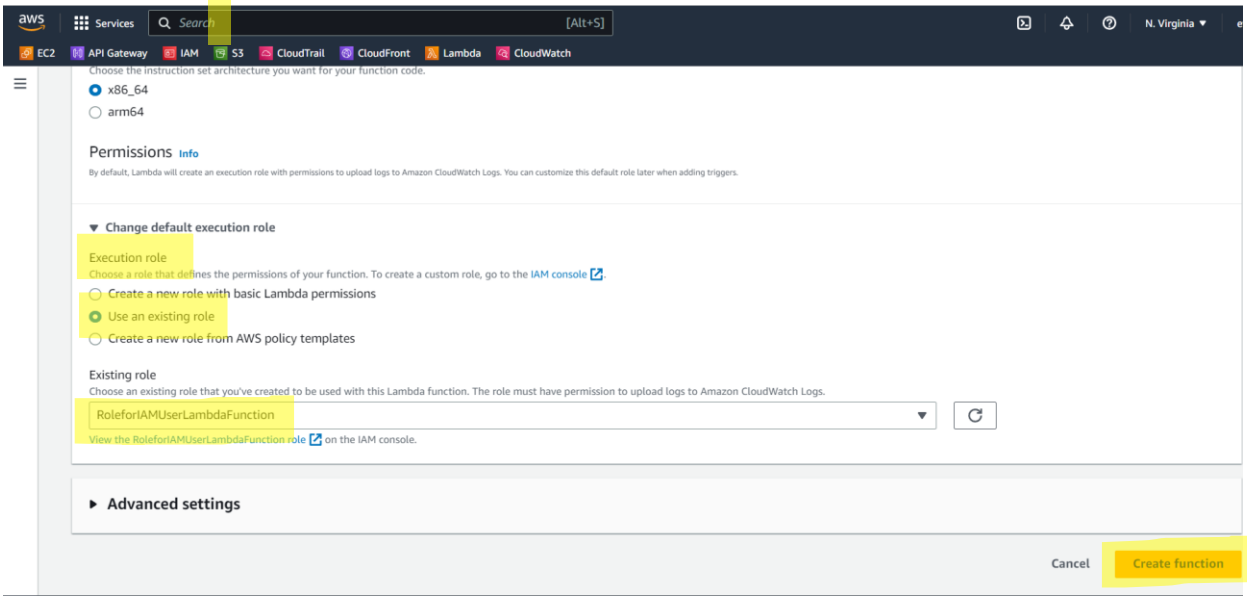
- Give your **lambda function name** for your unique identify purpose.

- Also select the runtime language i.e**. PYTHON 3.9**

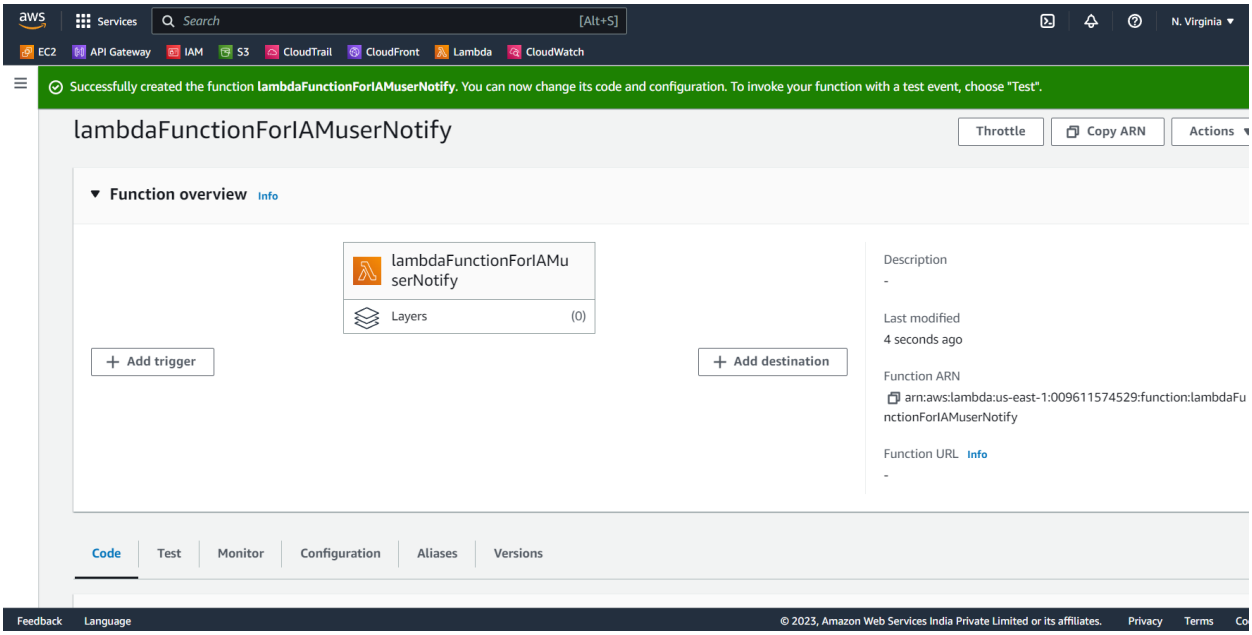


Now you can scroll down

- **Make sure you** have selected the option i.e. **Use an existing role**.
- And select your **existing role** which you have created you just before few time name as **RoleforIAMUserLambdaFunction**.



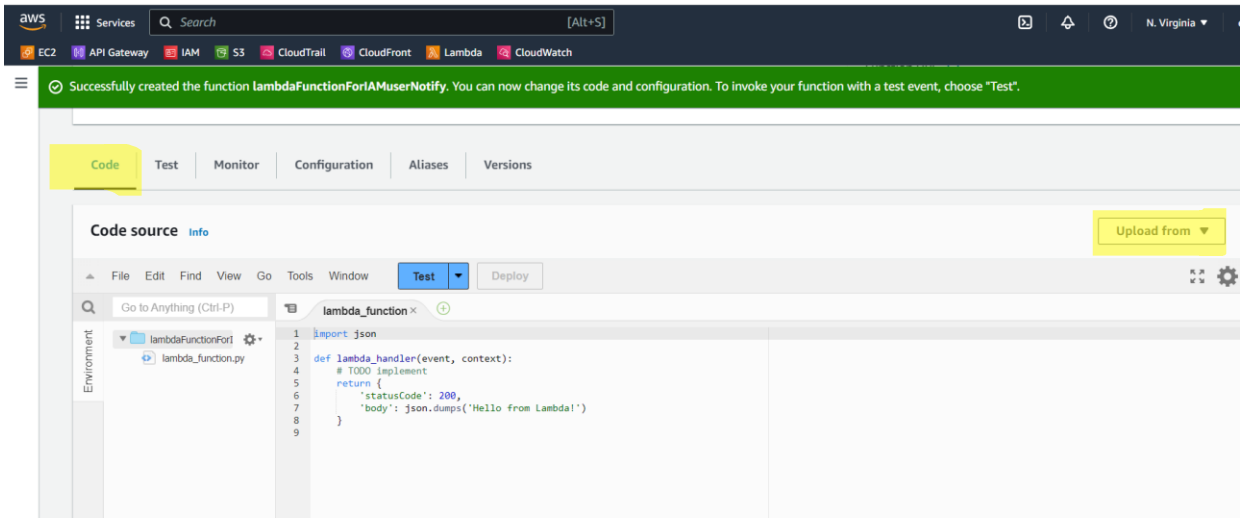
- **Finally click on the create function button.** When the function is created successfully then it shows like that the below screenshot:

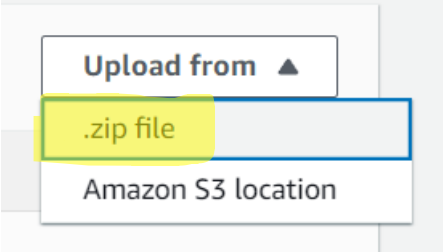


ALL SET ABOUT THE LAMBDA CREATED

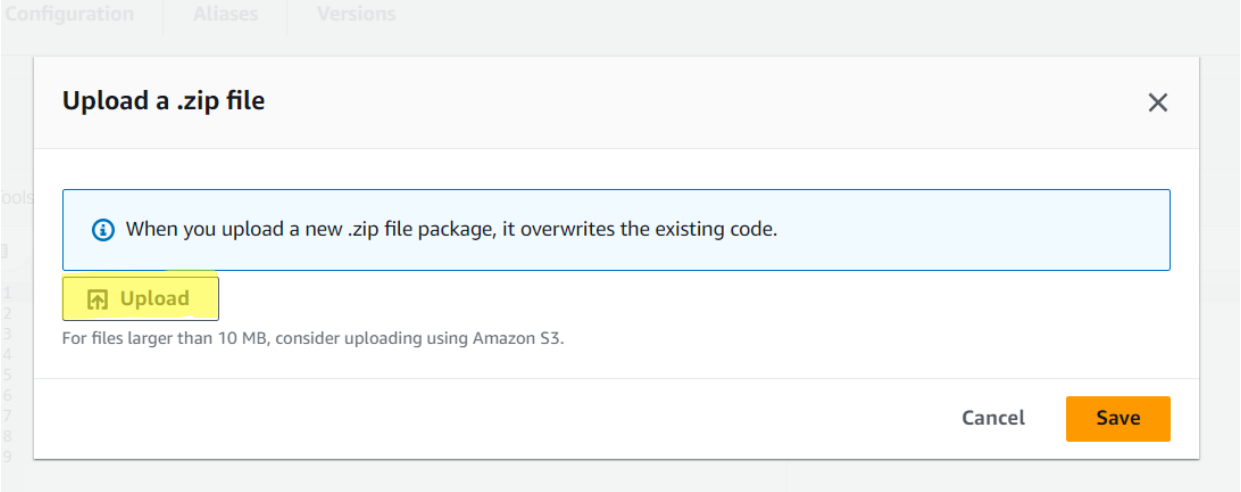
Now We have to create a Lambda Function code in python with the follow steps:

- **First** you can just click on the **upload from** button on the right side

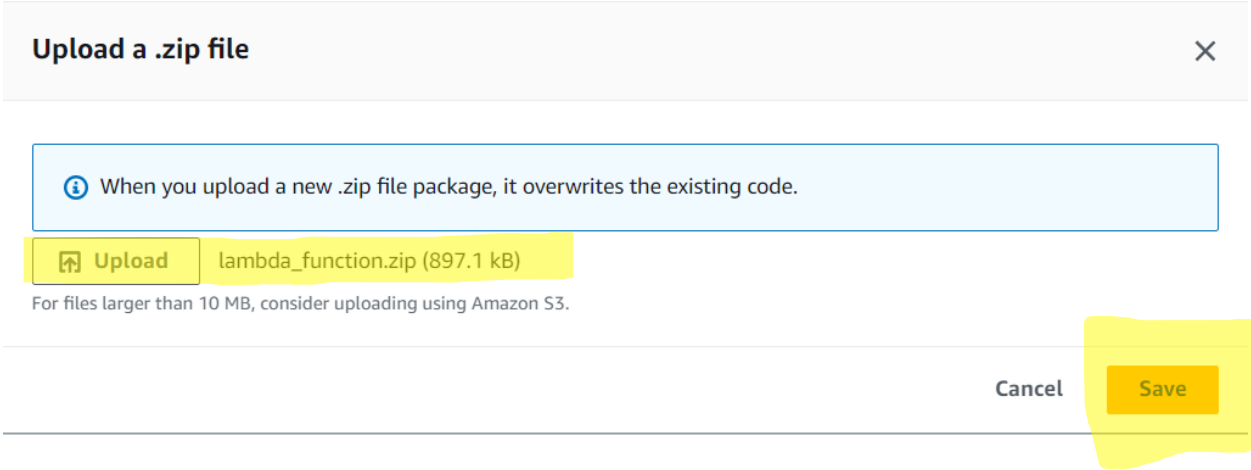




- Just click on .zip file
- And then press upload and upload the file which was given in the provided link.

