Microservicios

Trabajo Práctico

New Horizons®
Computer Learning Centers

24 de julio del 2019

Caso práctico

La empresa CLIENTE S.A.C. ha contratado los servicios de los alumnos del del curso de microservicios para desarrollar el backend de un aplicativo bajo arquitectura de microservicios. Durante la reunión se acordó desarrollar los servicios según las siguientes especificaciones técnicas.

Path: "/producto"
Método : POST

- Request:

```
{
    "id": 0,
    "descripcion" : "String",
    "categoria" : "String",
    "precio_unitario" : 0.0,
    "stock_actual" : 0,
    "stock_minimo" : 0,
    "estado " : 1
}
```

- Response:

```
{
    "codigo_servicio": "0000",
    "descripcion" : "Producto registrado con éxito"
}
```

Path: "/producto"
Método : GET

- Request: **Ninguno**

- Response:

```
{
    "codigo_servicio": "0000",
    "productos" : [
        {...},
        {...},
        {...}
    ]
}
```

Path: "/producto/{id}"
Método: GET

- Request: **id**
- Response:

```
{
"codigo_servicio": "0000",
"producto" : {
    "id": 0,
    "descripcion" : "String",
    "categoria" : "String",
    "precio_unitario" : 0.0,
    "stock_actual" : 0,
    "stock_minimo" : 0,
    "estado " : 1
    }
}
```

Path: "/producto/{id}"
Método: DELETE

- Request: **id**
- Response:

```
{
    "codigo_servicio": "0000",
    "descripcion" : "Producto eliminado con éxito"
}
```

New Horizons Computer Learning Centers

Caso práctico

Path: "/cliente"
Método : POST

- Request:

```
{
    "id": 0,
    "nombres" : "String",
    "apellido_pat" : "String",
    "apellido_mat" : "String",
    "sexo" : "String",
    "direccion" : "String",
    "estado " : 1
}
```

- Response:

```
{
    "codigo_servicio": "0000",
    "descripcion" : "Cliente registrado con éxito"
}
```

Path: "/cliente"
Método : GET

- Request: **Ninguno**

- Response:

```
{
    "codigo_servicio": "0000",
    "clientes" : [
        {...},
        {...},
        {...}
    ]
}
```

Path: "/cliente/{id}"
Método: GET

- Request: **id**
- Response:

```
{
"codigo_servicio": "0000",
"cliente" : {
    "id": 0,
    "nombres" : "String",
    "apellido_pat" : "String",
    "apellido_mat" : 0.0,
    "sexo" : 0,
    "direccion" : 0,
    "estado " : 1
    }
}
```

Path: "/cliente/{id}"
Método: DELETE

- Request: **id**
- Response:

```
{
    "codigo_servicio": "0000",
    "descripcion" : "Cliente eliminado con éxito"
}
```

New Horizons Computer Learning Centers

- Para facilitar el uso de los servicios a los desarrolladores front-end, es necesario utilizar un documentador de APIS y se aceptó la el uso de Swagger 2.

- Se indicó además el uso de la base de datos MySQL.
  - Nombre de la base de datos: “**microservicios”.**
  - Servidor: “Su máquina virtual”.
  - Configuración: **Debe aceptar conexiones externas.**

- El cliente ha manifestado que los servicios pueden hacerse utilizando maven o gradle (el que sea más cómodo para el desarrollador).

- Para el despliegue de la aplicación es necesario el despliegue en **contenedores (utilizando Docker)**, debido a su portabilidad y bajo consumo de recursos.

- Se debe utilizar un repositorio **GitLab** para subir las fuentes del proyecto. **(Debe ser público para evitar la solicitud de credenciales).**

- El cliente requiere que cualquier cambio que se realice al producto se visualice en el menor tiempo posible en producción, motivo por el cual se solicita la implementación de un sistema de Integración y Entrega Continua (CI / CD). Se debe utilizar **Jenkins** para realizar este proceso.

- Notificar los pasos del proceso CI / CD en Slack (Opcional).

Nota: Para la integración continua, tomar como referencia los pasos del laboratorio de la clase 04.

RECURSOS

New Horizons®
Computer Learning Centers

Script del Pipeline **(con maven)**

```
node {
    def mvnHome

    stage('Preparation') {
        git '<<ruta_git_proyecto>>'
        mvnHome = tool 'M2'
    }

    stage('Test') {
        try {
            sh "'${mvnHome}/bin/mvn' test"
        } catch (e) {
            throw e
        }
    }

    stage('Build') {
        try {
            sh "'${mvnHome}/bin/mvn' clean package -DskipTests"
        } catch (e) {
            throw e
        }
    }

    stage('Results') {
        try {
            archive 'target/*.jar'
        } catch (e) {
            throw e
        }
    }

    stage('Deployment') {
        try {
            sh '/var/lib/jenkins/workspace/<<nombre_pipeline>>/runDeployment.sh'
        } catch (e) {
            throw e
        }
    }
}
```

New Horizons Computer Learning Centers

Script del Pipeline **(con gradle)**

```
node {
    def gradleHome

    stage('Preparation') {
        git '<<ruta_git_proyecto>>'
        gradleHome = tool 'Gradle'
    }

    stage('Test') {
        try {
            sh "'${gradleHome}/bin/gradle'  clean check"
        } catch (e) {
            throw e
        }
    }

    stage('Build') {
        try {
            sh "'${gradleHome}/bin/gradle' clean build"
        } catch (e) {
            throw e
        }
    }

    stage('Results') {
        try {
            archive 'build/libs/*.jar'
        } catch (e) {
            throw e
        }
    }

    stage('Deployment') {
        try {
            sh '/var/lib/jenkins/workspace/<<nombre_pipeline>>/runDeployment.sh'
        } catch (e) {
            throw e
        }
    }
}
```

New Horizons Computer Learning Centers

Dockerfile

```
FROM openjdk:8-jdk-alpine
EXPOSE <<puerto_applicacion>>
ADD app.jar testapp.jar
ENTRYPOINT
["java","-jar","testapp.jar"]
```

runDeployment.sh **(con maven)**

```
#!/bin/bash -ex
echo "Deploying app.jar to docker folder"
packageName=`ls target/<<nombre_jar>>*.jar`
versionid=1.0.1
versionname=latest
version=`echo $versionid-$versionname`
echo "version: $version"
cp -r $packageName deployment/app.jar
dockerImageName=app
dockerpid=`docker ps -a | grep $dockerImageName | grep "Up" | awk -F " " '{ print $1 }'`
if [[ $dockerpid != "" ]]; then
docker kill $dockerpid
docker rm $dockerpid
fi
docker build -t $dockerImageName deployment/.
docker run -d -p <<puerto_host>>:<<puerto_applicacion>> $dockerImageName
```

New Horizons Computer Learning Centers

runDeployment.sh **(con gradle)**

```
#!/bin/bash -ex
echo "Deploying app.jar to docker folder"
packageName=`ls build/libs/<<nombre_jar>>*.jar`
versionid=1.0.1
versionname=latest
version=`echo $versionid-$versionname`
echo "version: $version"
cp -r $packageName deployment/app.jar
dockerImageName=app
dockerpid=`docker ps -a | grep $dockerImageName | grep "Up" | awk -F " " '{ print $1 }'`
if [[ $dockerpid != "" ]]; then
docker kill $dockerpid
docker rm $dockerpid
fi
docker build -t $dockerImageName deployment/.
docker run -d -p <<puerto_host>>:<<puerto_applicacion>> $dockerImageName
```

New Horizons Computer Learning Centers

Gracias

www.newhorizons.edu.pe