

Serverless Remainder App

M.SreeNidhi
2100030791

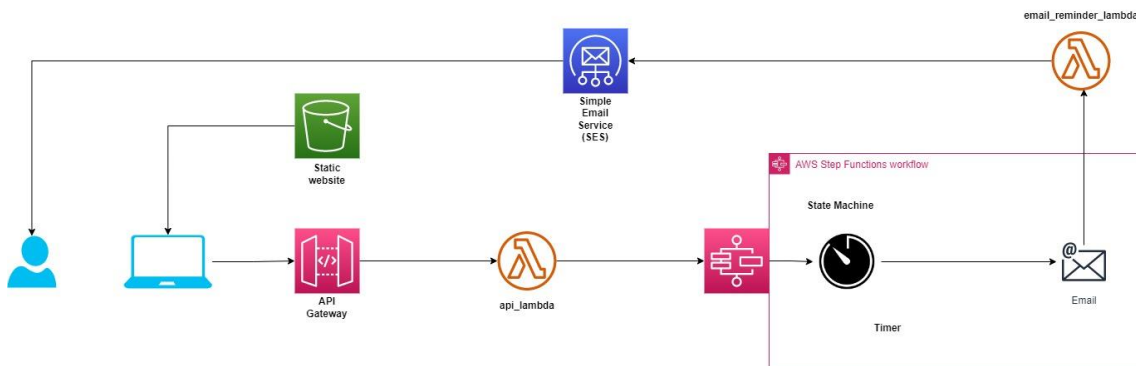
Introduction:

In this, we will be implementing a serverless reminder application. The Web application will load from an S3 bucket as a static website and run in browser, communicate with Lambda and Step functions via an API Gateway Endpoint. Using the application you will be able to configure reminders to be send using email.

Services covered:

- Simple Email service
- CloudFormation
- IAM
- Lambda Function
- Step Functions
- API gateway
- S3

- Make sure the Region is set to US East (N. Virginia) - us-east-1 while using all the services.

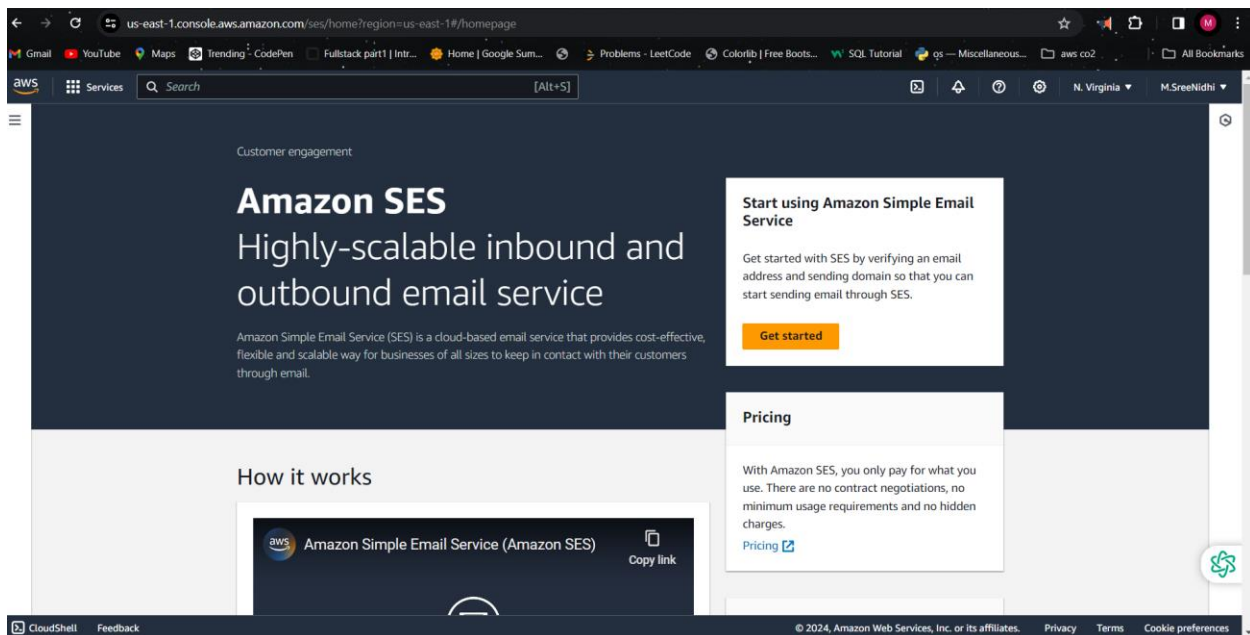


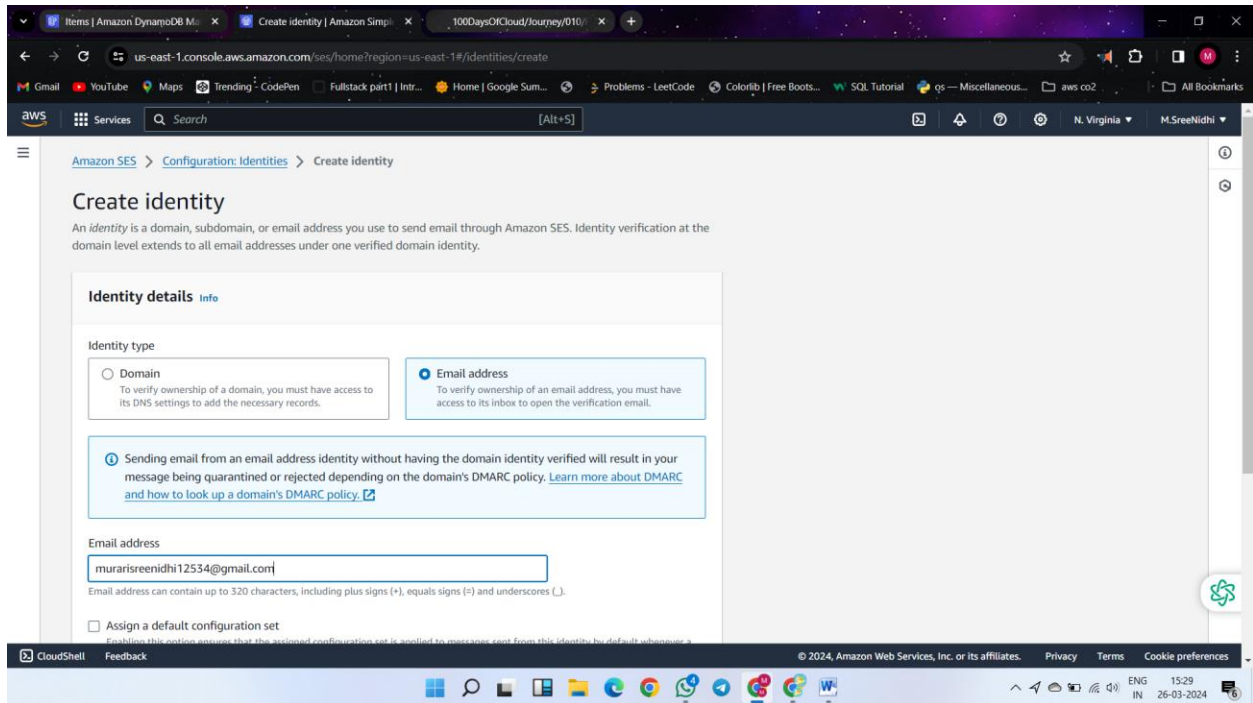
REMINDO

Step-1: Configure Simple Email Service (SES)