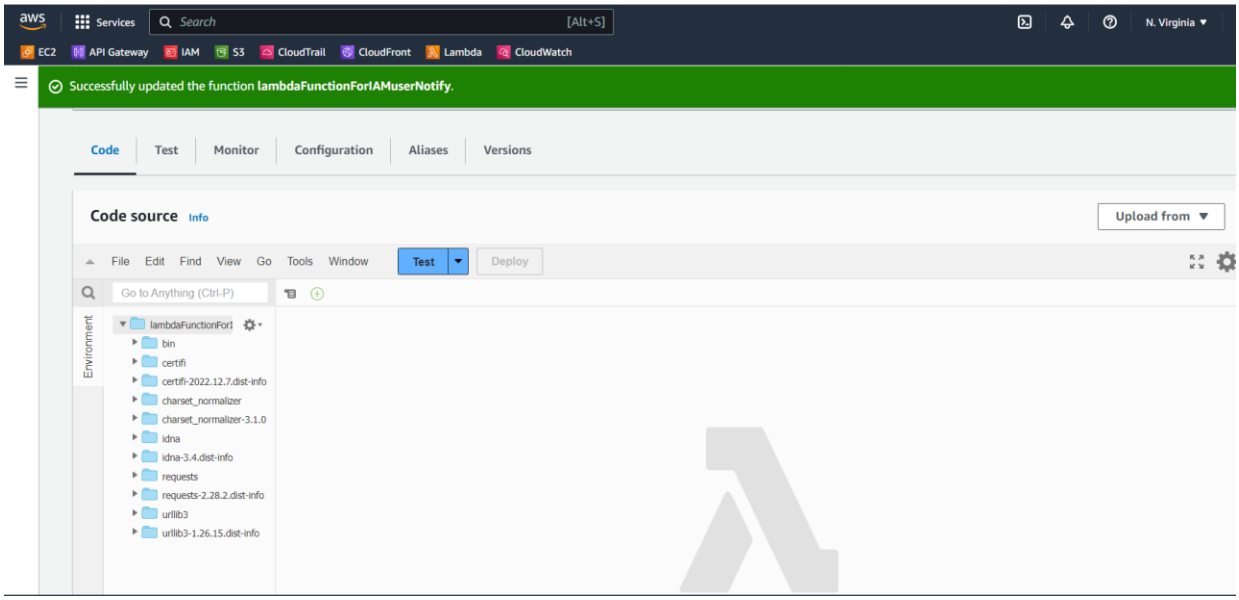


Code: https://drive.google.com/drive/folders/1BiltuSGWxPeD0QDDTmNEI7vJ21p8fMdq?usp=share_link

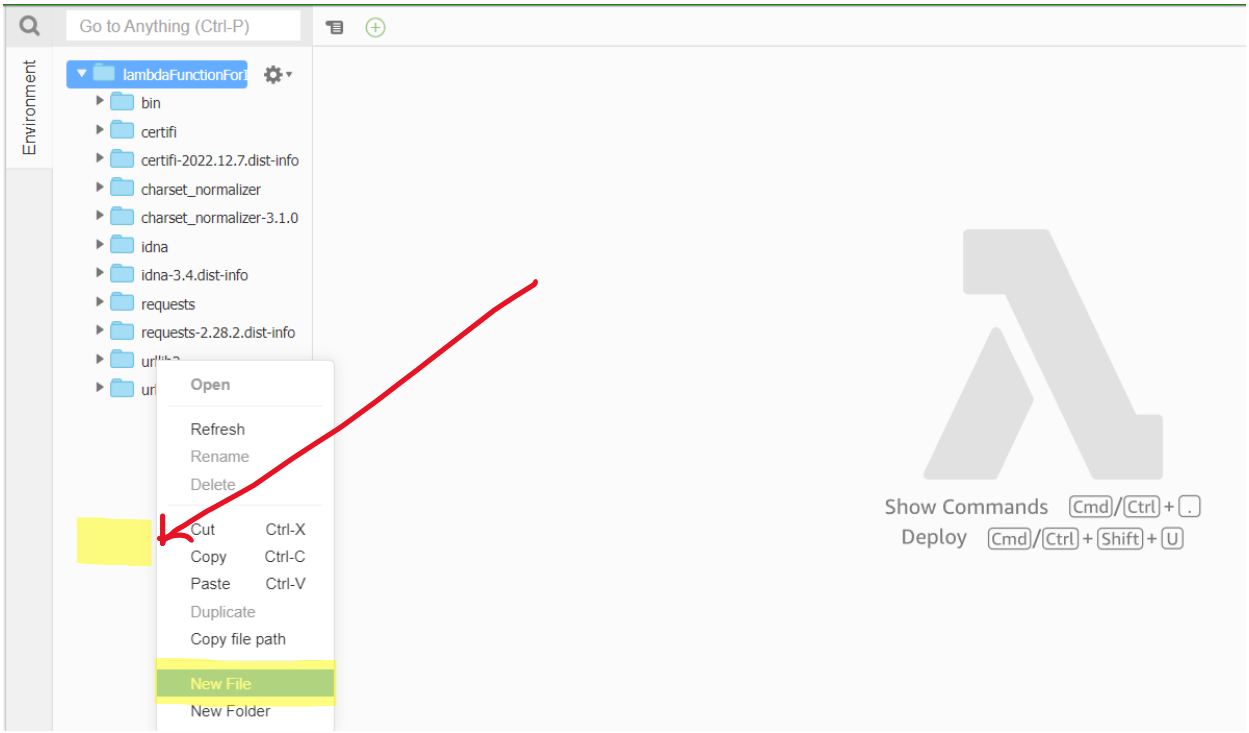


- Finally click on the **save button**

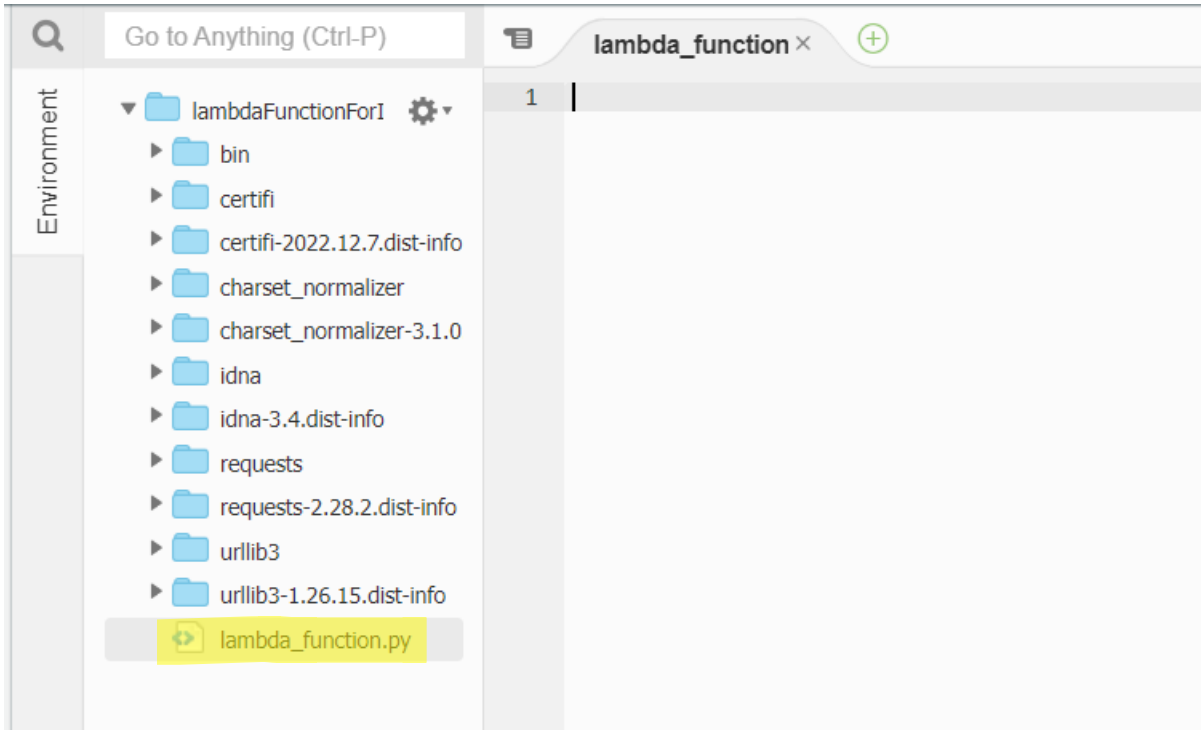
- After upload the file there it look like that



Now you have to create a file here with name as **lambda_function.py**



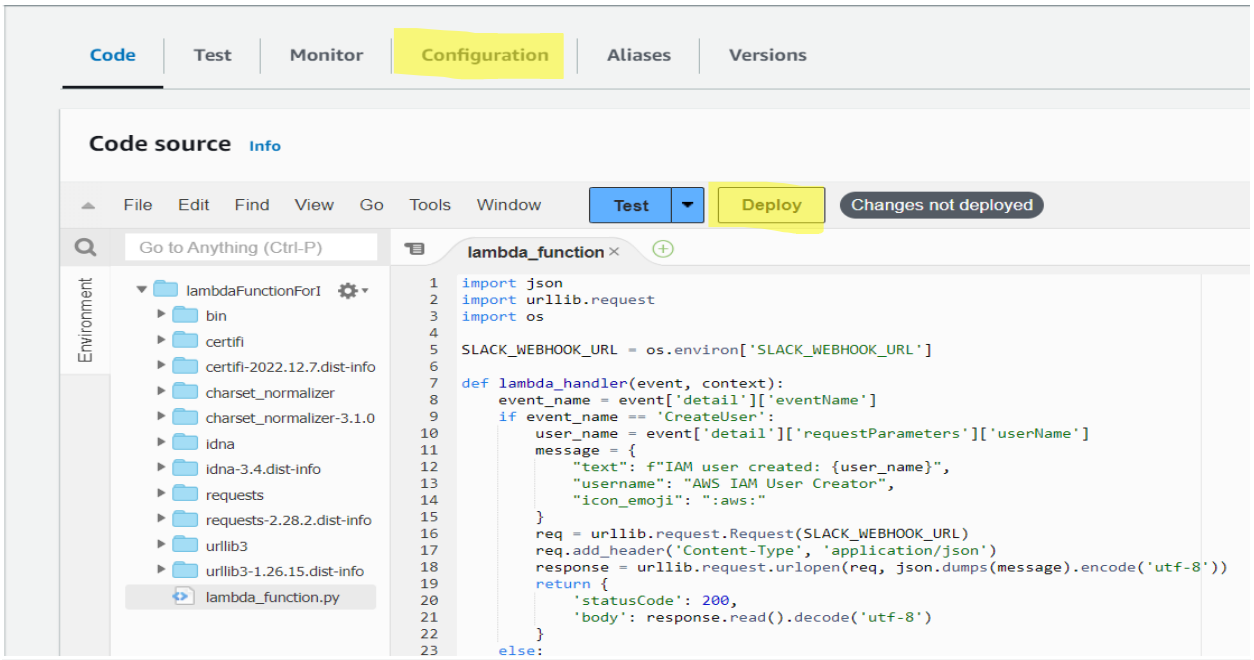
- First just right click on the **empty space where the red arrow shows the location** and then select on the **new file** and named as **lambda_function.py**

