

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

SISTEMAS OPERATIVOS I

SECCIÓN N



## INFORME Y VIDEO DE CONFIGURACIONES

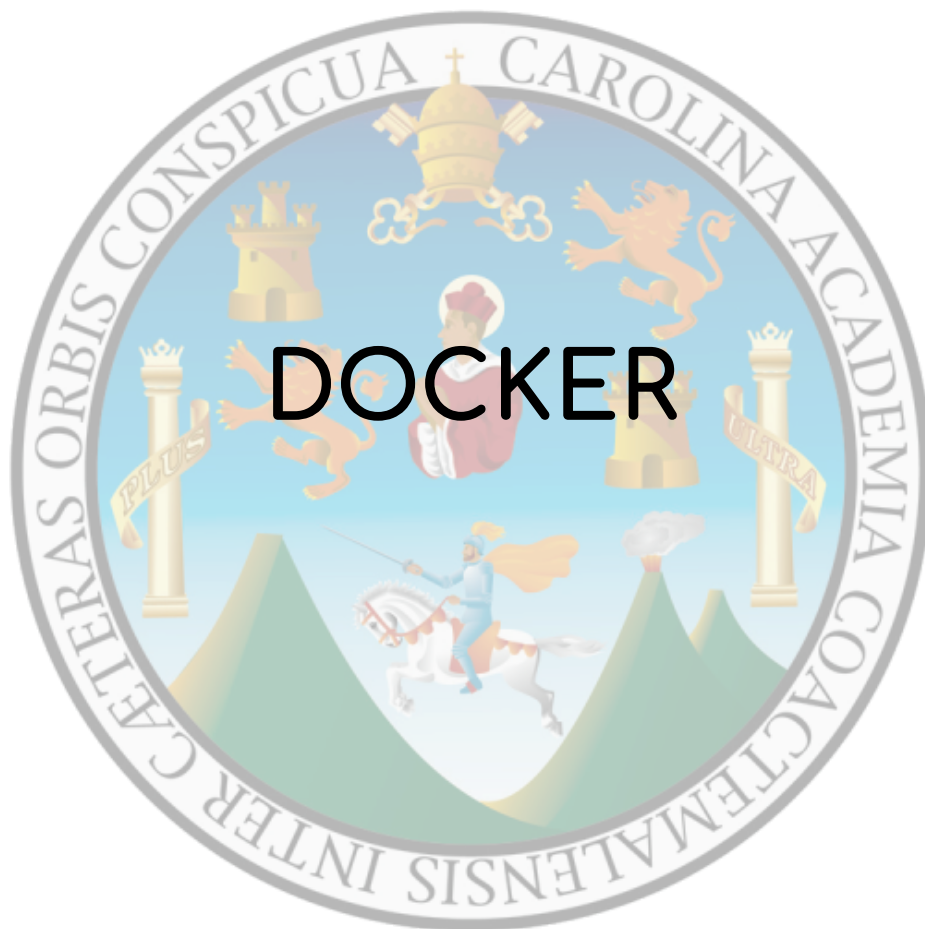
Estudiante	Carné No.
Harry Aaron Gómez Sanic	202103718

Guatemala, 02 de Enero de 2024

# **VIDEO DE CONFIGURACIONES**

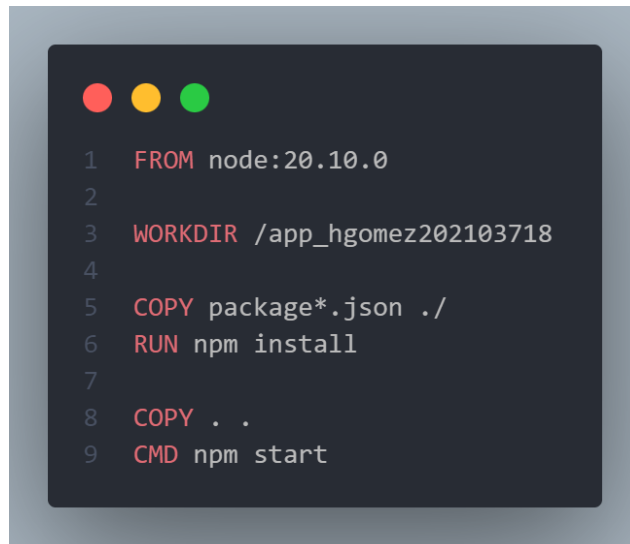
<https://youtu.be/C3k5gqaWUbl>





Se necesitará tener instalado docker ya que se creará la imagen del backend (nodejs).

