LAMBDA

[I) Lý thuyết: 1](#_Toc25152808)

[II) Tải và cài đặt AWS Command Line Interface (dùng để gõ lệnh aws) 1](#_Toc25152809)

[III) Cách cài đặt serverless online và upload source code 1](#_Toc25152810)

[IV) Cách cài đặt serverless và dynamodb offline: 3](#_Toc25152811)

[V) Docker 5](#_Toc25152812)

[VI) Upload source Express lên lambda và connect aws rds mysql với navicat 5](#_Toc25152813)

**LAMBDA**

# I) Lý thuyết:

- Là 1 serverless (lập trình viên không phải cài đặt / cấu hình server - chỉ cần viết code)  
- Cho phép người dùng upload hoặc paste code lên và chạy.

# II) Tải và cài đặt AWS Command Line Interface (dùng để gõ lệnh aws)

1. Vào link: <https://aws.amazon.com/vi/cli/>
2. Có link tải cho Windows và Mac/Linux
3. Tiến hành cài đặt như bình thường, sau đó có thể mở cmd để sử dụng
4. Chạy lệnh sau để connect account amazon:

- aws configure

Các lệnh thông dụng của amazon:

* aws dynamodb list-tables // show list table trên amazon
* aws dynamodb list-tables --endpoint-url <http://localhost:8000> // chạy lệnh aws trên server local

# III) Cách cài đặt serverless online và upload source code

Chạy lệnh sau (cài đặt serverless):

npm install -g serverless

Sau đó chek xem đã cài được chưa:

**serverless -v hoặc sls –v**

**Tạo 1 project serverless:**

**sls create --template aws-nodejs --path name\_path --name name\_project**

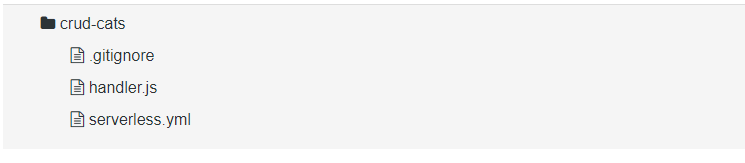
**Trong đó:**

**–template:** là tên của mẫu kiến trúc có sẵn, ở đây mình chọn **node.js**, còn rất nhiều mẫu khác tùy theo ngôn ngữ mà bạn sử dụng ở đây:

<https://serverless.com/framework/docs/providers/aws/cli-reference/create/>

**–path:** tên đường dẫn đến **thư mục source code**

**–name:** tên của service trong file **serverless.yml**

Sau khi tạo thành công thì bạn sẽ có một thư mục chứa các file tương tự như thế này: 

**.gitignore**: khai báo cho git biết những tập tin bị bỏ qua, không được push lên server.

**handler.js**: Là nơi mà chúng ta sẽ định nghĩa các hàm lambda (Lambda Function).

**serverless.yml**: Là nơi chúng ta sẽ khai báo cấu hình cho ứng dụng, file này thông thường có 3 phần chính sau:

**Provider**: Sử dụng để công khai các cấu hình cụ thể cho nhà cung cấp dịch vụ **Cloud**, ví dụ như cấu hình tên nhà cung cấp, môi trường runtime, khu vực sử dụng…vv

**Functions**: Chúng ta sẽ chỉ định các **Function logic** chức năng tại đây.

**Resources**: Phần này sẽ khai báo các tài nguyên để cho các Functions của bạn sử dụng được. Tài nguyên sẽ được khai báo bởi một dịch vụ của AWS có tên là CloudFormation.

- Tiếp theo login account amazon vào project:

export AWS\_ACCESS\_KEY\_ID=<your-key-here>

export AWS\_SECRET\_ACCESS\_KEY=<your-secret-key-here>

Kiểm tra thông tin nè:

printenv | grep AWS

- Chạy lệnh sau để upload source:

sls deploy // => up toàn bộ source

serverless deploy function -f tên\_function // => chỉ update 1 function, không cần up lại toàn bộ source, nhanh hơn nè

# IV) Cách cài đặt serverless và dynamodb offline:

**Serverless:**

1. Cài đặt plugin (mô phỏng AWS lambda và API Gateway cục bộ):

npm install serverless-offline --save-dev

1. Thêm plugin vào file .yml:

plugins:

* serverless-dynamodb-local (cái này của db, thêm sau nhé)
* serverless-offline

1. Chạy lệnh run server:

sls offline

1. Link config (tham khảo thêm nè)

<https://serverless.com/framework/docs/providers/aws/guide/resources/#configuration>

1. Thiết lập môi trường serverless

https://serverless.com/framework/docs/providers/aws/guide/variables/#referencing-environment-variables

DynomaDB:

1. Cài đặt plugin (cài đặt package dynamodb):

npm install serverless-dynamodb-local --save-dev

1. Thêm plugin vào file .yml:

plugins:

serverless-dynamodb-local (ở trên đã thêm rồi thì thôi nhen)