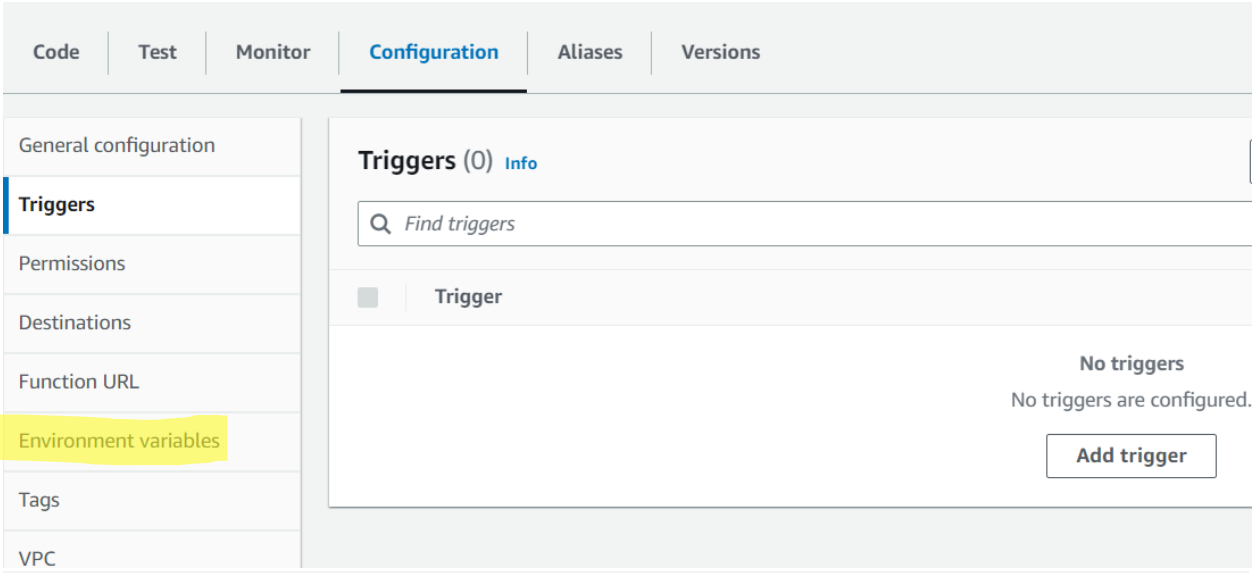


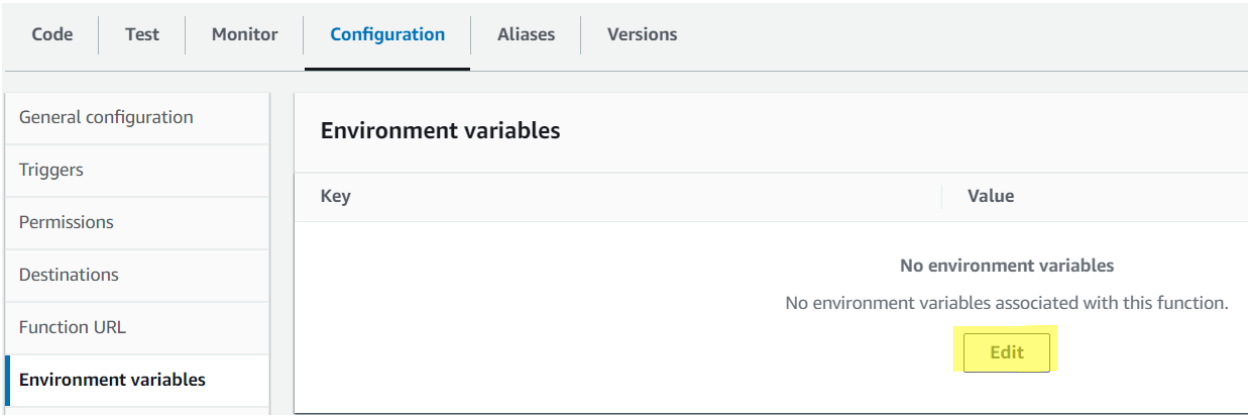
Now just copy and paste the code for lambda function

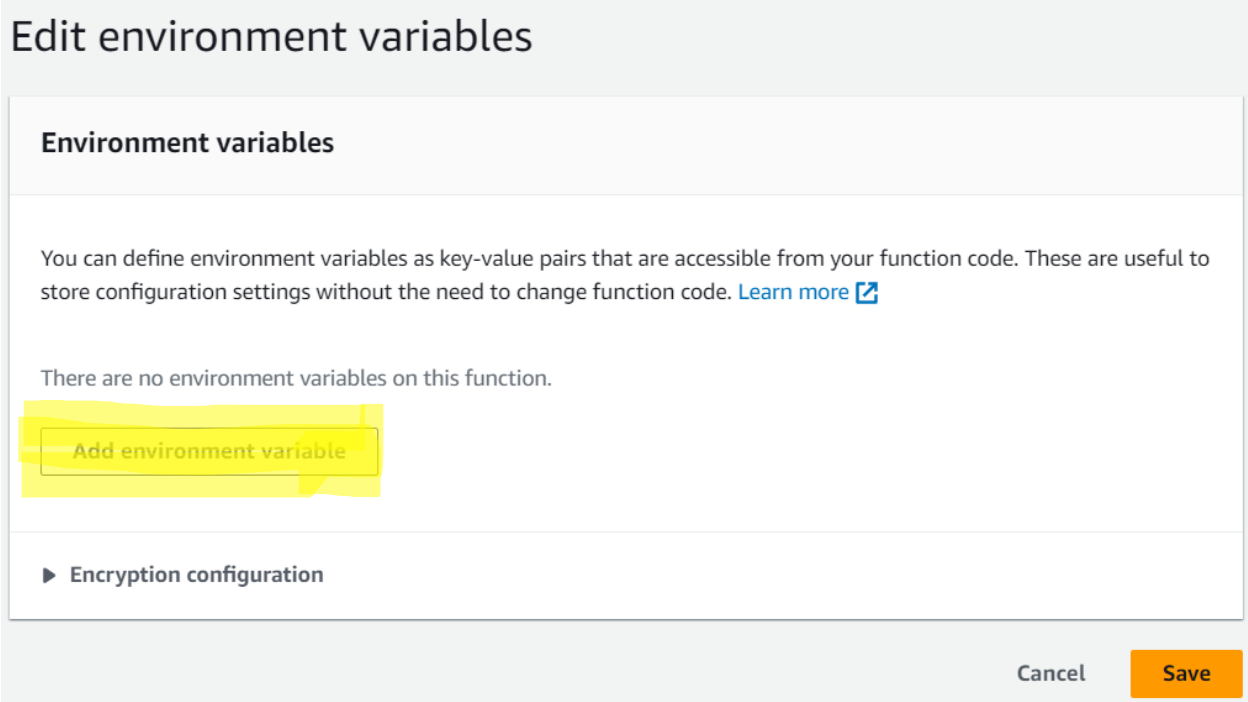
```
import json
import urllib.request
import os

SLACK_WEBHOOK_URL = os.environ['SLACK_WEBHOOK_URL']

def lambda_handler(event, context):
    event_name = event['detail']['eventName']
    if event_name == 'CreateUser':
        user_name = event['detail']['requestParameters']['userName']
        message = {
            "text": f"IAM user created: {user_name}",
            "username": "AWS IAM User Creator",
            "icon_emoji": ":aws:"
        }
        req = urllib.request.Request(SLACK_WEBHOOK_URL)
        req.add_header('Content-Type', 'application/json')
        response = urllib.request.urlopen(req,
json.dumps(message).encode('utf-8'))
        return {
            'statusCode': 200,
            'body': response.read().decode('utf-8')
        }
    else:
        return {
            'statusCode': 200,
            'body': 'Not a CreateUser event'
        }
```

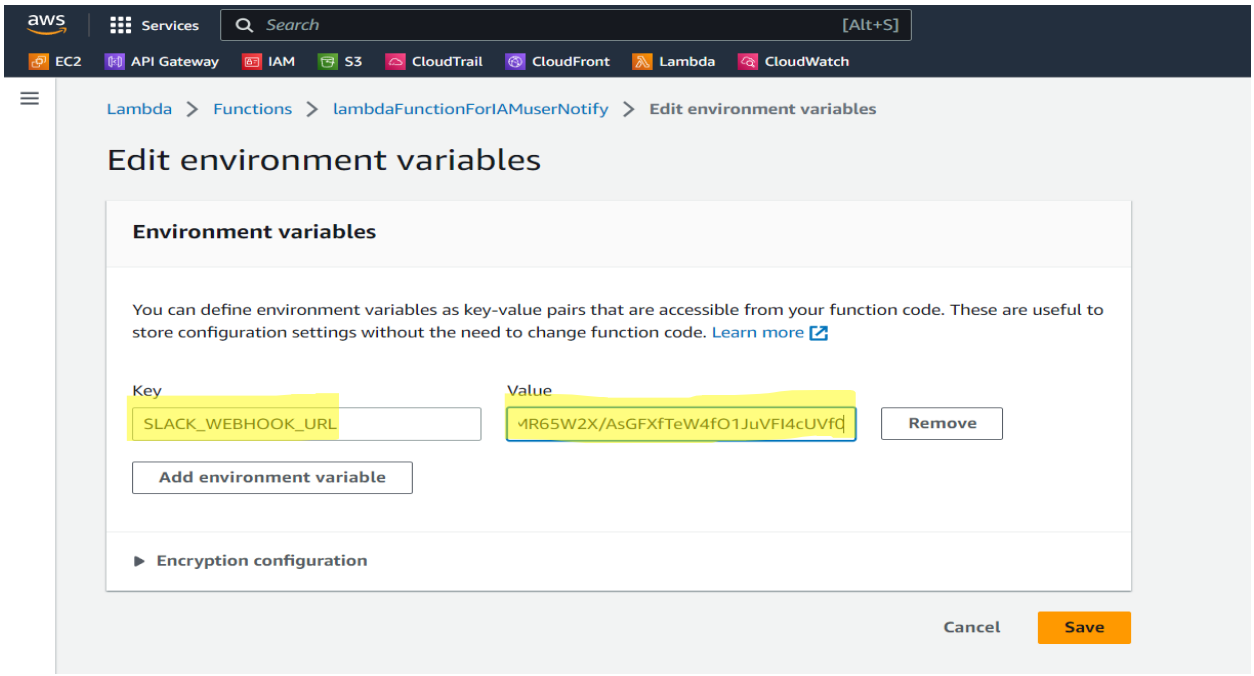
- Click on the **deploy button** and then click on the **configuration button** and then click on the **evironmental button** → add the **environmental variable**.


