05 - 06 - Escalado en K8s

Tipo	Area Temática	Duración
Manual	Administración de aplicaciones con Kubernetes	2 horas

Tema:

Despliegue de aplicaciones con autoescalado y balanceo de carga

Descripción:

Guía para realizar diferentes pruebas de concepto utilizando Kubernetes con la aplicación creada inicialmente. Las pruebas incluyen HPA (Horizontal Pod Autoscaler), reinicio de pods, y escalados a demanda.

Prerequisitos:

- Git
- NodeJS
- Kubectl
- Docker
- Minikube
- Haber completado el despliegue de la aplicación web en Kubernetes siguiendo la guía anterior.
- Tener configurado un clúster de Kubernetes con `kubectl` accesible.
- Conocimientos básicos sobre Kubernetes y escalabilidad.

Recursos:

- Aplicación web alojada en GitHub

Pasos:

Paso 1: Implementar el HPA (Horizontal Pod Autoscaler)

1. **Crear un archivo de HPA (`hpa.yaml`)** para escalar según el uso de la CPU:

```
apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
  name: web-app-hpa
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: web-app-deployment
  minReplicas: 2
  maxReplicas: 10
  metrics:
  - type: Resource
    resource:
      name: cpu
      target:
        type: Utilization
        averageUtilization: 50
```

2. **Aplicar el archivo HPA**:

```
kubectl apply -f hpa.yaml
```

3. **Generar carga en la aplicación** para activar el escalado automático:
   - Usa herramientas como `kubectl run -i --tty load-generator --image=busybox -- /bin/sh -c "while true; do wget -q -O- http://web-app-service; done"` para generar tráfico hacia la aplicación.
4. **Verificar el estado del HPA**:

```
kubectl get hpa
```

Paso 2: Reiniciar Pods Manualmente

Reiniciar pods puede ser útil para aplicar cambios de configuración o actualizar imágenes de contenedores.

1. **Obtener el nombre de un pod** en ejecución:

```
kubectl get pods
```

2. **Reiniciar el pod**:

```
kubectl delete pod <pod-name>
```

3. **Verificar que el pod se haya reiniciado**:

```
kubectl get pods
```

Paso 3: Escalado a Demanda

Puedes escalar el número de réplicas manualmente según las necesidades de tráfico o de recursos.

1. **Escalar manualmente el despliegue**:

```
kubectl scale deployment web-app-deployment --replicas=5
```

2. **Verificar que los pods adicionales se hayan creado**:

```
kubectl get pods
```

3. **Probar la aplicación escalada**:
   - Accede a la aplicación web y verifica que sigue funcionando con el número incrementado de réplicas.
4. **Reducir el número de réplicas** si es necesario:

```
kubectl scale deployment web-app-deployment --replicas=2
```

Paso 4: Simular Fallos y Recuperación

Simula fallos en los pods para ver cómo Kubernetes maneja la recuperación automática.

1. **Eliminar un pod de manera deliberada**:

```
kubectl delete pod <pod-name>
```

2. **Observar cómo Kubernetes crea automáticamente un nuevo pod** para reemplazar el eliminado:

```
kubectl get pods
```

Conclusión

Estas pruebas de concepto permiten a los estudiantes comprender cómo Kubernetes gestiona el escalado automático con HPA, el manejo de fallos y la recuperación, así como el escalado manual en función de la demanda. Realizar

estas pruebas les dará una visión más profunda de cómo mantener y gestionar aplicaciones en un entorno de producción.