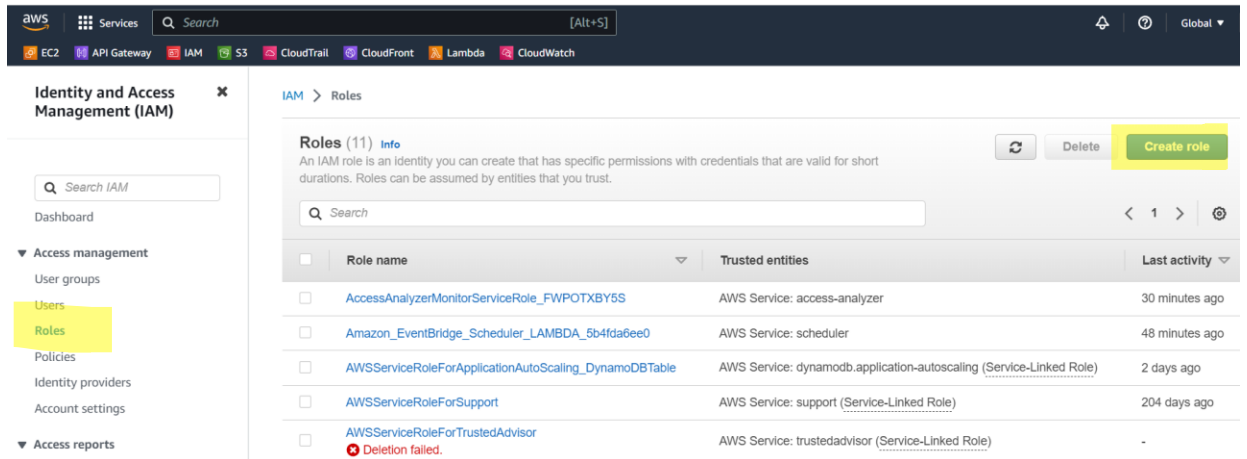
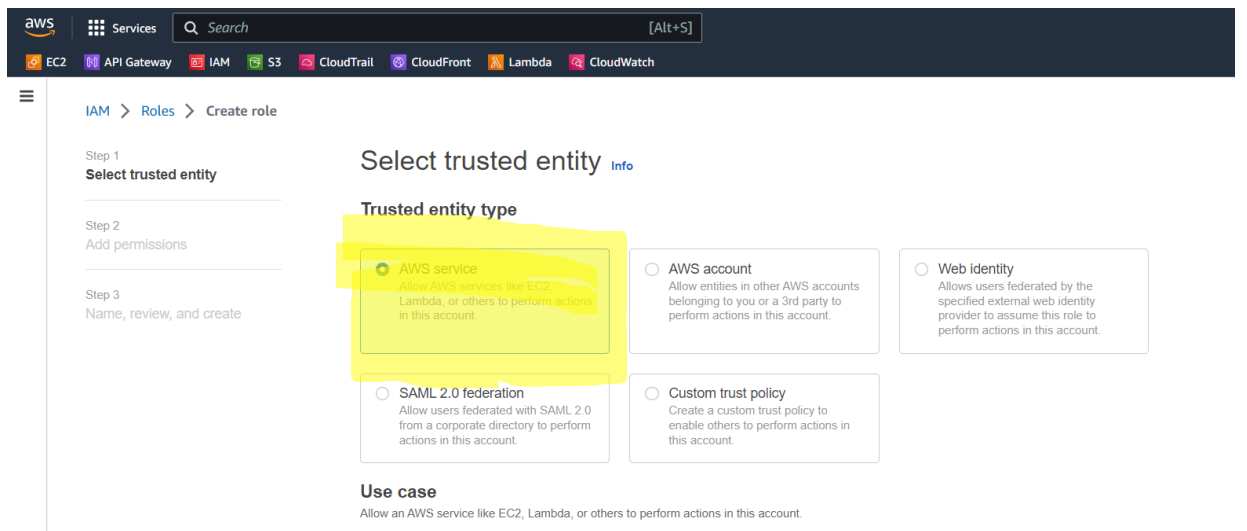


# Create a lambda function to notify on slack channel with webhook of slack channel with Event Bridge and IAM Role

- 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**

### Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

#### Common use cases

☐ EC2

Allows EC2 instances to call AWS services on your behalf.

☒ Lambda

Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:

Choose a service to view use case

Cancel

Next

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

- 


### Add permissions [Info](#)

**Permissions policies** (Selected 2/825) [Info](#)

Choose one or more policies to attach to your new role.

7 matches < 1 > ⚙

"AWSLambdaBasicExecutionRole" X Clear filters


<input checked="" type="checkbox"/>	Policy name <a href="#">↗</a>	Type	Description
<input checked="" type="checkbox"/>	 AWSLambdaBasicExecutionRole	AWS m...	Provides write permissions to CloudWatch Logs.

**Permissions policies** (Selected 2/825) [Info](#)

Choose one or more policies to attach to your new role.

1 match < 1 > ⚙

"AmazonAPIGatewayInvokeFullAccess" X Clear filters

<input checked="" type="checkbox"/>	Policy name <a href="#">↗</a>	Type	Description
<input checked="" type="checkbox"/>	 AmazonAPIGatewayInvokeFullAccess	AWS m...	Provides full access to invoke APIs in Amazon API Gateway.

▶ **Set permissions boundary - optional** [Info](#)

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

Cancel

Previous

Next

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

aws Services Search [Alt+S]

EC2 API Gateway IAM S3 CloudTrail CloudFront Lambda CloudWatch

IAM > Roles > Create role

Step 1  
Select trusted entity

Step 2  
Add permissions

Step 3  
**Name, review, and create**

## Name, review, and create

### Role details

Role name  
Enter a meaningful name to identify this role.  
RoleforIAMUserLambdaFunction  
Maximum 64 characters. Use alphanumeric and '+', '@', '\_' characters.

Description  
Add a short explanation for this role.  
Allows Lambda functions to call AWS services on your behalf.  
Maximum 1000 characters. Use alphanumeric and '+', '@', '\_' characters.