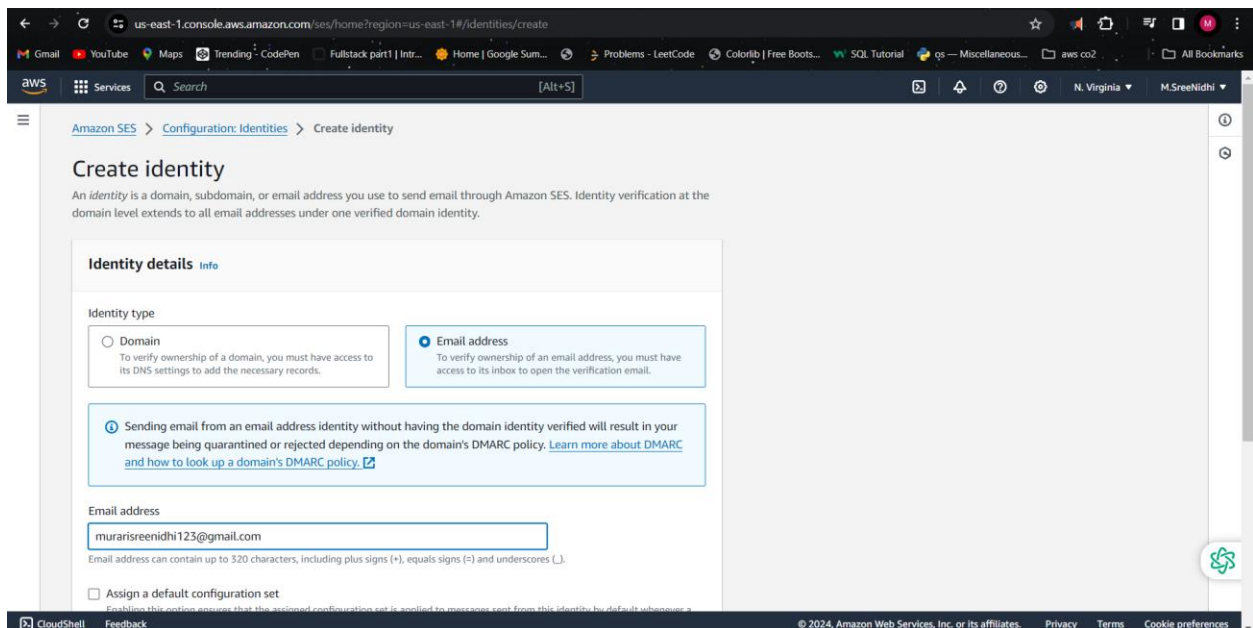
We need two emails for this. One for the web app to send the email from and another for the reminder from the app to be sent to (user).

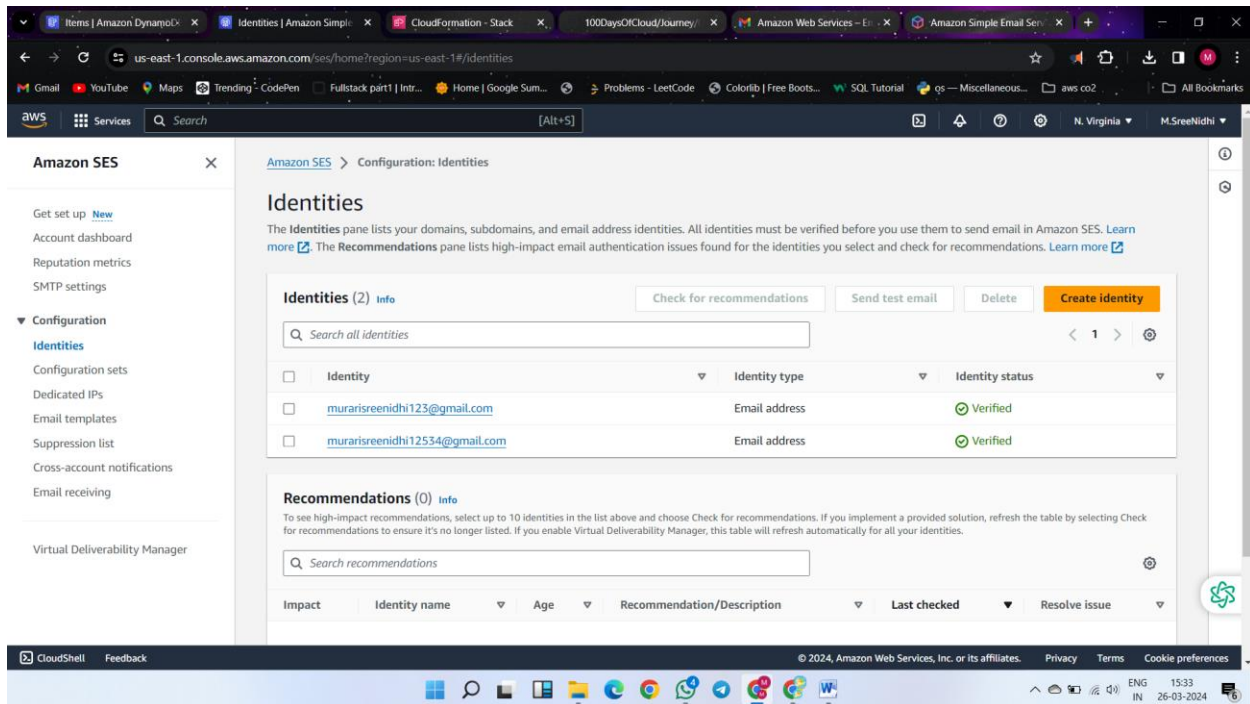
- Search SES. Click verified identities on left side.
- Create identity.
- Check the 'Email Address' checkbox.
- Enter the email address that you want the application to use to send the reminder.
- Create Identity.





Create one more identity with which you want to receive the reminder



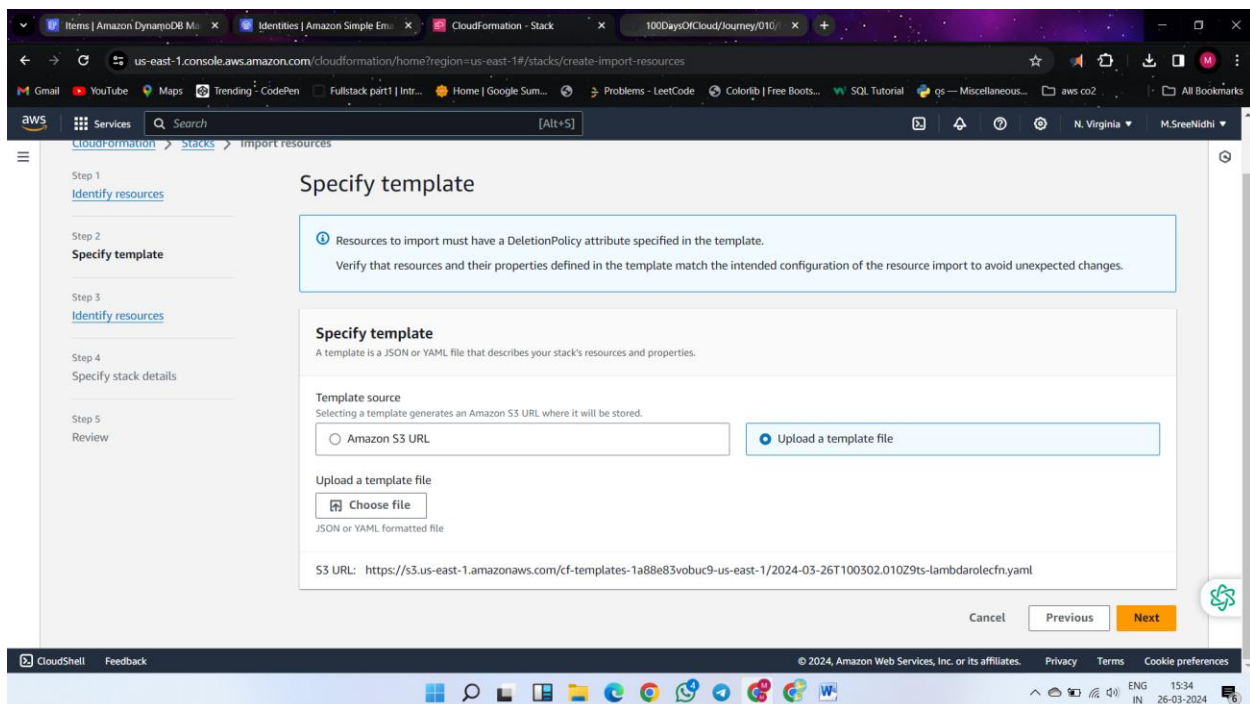
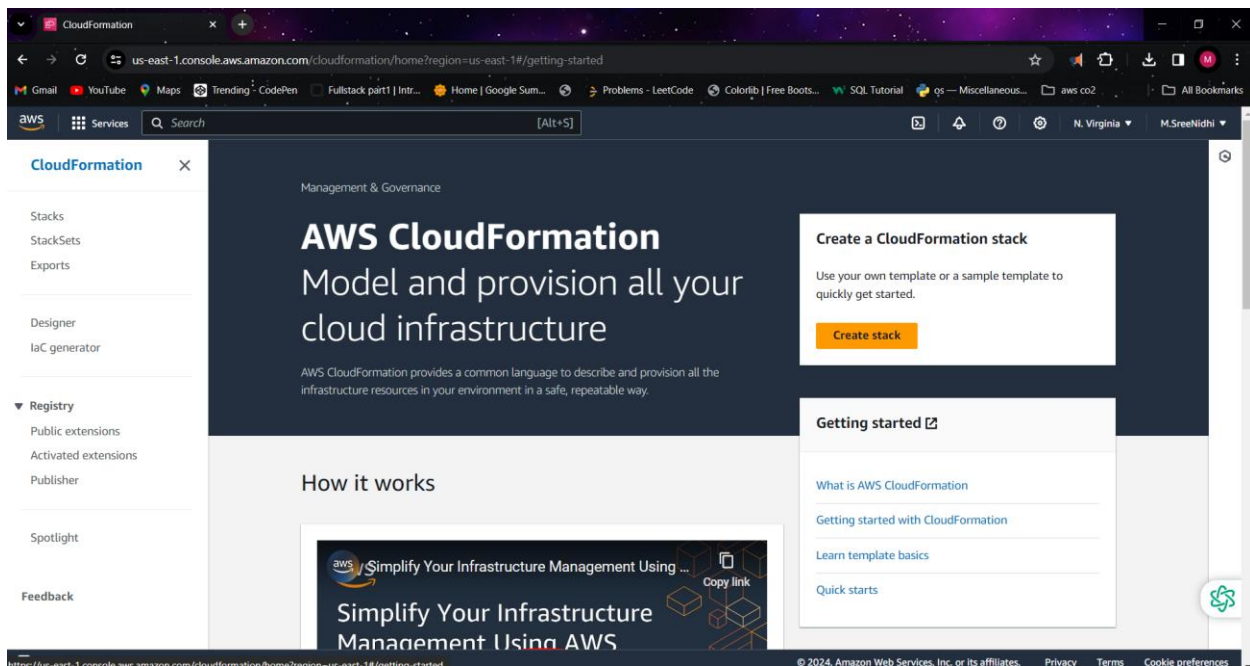


