

YAML

Antonio Espín Herranz

Introducción a YAML

#Las cadenas no requieren comillas:

Título: Introducción a YAML

Pero se pueden usar:

title-w-quotes: 'Introducción a YAML'

Las cadenas multilínea comienzan con | (no usar los tabs.)

ejecutar: |

npm ci

npm build

prueba npm

Introducción a YAML

#Secuencias

#Las secuencias nos permiten definir listas en YAML:

Una lista de números usando guiones:

números:

- uno
- dos
- Tres

La versión en línea:

números: [uno, dos, tres]

Introducción a YAML

#Valores anidados

#Podemos usar todos los tipos anteriores para crear un objeto con valores anidados, así:

Lista de libros

- 1984:

autor: George Orwell

publicado en: 1949-06-08

recuento de páginas: 328

descripción: |

Una novela, a menudo publicada como 1984, es una novela distópica del novelista inglés George Orwell.

Fue publicado en junio de 1949 por Secker & Warburg como noveno y último b de Orwell.

- el Hobbit:

autor: J. R. R. Tolkien

publicado en: 1937-09-21

recuento de páginas: 310

descripción: |

The Hobbit, o There and Back Again es una novela de fantasía para niños del autor inglés J. R. R.

Archivos YAML de kubernetes

Estructura

- **apiVersion:** Define la versión de la API de Kubernetes que se usará para el objeto.
- **kind:** Especifica el tipo de objeto que estás creando (por ejemplo: Pod, Service, Deployment, etc.).
- **metadata:** Contiene información adicional como el nombre del objeto y etiquetas (labels).
- **spec:** Describe las especificaciones y configuraciones del objeto. El contenido dentro de spec depende del tipo de objeto que estás definiendo.

Ejemplo de un POD

- **apiVersion:** v1: Utiliza la versión 1 de la API.
- **kind:** Pod: El objeto que se va a crear es un Pod.
- **metadata:**
 - name: Se asigna el nombre "ejemplo-pod" al Pod.
 - labels: Se añade una etiqueta con clave app y valor ejemplo-app.
- **spec:**
 - containers: Lista de contenedores dentro del Pod. En este caso, solo hay uno llamado ejemplo-contenedor.
 - image: El contenedor utiliza la imagen de Docker nginx:latest.
 - ports: Expone el puerto 80 del contenedor.

```
apiVersion: v1
kind: Pod
metadata:
  name: ejemplo-pod
  labels:
    app: ejemplo-app
spec:
  containers:
  - name: ejemplo-contenedor
    image: nginx:latest
    ports:
    - containerPort: 80
```

labels

- Se usan para agruparlos y hacer consultas o referenciar más fácilmente diferentes tipos de objetos. Ejemplos:
 - “frontend” o “backend” asociados a la clave “capa”
 - “pruebas” o “produccion” asociado a la clave “tipo”
- Es posible etiquetar varios pods de la infraestructura y hacer una operación sobremuchos nodos
 - Con consultas o referencias que usen las etiquetas de los pods que los contienen: capa=frontend AND tipo=pruebas
 - Se facilita aplicar determinadas operaciones sobre conjuntos de contenedores

Ejemplo de un service

- apiVersion: v1
- kind: Service
- metadata:
 - name: ejemplo-service
- spec:
 - selector:
 - app: ejemplo
 - ports:
 - protocol: TCP
 - port: 80
 - targetPort: 80

Ejemplo de un deployment

- apiVersion: apps/v1
- kind: Deployment
- metadata:
 - name: ejemplo-deployment
- spec:
 - replicas: 3
 - selector:
 - matchLabels:
 - app: ejemplo
 - template:
 - metadata:
 - labels:
 - app: ejemplo
 - spec:
 - containers:
 - name: ejemplo-container
 - image: nginx
 - ports:
 - containerPort: 80