- Teniendo ya hecha la aplicación se creará el archivo dockerfile para dockerizarla. A continuación se muestra dicho archivo:



```
1 FROM node:20.10.0
2
3 WORKDIR /app_hgomez202103718
4
5 COPY package*.json ./
6 RUN npm install
7
8 COPY . .
9 CMD npm start
```

- Se creará la imagen del backend(nodejs) teniendo ya configurado el archivo dockerfile. Para este paso se necesitará saber el nombre de usuario de docker hub y el nombre que se desea asignar a la imagen, en este caso:

- nombre de usuario es: h4rr1ss
- nombre de la imagen a crear es finalhgomez202103718

**docker build -t h4rr1ss/finalhgomez202103718 .**

- Para verificar la creación de la imagen se coloca:

**docker images**

- Por si se desea probar la imagen creada solo tomar en cuenta el puerto, en este caso 3000:

**docker run -d -p 3000:3000 h4rr1ss/finalhgomez202103718**

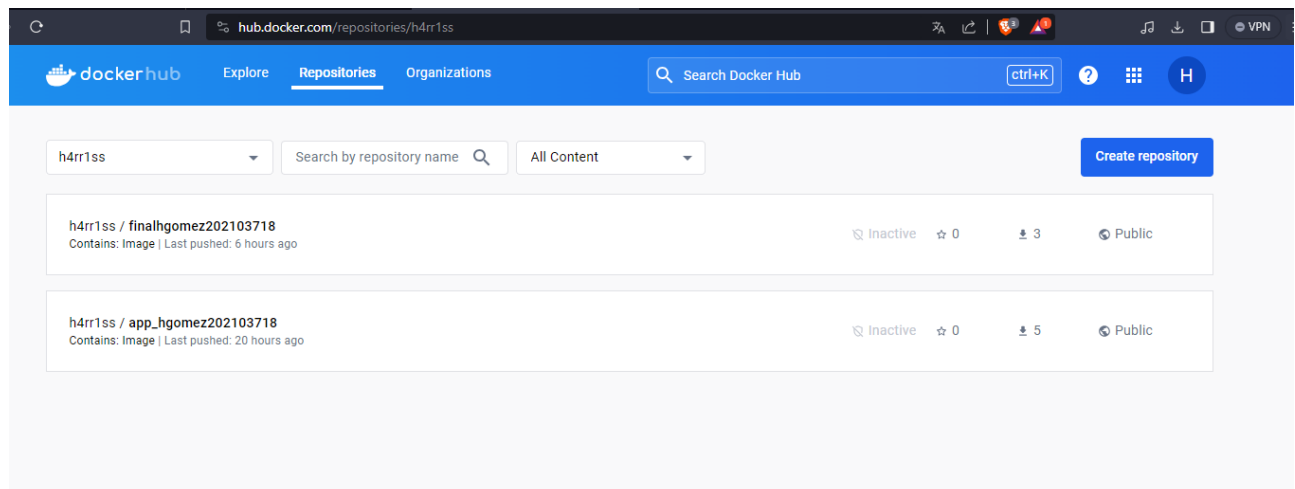
- Teniendo la image creada ahora se debe hacer push al repositorio en docker hub, pero antes se necesita estar logueado en docker

### **docker login**

- Hacer push de imagen a docker hub:

**docker push h4rr1ss/finalhgomez202103718:latest**

- Verificar si se encuentra la imagen nueva en el repositorio dirigiéndose al docker hub