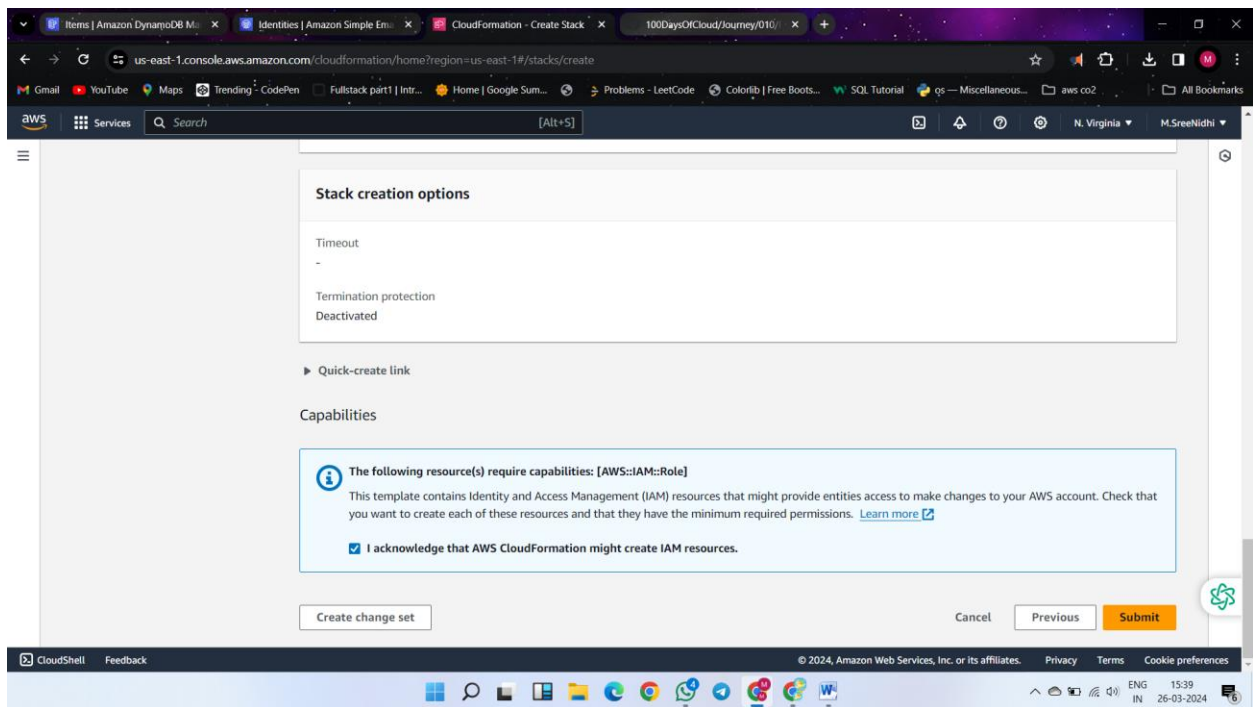
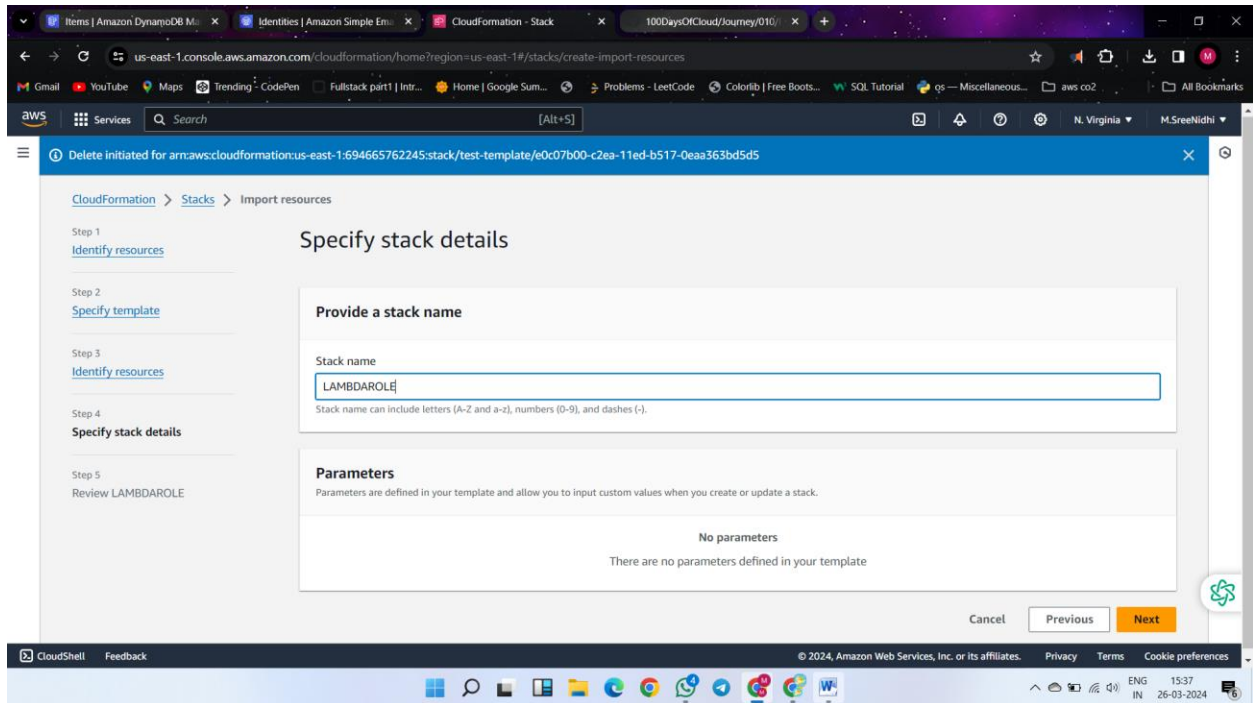
Step-2: Add a email Lambda function to use SES to send emails for the serverless application

