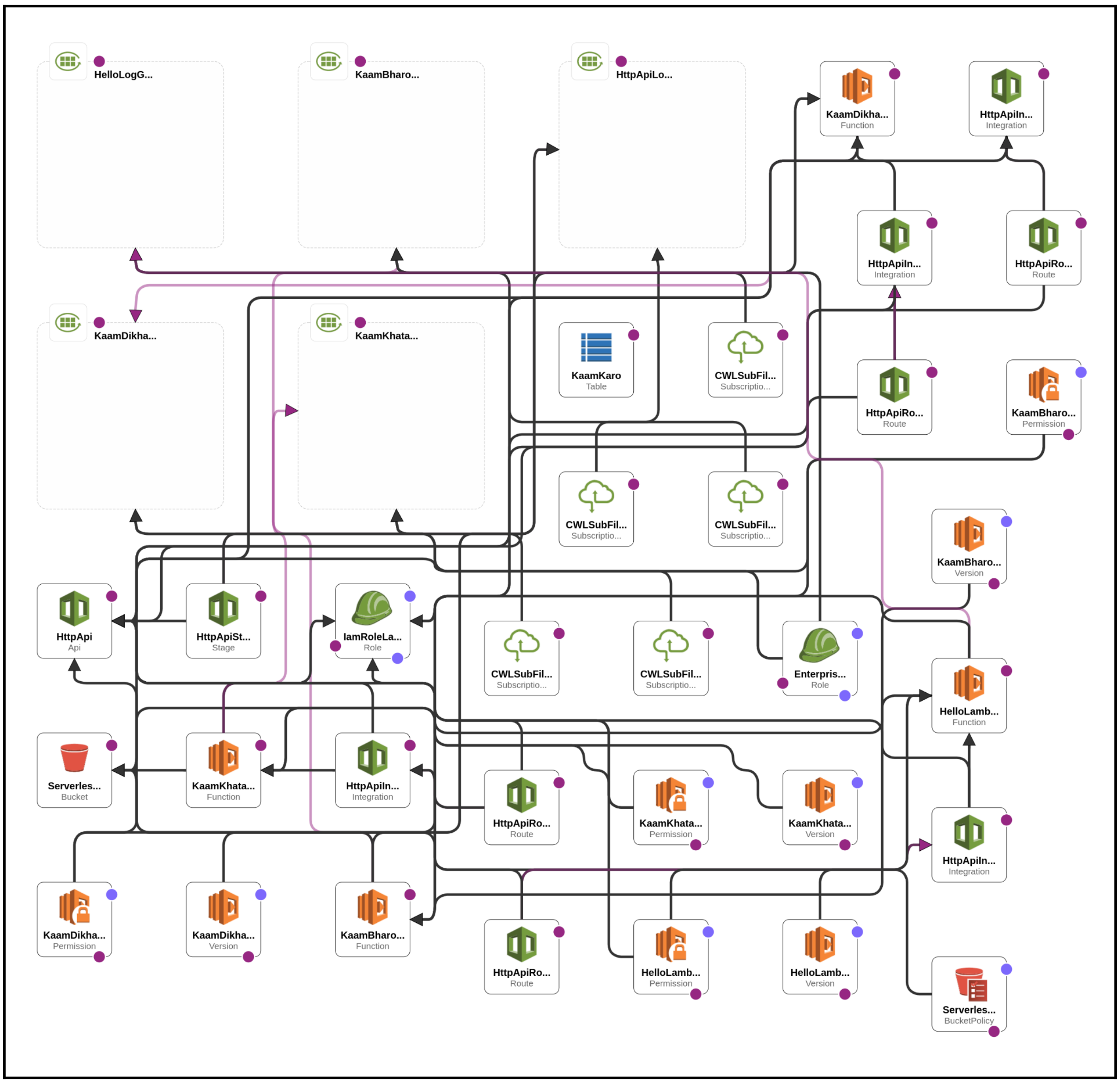
Project Name : Deployed serverless backend API based application using AWS cloudformation for provisioning & decommissioning of Infrastructure as code

Required services : AWS Cloud , Cloud-Formation , IAM , API Gateway , Cloud Watch,Serverless framework , EC2 ,DynamoDB

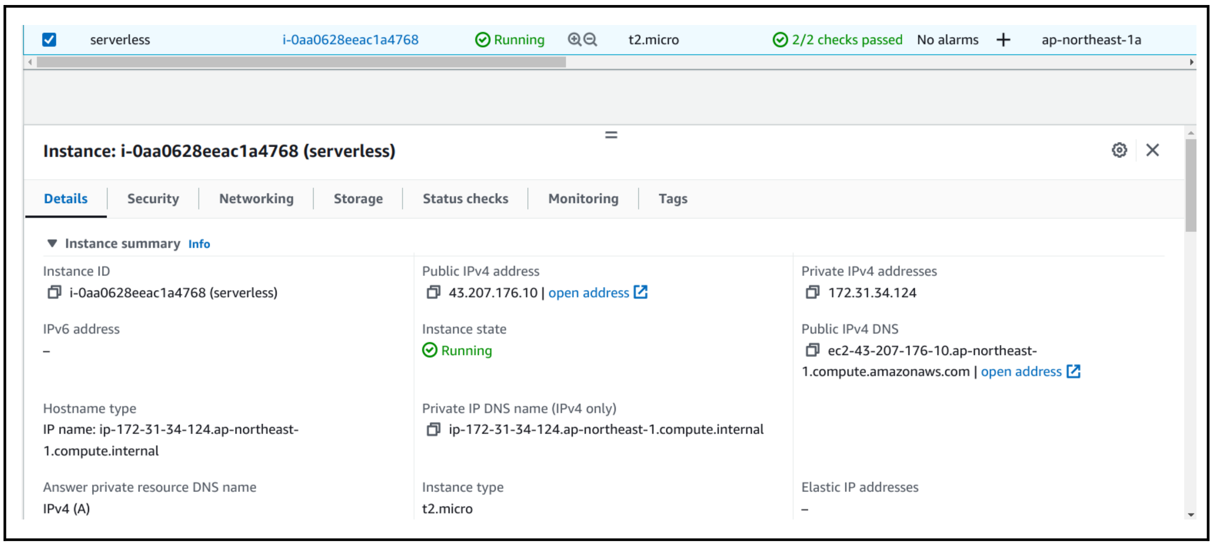
Architecture :



Required Steps :

1. Let’s make an EC2 instance & connect first . (image : ubuntu with t2.micro free tier)

Details :



	Name	Instance ID	State		Type	Status check	Alarm		Availability Zone
☑	serverless	i-0aa0628eeac1a4768	✓ Running		t2.micro	✓ 2/2 checks passed	No alarms	+	ap-northeast-1a