Code
Test
Monitor
Configuration
Aliases
Versions
Code source Info
File Edit Find View Go Tools Window
Test
Deploy
Changes not deployed
Go to Anything (Ctrl-P)
lambda_function
Environment
lambdaFunctionForI
bin
certifi
certifi-2022.12.7.dist-info
charset_normalizer
charset_normalizer-3.1.0
idna
idna-3.4.dist-info
requests
requests-2.28.2.dist-info
urllib3
urllib3-1.26.15.dist-info
lambda_function.py
import json
import urllib.request
import os
SLACK_WEBHOOK_URL = os.environ['SLACK_WEBHOOK_URL']
def lambda_handler(event, context):
event_name = event['detail']['eventName']
if event_name == 'CreateUser':
user_name = event['detail']['requestParameters']['userName']
message = {
"text": f"IAM user created: {user_name}",
"username": "AWS IAM User Creator",
"icon_emoji": ":aws:"
}
req = urllib.request.Request(SLACK_WEBHOOK_URL)
req.add_header('Content-Type', 'application/json')
response = urllib.request.urlopen(req, json.dumps(message).encode('utf-8'))
return {
'statusCode': 200,
'body': response.read().decode('utf-8')
}
else:


Code
Test
Monitor
Configuration
Aliases
Versions
General configuration
Triggers
Permissions
Destinations
Function URL
Environment variables
Tags
VPC
Triggers (0) Info
Find triggers
Trigger
No triggers
No triggers are configured.
Add trigger

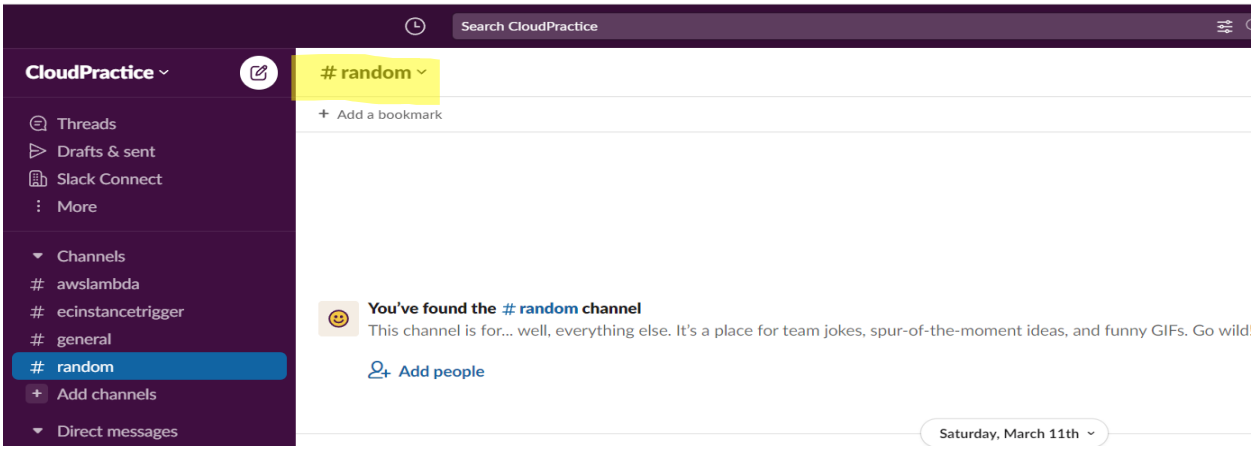

Code
Test
Monitor
Configuration
Aliases
Versions
General configuration
Triggers
Permissions
Destinations
Function URL
Environment variables
Environment variables
Key
Value
No environment variables
No environment variables associated with this function.
Edit



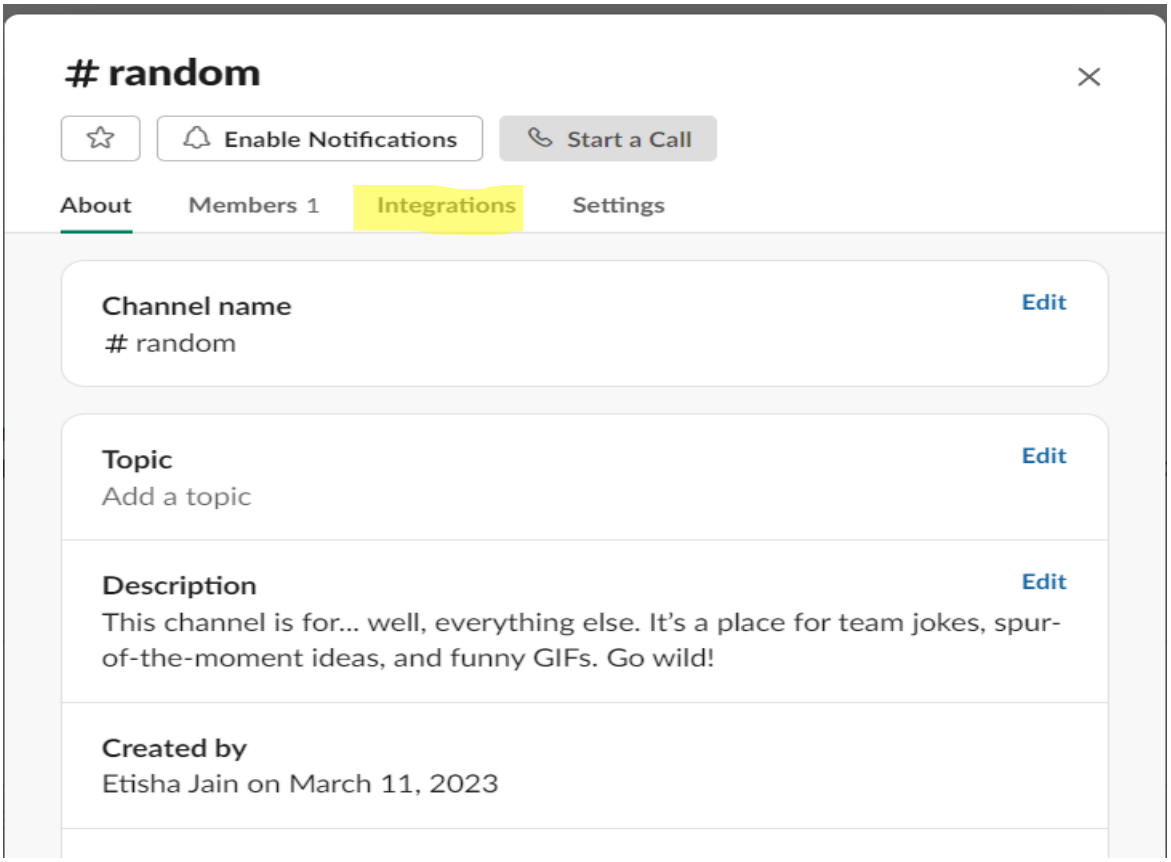


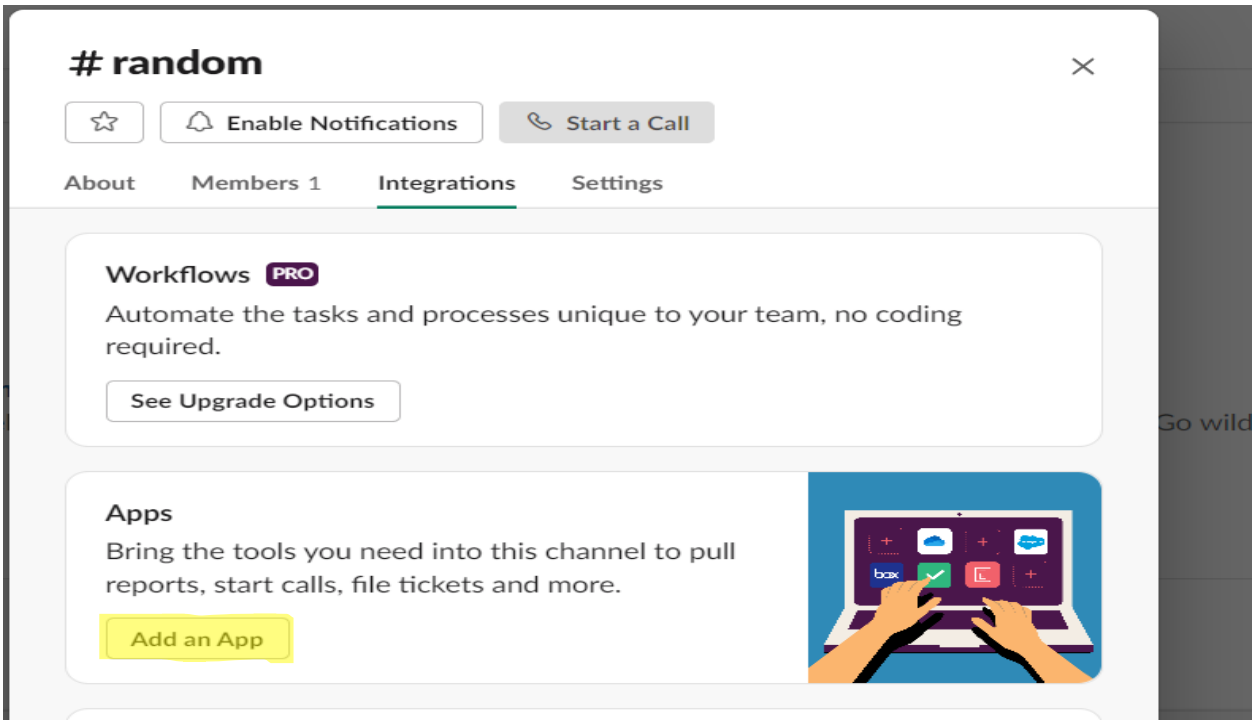
- Just give the variable name and its value i.e webhook url
- Just wait I have forgot to create a webhook url → just follow steps to create a webhook url.
- Just go inside which channel where you want to your message trigger.

