

Create Serverless application.

First install serverless package globally in your machine.

Npm install -g serverless

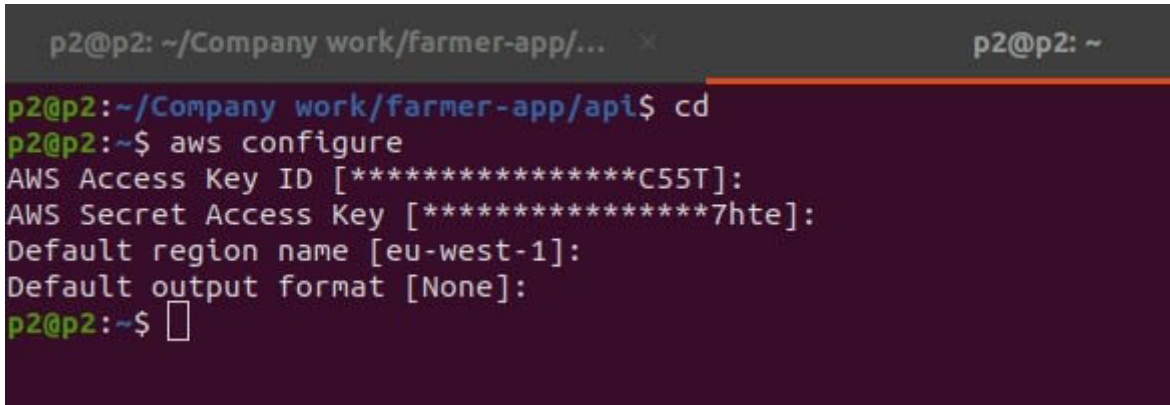
install aws-cli in your local machine.

Npm install -save aws-cli

Open terminal

type "aws configure"

set your aws account's secretkey and access key with region.

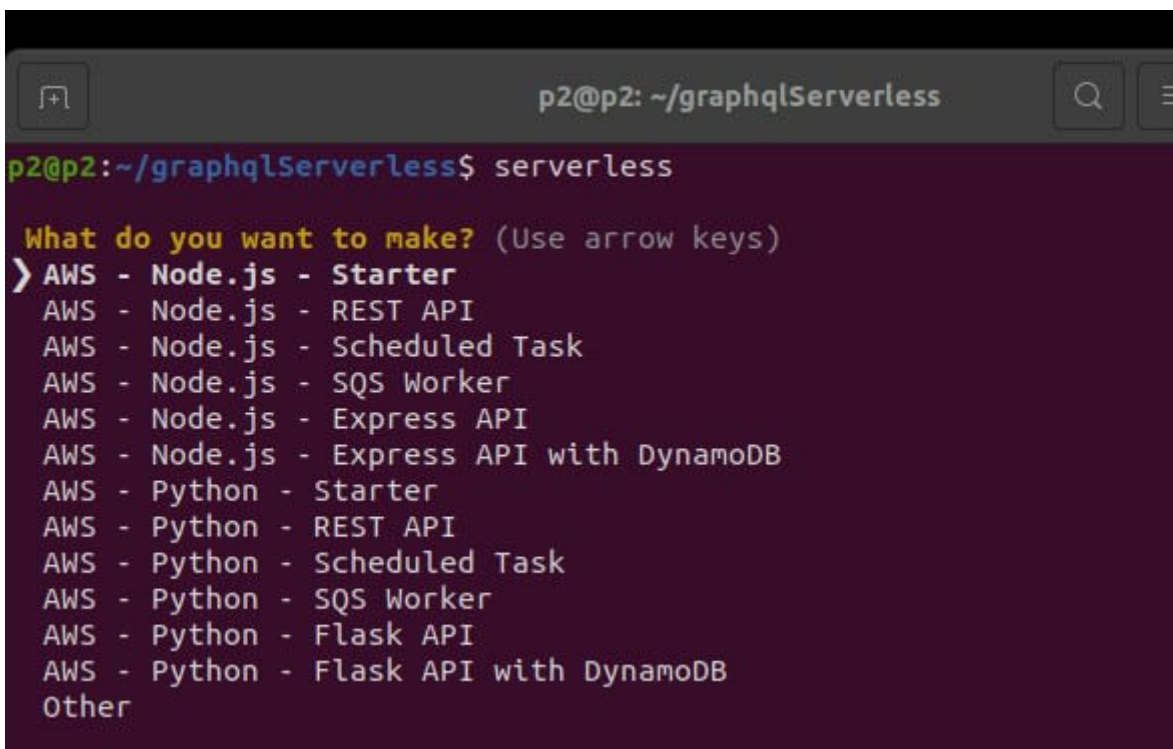
A terminal window with a dark background. The prompt is 'p2@p2: ~/Company work/farmer-app/...'. The user enters 'cd' and then 'aws configure'. The terminal shows the following prompts and inputs: 'AWS Access Key ID [*****C55T]:', 'AWS Secret Access Key [*****7hte]:', 'Default region name [eu-west-1]:', and 'Default output format [None]:'. The prompt returns to 'p2@p2: ~\$' with a cursor.

```
p2@p2: ~/Company work/farmer-app/... x p2@p2: ~
p2@p2:~/Company work/farmer-app/api$ cd
p2@p2:~$ aws configure
AWS Access Key ID [*****C55T]:
AWS Secret Access Key [*****7hte]:
Default region name [eu-west-1]:
Default output format [None]:
p2@p2:~$
```