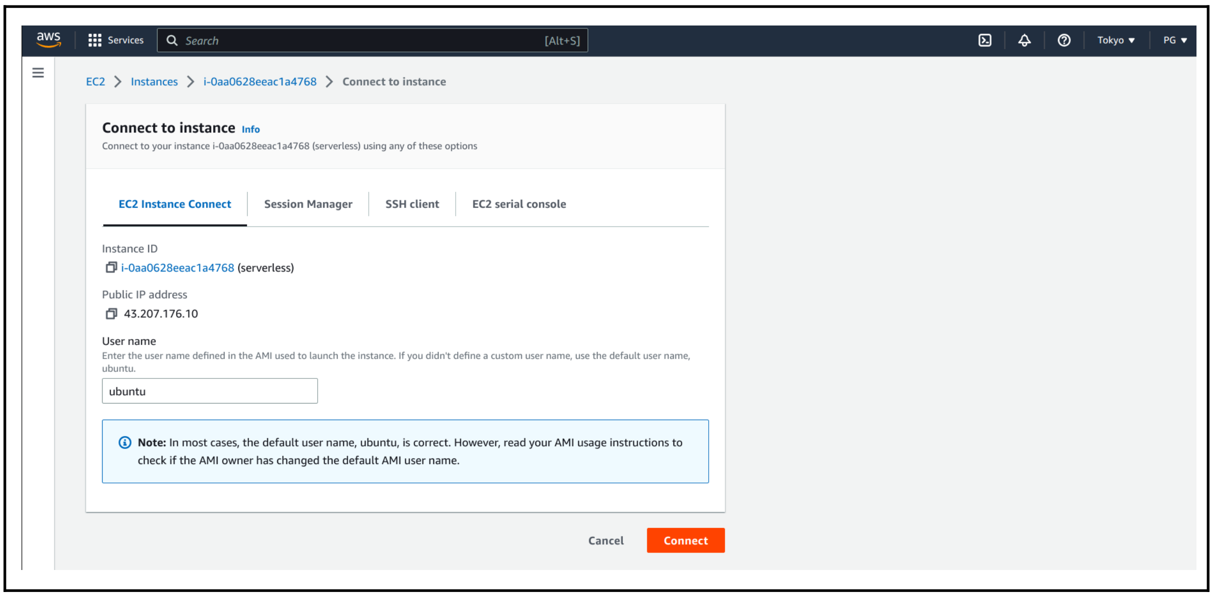
Instance: i-0aa0628eeac1a4768 (serverless)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

▼ Instance summary Info

Instance ID i-0aa0628eeac1a4768 (serverless)	Public IPv4 address 43.207.176.10 | open address	Private IPv4 addresses 172.31.34.124
IPv6 address –	Instance state ✓ Running	Public IPv4 DNS ec2-43-207-176-10.ap-northeast-1.compute.amazonaws.com | open address
Hostname type IP name: ip-172-31-34-124.ap-northeast-1.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-34-124.ap-northeast-1.compute.internal	
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	Elastic IP addresses –

In order to connect to an instance select ec2 > click connect > choose connect > open cli in browser , similarly you can connect using ssh as well



EC2 > Instances > i-0aa0628eeac1a4768 > Connect to instance

Connect to instance Info

Connect to your instance i-0aa0628eeac1a4768 (serverless) using any of these options

EC2 Instance Connect | Session Manager | SSH client | EC2 serial console

Instance ID
i-0aa0628eeac1a4768 (serverless)

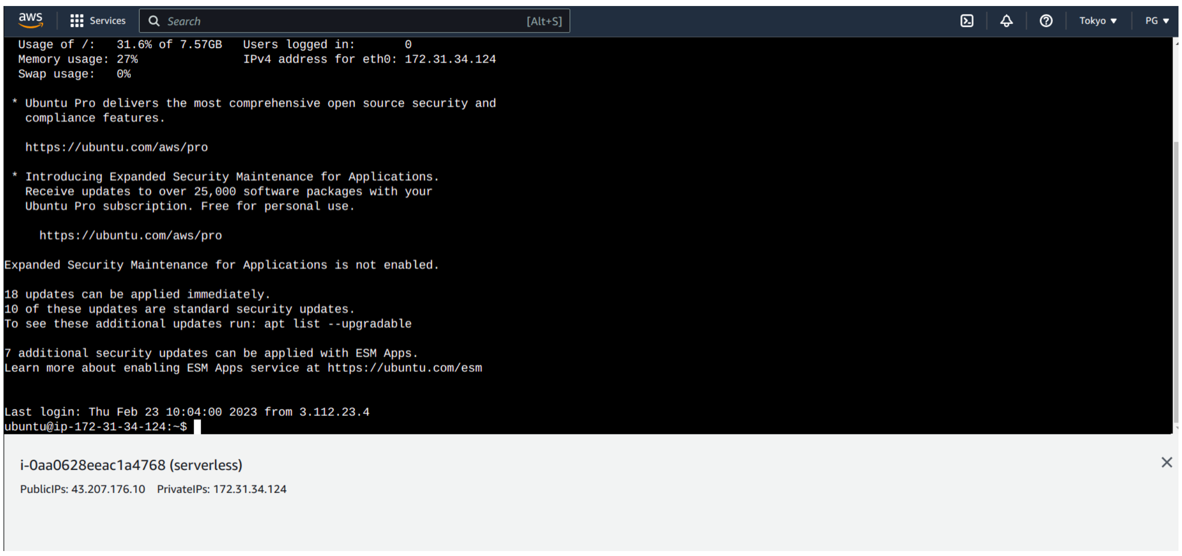
Public IP address
43.207.176.10

User name
Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ubuntu.

ubuntu

Note: In most cases, the default user name, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Cancel | Connect



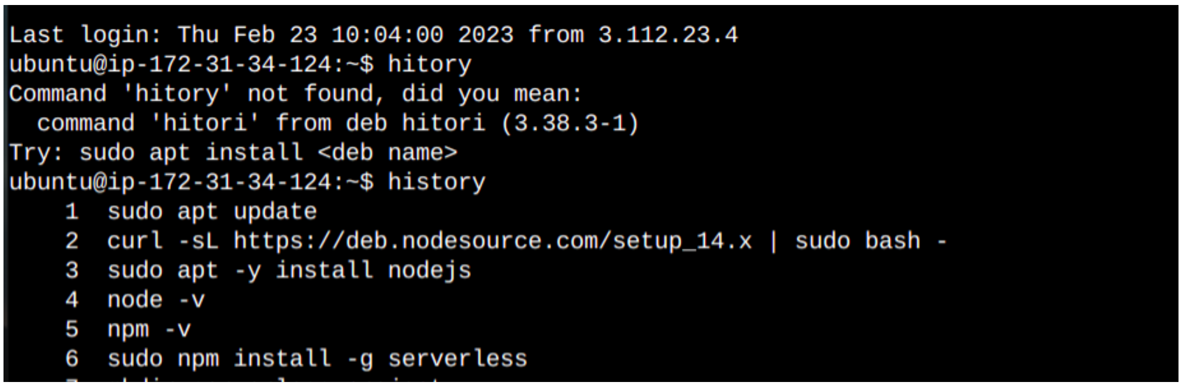
```
  Usage of /:   31.6% of 7.57GB   Users logged in:       0
  Memory usage: 27%               IPv4 address for eth0: 172.31.34.124
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

     https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

18 updates can be applied immediately.
10 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

7 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


Last login: Thu Feb 23 10:04:00 2023 from 3.112.23.4
ubuntu@ip-172-31-34-124:~$
```

i-0aa0628eeac1a4768 (serverless)