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

Step 2: Add permissions

Edit

Permissions policy summary

Policy name <a href="#">↗</a>	Type	Attached as
AWSLambdaBasicExecutionRole	AWS managed	Permissions policy
AmazonAPIGatewayInvokeFullAccess	AWS managed	Permissions policy

Step 1: Select trusted entities

Edit

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "sts:AssumeRole"
8       ],
9       "Principal": {
10        "Service": [
11          "lambda.amazonaws.com"
12        ]
13      }
14    ]
15  }
16 }
```

Tags

Add tags - optional [Info](#)

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add tag

You can add up to 50 more tags.

Cancel

Previous

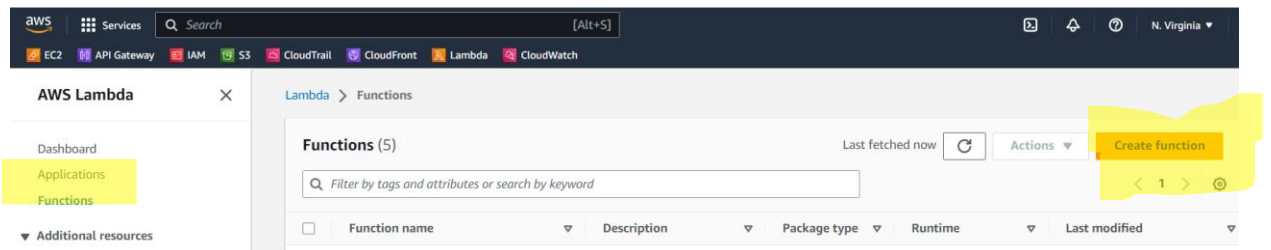
Create role

- 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 Scratch**
- Give your **lambda function name** for your unique identify purpose.
- Also select the runtime language i.e. **PYTHON 3.9**

aws Services Search [Alt+S]

EC2 API Gateway IAM S3 CloudTrail CloudFront Lambda CloudWatch

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

### Basic information

**Function name**  
Enter a name that describes the purpose of your function.  
  