WANT TO CREATE A WEBHOOK URL

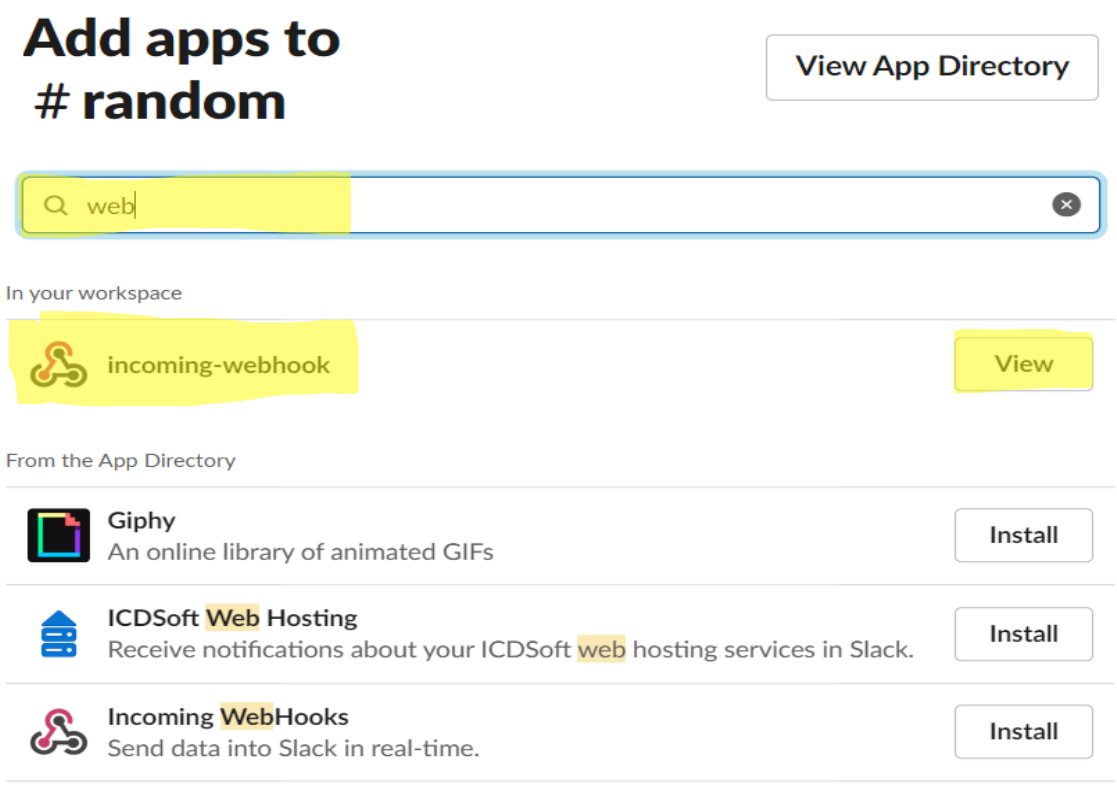


Just click on the highlight part random and just follow the highlight part.

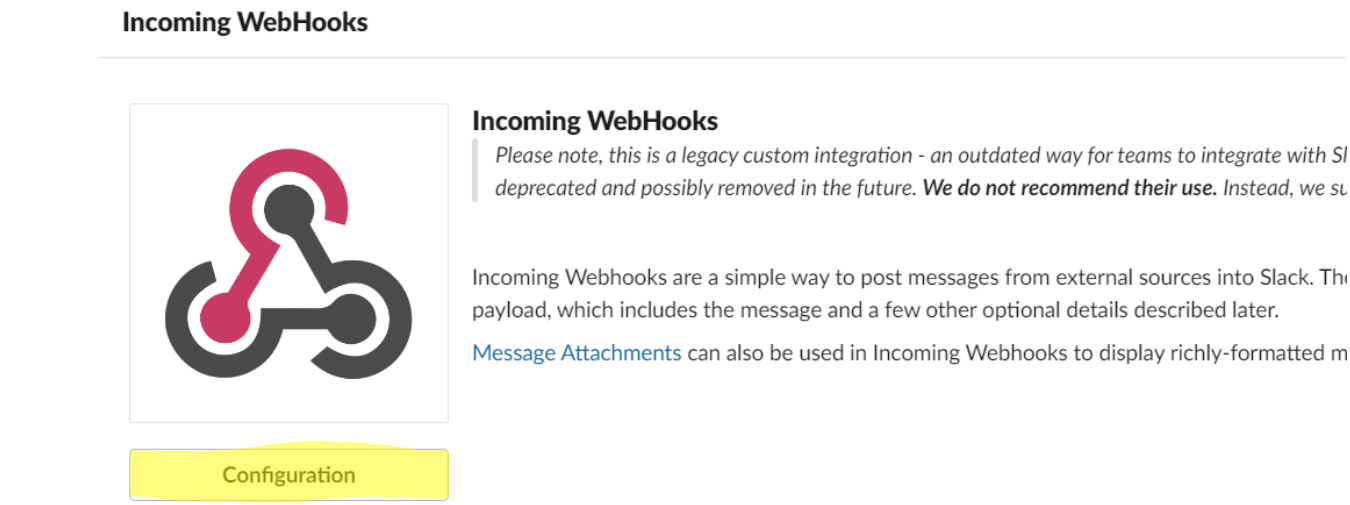




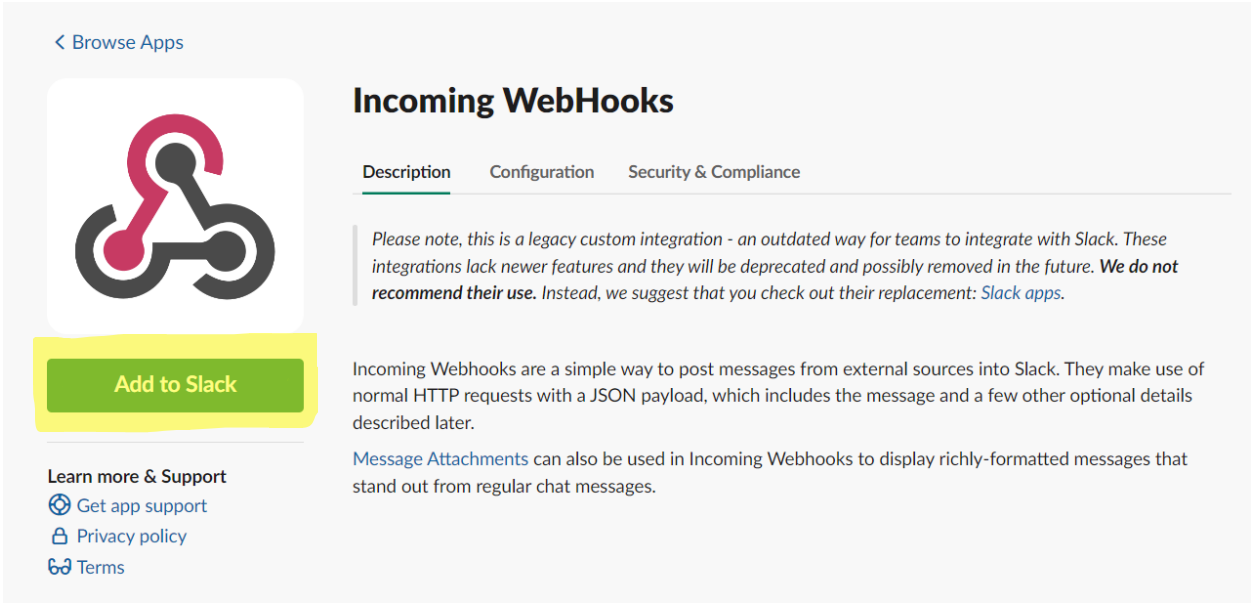
- Add an app after click on the button



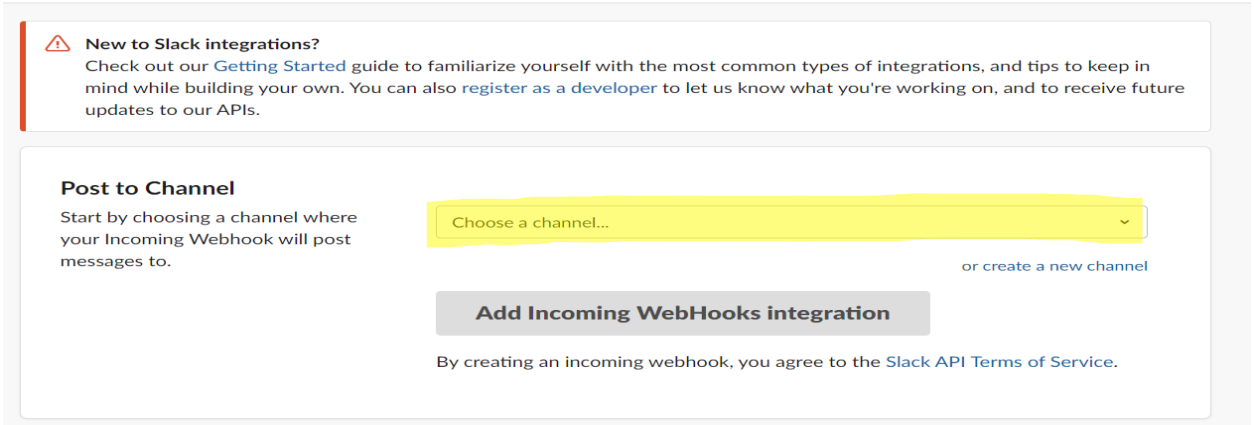
- Search the incoming-webhook install and view the hook

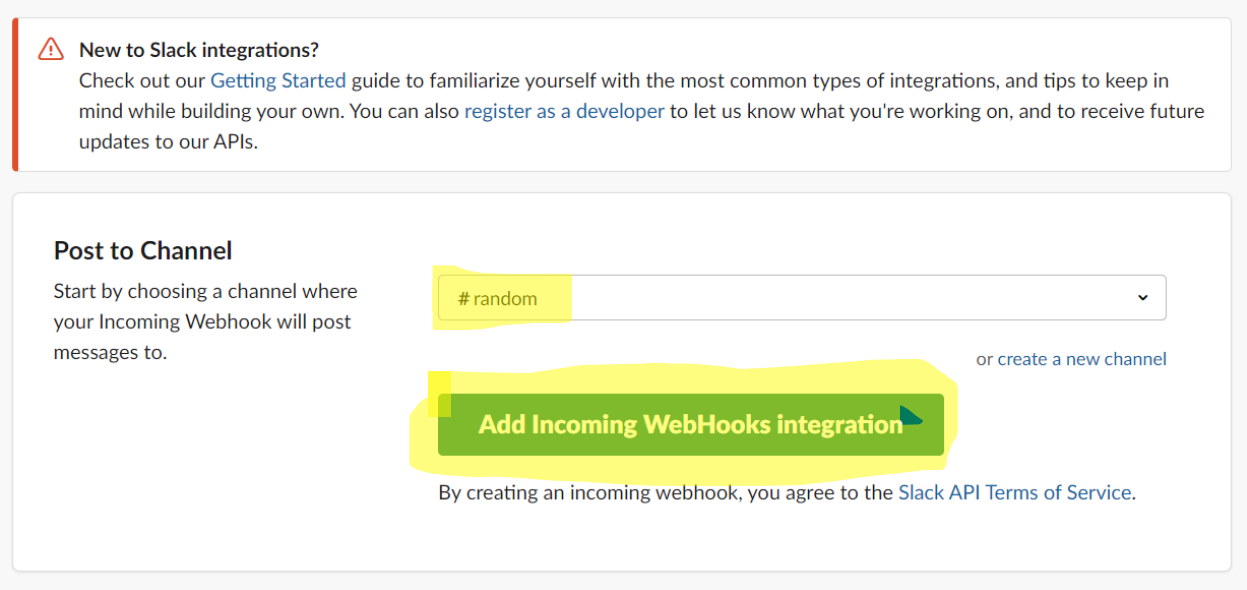


- Click on the configure button to configure the webhook url.

