

Universidad Tecnológica de Santiago

UTESA

Facultad de Arquitectura e Ingeniería



Presentado a:

IVAN MENDOZA

Asignatura:

INF-025-001

ALGORITMOS PARALELOS

Presentado por:

Yoel Núñez 2-17-0171

21 de agosto de 2023

Santiago de los caballeros,

República Dominicana

Kubernetes, comúnmente abreviado como K8s, es una plataforma de código abierto diseñada para automatizar la implementación, el escalado y la gestión de aplicaciones contenerizadas. Fue desarrollado por Google y posteriormente donado a la Cloud Native Computing Foundation (CNCF). Kubernetes facilita la orquestación de contenedores, lo que permite a los desarrolladores y equipos de operaciones gestionar aplicaciones de manera eficiente y escalable, independientemente del entorno en el que se ejecuten.

Características principales:

1. **Orquestación de Contenedores:** Kubernetes automatiza la implementación, el escalado y la administración de contenedores, como Docker, proporcionando una base sólida para el desarrollo y la ejecución de aplicaciones en un entorno contenerizado.
2. **Automatización:** Kubernetes permite la automatización de tareas, como la escalada automática de aplicaciones según la demanda o la recuperación automática en caso de fallas.
3. **Escalabilidad:** Permite escalar aplicaciones de manera flexible, aumentando o disminuyendo el número de instancias de contenedores en función de la carga.
4. **Despliegue Declarativo:** Los desarrolladores definen el estado deseado de la aplicación en archivos de configuración YAML o JSON. Kubernetes se encarga de llevar el estado real al estado deseado de manera declarativa.
5. **Descubrimiento de Servicios:** Ofrece un sistema de nombres y descubrimiento de servicios que permite a las aplicaciones encontrar y comunicarse entre sí de manera transparente.
6. **Balanceo de Carga:** Distribuye el tráfico de red entre las instancias de la aplicación, mejorando la disponibilidad y el rendimiento.
7. **Actualizaciones Continuas:** Permite realizar actualizaciones continuas sin interrumpir el servicio, garantizando una alta disponibilidad.

Comandos de Kubernetes: Aquí hay algunos comandos esenciales de Kubernetes que los administradores y desarrolladores suelen utilizar:

1. **kubectl create:** Crea recursos en Kubernetes a partir de definiciones YAML o JSON.
2. **kubectl get:** Muestra información sobre los recursos en el clúster.
3. **kubectl describe:** Proporciona información detallada sobre un recurso específico.
4. **kubectl apply:** Aplica o actualiza las definiciones de recursos en el clúster.
5. **kubectl delete:** Elimina recursos del clúster.

6. **kubectl scale:** Cambia el número de réplicas de un recurso.
7. **kubectl logs:** Muestra los registros de un contenedor en un pod.
8. **kubectl exec:** Permite ejecutar comandos en contenedores en ejecución dentro de un pod.
9. **kubectl port-forward:** Reenvía puertos locales a un pod.
10. **kubectl rollout:** Gestiona las implementaciones y actualizaciones de aplicaciones.

Comandos para Crear la Practica

1. Kind Create Clusters

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versi3n 10.0.19045.3324]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\pc\Desktop\Universidad\Algoritmos Paralelos\PracticaKubernetesNodes>kind create cluster
Creating cluster "kind" ...
  • Ensuring node image (kindest/node:v1.27.3) [ ] ...
  [ ] Ensuring node image (kindest/node:v1.27.3) [ ]
  • Preparing nodes [ ] ...
  [ ] Preparing nodes [ ]
  • Writing configuration [ ] ...
  [ ] Writing configuration [ ]
  • Starting control-plane [ ] ...
  [ ] Starting control-plane [ ]
  • Installing CNI [ ] ...
  [ ] Installing CNI [ ]
  • Installing StorageClass [ ] ...
  [ ] Installing StorageClass [ ]
Set kubect1 context to "kind-kind"
You can now use your cluster with:

kubect1 cluster-info --context kind-kind

Not sure what to do next? [ ] Check out https://kind.sigs.k8s.io/docs/user/quick-start/

C:\Users\pc\Desktop\Universidad\Algoritmos Paralelos\PracticaKubernetesNodes>
```

2. `docker build -t <image-name>` tanto para el pod1 como para el pod2

```
C:\Windows\System32\cmd.exe - docker build -t poduno-image:latest
```

Microsoft Windows [Versión 10.0.19045.3324]
c) Microsoft Corporation. Todos los derechos reservados.