PublicIPs: 43.207.176.10   PrivateIPs: 172.31.34.124

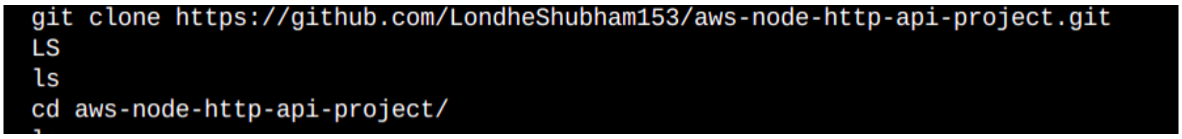
Here am connected

2. Since this project will be serverless we need serverless to be installed on our EC2 instance for that we are using following commands on EC2 cli



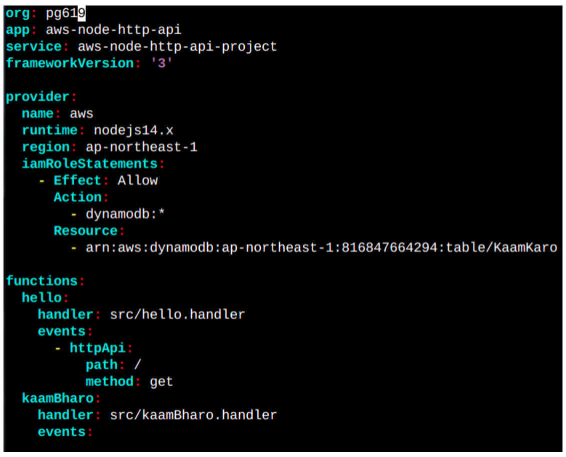
```
Last login: Thu Feb 23 10:04:00 2023 from 3.112.23.4
ubuntu@ip-172-31-34-124:~$ hitory
Command 'hitory' not found, did you mean:
  command 'hitori' from deb hitori (3.38.3-1)
Try: sudo apt install <deb name>
ubuntu@ip-172-31-34-124:~$ history
    1  sudo apt update
    2  curl -sL https://deb.nodesource.com/setup_14.x | sudo bash -
    3  sudo apt -y install nodejs
    4  node -v
    5  npm -v
    6  sudo npm install -g serverless
```

3. Now let's clone the application repository from github for this using [aws-node-todo-application]() code . inside a folder that we have created already on ec2 for serverless .

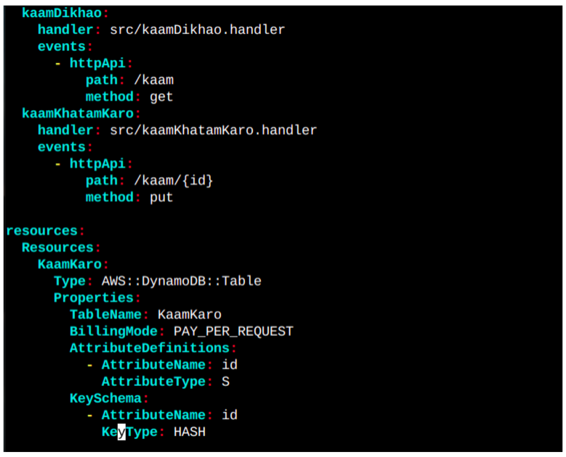


```
  git clone https://github.com/LondheShubham153/aws-node-http-api-project.git
  LS
  ls
  cd aws-node-http-api-project/
```

And we can see all folders inside this repo and also for dynamoDB i have created serverless.yml file where i have specified db config along with bucket details .



```yaml
org: pg619
app: aws-node-http-api
service: aws-node-http-api-project
frameworkVersion: '3'

provider:
  name: aws
  runtime: nodejs14.x
  region: ap-northeast-1
  iamRoleStatements:
    - Effect: Allow
      Action:
        - dynamodb:*
      Resource:
        - arn:aws:dynamodb:ap-northeast-1:816847664294:table/KaamKaro

functions:
  hello:
    handler: src/hello.handler
    events:
      - httpApi:
          path: /
          method: get
  kaamBharo:
    handler: src/kaamBharo.handler
    events:
```



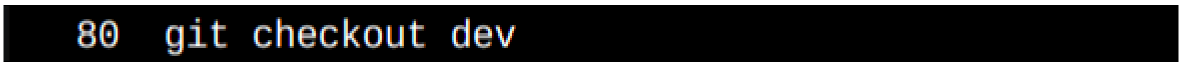
```yaml
  kaamDikhao:
    handler: src/kaamDikhao.handler
    events:
      - httpApi:
          path: /kaam
          method: get
  kaamKhatamKaro:
    handler: src/kaamKhatamKaro.handler
    events:
      - httpApi:
          path: /kaam/{id}
          method: put

resources:
  Resources:
    KaamKaro:
      Type: AWS::DynamoDB::Table
      Properties:
        TableName: KaamKaro
        BillingMode: PAY_PER_REQUEST
        AttributeDefinitions:
          - AttributeName: id
            AttributeType: S
        KeySchema:
          - AttributeName: id
            KeyType: HASH
```

So here I have defined : DynamoDB table name , billing-mode,attribute-type , key_type along with region , code handler authentication as well.

Now let's try to deploy this in order to make DynamoDB table for that : sls deploy

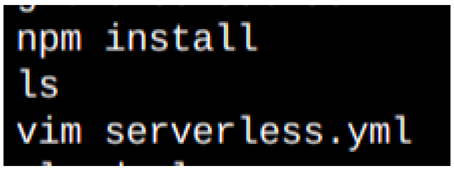
command required and it will give us api's that have specified in serverless.yml file

4. Now in github since main codes are in dev branch we need to switch to dev branch using : git checkout dev



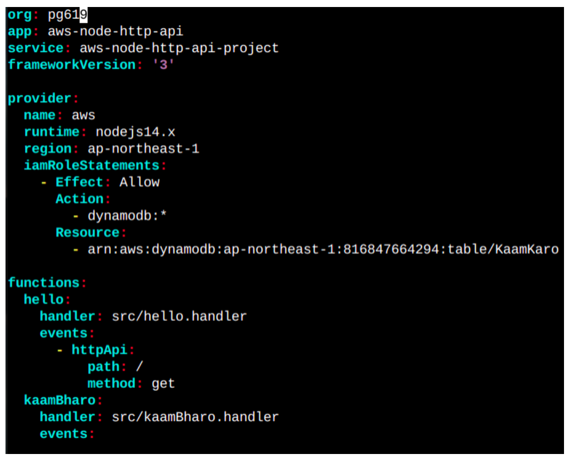
```
80  git checkout dev
```

5. Now since in the code uuid & aws-sdk services used hence we need to configure our dev branch with dependencies for that **npm install** command required :