# ARCHIVOS YML (KUBERNETES)

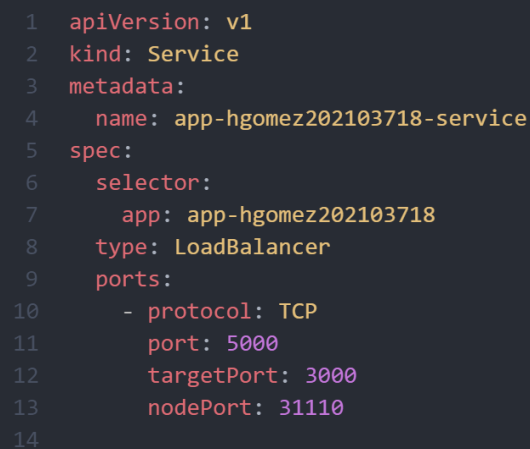
Se deben de crear una serie de archivos para la implementación, estos tienen extensión .yaml. Todos los archivos yaml del documento se explican con más detalle en el video de configuración.

- **Deployment.yaml:** Este tipo de archivo se utiliza para definir los recursos de implementación (Deployments) en Kubernetes. Un Deployment es un recurso que controla la creación y actualización de instancias de aplicaciones en el clúster.

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: app-hgomez202103718-deployment
5  spec:
6    replicas: 1
7    selector:
8      matchLabels:
9        app: app-hgomez202103718
10   template:
11     metadata:
12       labels:
13         app: app-hgomez202103718
14     spec:
15       containers:
16         - name: app-hgomez202103718-container
17           image: h4rr1ss/finalhgomez202103718:latest
18           ports:
19             - containerPort: 3000
20         - name: mysqldb-hgomez202103718
21           image: mysql
22           env:
23             - name: MYSQL_ROOT_PASSWORD
24               value: "12345"
25             - name: MYSQL_DATABASE
26               value: "hgomez202103718"
27
```

El archivo `service.yml` en Kubernetes es un manifiesto YAML que se utiliza para definir y crear un servicio. Un servicio en Kubernetes permite exponer una aplicación o un conjunto de aplicaciones como un servicio de red. En este caso son dos servicios.

- **serviceNODE.yml:**

A terminal window with a dark background and three colored window control buttons (red, yellow, green) at the top left. It displays a YAML manifest for a Kubernetes Service.

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: app-hgomez202103718-service
5  spec:
6    selector:
7      app: app-hgomez202103718
8    type: LoadBalancer
9    ports:
10     - protocol: TCP
11       port: 5000
12       targetPort: 3000
13       nodePort: 31110
14
```

- **serviceSQL.yml:**

A terminal window with a dark background and three colored window control buttons (red, yellow, green) at the top left. It displays a YAML manifest for a Kubernetes Service.

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: mysql-hgomez202103718-service
5  spec:
6    selector:
7      app: app-hgomez202103718
8    ports:
9     - protocol: TCP
10       port: 3306 # Usar el puerto estándar MySQL
11       targetPort: 3306
12     type: NodePort
13
```





Es necesario contar con la instalación de Minikube para utilizar Kubernetes, y dicha instalación puede variar según el sistema operativo que estés utilizando.

- Verificar que se tenga instalado minikube

**minikube version**

- Verificar el estado de minikube

**minikube status**

- En caso no está inicializado, inicializarlo

**minikube start**

Teniendo el cluster de minikube instalado e inicializado se procede a desplegar la aplicación en kubernetes.

- Teniendo el archivo deployment.yml se aplicará kubectl:

**kubectl apply -f deployment.yml**

- Para verificar el deployment se ejecuta el comando y ver que la columna llamada “READY” debe de estar 1/1 o dependiendo el número que se haya colocado, de lo contrario se debe de esperar:

**kubectl get deployment**

- Se pueden ver los pods con el siguiente comando:

**kubectl get pods**

- Para el servicio también se realiza el mismo procedimiento que el deployment. Cada uno de los servicios:

**kubectl apply -f serviceNODE.yml**

**kubectl apply -f serviceSQL.yml**

- Para verificar los servicios, se coloca el siguiente comando:

**kubectl get svc**

- Si no ocurre ningún error en el proceso, para ejecutar cualquiera de los servicios se coloca:

**minikube service *nombreAsignadoAlServicio***