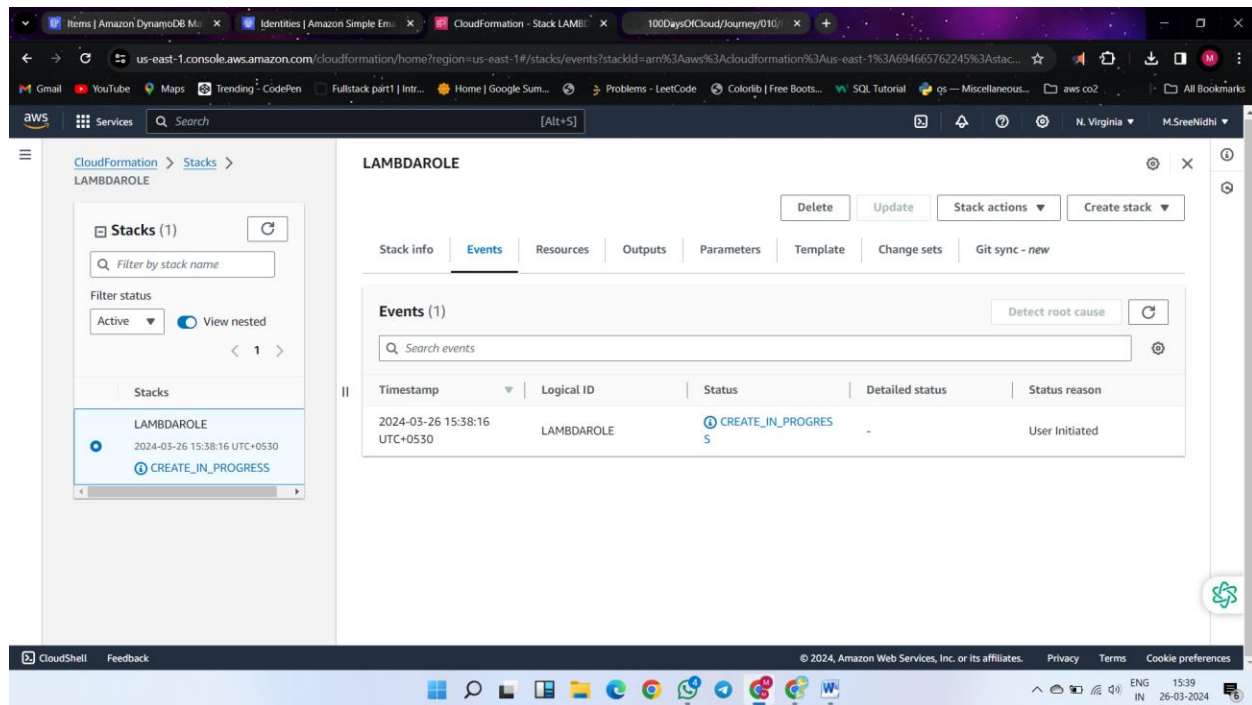
In this stage, we need to create a Lambda function which will be used by the serverless application to create an email and then send it using SES. Before that we need to create an IAM role which Lambda will use to interact with other AWS services.

- **Create a Lambda Role in CloudFormation using the yaml file. Make sure you unzip the file.**
 - Search CloudFormation and select Create Stack.
 - Select Template is Ready option and select upload a template file and choose the lambdarolecfm.yaml file and upload it and click Next.
 - Enter Stack name as LAMBDA_ROLE, click Next till you reach the last section.

In the last section, scroll down and check the 'I acknowledge that AWS CloudFormation might create IAM resources' box and then click Create Stack.







- **Create the Lambda Function.**

- Search Lambda and click on Create Function.
- Select Author from scratch. For Function name enter email_reminder_lambda .
- For runtime click the dropdown and select Python 3.9.
- Expand Change default execution role, select Use an existing role, click the Existing role dropdown and pick LambdaRole.
- Create the function.

us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/begin

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.

Get started

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

[Create a function](#)

How it works

[Run](#) Next: Lambda responds to events

.NET Java **Node.js** Python Ruby Custom runtime

```
1 exports.handler = async (event) => {
2   console.log(event);
3   return 'Hello from Lambda!';
4 }
```

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/create/function?firstrun=true

Create function

Choose one of the following options to create your function.

☒ **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.

email_reminder_lambda

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

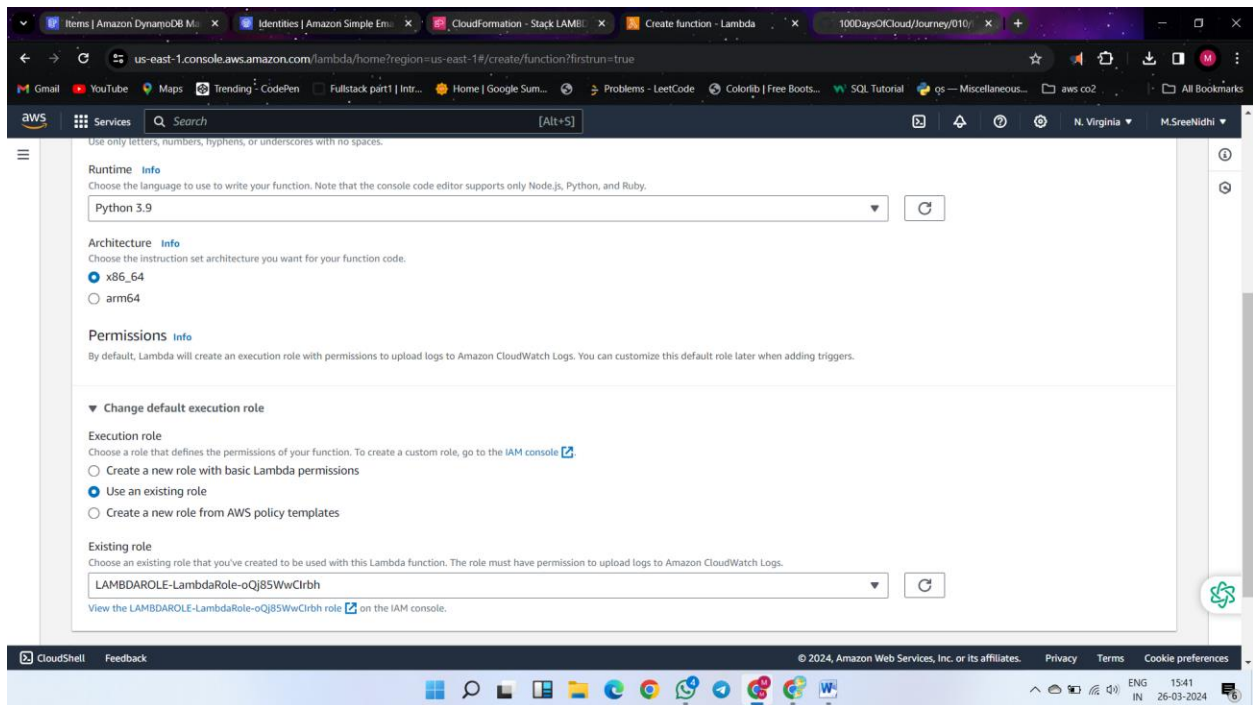
Architecture [Info](#)
Choose the instruction set architecture you want for your function code.

☒ **x86_64**

☐ arm64

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



```
import boto3, os, json
```

```
FROM_EMAIL_ADDRESS = 'REPLACE_ME'
```

```
ses = boto3.client('ses')
```

```
def lambda_handler(event, context):
```

```
    # Print event data to logs ..
```

```
print("Received event: " + json.dumps(event))
```

```
# Publish message directly to email, provided by EmailOnly or EmailPar
```

TASK

```
ses.send_email( Source=FROM_EMAIL_ADDRESS,
```

```
Destination={ 'ToAddresses': [ event['Input']['email'] ] },
```

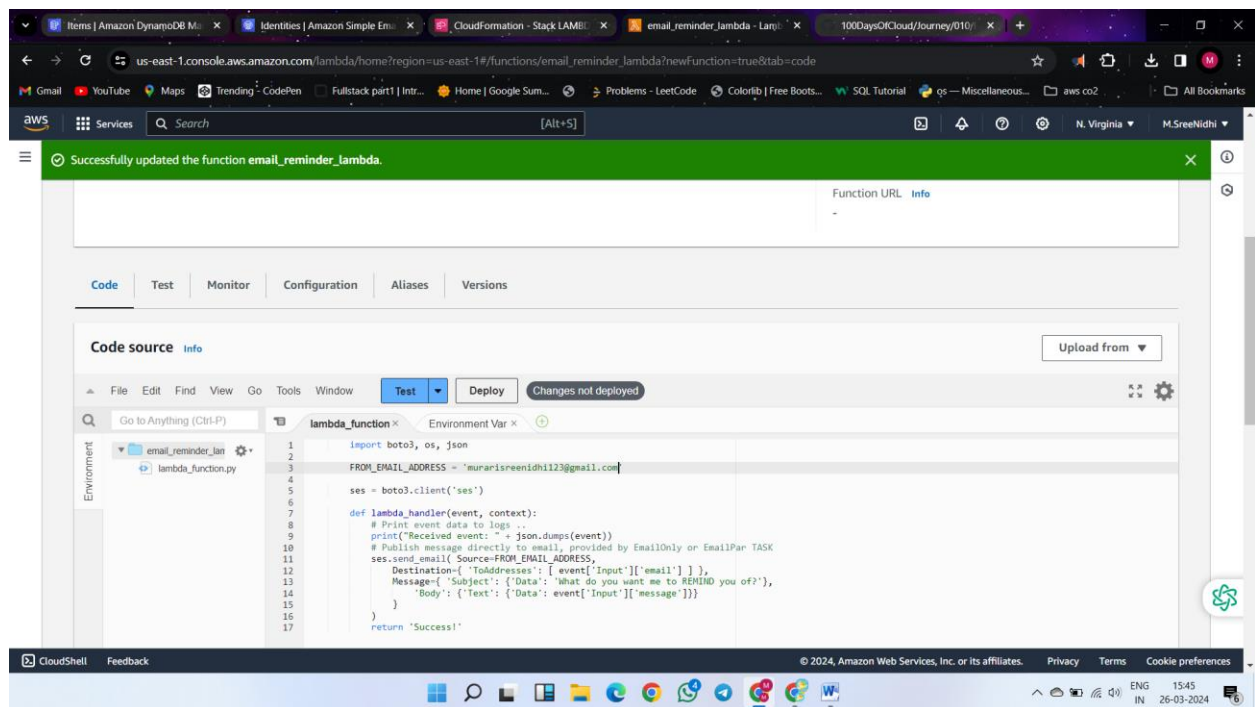
```
Message={ 'Subject': { 'Data': 'What do you want me to REMIND you of?'},
```

```
'Body': { 'Text': { 'Data': event['Input']['message'] }}
```