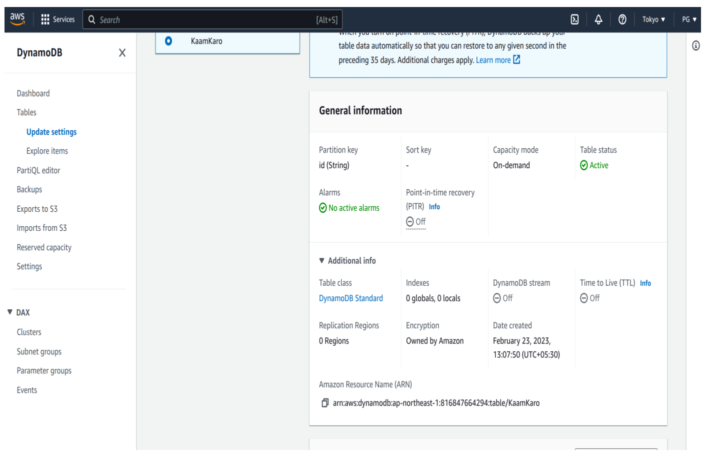


```
npm install
ls
vim serverless.yml
```

Now let's make some changes in serverless.yaml as well with DynamoDB table details like region and ARN



```yaml
org: pg619
app: aws-node-http-api
service: aws-node-http-api-project
frameworkVersion: '3'

provider:
  name: aws
  runtime: nodejs14.x
  region: ap-northeast-1
  iamRoleStatements:
    - Effect: Allow
      Action:
        - dynamodb:*
      Resource:
        - arn:aws:dynamodb:ap-northeast-1:816847664294:table/KaamKaro

functions:
  hello:
    handler: src/hello.handler
    events:
      - httpApi:
          path: /
          method: get
  kaamBharo:
    handler: src/kaamBharo.handler
    events:
```



KaamKaro

table data automatically so that you can restore to any given second in the preceding 35 days. Additional charges apply. Learn more

General information

Partition key: id (String)
Sort key: -
Capacity mode: On-demand
Table status: Active
Alarms: No active alarms
Point-in-time recovery (PITR) Info: Off

Additional info

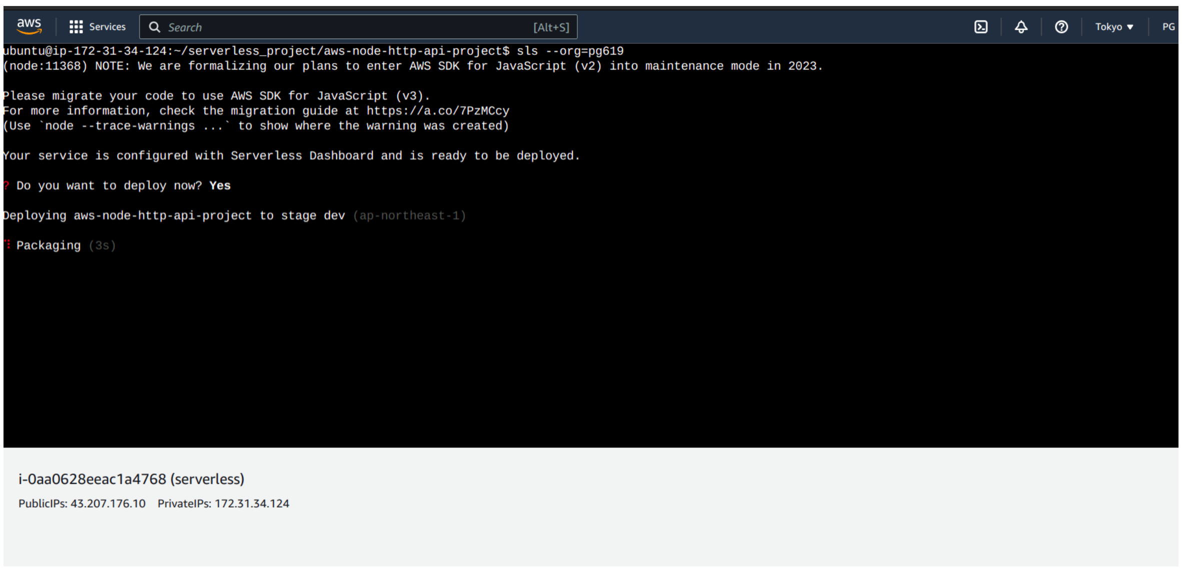
Table class: DynamoDB Standard
Indexes: 0 globals, 0 locals
DynamoDB stream: Off
Time to Live (TTL) Info: Off
Replication Regions: 0 Regions
Encryption: Owned by Amazon
Date created: February 23, 2023, 13:07:50 (UTC+05:30)

Amazon Resource Name (ARN)
arn:aws:dynamodb:ap-northeast-1:816847664294:table/KaamKaro

6. Now let’s deploy this on serverless : (I already created account in serverless framework website and have set my org with pg619 which is specified in serverless.yml file and required for aws-cli login and deploy)

For this command : **sls –org=pg619 > choose new api web project > specified your name and details and deploy as Y**

