PODS en Kubernetes

Antonio Espín Herranz

PODS: El objeto más básico de Kubernetes

- Unidad mínima de trabajo que tendremos en Kubernetes.
- Como se crean
- Ciclo de Vida del POD
- Configurar sus propiedades
- Archivos YAML
- Modo imperativo vs declarativo (preferida por kubernetes)

Ciclo de vida de una aplicación

Un **POD** es un envoltorio de un contenedor

ESTADO DESEADO → Adapta el estado real al deseado, crea réplicas si es necesario. Gestión de recursos automático.

Trabajar mejor con los MANIFEST

PODS

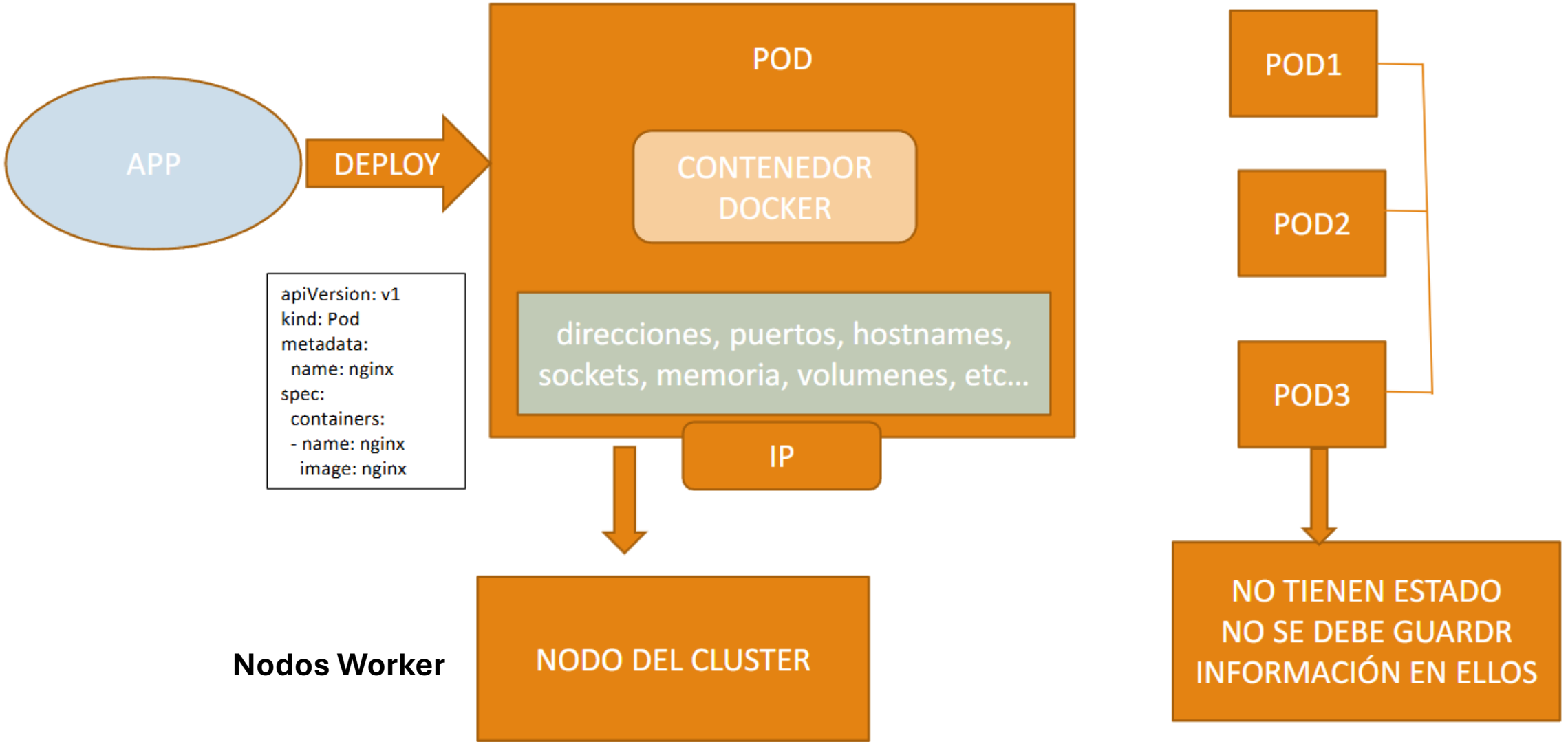
- El **POD** es el objeto mínimo dentro de **kubernetes**, es equivalente a los **containers** dentro de **Docker**.

- El POD otorga una serie de funcionalidades a los contenedores de Docker.

- **Dentro de un POD podemos tener más de un contenedor**
- **Pero el 99% todos los POD sólo tendrá un contenedor.**

PODS

Los PODS
Se pueden replicar
No tienen estado



Múltiples contenedores en PODS

- No es lo habitual pero el 99% tendremos en cada POD un solo contenedor.
- Seguimos con la misma filosofía que en Docker.
- Si en el mismo POD tuviéramos
  - Frontal Apache Web
  - Middleware Tomcat Java
  - Backend PostGreSQL

  - Es un error, rompemos el concepto de Microservicio (
    - Un microservicio solo hace una cosa y la hace bien
    - Que sea autónomo
    - Y autosuficiente

Múltiples contenedores en PODS

- Un POD es un envoltorio con una serie de propiedades.
- El ciclo de vida del supuesto anterior es un error
- ¿por ejemplo, un backup?
- Para hacer una actualización tendría que parar todo lo demás.
- Y todo tendría una única dirección IP

- Cuando tenemos un componente que está formado por dos módulos, en este caso si podría tener
  - Una aplicación que gestiona mensajes
  - Otro componente que recibía los mensajes.