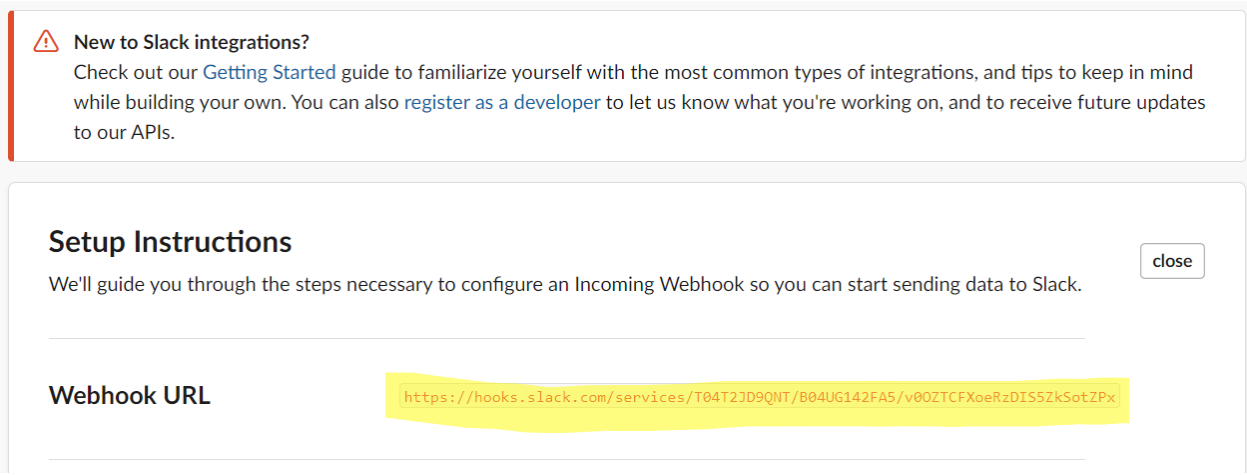


- **Just click to add to Slack button to add it and now choose your favourate channel where you want to trigger your notification message.**





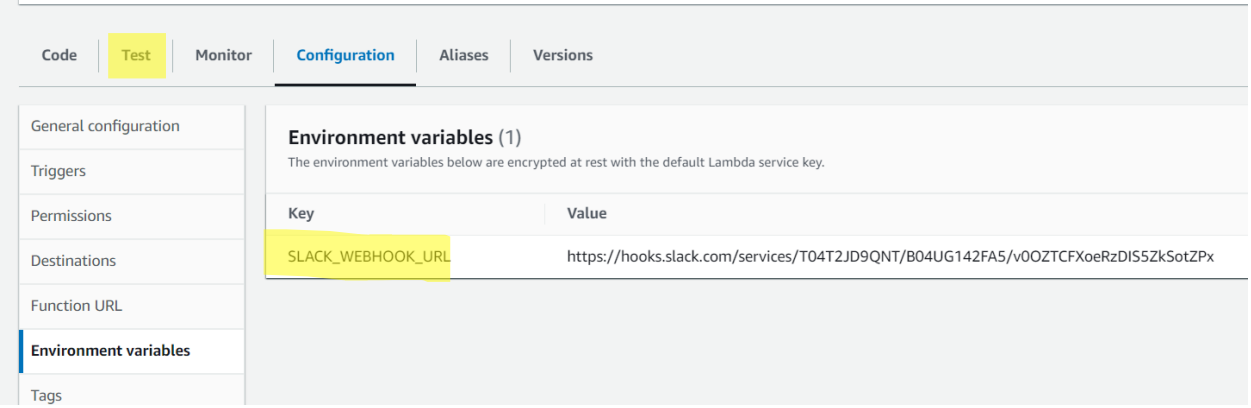
- After selected the channel random then you can add the buttom **add incoming webhooks integration.**



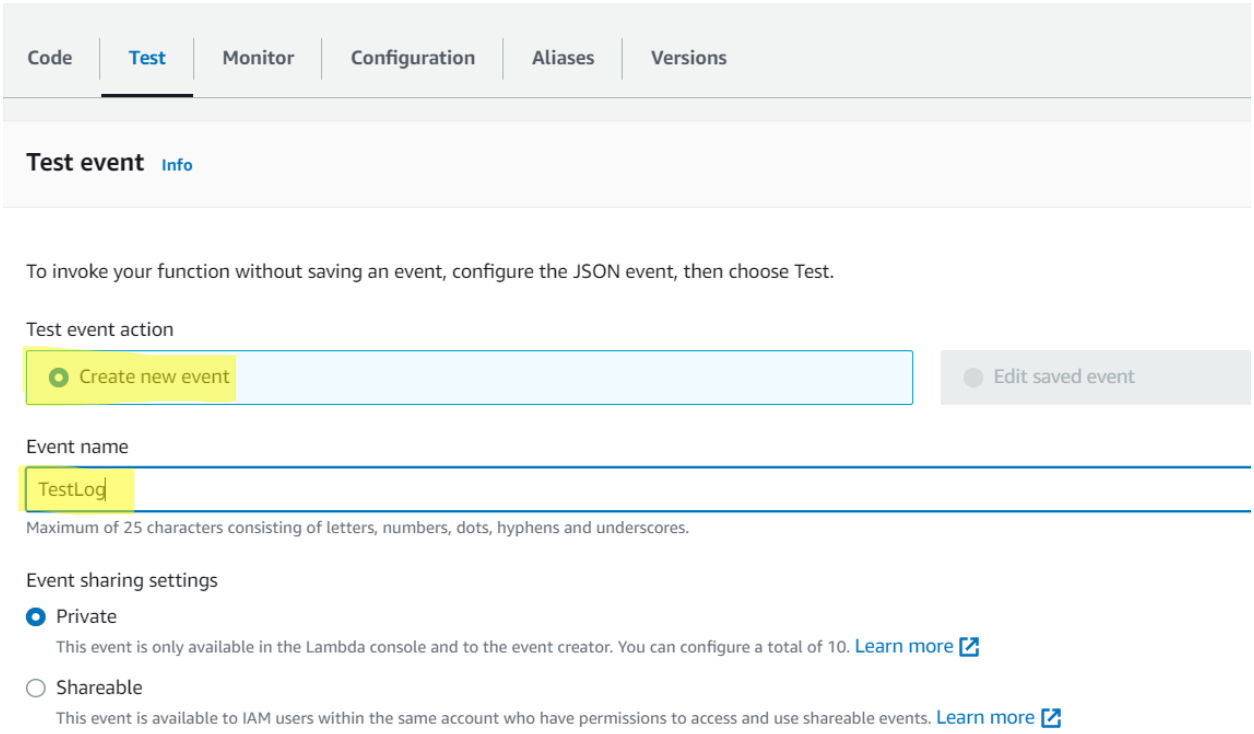
- Now **congratulation** you have created the webhook url for trigger a message through the lambda function now use the url in **lambda__function.py file.**

ALL SET THE URL IN SLACK CHANNEL FOR TRIGGER.

- After created a webhook url and save the environment
- Now have to click on the test tab

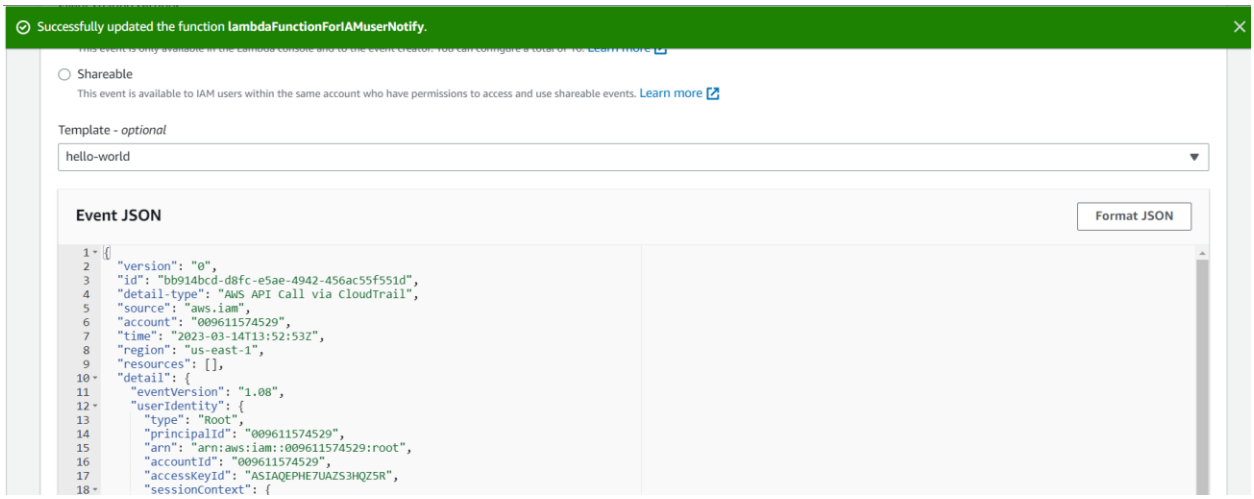


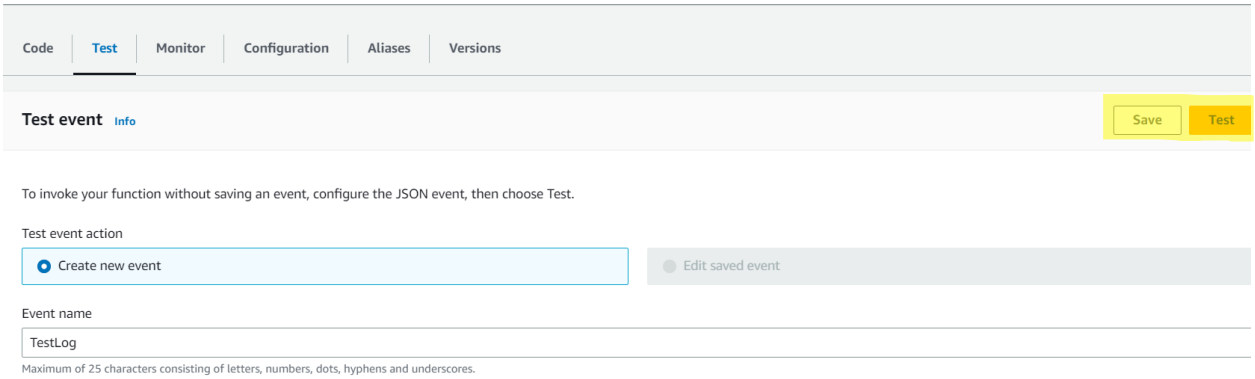
- Now have to create a new testCase named as **TestLog.**