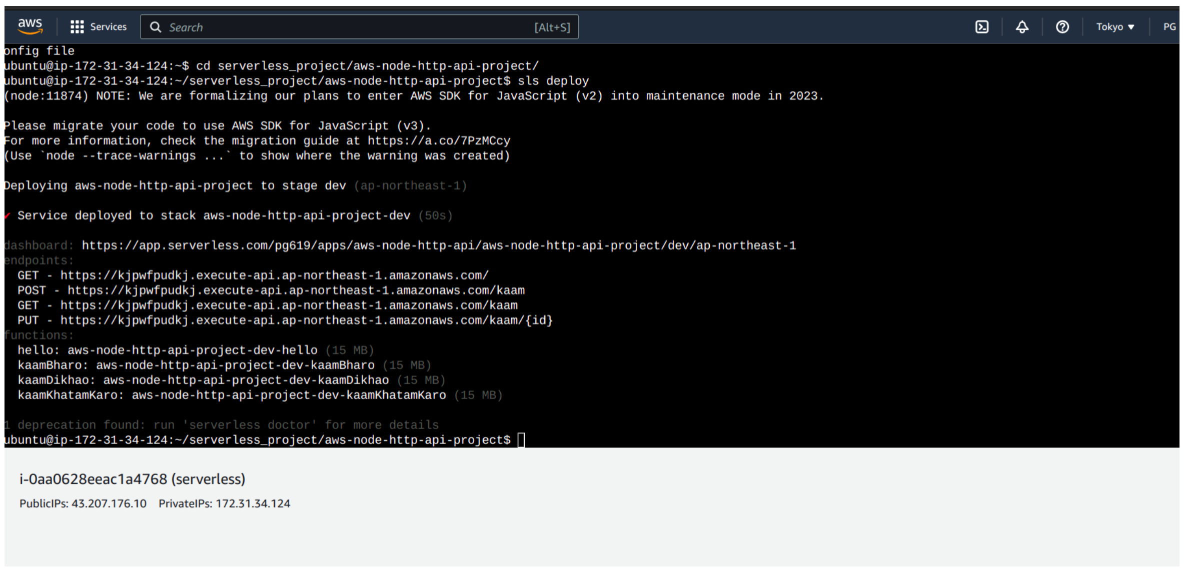
It will create cloudFormation stack and validate details :



```
ubuntu@ip-172-31-34-124:~/serverless_project/aws-node-http-api-project$ sls --org=pg619
(node:11368) NOTE: We are formalizing our plans to enter AWS SDK for JavaScript (v2) into maintenance mode in 2023.

Please migrate your code to use AWS SDK for JavaScript (v3).
For more information, check the migration guide at https://a.co/7PzMCcy
(Use `node --trace-warnings ...` to show where the warning was created)

Your service is configured with Serverless Dashboard and is ready to be deployed.

? Do you want to deploy now? Yes

Deploying aws-node-http-api-project to stage dev (ap-northeast-1)

⠋ Packaging (3s)
```

i-0aa0628eeac1a4768 (serverless)
PublicIPs: 43.207.176.10   PrivateIPs: 172.31.34.124

7. Now let’s deploy into a serverless dashboard using **sls deploy** where we will get a specified 4 api link using that we need to test on postman .



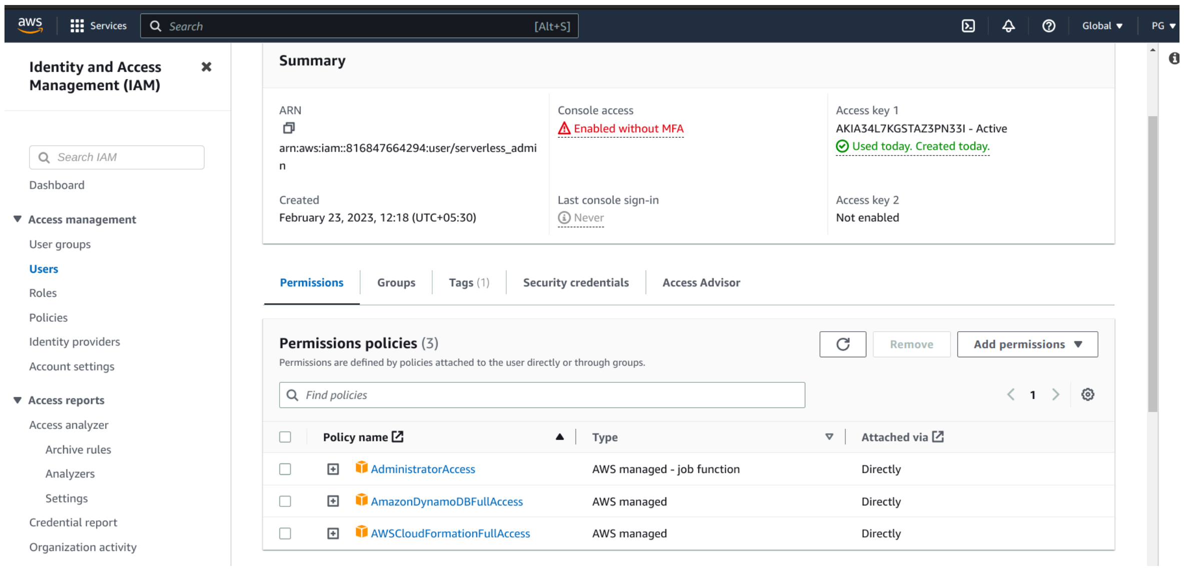
```
onfig file
ubuntu@ip-172-31-34-124:~$ cd serverless_project/aws-node-http-api-project/
ubuntu@ip-172-31-34-124:~/serverless_project/aws-node-http-api-project$ sls deploy
(node:11874) NOTE: We are formalizing our plans to enter AWS SDK for JavaScript (v2) into maintenance mode in 2023.

Please migrate your code to use AWS SDK for JavaScript (v3).
For more information, check the migration guide at https://a.co/7PzMCcy
(Use `node --trace-warnings ...` to show where the warning was created)

Deploying aws-node-http-api-project to stage dev (ap-northeast-1)

✓ Service deployed to stack aws-node-http-api-project-dev (50s)

dashboard: https://app.serverless.com/pg619/apps/aws-node-http-api/aws-node-http-api-project/dev/ap-northeast-1
endpoints:
  GET - https://kjpwfpudkj.execute-api.ap-northeast-1.amazonaws.com/
  POST - https://kjpwfpudkj.execute-api.ap-northeast-1.amazonaws.com/kaam
  GET - https://kjpwfpudkj.execute-api.ap-northeast-1.amazonaws.com/kaam
  PUT - https://kjpwfpudkj.execute-api.ap-northeast-1.amazonaws.com/kaam/{id}
functions:
  hello: aws-node-http-api-project-dev-hello (15 MB)
  kaamBharo: aws-node-http-api-project-dev-kaamBharo (15 MB)
  kaamDikhao: aws-node-http-api-project-dev-kaamDikhao (15 MB)
  kaamKhatamKaro: aws-node-http-api-project-dev-kaamKhatamKaro (15 MB)

1 deprecation found: run 'serverless doctor' for more details
ubuntu@ip-172-31-34-124:~/serverless_project/aws-node-http-api-project$
```

i-0aa0628eeac1a4768 (serverless)

PublicIPs: 43.207.176.10 PrivateIPs: 172.31.34.124

As you can see we got GET , POST , PUT , GET api endpoints

8. Now when we created user we have given permission like :



Identity and Access Management (IAM)

Summary

ARN: arn:aws:iam::816847664294:user/serverless_admin

Created: February 23, 2023, 12:18 (UTC+05:30)

Console access: Enabled without MFA

Last console sign-in: Never

Access key 1: AKIA34L7KGSTAZ3PN33I - Active, Used today. Created today.

