

K52. Creación de un Deployment para una aplicación web

José Manuel Rodríguez Guerrero

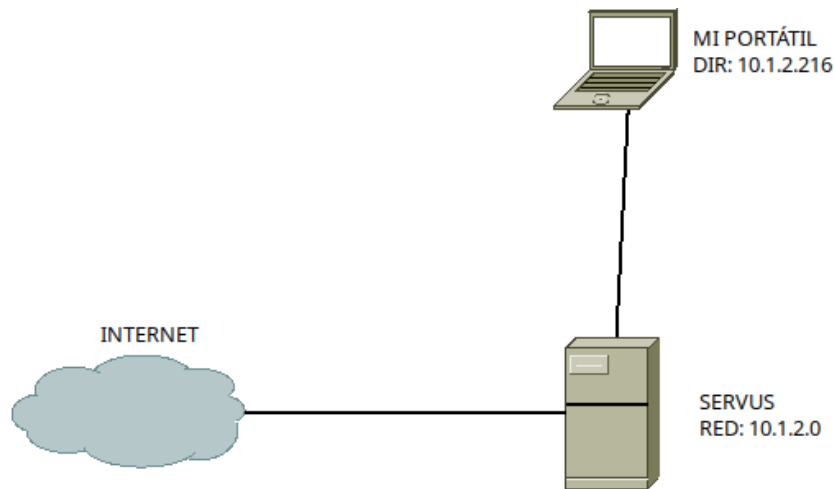
Noviembre 2022

Índice

1. Mapa de red	2
2. Creamos fichero yaml con la descripción del recurso Deployment	2
3. Creamos el Deployment	3
4. Comprobamos que los recursos se han creado	3
5. Vemos la información de forma detallada del Deployment creado	4
6. Creamos redirección utilizando port-forward	4
7. Accedemos a los logs del despliegue	5
8. Eliminamos el deployment	5
9. Bibliografía	6

1. Mapa de red

Partimos de un escenario que tendrá los siguientes dispositivos conectados a la red:



El SO del equipo anfitrión tendrá Ubuntu 22.04

2. Creamos fichero yaml con la descripción del recurso Deployment

Vamos a descargar el archivo de configuración en formato yaml y le indicaremos nombres distintos para el Deployment donde creará 2 réplicas y la imagen que desplegará será iesgn/test_web:latest. También tendremos que indicar una etiqueta en la especificación del pod para que coincida con el selector del deployment.

```
Archivo  Editar  Ver  Marcadores  Complementos  Preferencias  Ayuda
GNU nano 6.2  nginx-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deployment-joserogu
  labels:
    app: nginx-joserogu
spec:
  revisionHistoryLimit: 2
  strategy:
    type: RollingUpdate
  replicas: 2
  selector:
    matchLabels:
      app: nginx-joserogu
  template:
    metadata:
      labels:
        app: nginx-joserogu
    spec:
      containers:
        - image: iesgn/test_web:latest
          name: contendor-joserogu
          ports:
            - name: http
              containerPort: 80
```

3. Creamos el Deployment

En este paso usamos el comando `kubectl apply -f nginx-deployment.yaml` para crear el deployment.

```
Archivo Editar Ver Marcadores Complementos Preferencias Ayuda
josema ~/Imágenes/K52 kubectl apply -f nginx-deployment.yaml
deployment.apps/deployment-joserogu created
josema ~/Imágenes/K52 |
```

4. Comprobamos que los recursos se han creado

Mediante el comando `kubectl get pods` y `kubectl get pods -o wide` podremos comprobar que se han creado los deploys, replicaset y pods correctamente.

```
Archivo Editar Ver Marcadores Complementos Preferencias Ayuda
josema ~/Imágenes/K52 kubectl get pods
NAME                                READY    STATUS    RESTARTS   AGE
deployment-joserogu-cc8b9676f-57jbp  1/1      Running   0           2m46s
deployment-joserogu-cc8b9676f-bt5z2  1/1      Running   0           2m46s
josema ~/Imágenes/K52 kubectl get pods -o wide
NAME                                READY    STATUS    RESTARTS   AGE    IP            NODE
NOMINATED NODE    READINESS GATES
deployment-joserogu-cc8b9676f-57jbp  1/1      Running   0           2m54s  172.17.0.4    minikube
<none>              <none>
deployment-joserogu-cc8b9676f-bt5z2  1/1      Running   0           2m54s  172.17.0.3    minikube
<none>              <none>
josema ~/Imágenes/K52 |
```

También podemos ver los recursos que hemos creado con `kubectl get all`.

```
Archivo Editar Ver Marcadores Complementos Preferencias Ayuda
josema ~/Imágenes/K52 kubectl get deploy,rs,pod
NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
deployment.apps/deployment-joserogu  2/2      2              2            62s

NAME                                DESIRED    CURRENT    READY    AGE
replicaset.apps/deployment-joserogu-cc8b9676f  2          2          2        62s

NAME                                READY    STATUS    RESTARTS   AGE
pod/deployment-joserogu-cc8b9676f-57jbp  1/1      Running   0           62s
pod/deployment-joserogu-cc8b9676f-bt5z2  1/1      Running   0           62s
josema ~/Imágenes/K52 kubectl get all
NAME                                READY    STATUS    RESTARTS   AGE
pod/deployment-joserogu-cc8b9676f-57jbp  1/1      Running   0           69s
pod/deployment-joserogu-cc8b9676f-bt5z2  1/1      Running   0           69s

NAME                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
service/kubernetes  ClusterIP   10.96.0.1     <none>         443/TCP    2d21h

NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
deployment.apps/deployment-joserogu  2/2      2              2            69s

NAME                                DESIRED    CURRENT    READY    AGE
replicaset.apps/deployment-joserogu-cc8b9676f  2          2          2        69s
josema ~/Imágenes/K52 |
```

