Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Sistemas Operativos 1
Guillermo Alfredo Peitzner Estrada - 201504468.

Manual de Configuraciones

Node.JS API REST

DynamoDB

```javascript
const AWS = require("aws-sdk");
const DDB = new AWS.DynamoDB({
  accessKeyId: "AKIA5M3XKPWMTYJ5SMH7",
  secretAccessKey: "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",
  region: "us-east-2",
});

function insert(params) {
  return new Promise((resolve, reject) => {
    DDB.putItem(params, (err, data) => {
      if (err) {
        reject(err);
      } else {
        resolve(data);
      }
    });
  });
}

function getAll(params) {
  return new Promise((resolve, reject) => {
    DDB.scan(params, (err, data) => {
      if (err) {
        reject(err);
      } else {
        resolve(data);
      }
    });
  });
}
```

Express

```
const express = require("express");
const bodyParser = require("body-parser");
const cors = require("cors");
const app = new express();
app.use(bodyParser.json());
app.use(cors());

app.get("/", async (req, res) => {
  try {
    const params = {
      TableName: "FinalExam",
    };
    const results = await getAll(params);
    res.json(
      results.Items.map((data) => {
        return {
          Id: data.Id.S,
          Text: data.Text.S,
        };
      })
    );
  } catch (error) {
    res.json({ message: "error" });
  }
});

app.post("/", async (req, res) => {
  try {
    const { text } = req.body;
    const params = {
      TableName: "FinalExam",
      Item: {
        Id: { S: Date.now().toString() },
        Text: { S: text },
      },
    };
    await insert(params);
    res.json(params);
  } catch (error) {
    res.json({ message: "error" });
  }
});

app.listen(process.env.port || 8080);
```

Dockerfile

```dockerfile
FROM node:12
WORKDIR /usr/src/app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 8080
CMD [ "node", "server.js" ]
```

deployment.yml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: test-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: api
  template:
    metadata:
      labels:
        app: api
    spec:
      containers:
      - name: test-container
        image: gcr.io/kubproject-292502/test:1.0
        ports:
        - containerPort: 8080
```

service.yml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: api-service
spec:
  selector:
    app: api
  ports:
    - port: 8080
      targetPort: 8080
  type: LoadBalancer
```