Now open location when you create project.and open terminal

type "serverless"

and next follow the step.

A terminal window with a dark background. The prompt is 'p2@p2: ~/graphqlServerless'. The user enters 'serverless'. The terminal shows a list of options for 'What do you want to make? (Use arrow keys)'. The options are: 'AWS - Node.js - Starter', 'AWS - Node.js - REST API', 'AWS - Node.js - Scheduled Task', 'AWS - Node.js - SQS Worker', 'AWS - Node.js - Express API', 'AWS - Node.js - Express API with DynamoDB', 'AWS - Python - Starter', 'AWS - Python - REST API', 'AWS - Python - Scheduled Task', 'AWS - Python - SQS Worker', 'AWS - Python - Flask API', 'AWS - Python - Flask API with DynamoDB', and 'Other'. The prompt returns to 'p2@p2: ~/graphqlServerless\$' with a cursor.

```
p2@p2: ~/graphqlServerless
p2@p2:~/graphqlServerless$ serverless

What do you want to make? (Use arrow keys)
> AWS - Node.js - Starter
  AWS - Node.js - REST API
  AWS - Node.js - Scheduled Task
  AWS - Node.js - SQS Worker
  AWS - Node.js - Express API
  AWS - Node.js - Express API with DynamoDB
  AWS - Python - Starter
  AWS - Python - REST API
  AWS - Python - Scheduled Task
  AWS - Python - SQS Worker
  AWS - Python - Flask API
  AWS - Python - Flask API with DynamoDB
  Other
p2@p2:~/graphqlServerless$
```

```

z@p2:~/graphqlServerless$ serverless

What do you want to make? AWS - Node.js - Starter
What do you want to call this project? demo

Downloading "aws-node" template...

Project successfully created in demo folder

You are not logged in or you do not have a Serverless account.

Do you want to login/register to Serverless Dashboard? No

Do you want to deploy your project? No

Your project is ready for deployment and available in ./demo

Run serverless deploy in the project directory
  Deploy your newly created service

Run serverless info in the project directory after deployment
  View your endpoints and services

Run serverless invoke and serverless logs in the project directory after deployment
  Invoke your functions directly and view the logs

Run serverless in the project directory
  Add metrics, alerts, and a log explorer, by enabling the dashboard functionality

z@p2:~/graphqlServerless$ 

```

After successful install project.

Cd demo & open in VS code.

Now you can see yml file and handler.js file.

Now run your project use this command.

Sls deploy or serverless deploy

After successfully run project you have getting endpoint. And run into browser or postman.

Now you have to check your lambda function successfully run or not in your aws accounts.
Into lambda service.

And also check logs into cloundwatch.

If you have run project in local.

npm install serverless-offline --save-dev
plugins:

- serverless-offline

sls offline