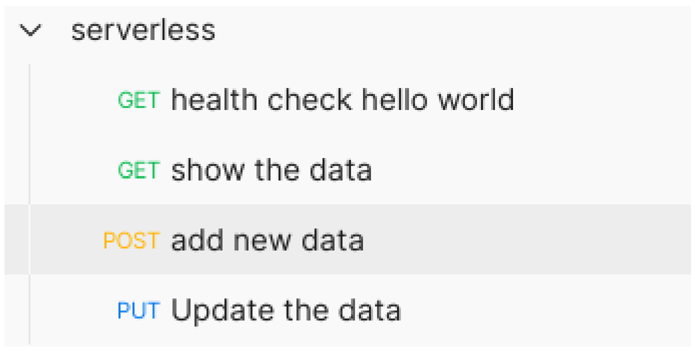
Access key 2: Not enabled

Permissions policies (3)

Permissions are defined by policies attached to the user directly or through groups.

Policy name	Type	Attached via
AdministratorAccess	AWS managed - job function	Directly
AmazonDynamoDBFullAccess	AWS managed	Directly
AWSCloudFormationFullAccess	AWS managed	Directly

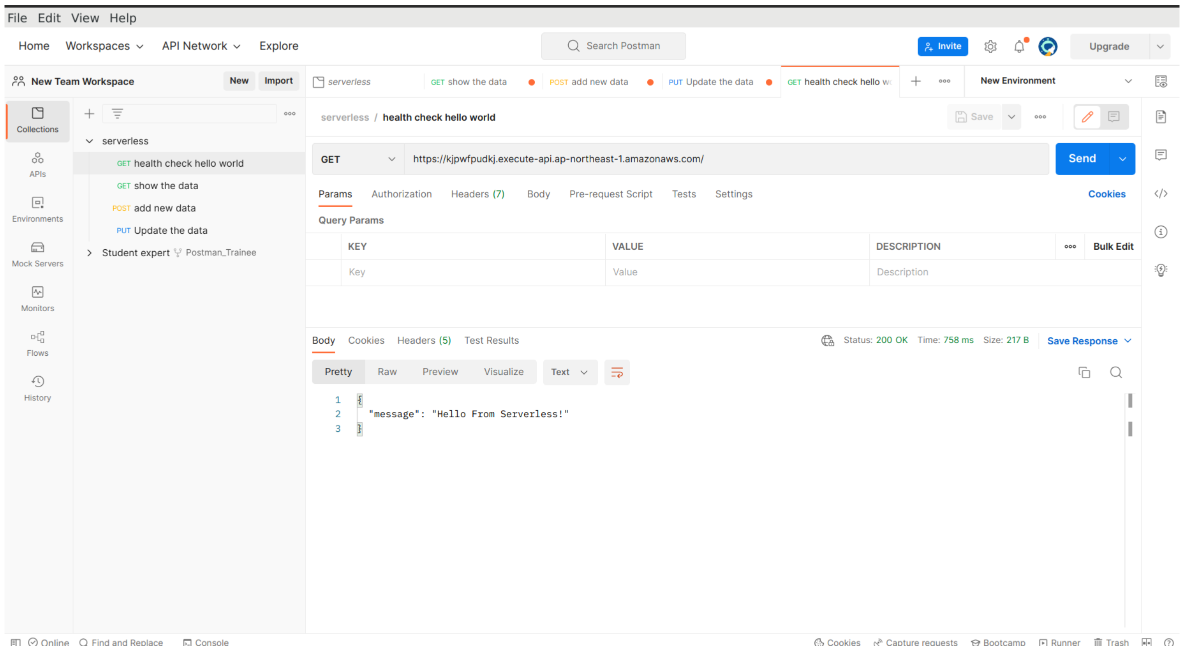
Let’s test POST method in postman using some sample payload , for that open postman > add collection > under collection specify 4 API Endpoints like this way :



serverless
- GET health check hello world
- GET show the data
- POST add new data
- PUT Update the data

Now let’s manipulate api endpoints :

GET Method :

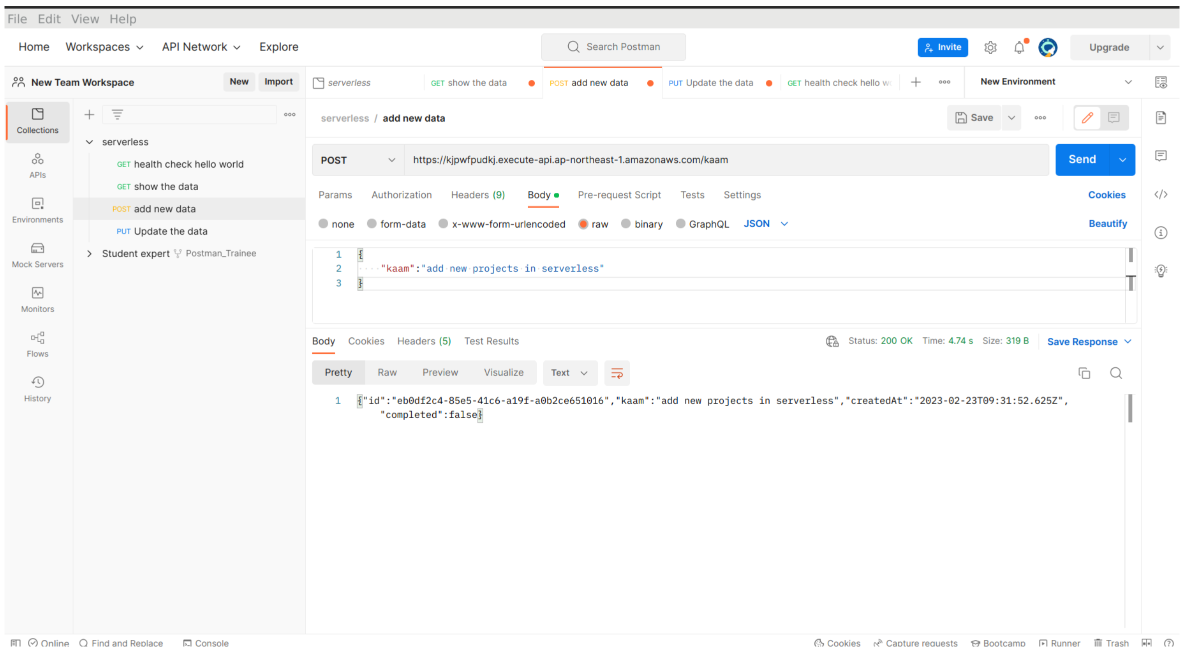

File Edit View Help
Home Workspaces API Network Explore
Search Postman
Invite
Upgrade
New Team Workspace New Import
serverless
GET show the data
POST add new data
PUT Update the data
GET health check hello w
New Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
serverless
GET health check hello world
GET show the data
POST add new data
PUT Update the data
Student expert Postman_Trainee
serverless / health check hello world
Save
Send
Params Authorization Headers (7) Body Pre-request Script Tests Settings
Cookies
Query Params
KEY VALUE DESCRIPTION Bulk Edit
Key Value Description
Body Cookies Headers (5) Test Results
Status: 200 OK Time: 758 ms Size: 217 B
Save Response
Pretty Raw Preview Visualize Text
Online Find and Replace Console
Cookies Capture requests Bootcamp Runner Trash

```
GET https://kjpwfpudkj.execute-api.ap-northeast-1.amazonaws.com/
```

```
{
  "message": "Hello From Serverless!"
}
```

Yes we are able to fetch details as you can see above mentioned pic.