 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 3.9

**Architecture** [Info](#)  
Choose the instruction set architecture you want for your function code.  
☒ x86\_64  
☐ arm64

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**.

aws Services Search [Alt+S]

EC2 API Gateway IAM S3 CloudTrail CloudFront Lambda CloudWatch

Choose the instruction set architecture you want for your function code.

☒ x86\_64  
☐ arm64

### Permissions [Info](#)

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ **Change default execution role**

**Execution role**  
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

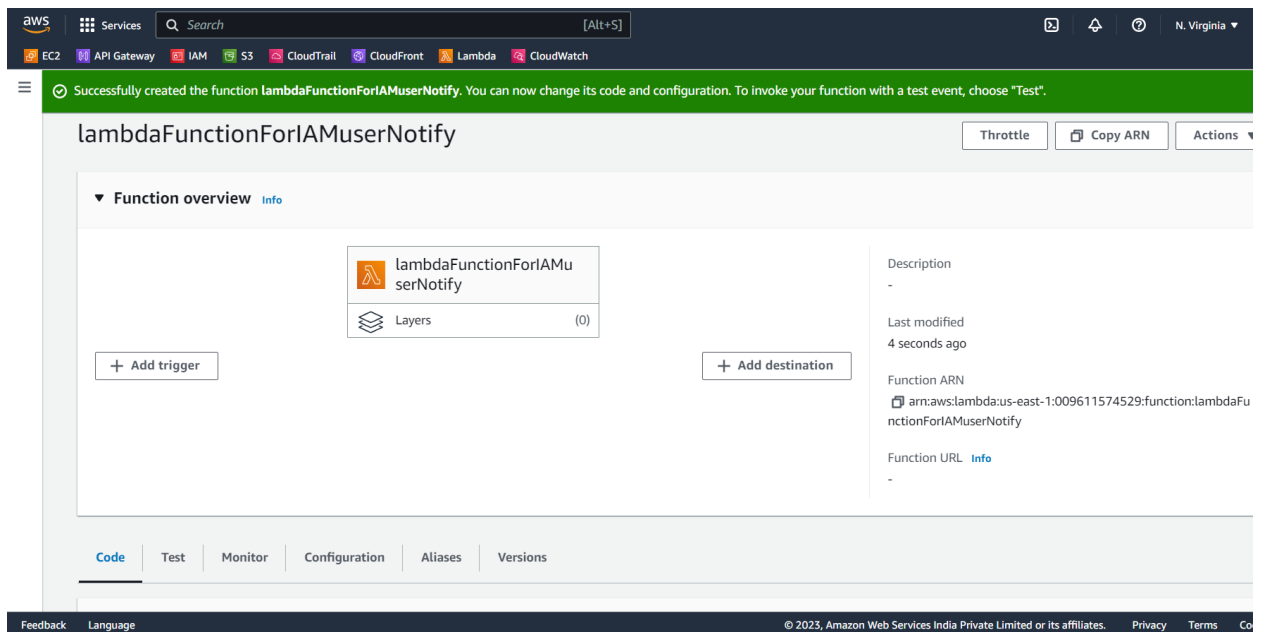
☐ Create a new role with basic Lambda permissions
 ☒ **Use an existing role**
☐ Create a new role from AWS policy templates

**Existing role**  
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.  
  
[View the RoleforIAMUserLambdaFunction role](#) on the IAM console.

► **Advanced settings**

Cancel **Create function**

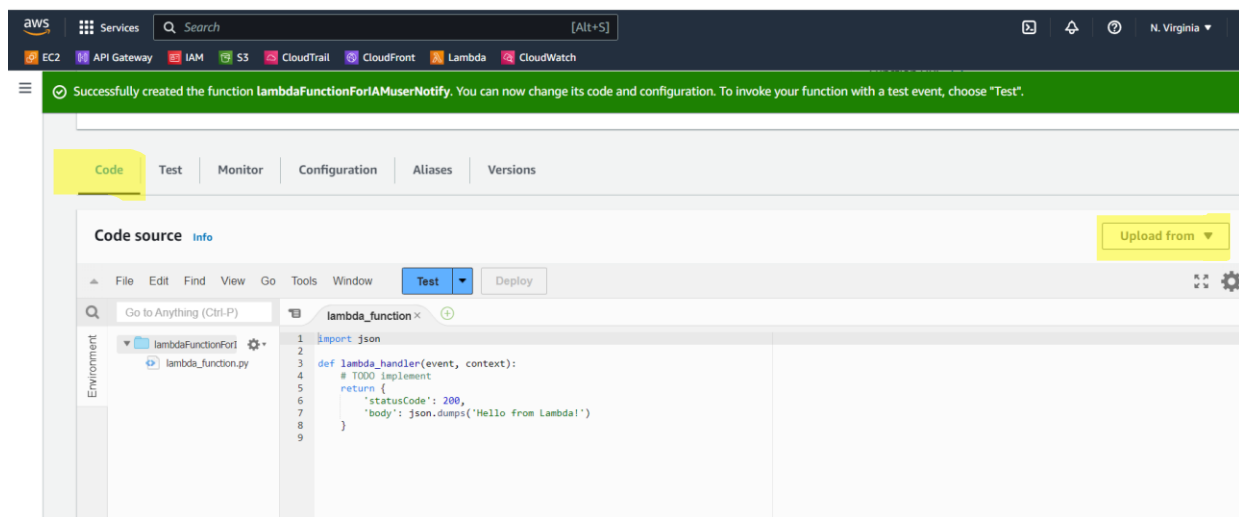
- **Finally click on the create function button.** When the function is created successfully then it shows like that the below screenshot:

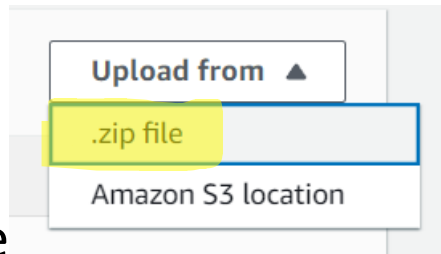


ALL SET ABOUT THE LAMBDA CREATED