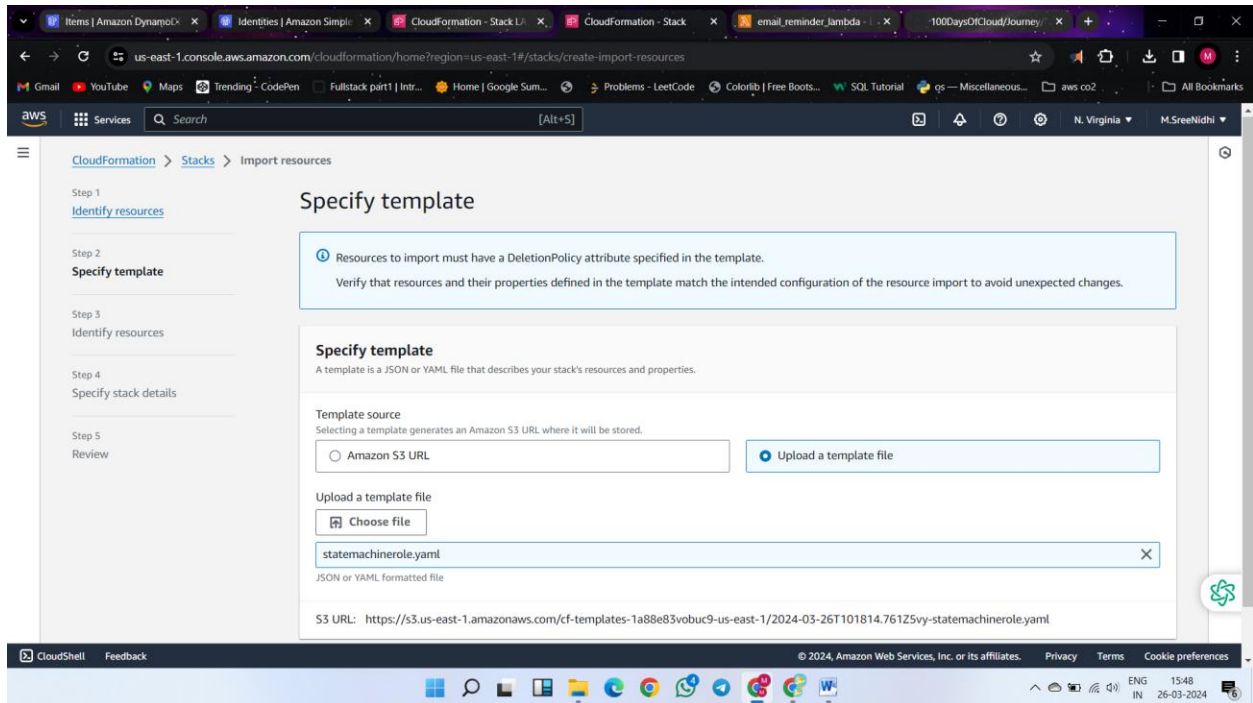
```
}
```

```
)
```

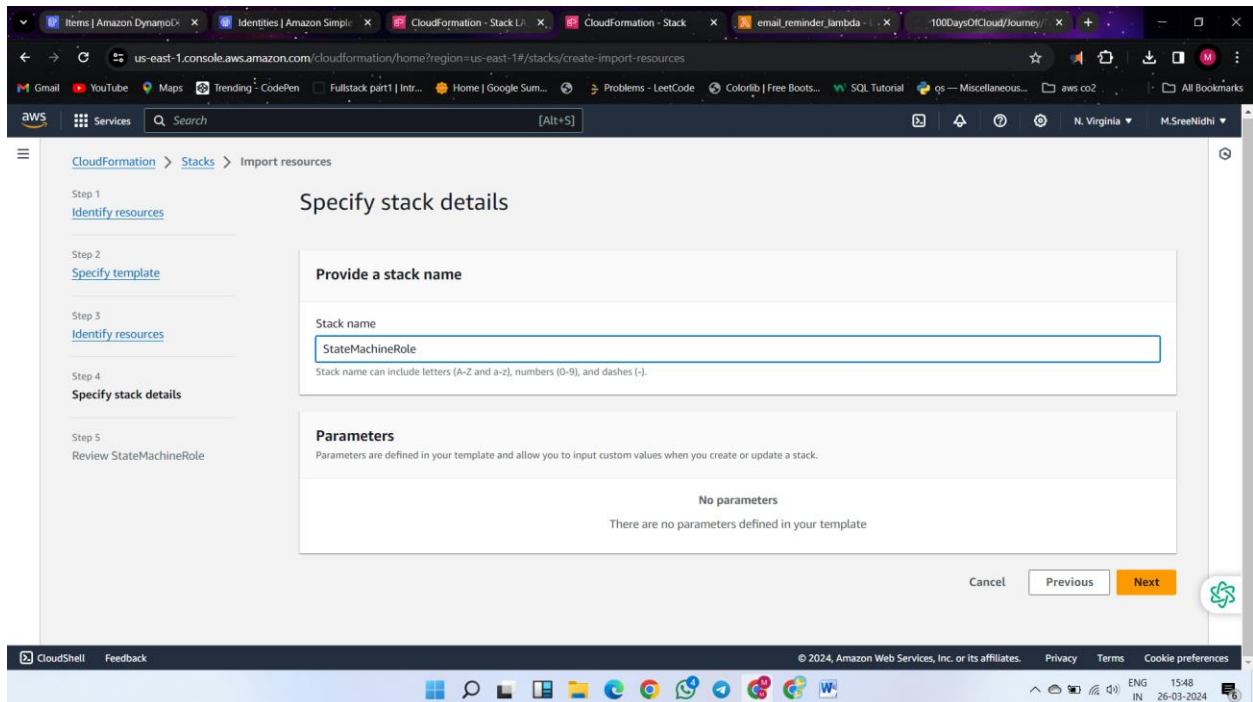
```
return 'Success!'
```

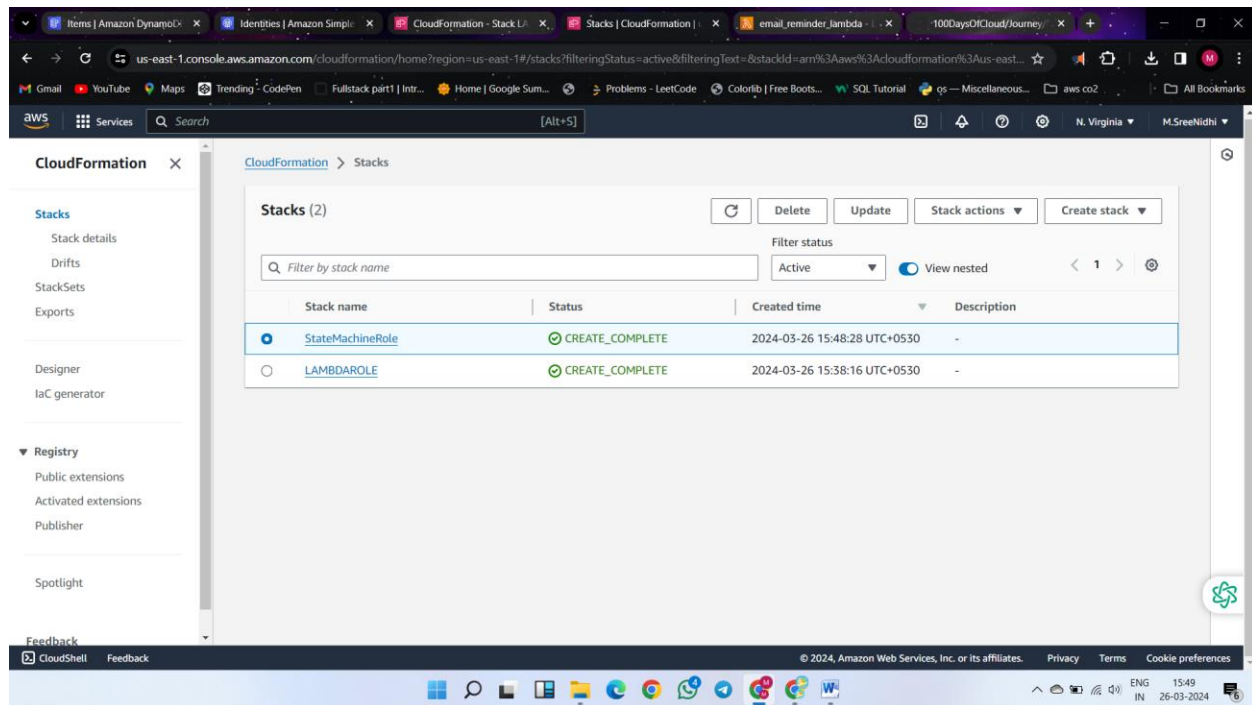


Create a new stack in cloudformation



we need to create an IAM role which the machine that we create will use to interact with other AWS services.





Step-3: Implement and configure the state machine, the core of the application

In step functions service, create a new state machine

us-east-1.console.aws.amazon.com/states/home?region=us-east-1#/statemachines

Step Functions

State machines (0)

☐ View execution counts

Execution counts are based on the most recent 1000 executions

Search for state machines Any type

Name	Type	Creation date	Status
No state machines			

https://us-east-1.console.aws.amazon.com/states/home?region=us-east-1#/statemachines

us-east-1.console.aws.amazon.com/states/home?region=us-east-1#/v2/statemachines/create

Step Functions

MyStateMachine-1v3w14jok

Undo Redo Format Copy Commands

Zoom in Zoom out Center Feedback

```
1 {
2   "comment": "Remindo - using Lambda for email.",
3   "startAt": "Timer",
4   "states": {
5     "Timer": {
6       "type": "wait",
7       "secondsPath": "$.waitSeconds",
8       "next": "Email"
9     },
10    "Email": {
11      "type": "Task",
12      "resource": "arn:aws:states:::lambda:invoke",
13      "parameters": {
14        "FunctionName": "EMAIL_LAMBDA_ARN",
15        "payload": {
16          "input.$": "$"
17        }
18      },
19      "next": "NextState"
20    },
21    "NextState": {