POST Method :


File Edit View Help
Home Workspaces API Network Explore
Search Postman
Invite
Upgrade
New Team Workspace New Import
serverless
GET show the data
POST add new data
PUT Update the data
GET health check hello w
New Environment
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
serverless
GET health check hello world
GET show the data
POST add new data
PUT Update the data
Student expert Postman_Trainee
serverless / add new data
Save
Send
Params Authorization Headers (9) Body Pre-request Script Tests Settings
Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON
Beautify
Body Cookies Headers (5) Test Results
Status: 200 OK Time: 4.74 s Size: 319 B
Save Response
Pretty Raw Preview Visualize Text
Online Find and Replace Console
Cookies Capture requests Bootcamp Runner Trash

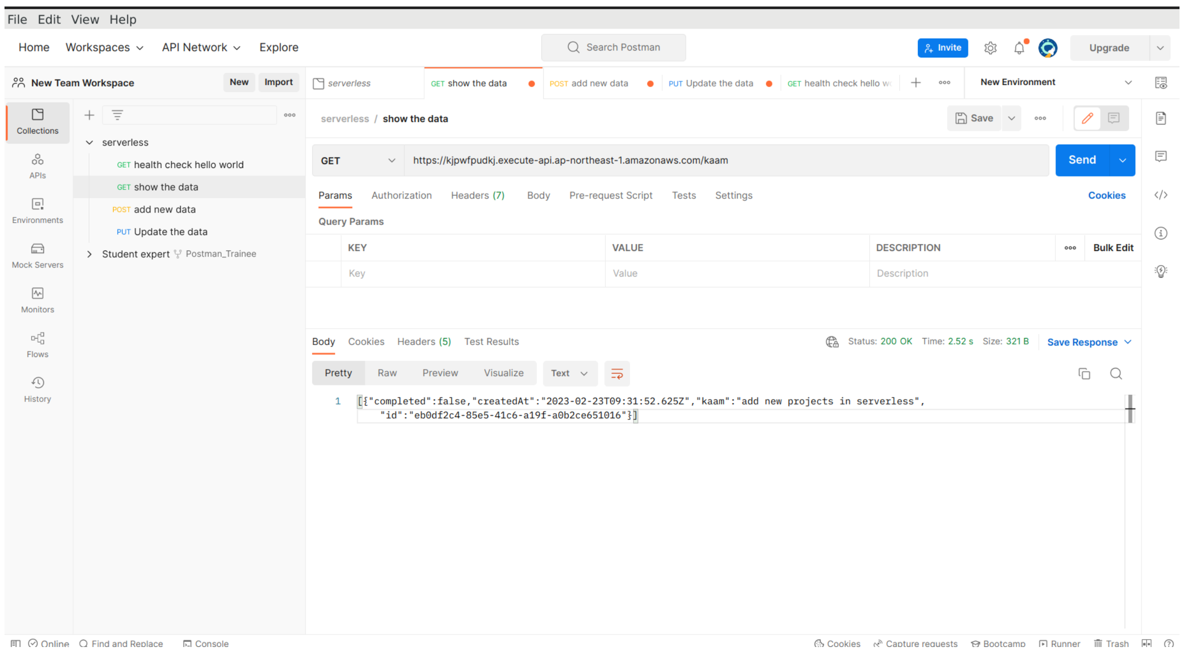
```
POST https://kjpwfpudkj.execute-api.ap-northeast-1.amazonaws.com/kaam
```

```
{
    "kaam":"add new projects in serverless"
}
```

```
{"id":"eb0df2c4-85e5-41c6-a19f-a0b2ce651016","kaam":"add new projects in serverless","createdAt":"2023-02-23T09:31:52.625Z",
    "completed":false}
```

Yes with POST method also we are able to push some data into endpoint

Now let’s check whole data :


File Edit View Help
Home Workspaces API Network Explore
Search Postman
Invite
Upgrade
New Team Workspace
New Import
Collections
APIs
Environments
Mock Servers
Monitors
Flows
History
serverless
GET health check hello world
GET show the data
POST add new data
PUT Update the data
Student expert Postman_Trainee
New Environment
Save
Send
Params Authorization Headers (7) Body Pre-request Script Tests Settings
Cookies
Query Params
KEY VALUE DESCRIPTION Bulk Edit
Pretty Raw Preview Visualize Text
Online Find and Replace Console
Cookies Capture requests Bootcamp Runner Trash

serverless / show the data

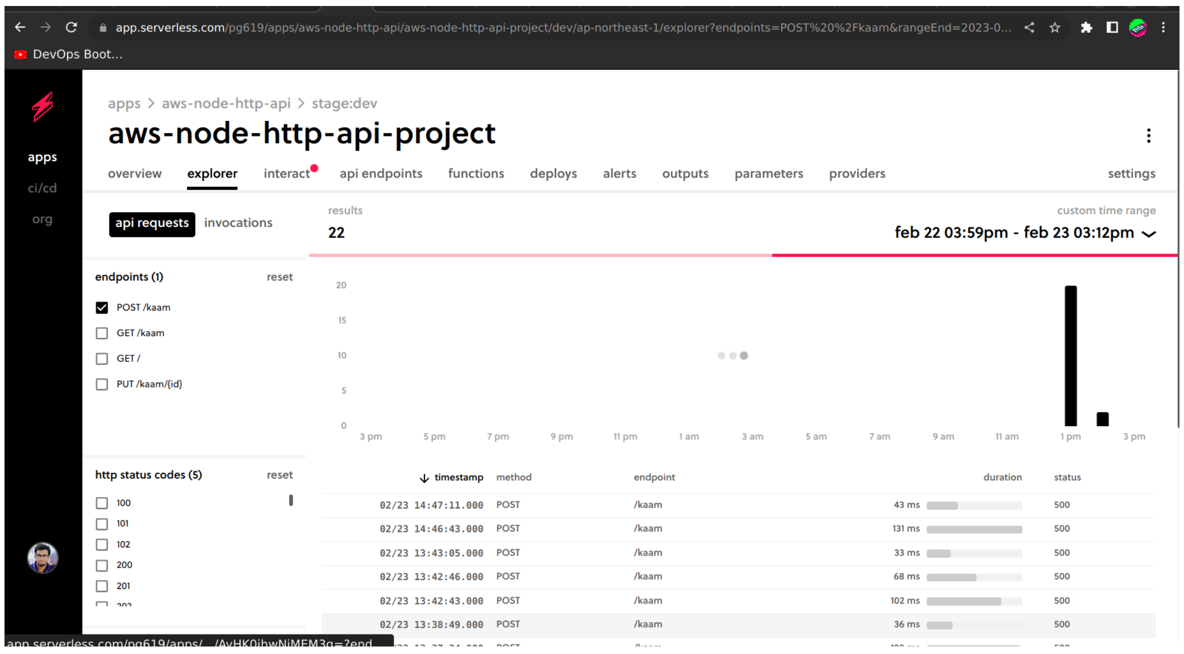
GET https://kjpwfpudkj.execute-api.ap-northeast-1.amazonaws.com/kaam