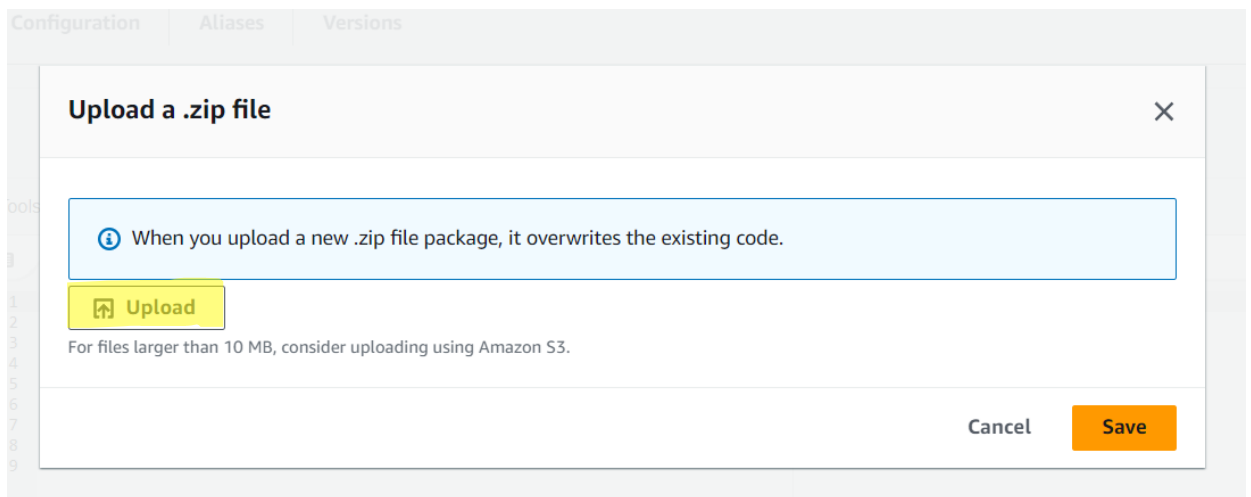
Now We have to create a Lambda Function code in python with the follow steps:

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

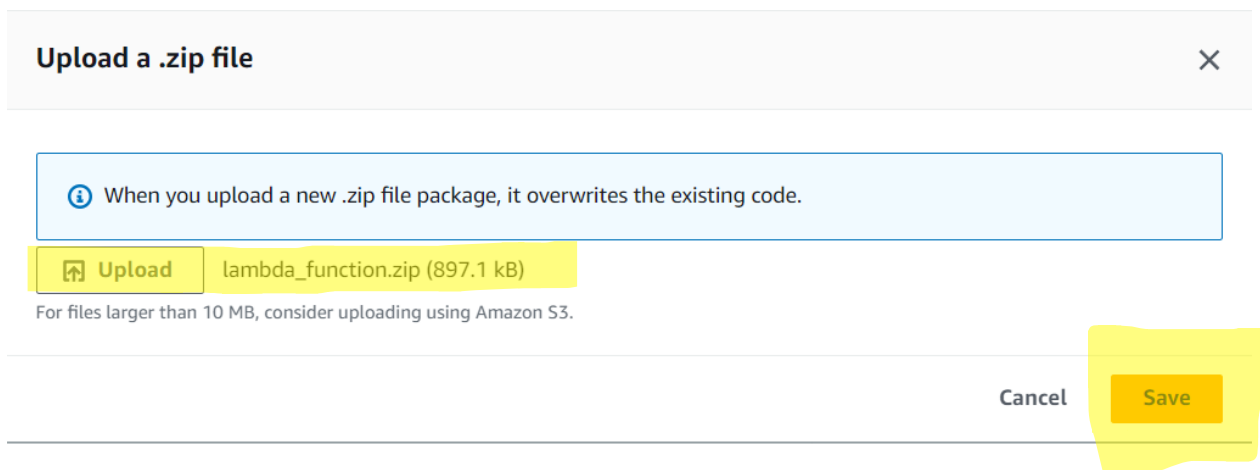




- Just click on .zip file
- And then press upload and upload the file which was given in the provided link.

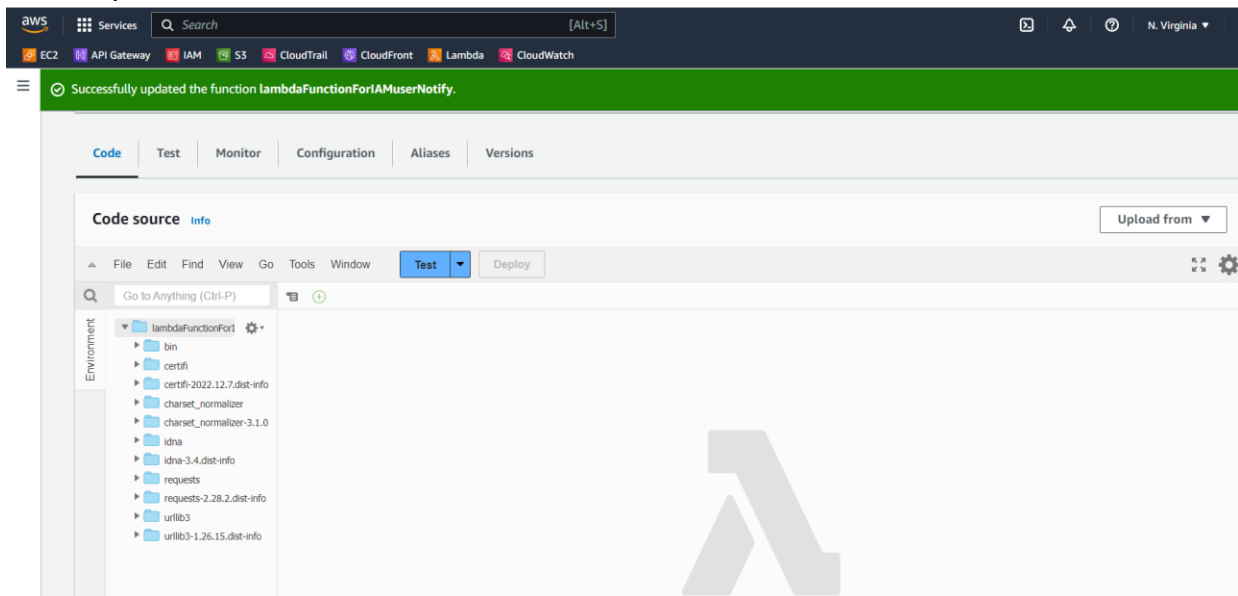


Code: [https://drive.google.com/drive/folders/1BiltuSGWxPeD0QDDTmNEI7vJ21p8fMdg?usp=share\\_link](https://drive.google.com/drive/folders/1BiltuSGWxPeD0QDDTmNEI7vJ21p8fMdg?usp=share_link)

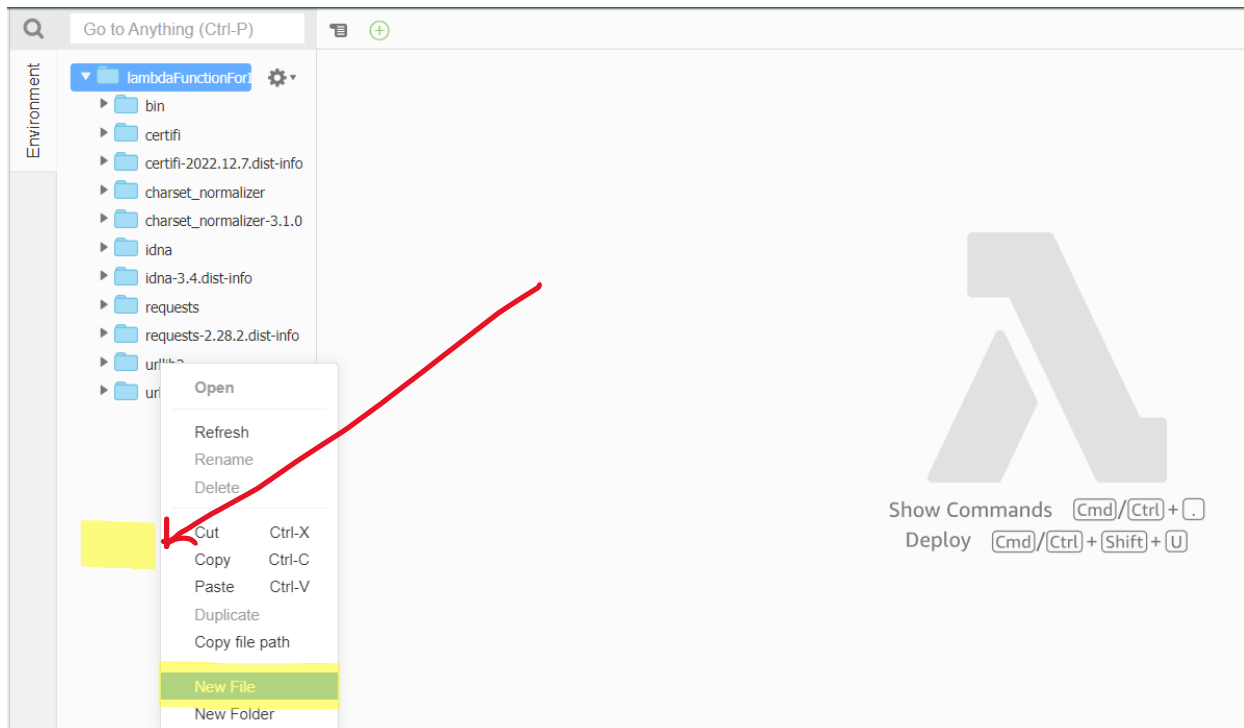


- Finally click on the **save button**

- After upload the file there it look like that

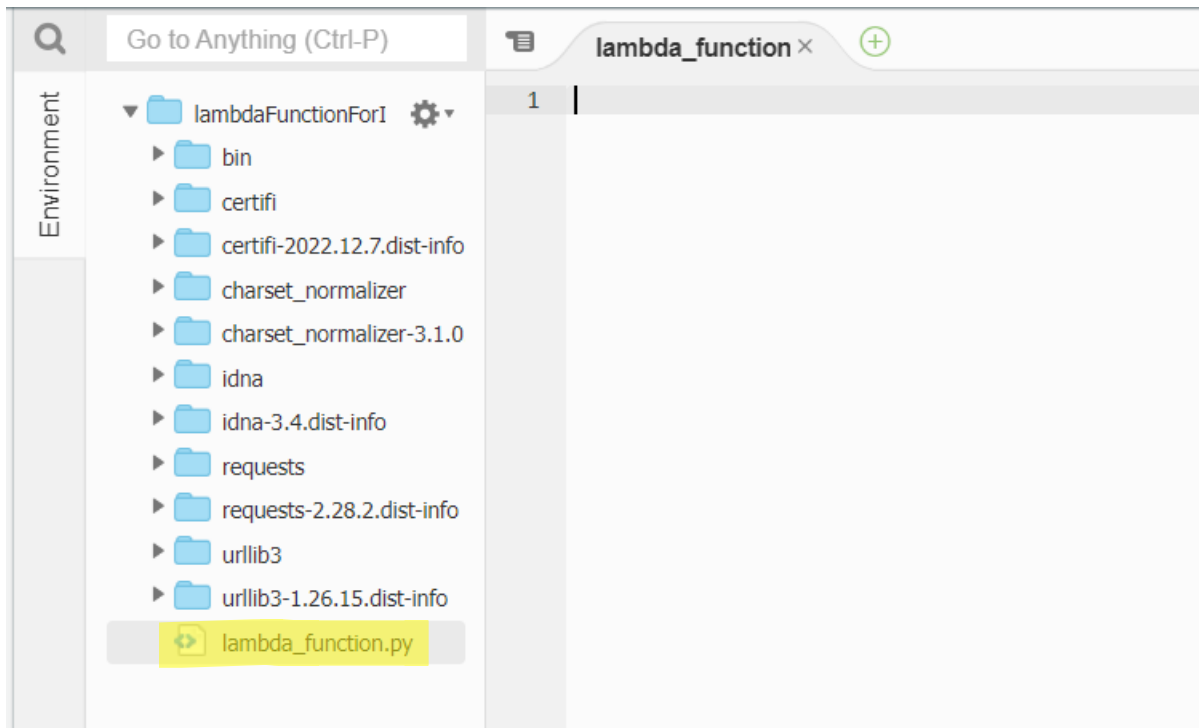


Now you have to create a file here with name as **lambda\_function.py**





- 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 **environmental button** → add the **environmental variable**.

CodeTestMonitorConfigurationAliasesVersions

Code sourceInfo

FileEditFindViewGoToolsWindowTestDeployChanges not deployed

Go to Anything (Ctrl-P)

Environment

- lambdaFunctionFor1
  - 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

lambda\_function