22      "type": "Pass",
23      "end": true
24    }
25  }
26 }
```

Start

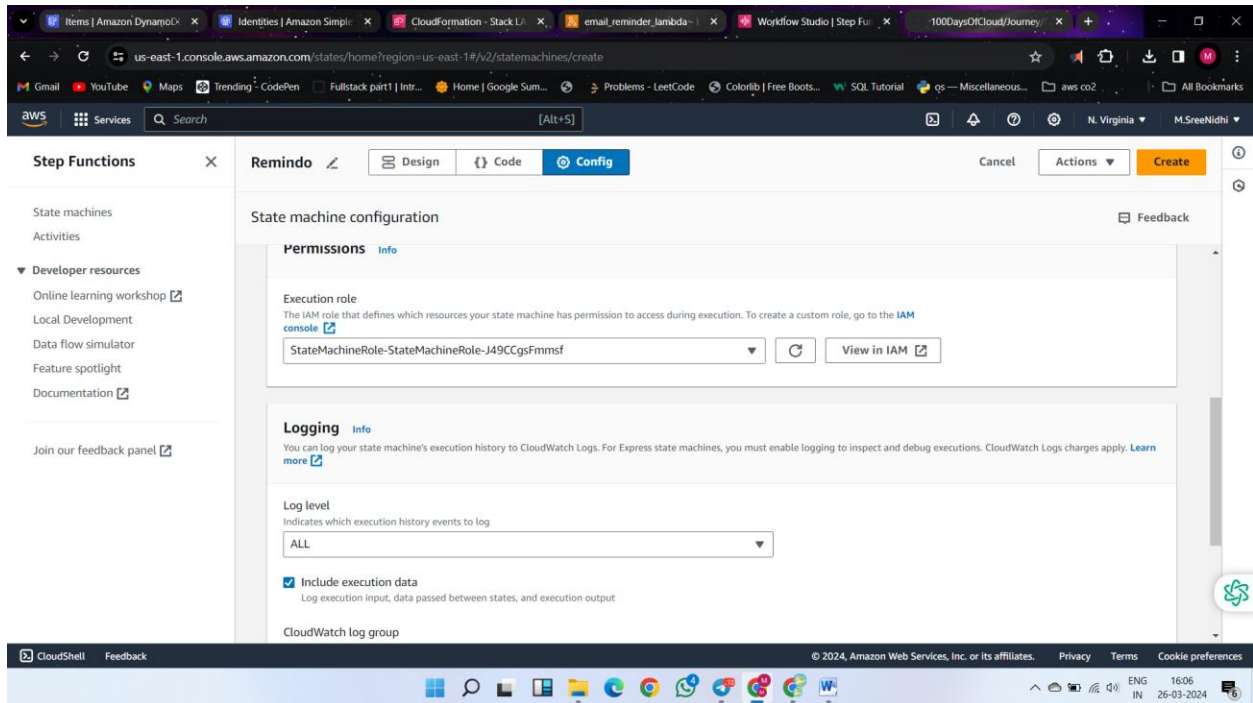
Wait state
Timer

Lambda: Invoke
Email

Pass state
NextState

End

CloudShell Feedback



Step-4: Implement the API Gateway, API and supporting Lambda function

In this stage you will be creating the front end API for the serverless application. The front end loads from S3, runs in your browser and communicates with this API. It uses API Gateway for the API Endpoint, and this uses Lambda to provide the backing compute. First you will create the supporting Lambda function and then the API Gateway.

- **Create the Lambda Function.**
 - Search Lambda and click on Create Function.
 - Select Author from scratch.
 - For Function name enter api_lambda .
 - For runtime click the dropdown and select Python 3.9.
 - Expand Change default execution role, select Use an existing role, click the Existing role dropdown and pick LambdaRole.

Create the function.

Create function [info](#)

Choose one of the following options to create your function.

- ☒ **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.

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.

Architecture [info](#)
Choose the instruction set architecture you want for your function code.
☒ **x86_64**
☐ arm64

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](#).

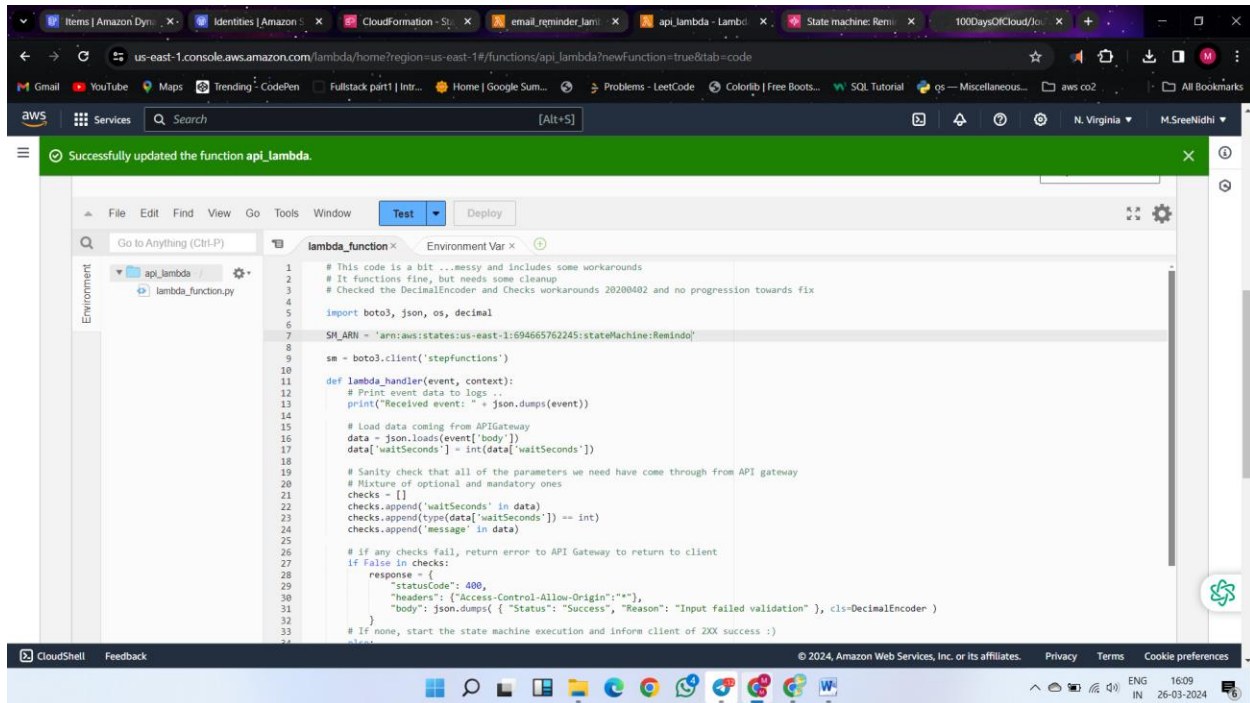
- ☐ Create a new role with basic Lambda permissions
- ☒ Use an existing role
- ☐ Create a new role from AWS policy templates