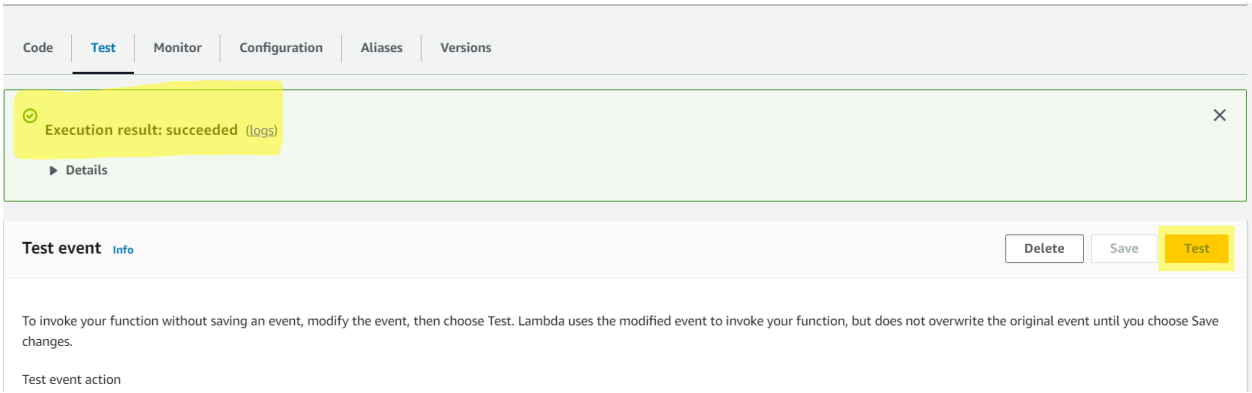
- Just scroll and paste the JSON to here and save it

```
{
   "version": "0",
   "id": "bb914bcd-d8fc-e5ae-4942-456ac55f551d",
   "detail-type": "AWS API Call via CloudTrail",
   "source": "aws.iam",
   "account": "009611574529",
   "time": "2023-03-14T13:52:53Z",
   "region": "us-east-1",
   "resources": [],
   "detail": {
      "eventVersion": "1.08",
      "userIdentity": {
         "type": "Root",
         "principalId": "009611574529",
         "arn": "arn:aws:iam::009611574529:root",
         "accountId": "009611574529",
         "accessKeyId": "ASIAQEPHE7UAZS3HQZ5R",
         "sessionContext": {
            "sessionIssuer": {},
            "webIdFederationData": {},
            "attributes": {
               "creationDate": "2023-03-14T03:53:46Z",
               "mfaAuthenticated": "false"
            }
         }
      },
      "eventTime": "2023-03-14T13:52:53Z",
      "eventSource": "iam.amazonaws.com",
      "eventName": "CreateUser",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "103.196.213.164",
      "userAgent": "AWS Internal",
      "requestParameters": {
         "userName": "demoVishu"
      },
      "responseElements": {
         "user": {
            "createDate": "Mar 14, 2023 1:52:53 PM",
            "userName": "demoVishu",
            "arn": "arn:aws:iam::009611574529:user/demoVishu",
            "path": "/",
            "userId": "AIDAQEPHE7UA22ZB5L5KL"
         }
      },
      "requestID": "010b45b2-c22b-41ea-9c2b-a29ee3316d2d",
      "eventID": "90c52c9f-f83b-4b34-a108-7589724383a1",
      "readOnly": false,
      "eventType": "AwsApiCall",
      "managementEvent": true,
      "recipientAccountId": "009611574529",
      "eventCategory": "Management",
      "sessionCredentialFromConsole": "true"
   }
}
```





- Save and test the code and check it your slack notification it send the message there.



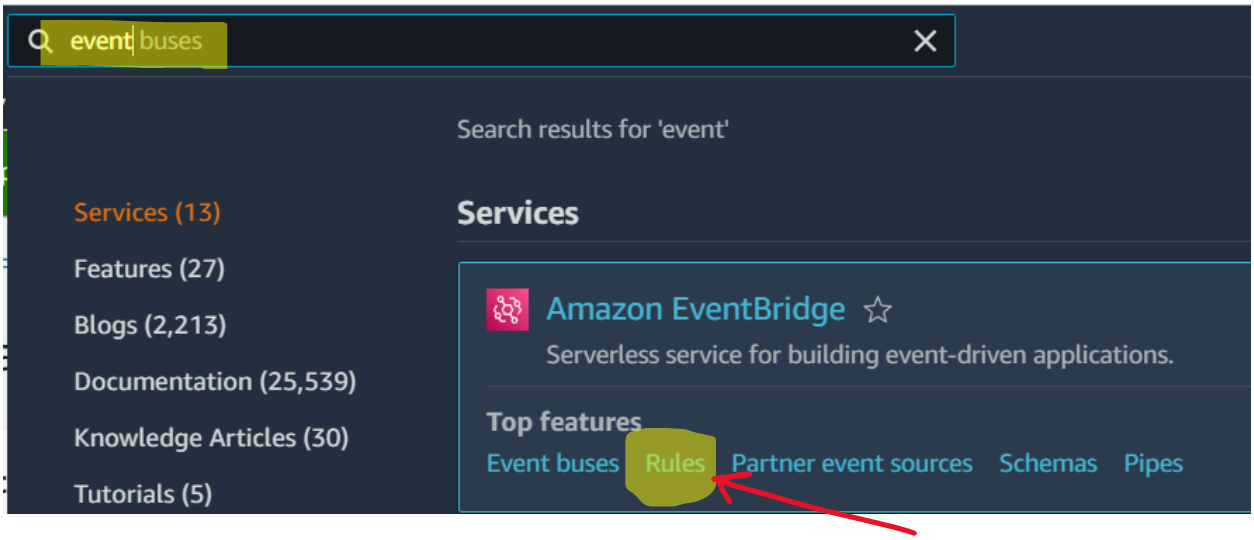
- **After save and test the JSON file then check on the slack it send the message there.**

:aws: **AWS IAM User Creator** APP 9:33 PM
IAM user created: demoVishu

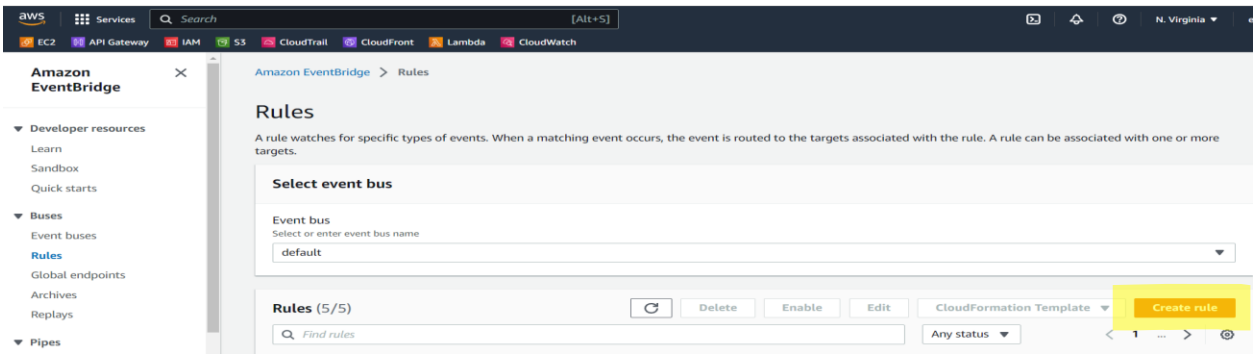
:aws: **AWS IAM User Creator** APP 9:39 PM
IAM user created: demoVishu

CONGRATULATION YOU HAVE GOT THE MESSAGE ON SLACK CHANNEL

NOW WE HAVE TO CREATE A EVENT BRIDGE
Which help use to trigger the lambda function when event occur.



- Search the **EventBridge** and just click on the **Rules**
- Just click on the **create rule**



Define rule detail Info

Rule detail

Name

SlackNotificationEventBridgeRule

Maximum of 64 characters consisting of numbers, lower/upper case letters, .,-,_.

Description - *optional*

Enter description

Event bus Info

Select the event bus this rule applies to, either the default event bus or a custom or partner event bus.

default

Enable the rule on the selected event bus

Rule type Info

- (●) Rule with an event pattern
  A rule that runs when an event matches the defined event pattern. EventBridge sends the event to the specified target.
- ( ) Schedule
  A rule that runs on a schedule