```
1 import json
2 import urllib.request
3 import os
4
5 SLACK_WEBHOOK_URL = os.environ['SLACK_WEBHOOK_URL']
6
7
8 def lambda_handler(event, context):
9     event_name = event['detail']['eventName']
10    if event_name == 'CreateUser':
11        user_name = event['detail']['requestParameters']['userName']
12        message = {
13            "text": f"IAM user created: {user_name}",
14            "username": "AWS IAM User Creator",
15            "icon_emoji": ":aws:"
16        }
17        req = urllib.request.Request(SLACK_WEBHOOK_URL)
18        req.add_header('Content-Type', 'application/json')
19        response = urllib.request.urlopen(req, json.dumps(message).encode('utf-8'))
20        return {
21            'statusCode': 200,
22            'body': response.read().decode('utf-8')
23        }
24    else:
```

CodeTestMonitorConfigurationAliasesVersions

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

CodeTestMonitorConfigurationAliasesVersions

General configuration

Triggers

Permissions

Destinations

Function URL

Environment variables

Environment variables

Key	Value
No environment variables No environment variables associated with this function. Edit	

## Edit environment variables

### Environment variables

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more](#)

There are no environment variables on this function.

Add environment variable

► Encryption configuration

Cancel

Save

aws Services Search [Alt+S]

EC2 API Gateway IAM S3 CloudTrail CloudFront Lambda CloudWatch

Lambda > Functions > lambdaFunctionForIAMUserNotify > Edit environment variables

### Edit environment variables

#### Environment variables

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more](#)

Key	Value	
SLACK_WEBHOOK_URL	4R65W2X/AsGFxfTeW4fO1JuVFI4cUVfQ	Remove

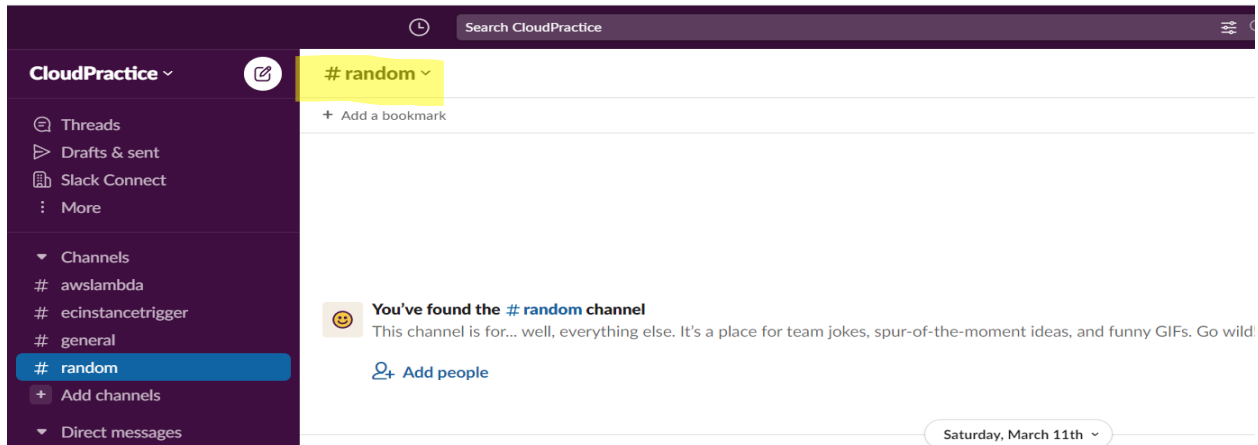
Add environment variable

► Encryption configuration

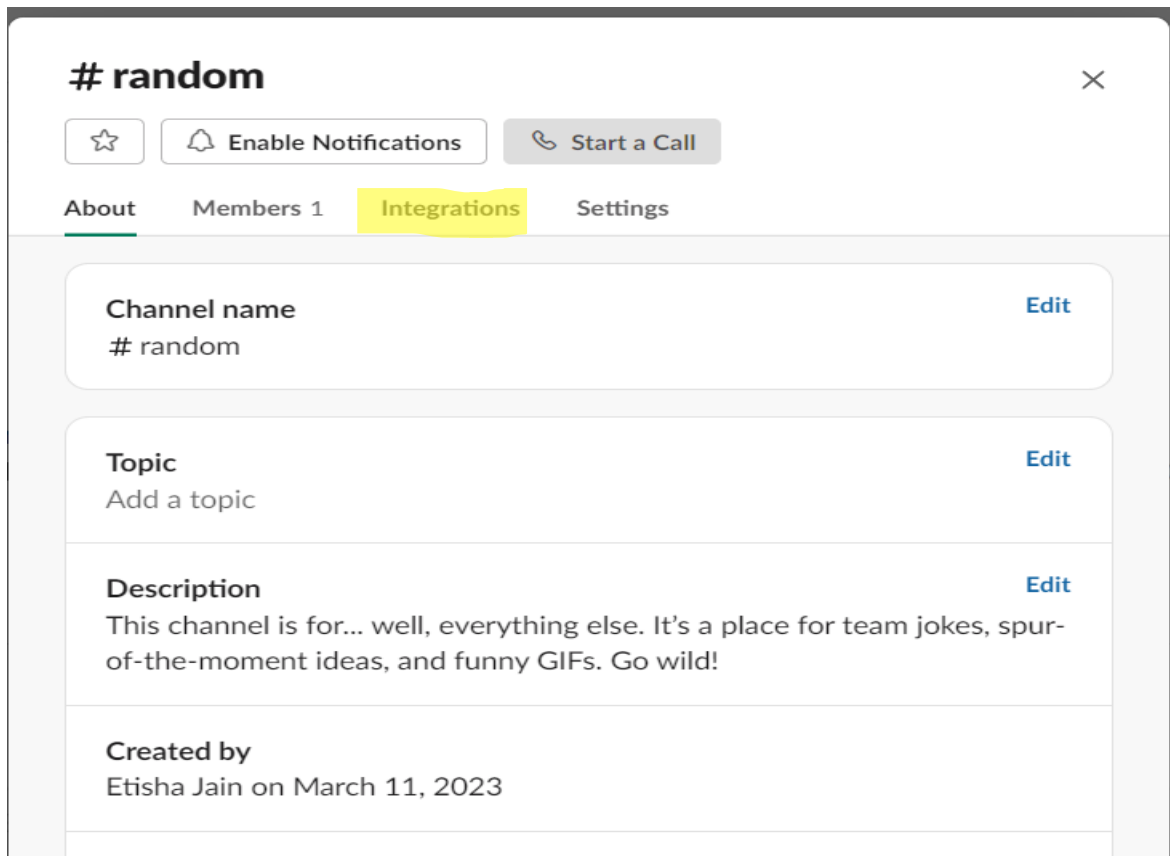
Cancel Save

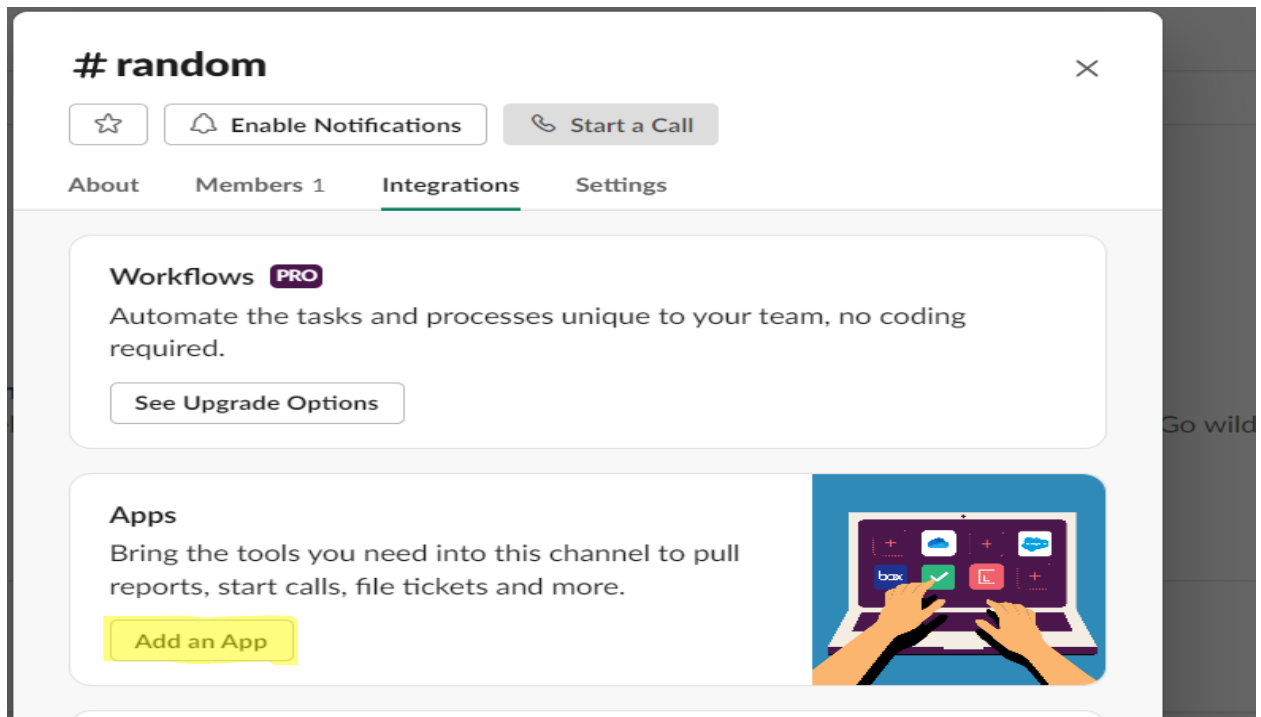
- Just give the variable name and its value i.e webhook url
- Just wait I have forgot to create a webhook url → just follow steps to create a webhook url.
- Just go inside which channel where you want to your message trigger.

## 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

## Add apps to # random

[View App Directory](#)

Q web

In your workspace



incoming-webhook

[View](#)

From the App Directory



Giphy

An online library of animated GIFs

[Install](#)



ICDSOft Web Hosting

Receive notifications about your ICDSoft web hosting services in Slack.

[Install](#)



Incoming WebHooks

Send data into Slack in real-time.

[Install](#)

- Search the incoming-webhook install and view the hook

## Incoming WebHooks



Configuration

### Incoming WebHooks

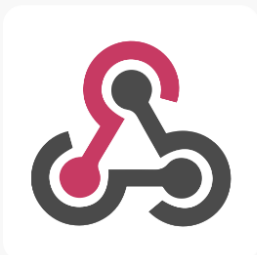
Please note, this is a legacy custom integration - an outdated way for teams to integrate with Slack. These integrations lack newer features and they will be deprecated and possibly removed in the future. **We do not recommend their use.** Instead, we suggest that you check out their replacement: [Slack apps](#).

Incoming Webhooks are a simple way to post messages from external sources into Slack. The payload, which includes the message and a few other optional details described later.

[Message Attachments](#) can also be used in Incoming Webhooks to display richly-formatted messages that stand out from regular chat messages.

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

[← Browse Apps](#)



Add to Slack

Learn more & Support

[Get app support](#)

[Privacy policy](#)

