



# TRABAJANDO CON KUBERNETES AVANZADO

## MÓDULO 2

## Módulo 2: Servicios, balanceo de carga y DNS externo

Exponer **workloads** a solicitudes

Introducción a los **balanceadores de carga**

Balanceadores de carga de **capa 7**

Balanceadores de carga de **capa 4**

Hacer que los **nombres de los servicios** estén disponibles externamente

## Exponer workloads a solicitudes (1)

Objeto	Descripción
Deployment	Permite <b>desplegar un Pod</b> en múltiples replicas y ajustar configuraciones sobre el despliegue
Service	Permite <b>exponer un Deployment</b> para ser consumido de forma más efectiva en los nodos

## Exponer workloads a solicitudes (2)

=== echo-deploy.yaml =====

```
apiVersion: apps/v1
kind: Deployment
```

```
metadata:
  name: echo-deploy
```

```
spec:
  replicas: 2
  selector:
    matchLabels:
      pod: hello-pod
  template:
```

```
    ...
```

=====

```
////////////////////////////////////
template:
  metadata:
    name: hello-pod
    labels:
      pod: hello-pod
  spec:
    containers:
      - name: echo-app
        image: dragonnomada/echo-env
        ports:
          - containerPort: 80
////////////////////////////////////
```

## Exponer workloads a solicitudes (3)

=== echo-service.yaml =====

apiVersion: v1

kind: **Service**

metadata:

name: **echo-service**

spec:

type: **ClusterIP**

selector:

matchLabels:

pod: **hello-pod**

ports:

...

=====

```
////////////////////////////////////  
  
ports:  
  - name: http  
    protocol: TCP  
    port: 8080  
    targetPort: 80  
  - name: https  
    protocol: TCP  
    port: 4443  
    targetPort: 443  
  
////////////////////////////////////
```

## Exponer workloads a solicitudes (4)

=== echo-service.yaml =====

apiVersion: v1

kind: **Service**

metadata:

name: **echo-service-node-port**

spec:

type: **NodePort**

selector:

matchLabels:

pod: **hello-pod**

ports:

...

=====

////////////////////////////////////

ports:

- name: http  
protocol: TCP  
port: 8080  
targetPort: **80**  
nodePort: **31111**
- name: https  
protocol: TCP  
port: 4433  
targetPort: **443**

////////////////////////////////////

## Exponer workloads a solicitudes (5)

```
=== echo-service.yaml =====
```

```
apiVersion: v1
```

```
kind: Service
```

```
metadata:
```

```
  name: echo-service-external-name
```

```
spec:
```

```
  type: ExternalName
```

```
  externalName: hello.example.com
```

```
=====
```

```
=== /etc/hosts =====
```

```
127.0.0.1 localhost
```

```
<internal-ip> hello.example.com
```

```
...
```

```
=====
```

# Introducción a los balanceadores de carga

Modelo OSI (Open System Interconnection)

Capa	Nombre	Descripción
7	Aplicación	Provee el tráfico en la aplicación (HTTP/S)
6	Presentación	Paquetes de datos y formularios
5	Sesión	Control del tráfico de peticiones
4	Transporte	Comunicación entre dispositivos (TCP/UDP)
3	Red	Ruteo entre dispositivos (IP)
2	Enlace	Pruebas entre comunicaciones físicas (MAC)
1	Física	Conexión física entre dispositivos



# Balanceadores de carga de capa 7

## [[ PRÁCTICA 1 ]]

<https://github.com/dragonnomada/kubelab/blob/main/d303-nginx-ingress.txt>

Balanceadores de carga de capa 4

[ [ PRÁCTICA 2 ] ]

<https://github.com/dragonnomada/kubelab/blob/main/d304-metal1b-load-balancer.txt>

Hacer que los nombres de los servicios estén disponibles externamente

## [ [ PRÁCTICA 2 ] ]

<https://github.com/dragonnomada/kubelab/blob/main/d304-metal1b-load-balancer.txt>