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 LAMBDA_ROLE-LambdaRole-oQJ85WwCrbh role](#) on the IAM console.

Advanced settings

[Cancel](#) [Create function](#)



- **Create an API Gateway.**
 - Click APIs on the menu on the left.
 - Locate the REST API box, and click Build.
 - Under Create new API ensure New API is selected. For API name* enter remindo
 - For Endpoint Type, pick Regional.
 - Click create API.
 - Click the Actions dropdown and Click Create Resource.
 - Under resource name enter remindo .
 - Make sure that Configure as proxy resource is NOT ticked.
 - Tick Enable API Gateway CORS.
 - Click Create Resource.
 - Ensure you have the /remindo resource selected, click Actions dropdown and click create method. In the small dropdown box which appears below /remindo select POST and click the tick symbol next to it.
 - Ensure for Integration Type that Lambda Function is selected.
 - Make sure us-east-1 is selected for Lambda Region. In the Lambda Function box, start typing api_lambda and it should autocomplete, click this auto complete (Make sure you pick api_lambda and not email reminder lambda).

- Make sure that Use Default Timeout box is ticked.
- Make sure that Use Lambda Proxy integration box is ticked.
- Click Save.
- Click Actions Dropdown and Deploy API.
- For Deployment Stage select New Stage. -For stage name and stage description enter prod.
- Click Deploy.

The screenshot shows the AWS Management Console interface for creating a new REST API. The browser address bar indicates the URL is `us-east-1.console.aws.amazon.com/apigateway/main/create-rest?experience=rest®ion=us-east-1`. The page title is "Create REST API".

API details

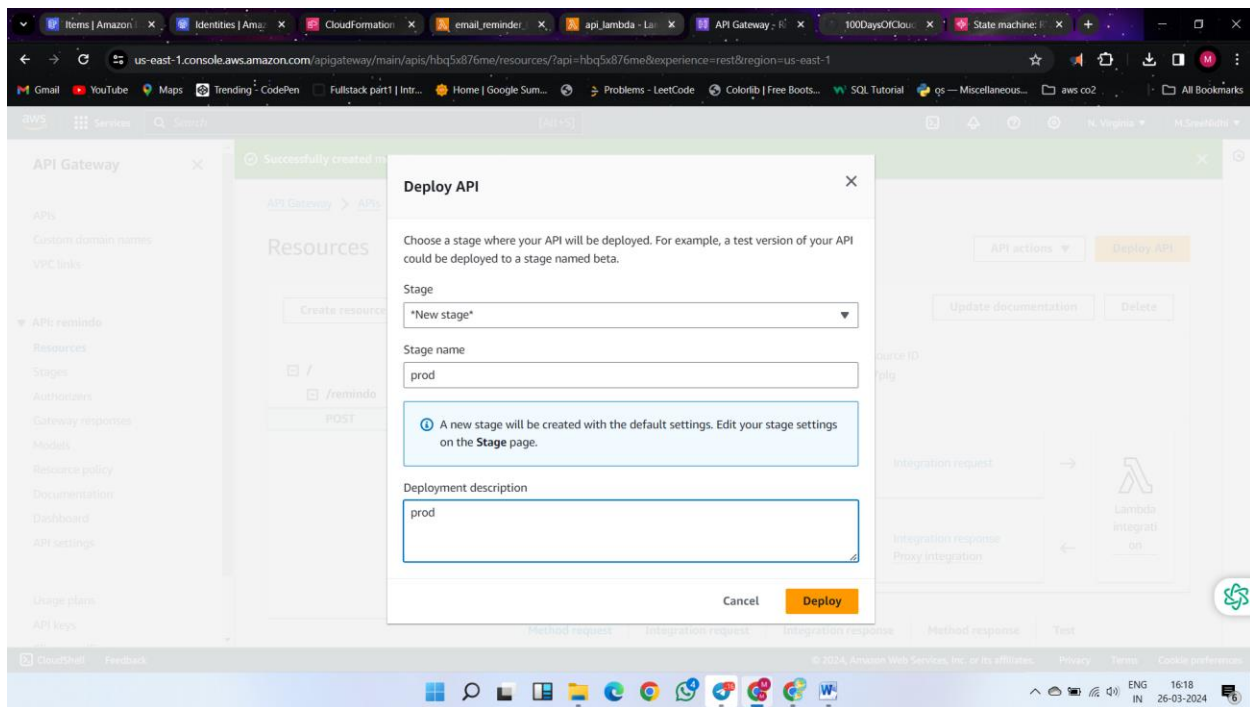
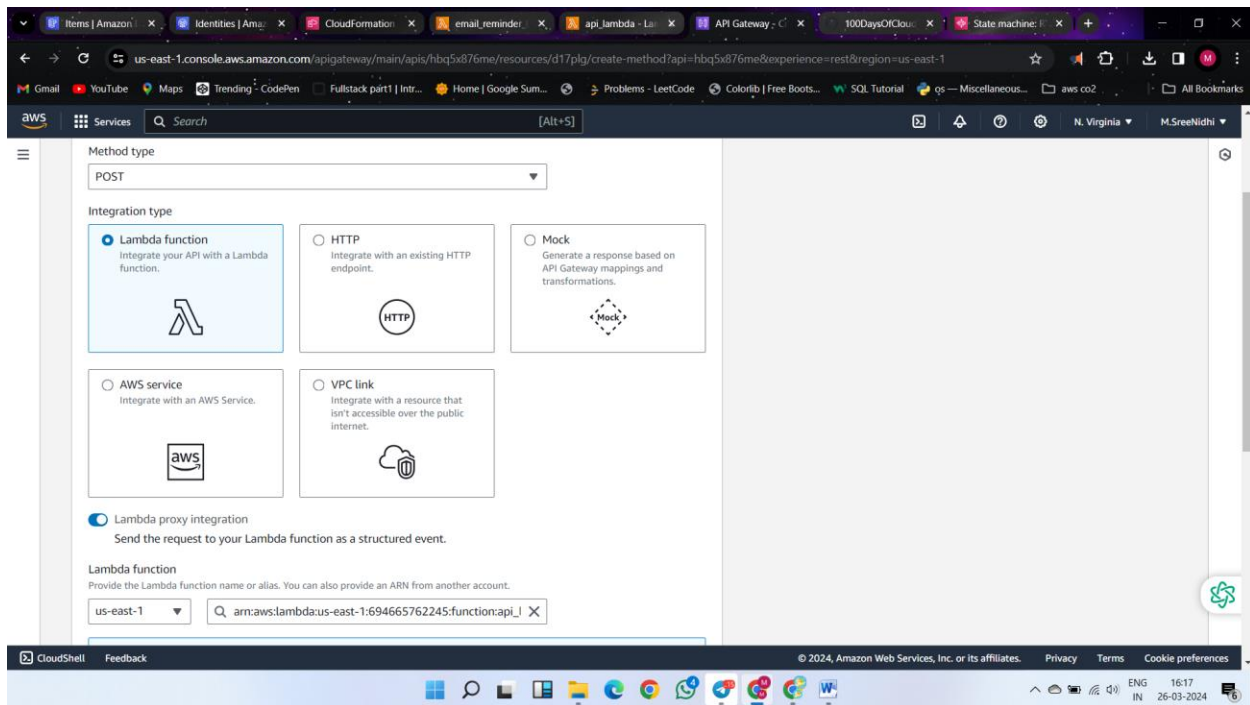
- ☒ **New API**
Create a new REST API.
- ☐ **Clone existing API**
Create a copy of an API in this AWS account.
- ☐ **Import API**
Import an API from an OpenAPI definition.
- ☐ **Example API**
Learn about API Gateway with an example API.