Cancel Next

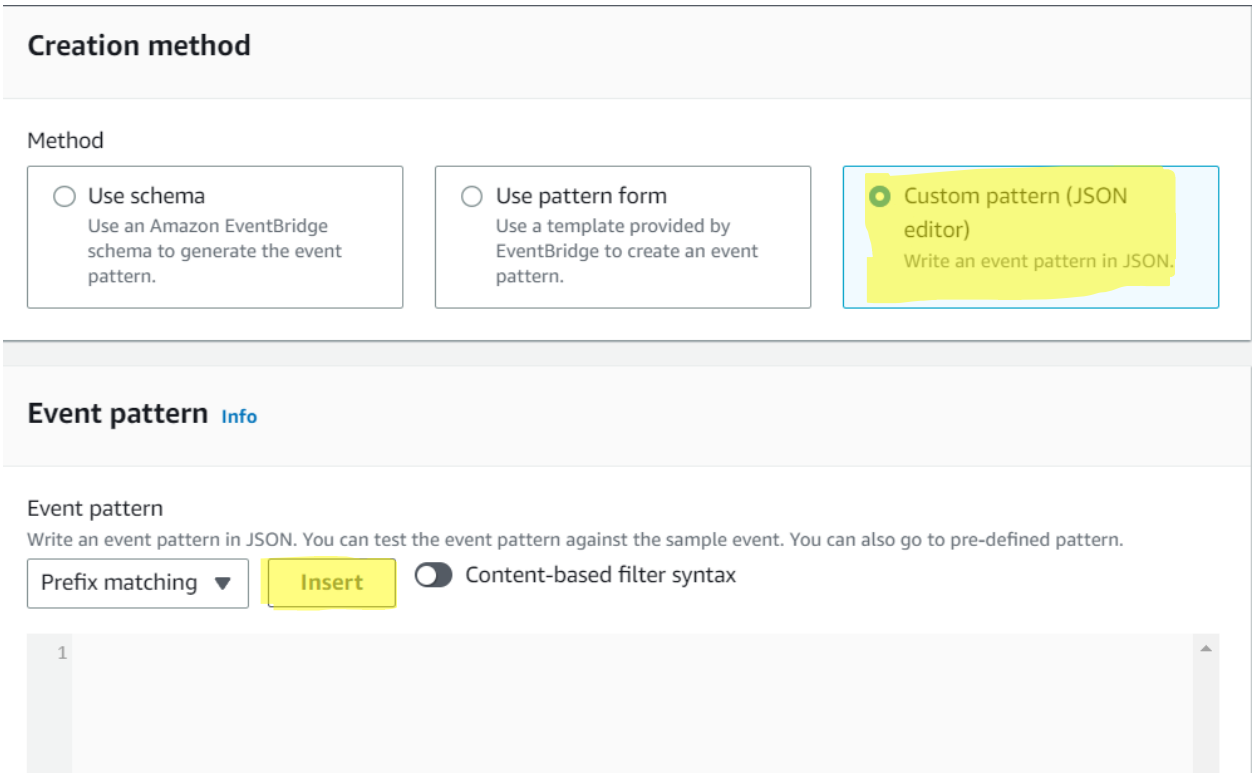
Build event pattern Info

Event source

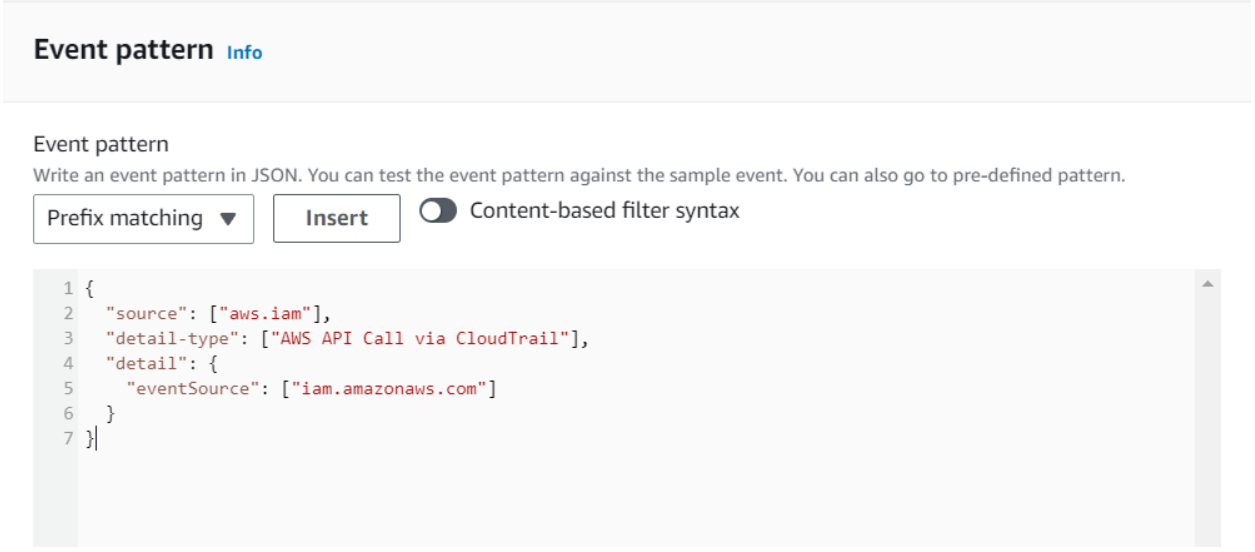
Event source

Select the event source from which events are sent.

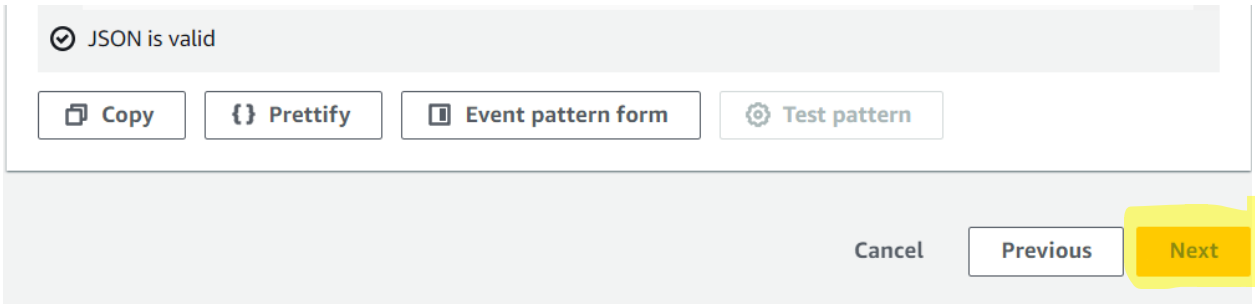
- (●) AWS events or EventBridge partner events
  Events sent from AWS services or EventBridge partners.
- ( ) Other
  Custom events or events sent from more than one source, e.g. events from AWS services and partners.
- ( ) All events
  All events sent to your account.



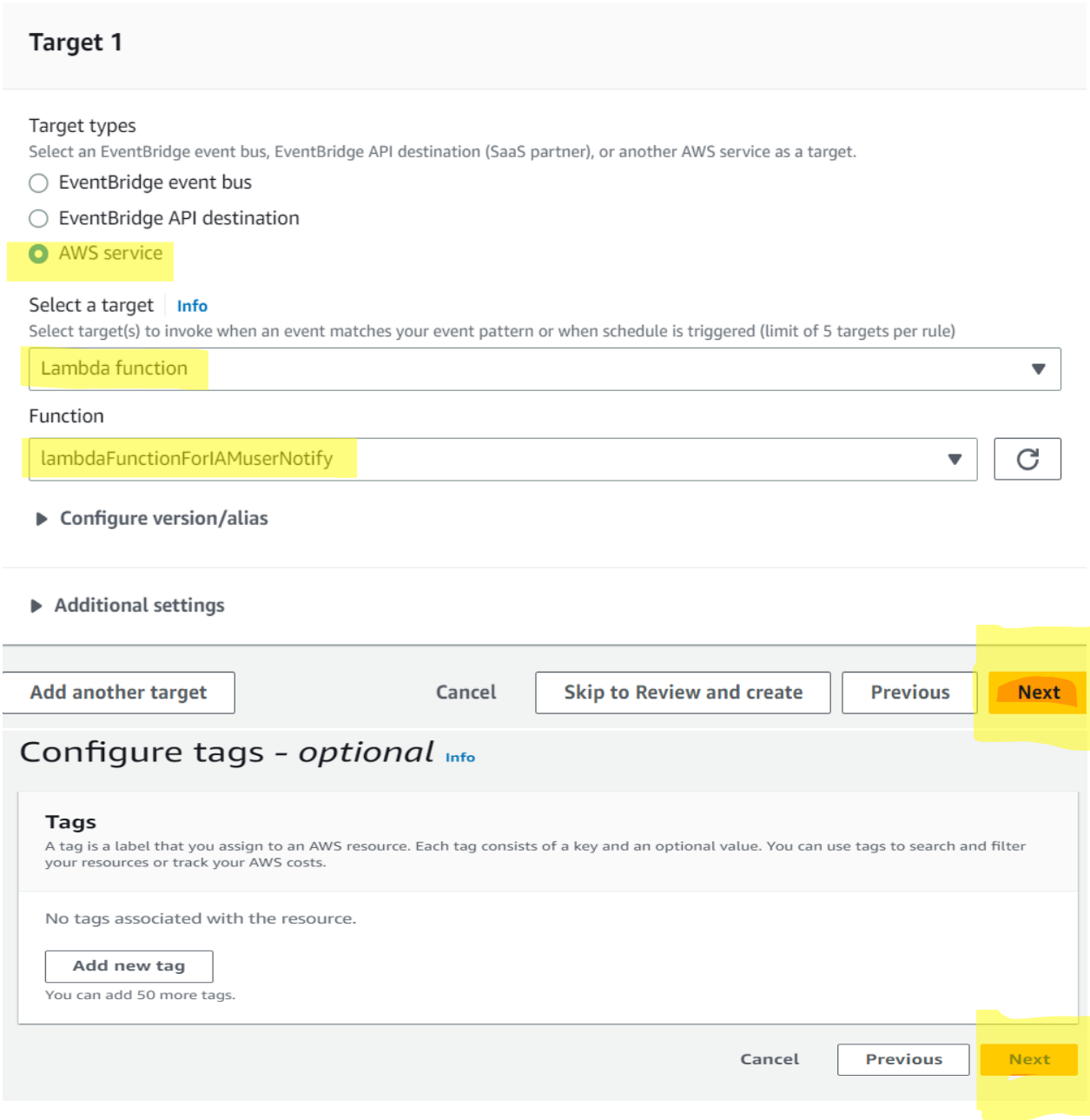
- **After click on the Insert button just insert the event patter .**




```
{
 "source": ["aws.iam"],
 "detail-type": ["AWS API Call via CloudTrail"],
 "detail": {
    "eventSource": ["iam.amazonaws.com"]
            }
 }
```

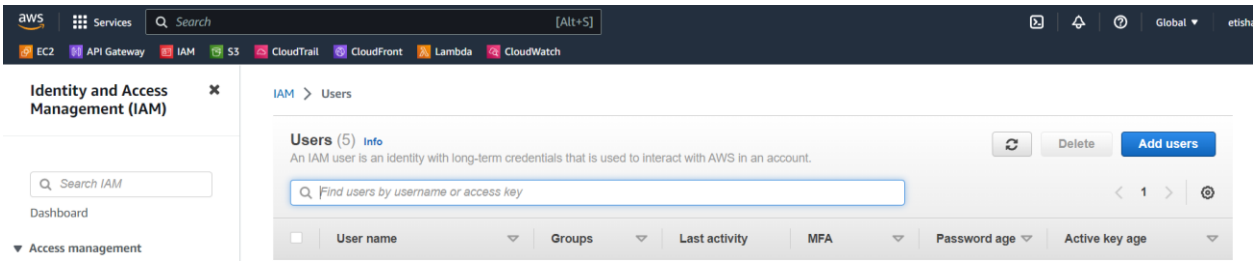


Now you have to create a NEXT button

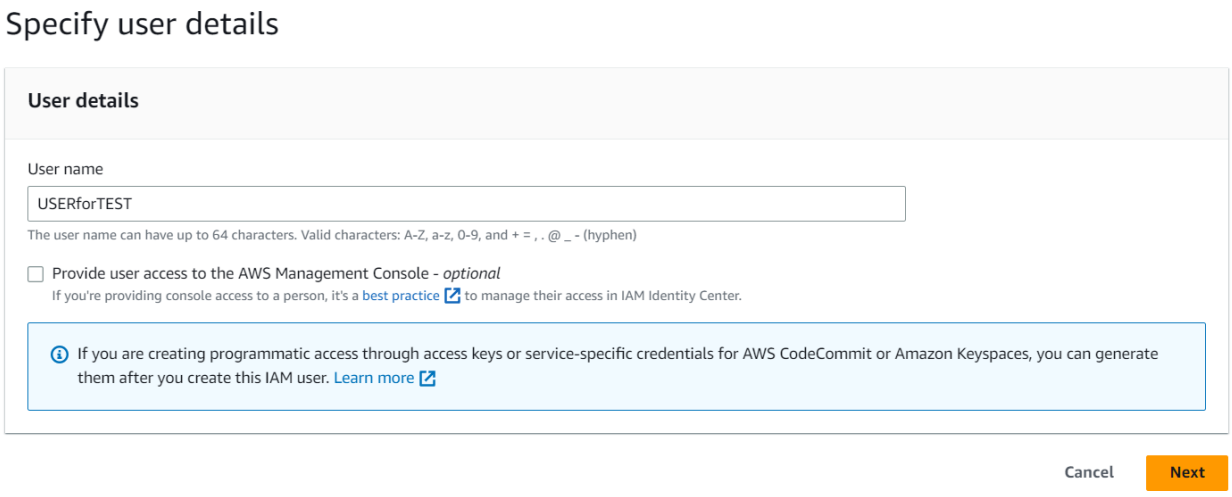


ALL SET TILL YET
NOW THE TIME TO TEST THE AUTOMATION PROCESS

Now you have to create a IAM USER
It automatically send a message on the SLACK Channel



Add user now



Just do it next → next → createUser
Then finally check your slack channel

:aws: **AWS IAM User Creator** APP 9:33 PM
IAM user created: demoVishu

:aws: **AWS IAM User Creator** APP 9:39 PM
IAM user created: demoVishu

:aws: **AWS IAM User Creator** APP 10:09 PM
IAM user created: TestingUSer

IAM user created: USERforTEST

Check the last message it send the notification

IAM user created USERforTEST

Congratulation you have done.