Paso 1: Crear una Aplicación Node.js

1. **Inicializa un nuevo proyecto Node.js**:

```
mkdir lab12
cd lab12
npm init -y
```

2. **Instalar Express** (o cualquier otro framework de tu elección):

```
npm install express
```

3. **Crea un archivo** `index.js` con el siguiente código:

```
const express = require('express');
const app = express();
const port = 3000;

app.get('/', (req, res) => {
  res.send('Practica 12!');
});

app.listen(port, () => {
  console.log(`Aplicación escuchando en el puerto ${port}`);
});
```

Paso 2: Contenerizar la Aplicación

1. **Crear un archivo** `Dockerfile` :

```
FROM node:latest
WORKDIR /usr/src/app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 3000
CMD ["node", "index.js"]
```

2. **Construir la imagen Docker**:

```
docker build -t [TU USUARIO DOCKERHUB]/practica12:latest .
```

3. **Subir tu imagen a tu repositorio**

```
docker push [TU USUARIO DOCKERHUB]/practica12:latest
```

Paso 2: Crear los objetos Deployment y Service

1. **Deployment:**

```yaml
# deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: practica12-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: practica12
  template:
    metadata:
      labels:
        app: practica12
    spec:
      containers:
      - name: practica12
        image: [USUARIO DOCKERHUB]/practica12
        ports:
        - containerPort: 3000
```

2. **Service**

```yaml
# service.yaml
apiVersion: v1
kind: Service
metadata:
  name: practica12
spec:
  type: LoadBalancer
  ports:
  - port: 8081
    protocol: TCP
    targetPort: 3000
  selector:
    app: practica12
```

Paso 3: Instalar Ingress-Nginx y crear reglas

1. **Instalar Ingress-Nginx**
   Para lograr este paso únicamente tienes que instalar el controlador de ingress oficial de kubernetes con el siguiente comando:

   ```
   kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.6.4
   ```

2. **Verificar el controlador ingress**

   ```
   kubectl -n ingress-nginx get pods
   ```

3. **Crea las reglas de ingress**

```yaml
# ingress-rules.yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: practica12-ingress
 annotations:
    kubernetes.io/ingress.class: nginx
spec:
  ingressClassName: nginx
  rules:
  - host: practica12.example.com
    http:
      paths:
      - path: /
        pathType: ImplementationSpecific
        backend:
          service:
            name: practica12
            port:
              number: 8081
```

Explicación Detallada

- **apiVersion**: Define la versión de la API de Kubernetes que se está utilizando. En este caso, `networking.k8s.io/v1` es la versión de la API para objetos Ingress.
- **kind**: El tipo de recurso que se está definiendo, que en este caso es `Ingress`. Ingress es un objeto de Kubernetes que gestiona el acceso externo a los servicios en un clúster, típicamente HTTP.
- **metadata**:
  - **name**: El nombre del recurso Ingress, aquí es `practica12-ingress`.
  - **annotations**:
    - **kubernetes.io/ingress.class**: Define la clase de Ingress a utilizar. En este caso, se utiliza `nginx`, lo que significa que se espera un Ingress Controller de NGINX para manejar este Ingress.
- **spec** (Especificación):
  - **rules**: Define las reglas para enrutar el tráfico entrante.
    - **host**: El nombre del host para el cual se aplicará esta regla, aquí es `practica12.example.com`. Esto significa que cualquier tráfico dirigido a este dominio será manejado por este Ingress.
    - **http**:
      - **paths**: Una lista de rutas y sus backends asociados.

- **path**: La ruta URL que este Ingress manejará, en este caso es `/`, que representa la raíz.
- **pathType**: Define cómo interpretar el `path`, `ImplementationSpecific` permite que el Ingress Controller decida cómo interpretarlo.
- **backend**: Define el backend al cual el tráfico debe ser enviado.
  - **service**:
    - **name**: El nombre del servicio al que se dirigirá el tráfico, aquí es `practica12`.
    - **port**:
      - **number**: El puerto del servicio al que se dirigirá el tráfico, aquí es `8081`.

4. **Modifica tu archivo hosts**

   Agrega la siguiente linea a tu archivo `hosts` que se encuentra en la ruta `C:\Windows\System32\drivers\etc`

   ```
   127.0.0.1 practica12.example.com
   ```