[Terms](#)

### Incoming WebHooks

Description

Configuration

Security & Compliance

Please note, this is a legacy custom integration - an outdated way for teams to integrate with Slack. These integrations lack newer features and they will be deprecated and possibly removed in the future. **We do not recommend their use.** Instead, we suggest that you check out their replacement: [Slack apps](#).

Incoming Webhooks are a simple way to post messages from external sources into Slack. They make use of normal HTTP requests with a JSON payload, which includes the message and a few other optional details described later.

[Message Attachments](#) can also be used in Incoming Webhooks to display richly-formatted messages that stand out from regular chat messages.

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



#### New to Slack integrations?

Check out our [Getting Started](#) guide to familiarize yourself with the most common types of integrations, and tips to keep in mind while building your own. You can also [register as a developer](#) to let us know what you're working on, and to receive future updates to our APIs.

#### Post to Channel

Start by choosing a channel where your Incoming Webhook will post messages to.

Choose a channel...

or [create a new channel](#)

Add Incoming WebHooks integration

By creating an incoming webhook, you agree to the [Slack API Terms of Service](#).



#### New to Slack integrations?

Check out our [Getting Started](#) guide to familiarize yourself with the most common types of integrations, and tips to keep in mind while building your own. You can also [register as a developer](#) to let us know what you're working on, and to receive future updates to our APIs.

### Post to Channel

Start by choosing a channel where your Incoming Webhook will post messages to.

# random

[or create a new channel](#)

**Add Incoming WebHooks integration**

By creating an incoming webhook, you agree to the [Slack API Terms of Service](#).

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



#### New to Slack integrations?

Check out our [Getting Started](#) guide to familiarize yourself with the most common types of integrations, and tips to keep in mind while building your own. You can also [register as a developer](#) to let us know what you're working on, and to receive future updates to our APIs.

### Setup Instructions

close

We'll guide you through the steps necessary to configure an Incoming Webhook so you can start sending data to Slack.

Webhook URL

<https://hooks.slack.com/services/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>

- 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

The screenshot shows the AWS Lambda console with the 'Configuration' tab selected. On the left sidebar, 'Environment variables' is highlighted. The main area displays 'Environment variables (1)' with a table containing one entry:

Key	Value
SLACK_WEBHOOK_URL	https://hooks.slack.com/services/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

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

The screenshot shows the AWS Lambda console with the 'Test' tab selected. The header 'Test event' is visible with an 'Info' link.

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

☒ Create new event

☐ Edit saved event

Event name

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings

- ☒ Private  
This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)
- ☐ Shareable  
This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

- 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"
}