- Pero lo habitual es tener cada componente en un POD distinto

Creación de POD

- Modo **imperativo** se basa en comandos
- Modo **declarativo** con un archivo de manifest en YAML

- Se crea y se ejecuta en el momento:
  - **Kubectl run nginx1 --image=nginx**
  - La imagen la recupera de un repositorio

  - Comprobar los POD
    - **Kubectl get pods**

  - Si da error la creación comprobar si está arrancado o no el cluster
  - minikube status
  - minikube start

Creación de POD

- Tenemos la opción:
- **Kubectl get pods –o wide**
  - Tenemos mas información.
  - La IP del PODS, el nodo donde se ejecuta y el estado (running)

  - Si lanzamos muy rápido el comando, puede que no le haya dado tiempo a crear el POD.

Propiedades del POD

- Crear otro por nginx2
- Kubectl describe nginx2
- Kubectl describe nginx1
- Podemos obtener un error como que no tenemos el recurso.

- Hay que poner previamente el componente:
  - **Kubectl describe pod/nginx1**
  - Da información similar a **Docker inspect**
  - Namespace default
  - Nodo, etiquetas, estado y la IP
  - Hace referencia también al contenedor
  - Condiciones del POD
  - Y los volúmenes

Lanzar comandos a los PODs

- Kubectl exec nginx1 -- **comando** (el comando separado de los guiones)

- Por ejemplo:
  - **kubectl exec nginx1 -- ls**
  - **Kubectl exec nginx1 -- uname –a (ver el S.O.)**

  - En modo **interactivo** (muy similar a Docker)
  - **Kubectl exec nginx1 –it --bash**
  - Salimos con **exit** del POD

Ver los Log del POD

- Kubectl run apache --image=httpd --port=8080

- Ver los **logs**:
  - **Kubectl logs apache**

- Con la opción --**tail=n** (los n últimos mensajes)

- Probar a mostrar la ayuda del comando:
  - **Kubectl logs --help**

Comprobaciones

- Comprobar si apache está funcionando
- **Kubectl exec apache -it – bash**

- Ahora lo podemos hacer desde dentro del POD
  - >wget localhost
  - Si no está el comando, se puede instalar igual que hacíamos en los contenedores.

  - Actualizar e instalar el comando **wget**:
  - **apt-get update**
  - **apt-get install wget**

  - Prueba: **wget localhost**
    - **Devuelve el index.html**

Formas de acceder desde fuera

- Acceso mediante:
  - Un **proxy**:
    - **Kubectl proxy**
  - Con un **servicio**
    - Más adelante
  - **Con port-forwarding**
    - Kubectl port-forward nginx 9999:80

Kubectl proxy

- Para probar el contenedor desde fuera:
  - Hay utilizar los deployments y services
  - Esta es una opción hasta que montamos todo (y ya podremos acceder desde fuera)

  - **Kubectl proxy**
    - Arranca un servidor y un puerto
    - Ir al navegador y comprobarlo → http://localhost:8001

    - Muestra todas las APIs:
    - Podemos probar (versión)
      - http://localhost/version
    - Para ver si el clúster está correcto o no:
      - http://localhost:8001/healthz

API – Vía Web

- http://localhost:8001/api/v1
  - Todos los recursos que forman parte del API
- http://localhost:8001/api/v1/namespaces
- http://localhost:8001/api/v1/namespaces/default
  - Nos da las características del namespace **default**
- http://localhost:8001/api/v1/namespaces/default/pods
  - Para los PODS del namespaces default (los namespaces agrupar objetos)
- http://localhost:8001/api/v1/namespaces/default/pods/apache
  - Para ver un POD concreto.

Acceder mediante Port Fowarding Recurso general

- **Kubectl port-forward nginx 9999:80**
- 9999 puerto local

```
C:\Users\Anton>kubectl port-forward nginx1 9999:80
Forwarding from 127.0.0.1:9999 -> 80
Forwarding from [::1]:9999 -> 80
```

- http://localhost:9999

Ver los PODS desde un nodo del clúster

- Nos conectamos al nodo de minikube:
- **minikube ssh**
- Prueba: **uname –a**

- Dentro podemos usar comando de Docker:
  - Docker ps
  - Docker ps | grep apache

- **NO Tocar directamente los componentes desde el nodo.**

- El clúster siempre se maneja con las herramientas:
  - **Kubectl, dashboard, kubeadm**

Crear un POD con un Manifest

- Crear una imagen con un Dockerfile
  - docker build –t piri12345/nginx:v1 .

- Subirlo a Docker hub

- Crear el POD con un manifest.
  - apiVersion:V1
  - Kind: Pod (tipo de objeto)
  - Metadata:
    - Name: nombre del pod
    - Labels:
      - Son etiquetas
  - Spec: → especificaciones
    - Containers:
      - Name: nginx
      - Image: la de Docker hub
- Para crear el recurso → **kubectl create –f nginx.yalm**
- Para ver la información → **kubectl describe pod/nginx**

PRACTICA 1