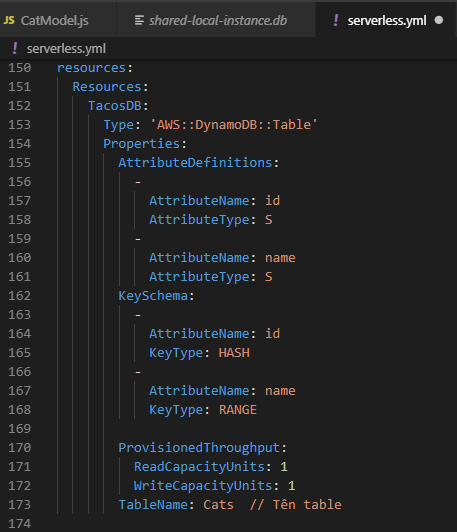
1. Install DynamoDB Local:

sls dynamodb install

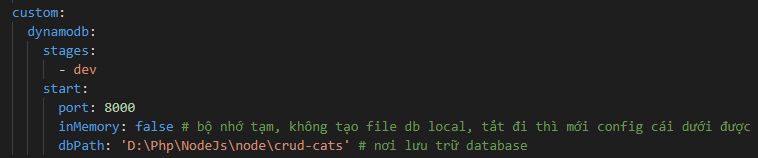
1. Có thể cài đặt thêm docker nếu cần:

docker pull lambci/lambda

1. Khai báo DynamoDb trong .yml

**No table of contents entries found.**

1. Khai báo DynamoDB chạy trên server dev



1. Cài đặt thư viện DynamoDB để dung trong code:

npm install --save serverless-dynamodb-client

let dynamodb = require('serverless-dynamodb-client'); // Cơ sở dữ liệu local

1. Run server

sls dynamodb start -p 8000

# V) Docker

Mở server

docker run -p 8000:8000 amazon/dynamodb-local # chạy server docker

Tắt server (chưa tối ưu lắm)

docker ps # get the id of the running container

docker stop <container> # kill it (gracefully)

# VI) Upload source Express lên lambda và connect aws rds mysql với navicat

1. Upload source

<https://serverless.com/framework/docs/providers/aws/cli-reference/deploy/>

B1: Cài đặt serverless local và chạy được nhé.

Lưu ý: chỉ sử dụng const abc = require(‘package’) thay cho import abc from ‘package’ => vì lên aws sẽ không chạy được

B2: Config env cho lamda:

+ App.js

Thêm thư viện serverless-http: const serverless = require('serverless-http');

Thay module.exports = app; thành module.exports.server = serverless(app);

+ serverless.yml (chưa có file này thì khởi tạo nhé)

Nội dung file .yml dung cho serverless:

service: first

provider:

  name: aws

  runtime: nodejs8.10

  stage: ${opt:stage, 'dev'} // khi chạy sls offline sẽ thêm hậu tố stage

region: us-east-1

  memorySize: 128

environment:

    MY\_SECRET: ${file(./src/config/env.${self:provider.stage}.json):CREDS}

plugins:

  - serverless-offline

functions:

  test:

    handler: src/app.server

    events:

      - http:

          path: /

          method: ANY

          cors: true

      - http:

          path: /{proxy+}

          method: ANY

          cors: true

Khi chạy lệnh:

sls offline –stage dev (phân biệt env sử dụng)

serverless deploy --stage pro (upload với môi trường là production)

+ src/model/mysql.js (Khởi tạo kết nối server mysql)

const con = mysql.createConnection({

    host     : process.env.RDS\_HOSTNAME,

    user     : process.env.RDS\_USERNAME,

    password : process.env.RDS\_PASSWORD,

    port     : process.env.RDS\_PORT,

    database : process.env.RDS\_DATABASE

});

Với các params RDS\_..., sẽ được config vào env trên aws

Vd:

* RDS\_HOSTNAME: là link ENDPOINT khi khởi tạo host trên aws RDS
* RDS\_USERNAME: là username khi tạo host
* RDS\_PASSWORD: là password khi tạo host
* RDS\_PORT: default sẽ là 3306, tạo xong sẽ có thông tin
* RDS\_DATABASE: tên database mình sẽ sử dụng

+ Upload có khả năng xảy ra lỗi, vào đây check nhé (cloudformation)

[cloudformation](https://us-east-1.console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/resources?filteringText=&filteringStatus=active&viewNested=true&hideStacks=false&stackId=arn%3Aaws%3Acloudformation%3Aus-east-1%3A030154202403%3Astack%2Ffirst-dev%2Fd7d5df50-05c7-11ea-8e4a-0ed829f607b4) => thằng này co tác dụng theo dõi tiến trình up

+ env.dev.json

{

    "CREDS": "dev"