API name
remindq

Description - optional

API endpoint type
Regional APIs are deployed in the current AWS Region. Edge-optimized APIs route requests to the nearest CloudFront Point of Presence. Private APIs are only accessible from VPCs.
Regional

The bottom of the screen shows the Windows taskbar with various application icons and the system clock displaying 16:11 on 26-03-2024.



Step-5: Implement the static frontend application and test functionality

In this stage, you will create an S3 bucket and static website hosting which will host the application front end. You will download the source files for the front end,

configure them to connect to your specific API gateway and then upload them to S3. Finally, you will run some application tests to verify its functionality.

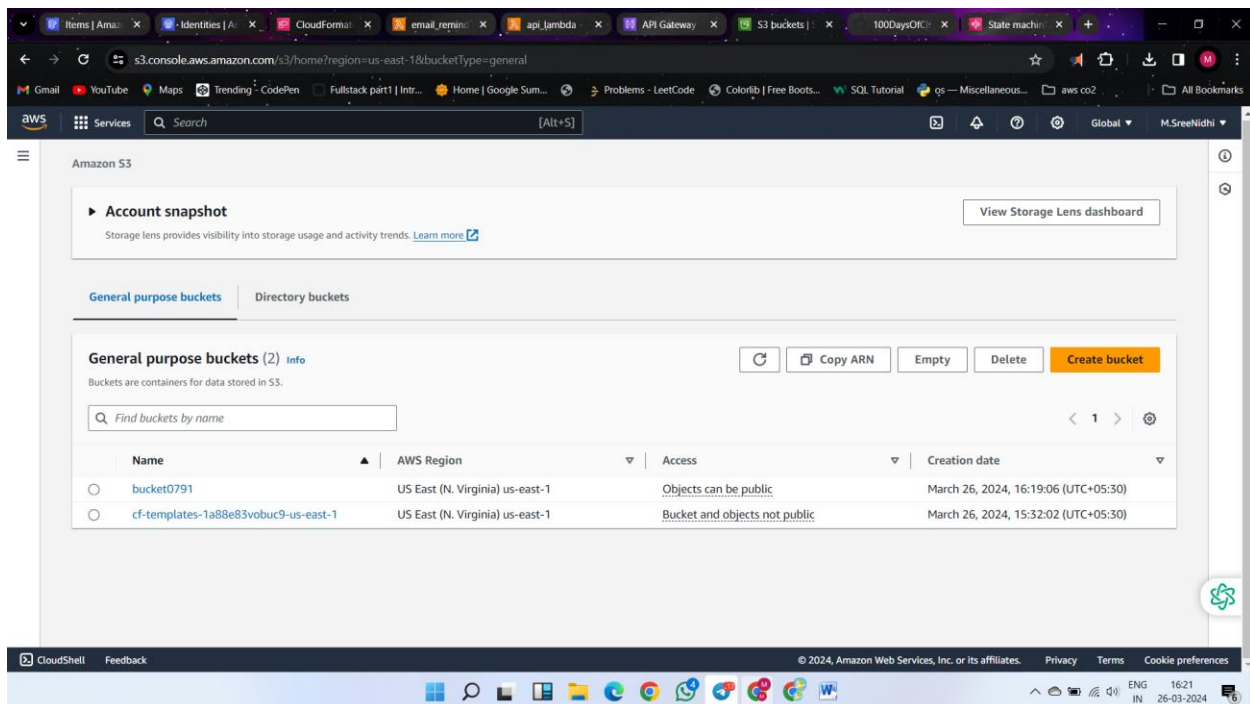
- **Create S3 Bucket.**

- Search S3. Click Create Bucket.
- Give a unique bucket name and ensure the region is set to US East (N.Virginia) us-east-1 .
- Uncheck Block all public access. Tick the box under Turning off block all public access might result in this bucket and the objects within becoming public.
- Create Bucket.
- Make it publicly accessible using a bucket policy.
- Go into the bucket you just created. Click the Permissions tab.
- Scroll down and in the Bucket Policy area, click Edit and in the box, paste the code below :

```
{  
    "Version":"2012-10-17",  
    "Statement":[  
        {  
            "Sid":"PublicRead",  
            "Effect":"Allow",  
            "Principal": "*",  
            "Action":["s3:GetObject"],  
            "Resource":["REPLACEME_BUCKET_ARN/*"]  
        }  
    ]  
}
```

- Replace REPLACEME_BUCKET_ARN with your bucket ARN and save changes.

- **Enable Static Hosting.**
 - Click on the Properties Tab.
 - Scroll down and locate Static website hosting and click Edit.
 - Select Enable Select Host a static website.
 - For both Index Document and Error Document enter index.html.
 - Click Save Changes.



Items | Ama... | Identities | A... | Cloudformat... | email_remin... | api_lambda... | API Gateways... | Edit bucket p... | 100DaysOfC... | State machi... | +

s3.console.aws.amazon.com/s3/bucket/bucket0791/property/policy/edit?region=us-east-1&bucketType=general

1 {
2 "Version": "2012-10-17",
3 "Statement": [
4 {
5 "Sid": "PublicRead",
6 "Effect": "Allow",
7 "Principal": "*",
8 "Action": ["s3:GetObject"],
9 "Resource": ["arn:aws:s3::bucket0791/*"]
10 }
11]
12 }

PublicRead Remove

Add actions

Choose a service

Filter services

Included

S3

Available

AMP

API Gateway

API Gateway V2

ASC

Access Analyzer

Account

Activate

Alexa for Business

Add a resource Add

Add a condition (optional) Add

+ Add new statement

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

ENG IN 16:22 26-03-2024

Items | Ama... | Identities | A... | Cloudformat... | email_remin... | api_lambda... | API Gateways... | Edit static w... | 100DaysOfC... | State machi... | +

s3.console.aws.amazon.com/s3/bucket/bucket0791/property/website/edit?region=us-east-1&bucketType=general

Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting

☐ Disable

☒ Enable

Hosting type

☒ Host a static website

Use the bucket endpoint as the web address. [Learn more](#)

☐ Redirect requests for an object

Redirect requests to another bucket or domain. [Learn more](#)

For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see [Using Amazon S3 Block Public Access](#)

Index document

Specify the home or default page of the website.

index.html

Error document - optional

This is returned when an error occurs.

index.html

Redirection rules - optional

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

ENG IN 16:23 26-03-2024

us-east-1.console.aws.amazon.com/apigateway/main/apis/hbq5x876me/stages?api=hbq5x876me&experience=rest®ion=us-east-1

API Gateway

APIs
Custom domain names
VPC links

▼ API: remindo

- Resources
- Stages**
- Authorizers
- Gateway responses
- Models
- Resource policy
- Documentation
- Dashboard
- API settings

Usage plans
API keys

Stages

Stage actions ▼ **Create stage**

prod

Stage details info

Stage name
prod

Rate
-

Web ACL
-

Cache cluster
Inactive

Burst
-

Client certificate
-

Default method-level caching
-

Copy
https://hbq5x876me.execute-api.us-east-1.amazonaws.com/prod

Active deployment
kogqgb on March 26, 2024, 16:17 (UTC+05:30)

Logs and tracing info

Edit

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

s3.console.aws.amazon.com/s3/buckets/bucket0791?region=us-east-1&bucketType=general&tab=objects

Amazon S3

Buckets > bucket0791

bucket0791

Objects Properties Permissions Metrics Management Access Points

Objects (4) info

Copy S3 URI Copy URL Download Open Delete Actions ▼ Create folder Upload

Find objects by prefix

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	index.html	html	March 26, 2024, 16:27:01 (UTC+05:30)	2.9 KB	Standard
<input type="checkbox"/>	main.css	css	March 26, 2024, 16:27:02 (UTC+05:30)	970.0 B	Standard
<input type="checkbox"/>	remindo.png	png	March 26, 2024, 16:27:03 (UTC+05:30)	19.9 KB	Standard
<input type="checkbox"/>	serverless.js	js	March 26, 2024, 16:27:04 (UTC+05:30)	1.6 KB	Standard

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