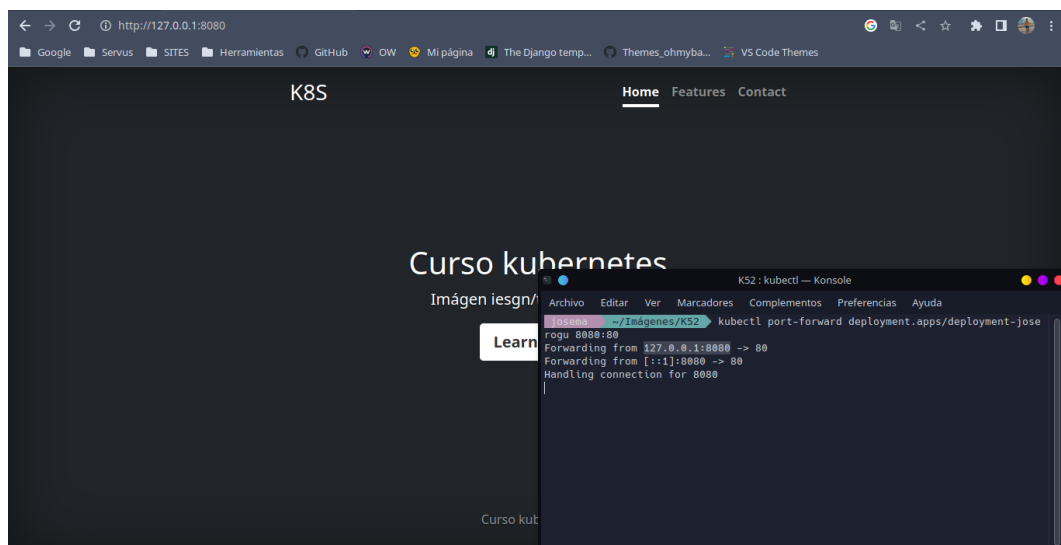
5. Vemos la información de forma detallada del Deployment creado

Para ello usaremos `kubectl describe deployment.apps/deployment-joserogu` donde podremos ver la información más detallada del deployment que creamos.

```
josema ~/Imágenes/K52 kubectl describe deployment.apps/deployment-joserogu
Name: deployment-joserogu
Namespace: default
CreationTimestamp: Mon, 31 Oct 2022 17:58:15 +0100
Labels: app=nginx-joserogu
Annotations: deployment.kubernetes.io/revision: 1
Selector: app=nginx-joserogu
Replicas: 2 desired | 2 updated | 2 total | 2 available | 0 unavailable
StrategyType: RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels: app=nginx-joserogu
  Containers:
    contendor-joserogu:
      Image: iesgn/test_web:latest
      Port: 80/TCP
      Host Port: 0/TCP
      Environment: <none>
      Mounts: <none>
      Volumes: <none>
  Conditions:
    Type           Status Reason
    ----           -
    Available       True  MinimumReplicasAvailable
    Progressing     True  NewReplicaSetAvailable
  OldReplicaSets: <none>
  NewReplicaSet: deployment-joserogu-cc8b9676f (2/2 replicas created)
  Events:
    Type      Reason              Age   From          Message
    ----      -
    Normal    ScalingReplicaSet   8m32s deployment-controller Scaled up replica set deployment-joserogu-cc8b9676f to 2
```

6. Creamos redirección utilizando port-forward

En este paso realizaremos una redirección con uso de port-forward donde utilizaremos el puerto 80 para ofrecer el servicio. Accedemos al navegador para ver que todo ha salido correctamente:



7. Accedemos a los logs del despliegue

Usamos el comando `kubectl logs deployment.apps/deployment-joserogu` para acceder a los logs pudiendo comprobar el acceso que hicimos anteriormente a través del navegador.

```
josema ~/Imágenes/K52 ➤ kubectl logs deployment.apps/deployment-joserogu
Found 2 pods, using pod/deployment-joserogu-cc8b9676f-57jbp
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.4. Set the 'Server
Name' directive globally to suppress this message
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.4. Set the 'Server
Name' directive globally to suppress this message
[Mon Oct 31 16:58:20.197742 2022] [mpm_event:notice] [pid 1:tid 140466702812288] AH00489: Apache/2.4.46 (Unix) configure
d -- resuming normal operations
[Mon Oct 31 16:58:20.199100 2022] [core:notice] [pid 1:tid 140466702812288] AH00094: Command line: 'httpd -D FOREGROUND'
127.0.0.1 - - [31/Oct/2022:17:03:51 +0000] "GET / HTTP/1.1" 200 2884
127.0.0.1 - - [31/Oct/2022:17:03:52 +0000] "GET /favicon.ico HTTP/1.1" 404 196
josema ~/Imágenes/K52 |
```

8. Eliminamos el deployment

En este paso eliminaremos el despliegue y comprobaremos que no ha quedado nada de él.

```
Archivo Editor VCL Marcadores Complementos Preferencias Ayuda
josema ~/Imágenes/K52 ➤ kubectl delete deployment.apps/deployment-joserogu
deployment.apps "deployment-joserogu" deleted
josema ~/Imágenes/K52 ➤ kubectl get pods
No resources found in default namespace.
josema ~/Imágenes/K52 |
```

9. Bibliografía

- [Gestión básica de un Deployment](#) - Site de Jaime
- [Video de youtube](#) - Curso de José Domingo