}

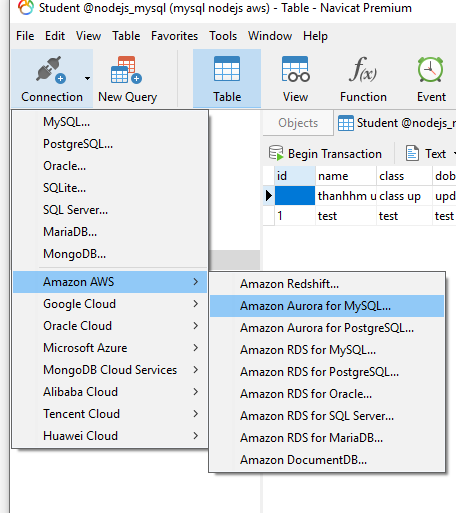
1. Connect navicat với aws mysql

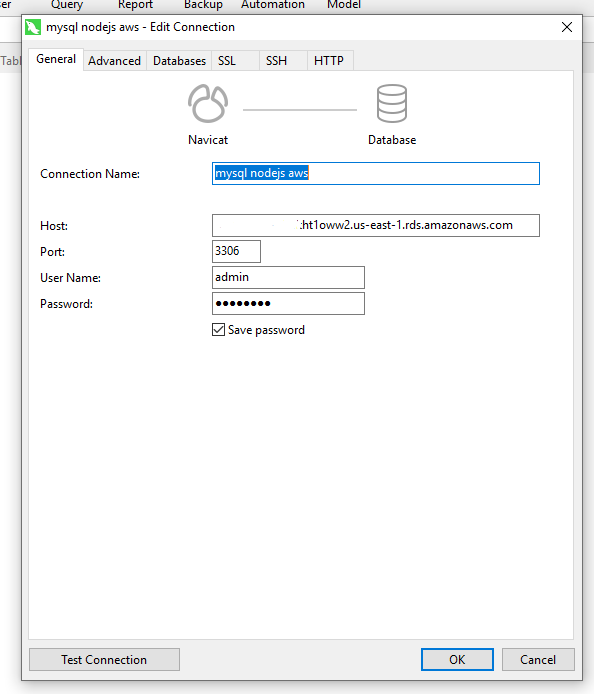
* Khi connect db navicat vào mysql aws thì cần thay đổi VPC security groups

Link VPC: [SecurityGroups](https://console.aws.amazon.com/vpc/home?region=us-east-1#SecurityGroups:sort=groupId)

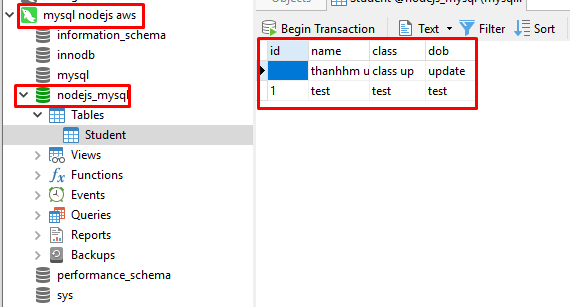
Đang chọn annywhere nên ip máy nào cũng connect được

* Điền các thông tin connect vào host trên amazon, eazy hen:





* Tên host, đặt dưới local thôi
* Tên database mình sẽ sử dụng



# VII) Query trong DynamoDB

1. Tạo bảng

Những câu lệnh sau sẽ dung để chạy cmd của DynamoDB

var params = {

TableName: 'Cats',

KeySchema: [ // The type of of schema. Must start with a HASH type, with an optional second RANGE.

{ // Required HASH type attribute

AttributeName: 'id',

KeyType: 'HASH',

},

{ // Optional RANGE key type for HASH + RANGE tables

AttributeName: 'name',

KeyType: 'RANGE',

}

],

AttributeDefinitions: [ // The names and types of all primary and index key attributes only

{

AttributeName: 'id',

AttributeType: 'S', // (S | N | B) for string, number, binary

},

{

AttributeName: 'name',

AttributeType: 'S', // (S | N | B) for string, number, binary

}

// ... more attributes ...

],

ProvisionedThroughput: { // required provisioned throughput for the table

ReadCapacityUnits: 1,

WriteCapacityUnits: 1,

}

};

dynamodb.createTable(params, function(err, data) {

if (err) ppJson(err); // an error occurred

else ppJson(data); // successful response

});

// ExpressionAttributeNames: sử dụng khi muốn gọi 1 ProjectionExpression là reserved (hiểu nôm na là định danh nó để có thể gọi được)

// Select data

// select nhiều record

var params = {

TableName: "Cats",

ProjectionExpression: "id, #name",

ExpressionAttributeNames: {

"#name": "name",

}

};

dynamodb.scan(params, function(err, data) {

if (err) ppJson(err); // an error occurred

else ppJson(data); // successful response

});

// select 1 record

var params = {

TableName: 'Cats',

KeyConditionExpression: 'id = :value', // a string representing a constraint on the attribute

ExpressionAttributeValues: { // a map of substitutions for all attribute values

':value': '1',

},

ScanIndexForward: true, // optional (true | false) defines direction of Query in the index

Limit: 5, // optional (limit the number of items to evaluate)

ConsistentRead: false, // optional (true | false)

Select: 'ALL\_ATTRIBUTES', // optional (ALL\_ATTRIBUTES | ALL\_PROJECTED\_ATTRIBUTES |

ReturnConsumedCapacity: 'NONE', // optional (NONE | TOTAL | INDEXES)

};

docClient.query(params, function(err, data) {

if (err) ppJson(err); // an error occurred

else ppJson(data); // successful response

});