```
: \Users\pc\Desktop\Universidad\Algoritmos Paralelos\P practicaKubernetesNodes>cd primero  
  
: \Users\pc\Desktop\Universidad\Algoritmos Paralelos\P practicaKubernetesNodes\primero>docker build -t poduno-image:latest  
  
[+] Building 16.6s (5/12)
```

	docker:default
=> [internal] load .dockerignore	0.1s
=> => transferring context: 2B	0.0s
=> [internal] load build definition from Dockerfile	0.1s
=> => transferring dockerfile: 390B	0.0s
=> [internal] load metadata for docker.io/library/node:14	1.9s
=> [auth] library/node:pull token for registry-1.docker.io	0.0s
=> [1/7] FROM docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd8461aa	14.5s
=> => resolve docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd8461aa	0.0s
=> => sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd8461aa 7768 / 7768	0.0s
=> => sha256:2cafa3fb0b6529ee4726b4cf599ec27ee557ea3dea7019182323b3779959927f 2.21kB / 2.21kB	0.0s
=> => sha256:2ffid7dc41c74a25258bfaf60b8adb0a727f84518f55f65ca845ebc747976c408 14.68MB / 50.45MB	14.4s
=> => sha256:3d2201bd995ccfc12851a50820de03d34a17011dcbb9ac9fdf3a50c952cb131 10.00MB / 10.00MB	10.8s
=> => sha256:1d1247afe662a2a5c85b0378dc8ba228c1735747db410bbeffb8e2d91445452 7.51kB / 7.51kB	0.0s
=> => sha256:b253aeafa7e0671bb6008dd01de101a38a045fff7bc65663b0fbf7c05cca5 7.86MB / 7.86MB	8.3s
=> => sha256:1de76e26b8103d05fab960e0f77951ff54b912b63429c34f5d6adf09f5f9ee2 5.24MB / 51.88MB	14.4s
=> => sha256:d9a8df5894511ce28a05e2925a75e8a4acb0634c39ad734dfdaba823d1b1569 2.10MB / 191.85MB	14.4s
=> [internal] load build context	0.2s
=> => transferring context: 3.53kB	0.1s

3. kind load docker-image nombre_imagen:tag con ambas imagenes

```
C:\Users\pc\Desktop\Universidad\Algoritmos Paralelos\PracticaKubernetesNodes\primero>kind load docker-image poduno-image:latest
Image: "poduno-image:latest" with ID "sha256:65c0fc54aa0e362ddb4c8d7152e852b1089de79c0da40fd10dd217c7102c1942" not yet present on node "kind-control-plane", loading...
C:\Users\pc\Desktop\Universidad\Algoritmos Paralelos\PracticaKubernetesNodes\primero>
```

4. kubectl apply -f

```
C:\Users\pc\Desktop\Universidad\Algoritmos Paralelos\PracticaKubernetesNodes\primero>kubectl apply -f 1erpod.yaml
pod/pod1 created

C:\Users\pc\Desktop\Universidad\Algoritmos Paralelos\PracticaKubernetesNodes\primero>kubectl apply -f 1erpodservice.yaml
service/poduno-service created

C:\Users\pc\Desktop\Universidad\Algoritmos Paralelos\PracticaKubernetesNodes\primero>
```

5. kubectl get pods y kubectl get services para ver los pods y servicios creados

```
C:\Users\pc\Desktop\Universidad\Algoritmos Paralelos\PracticaKubernetesNodes\primero>kubectl get pods
NAME      READY   STATUS             RESTARTS   AGE
pod1      0/1     CrashLoopBackOff   3 (51s ago)  94s

C:\Users\pc\Desktop\Universidad\Algoritmos Paralelos\PracticaKubernetesNodes\primero>kubectl get services
NAME            TYPE       CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
kubernetes      ClusterIP  10.96.0.1    <none>        443/TCP          9m44s
poduno-service  LoadBalancer  10.96.61.216 <pending>    40:31672/TCP     53s

C:\Users\pc\Desktop\Universidad\Algoritmos Paralelos\PracticaKubernetesNodes\primero>
```

6. kubectl port-forward ambos pod con sus puertos correspondientes

```
C:\Users\pc\Desktop\Universidad\Algoritmos Paralelos\PracticaKubernetesNodes\primero>kubectl port-forward pod1 8080:4000
Forwarding from 127.0.0.1:8080 -> 4000
Forwarding from [::1]:8080 -> 4000
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
Handling connection for 8080
```