

# 02 - Despliegue de aplicación en K8s

Tipo	Area Temática	Duración
Manual	Introducción a Kubernetes y EKS	1 hora

## Tema:

Despliegue de una aplicación básica en kubernetes

## Descripción:

Despliegue y configuración de una aplicación web en kubernetes, en donde se pueden evidenciar la forma en que se lleva una aplicación tradicional a un modelo contenerizado y orquestado en K8s.

## Prerequisitos:

- Git
- NodeJS
- Kubectl
- Docker
- Minikube

## Recursos:

- Aplicación web alojada en GitHub

## Pasos:

### Paso 1: Descargar la aplicación

- Descargar la aplicación de ejemplo que se encuentra en el repositorio en la carpeta '02\_app\_web'

```
git clone https://github.com/talentotech-resources/cloud_architecture
cd level3/resources/02_app_web/
```

- Realizar pruebas de la aplicación en ambiente local:

```
npm run start
```

### Paso 2: Crear imagen de la aplicación:

- Después de realizar las pruebas de la aplicación, realizar la creación de la imagen de docker:

```
docker build -t my-web-app:1.0 .
```

- Verifica que la imagen se haya creado:

```
docker images
```

## Paso 3: Desplegar en Kubernetes

### 1. Crear un archivo de despliegue ( `deployment.yaml` ):

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web-app-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: web-app
  template:
    metadata:
      labels:
        app: web-app
    spec:
      containers:
        - name: web-app
          image: my-web-app:1.0
          ports:
            - containerPort: 8080
```

### 2. Aplicar el archivo de despliegue:

```
kubectl apply -f deployment.yaml
```

### 3. Verificar que el despliegue esté corriendo:

```
kubectl get deployments
kubectl get pods
```

## Paso 4: Exponer la Aplicación como un Servicio

1. **Crear un archivo de servicio ( `service.yaml` ):**

```
apiVersion: v1
kind: Service
metadata:
  name: web-app-service
spec:
  type: LoadBalancer
  selector:
    app: web-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8080
```

2. **Aplicar el archivo de servicio:**

```
kubectl apply -f service.yaml
```

3. **Verificar que el servicio esté corriendo:**

```
kubectl get services
```

4. **Obtener la URL del servicio** (si estás usando Minikube, ejecuta `minikube service web-app-service --url`).

## Paso 5: Probar la Aplicación

1. **Accede a la URL proporcionada** para verificar que la aplicación esté funcionando correctamente.
2. **Verifica el estado de los pods:**
  - Para ver los logs de un pod específico:

```
kubectl logs <pod-name>
```

- Para acceder a un pod en ejecución:

```
kubectl exec -it <pod-name> -- /bin/bash
```

### 3. Escalar el despliegue:

- Incrementa o disminuye el número de réplicas:

```
kubectl scale deployment/web-app-deployment --replicas=5
```

- Verifica que los nuevos pods estén corriendo:

```
kubectl get pods
```

## Conclusión

Este paso a paso les permitirá a los estudiantes desplegar una aplicación web en Kubernetes, probar su funcionalidad, y entender cómo se manejan los pods, despliegues y servicios en un entorno Kubernetes. Pueden experimentar con diferentes configuraciones y escalas para comprender mejor el manejo de aplicaciones en contenedores.