```

Successfully updated the function lambdaFunctionForIAMUserNotify. X

Shareable

This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

Template - optional

hello-world

Event JSON

Format JSON

```
1 {
2   "version": "0",
3   "id": "bb914bcd-d8fc-e5ae-4942-456ac55f551d",
4   "detail-type": "AWS API Call via CloudTrail",
5   "source": "aws.iam",
6   "account": "009611574529",
7   "time": "2023-03-14T13:52:53Z",
8   "region": "us-east-1",
9   "resources": [],
10  "detail": {
11    "eventVersion": "1.08",
12    "userIdentity": {
13      "type": "Root",
14      "principalId": "009611574529",
15      "arn": "arn:aws:iam::009611574529:root",
16      "accountId": "009611574529",
17      "accessKeyId": "ASIAQEPHE7UAZS3HQZ5R",
18      "sessionContext": {
```

Code **Test** Monitor Configuration Aliases Versions

Test event [Info](#)

Save Test

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

Create new event

Edit saved event

Event name

TestLog

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

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

Code **Test** Monitor Configuration Aliases Versions

Execution result: succeeded (logs) X

Details

Test event [Info](#)

Delete Save Test

To invoke your function without saving an event, modify the event, then choose Test. Lambda uses the modified event to invoke your function, but does not overwrite the original event until you choose Save changes.