Body Cookies Headers (5) Test Results — Status: 200 OK Time: 2.52 s Size: 321 B

```
[{"completed":false,"createdAt":"2023-02-23T09:31:52.625Z","kaam":"add new projects in serverless",
  "id":"eb0df2c4-85e5-41c6-a19f-a0b2ce651016"}]
```

As you can see GET method also working fine and showing details

Now let’s check serverless framework dashboard :


DevOps Boot...
apps
ci/cd
org
overview explorer interact api endpoints functions deploys alerts outputs parameters providers settings
api requests invocations
reset
20
15
10
5
0
3 pm 5 pm 7 pm 9 pm 11 pm 1 am 3 am 5 am 7 am 9 am 11 am 1 pm 3 pm

app.serverless.com/pg619/apps/aws-node-http-api/aws-node-http-api-project/dev/ap-northeast-1/explorer?endpoints=POST%20%2Fkaam&rangeEnd=2023-0...

apps > aws-node-http-api > stage:dev

aws-node-http-api-project

results 22

custom time range feb 22 03:59pm - feb 23 03:12pm

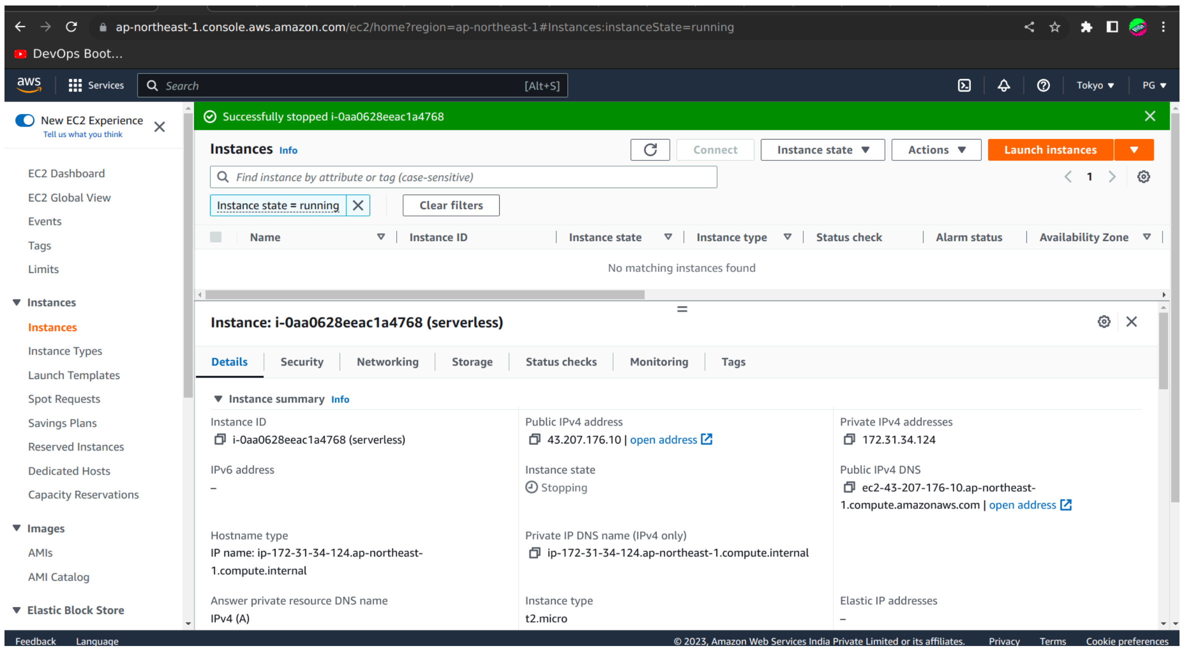
endpoints (1)
- [✓] POST /kaam
- [ ] GET /kaam
- [ ] GET /
- [ ] PUT /kaam/{id}

http status codes (5)
- [ ] 100
- [ ] 101
- [ ] 102
- [ ] 200
- [ ] 201

timestamp	method	endpoint	duration	status
02/23 14:47:11.000	POST	/kaam	43 ms	500
02/23 14:46:43.000	POST	/kaam	131 ms	500
02/23 13:43:05.000	POST	/kaam	33 ms	500
02/23 13:42:46.000	POST	/kaam	68 ms	500
02/23 13:42:43.000	POST	/kaam	102 ms	500
02/23 13:38:49.000	POST	/kaam	36 ms	500

app.serverless.com/pg619/apps/.../AvHK0ibwNiMEM3g=?end

9. Now for provisioning we need to stop our instance and need to check whether api endpoints are working or not : for this EC2 > select ec2 > instance state > stop


ap-northeast-1.console.aws.amazon.com/ec2/home?region=ap-northeast-1#Instances:instanceState=running

DevOps Boot...

Successfully stopped i-0aa0628eeac1a4768

Instances Info

Find instance by attribute or tag (case-sensitive)

Instance state = running | Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone

No matching instances found

Instance: i-0aa0628eeac1a4768 (serverless)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

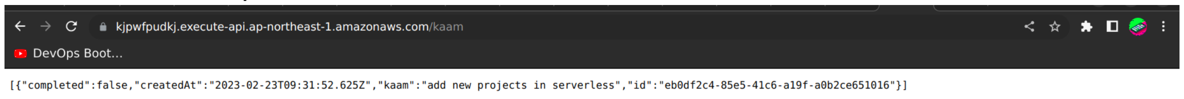
Instance summary Info

Instance ID i-0aa0628eeac1a4768 (serverless)	Public IPv4 address 43.207.176.10 | open address	Private IPv4 addresses 172.31.34.124
IPv6 address –	Instance state Stopping	Public IPv4 DNS ec2-43-207-176-10.ap-northeast-1.compute.amazonaws.com | open address
Hostname type IP name: ip-172-31-34-124.ap-northeast-1.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-34-124.ap-northeast-1.compute.internal	
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	Elastic IP addresses –

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

Stopped instance .

Now if i hit the endpoint to fetch details :


kjpwfpudkj.execute-api.ap-northeast-1.amazonaws.com/kaam

DevOps Boot...

```
[{"completed":false,"createdAt":"2023-02-23T09:31:52.625Z","kaam":"add new projects in serverless","id":"eb0df2c4-85e5-41c6-a19f-a0b2ce651016"}]
```

Yes serverless framework is working fine with provisioning status .

10. Now for decommissioning command required : sls remove so all cloudformation and stacks will be removed from aws cloud formation .