Test event action

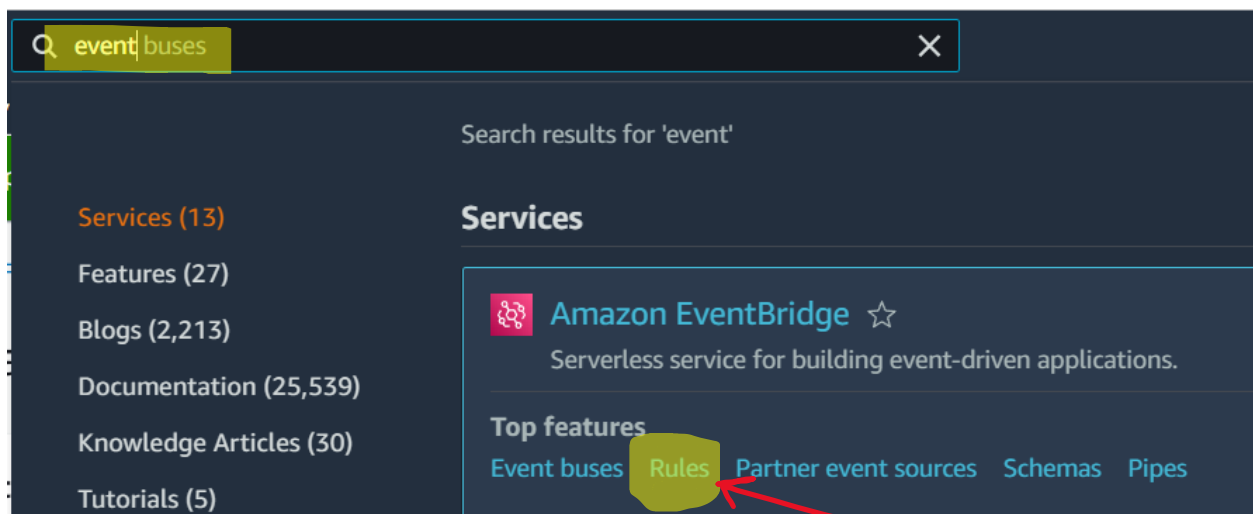
- After save and test the JSON file then check on the slack it send the message there.

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

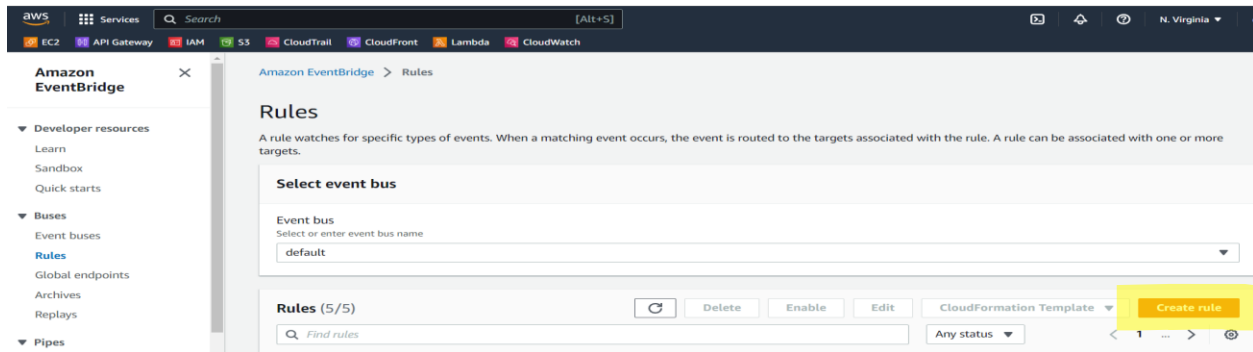
:aws: **AWS IAM User Creator** APP 9:39 PM  
IAM user created: demoVishu

## 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**

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

**Description - optional**

**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.

## Creation method

### Method

☐ Use schema

Use an Amazon EventBridge schema to generate the event pattern.

☐ Use pattern form

Use a template provided by EventBridge to create an event pattern.

☒ Custom pattern (JSON editor)  
Write an event pattern in JSON.

## Event pattern [Info](#)

### Event pattern

Write an event pattern in JSON. You can test the event pattern against the sample event. You can also go to pre-defined pattern.

Prefix matching ▼

Insert

☐ Content-based filter syntax

1

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

## Event pattern [Info](#)

### Event pattern

Write an event pattern in JSON. You can test the event pattern against the sample event. You can also go to pre-defined pattern.

Prefix matching ▼

Insert

☐ Content-based filter syntax

```
1 {
2   "source": ["aws.iam"],
3   "detail-type": ["AWS API Call via CloudTrail"],
4   "detail": {
5     "eventSource": ["iam.amazonaws.com"]
6   }
7 }
```

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

✓ JSON is valid

Copy

Prettify

Event pattern form

Test pattern

Cancel

Previous

Next

Now you have to create a NEXT button

## Target 1

### Target types

Select an EventBridge event bus, EventBridge API destination (SaaS partner), or another AWS service as a target.

- ☐ EventBridge event bus
- ☐ EventBridge API destination

☒ AWS service

Select a target [Info](#)

Select target(s) to invoke when an event matches your event pattern or when schedule is triggered (limit of 5 targets per rule)

Lambda function

### Function

lambdaFunctionForIAMUserNotify



► Configure version/alias

► Additional settings

Add another target

Cancel

Skip to Review and create

Previous

Next

## Configure tags - optional [Info](#)

### Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

Add new tag

You can add 50 more tags.

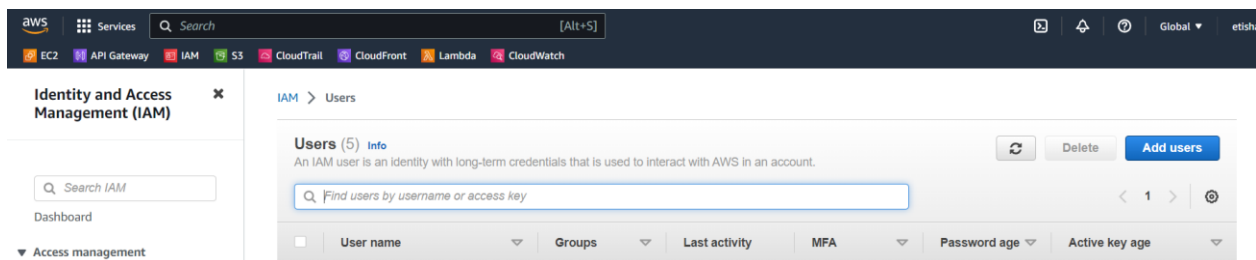
Cancel

Previous

Next

# 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**

Specify user details

**User details**

User name

USERforTEST

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ \_ - (hyphen)

☐ Provide user access to the AWS Management Console - *optional*  
If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel

Next

**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.**

