

CAS con OpenEBS

Preparando el equipo o la máquina virtual

Asegurarse que kubectl y Minikube están instalados según la bitácora en el sitio de la cátedra.

Instalar contrack:

```
sudo apt update
sudo apt install contrack
```

Para esta práctica necesitaremos un cluster minikube con privilegios de root (ALGO TOTALMENTE INSEGURO). Para ello primero eliminaremos cualquier cluster que hayamos creado y crearemos uno nuevo con privilegios de root:

```
sudo -E minikube start --vm-driver=none
```

Verificar que kubectl y kubernetes están configurados y funcionando

```
sudo kubectl get nodes
sudo kubectl get pods --all-namespaces
```

Instalar Open iSCSI

```
sudo apt update
sudo apt install open-iscsi
sudo service open-iscsi restart
```

Verificar que el servicio está funcionando con:

```
sudo service open-iscsi status
```

Podemos ver el nombre del iSCSI initiator con:

```
sudo cat /etc/iscsi/initiatorname.iscsi
```

Instalando OpenEBS en el cluster Kubernetes

Descargamos el manifiesto para instalar OpenEBS en nuestro cluster con:

```
curl -Ls https://github.com/openebs/openebs/raw/master/k8s/openebs-operator.yaml -o openebs-operator.yaml
```

Instalamos OpenEBS con:

```
sudo kubectl apply -f openebs-operator.yaml
```

Podemos monitorear el despliegue hasta que todos los deployments estén listos con:

```
sudo kubectl get deploy -n openebs --watch
```

Podemos ver todos los elementos de OpenEBS con:

```
sudo kubectl get all -n openebs
```

Creando un volumen persistente con OpenEBS

Descargamos el manifiesto para crear una storage class y un volumen persistente OpenEBS con una sola réplica:

```
curl -Ls https://raw.githubusercontent.com/openebs/openebs/master/k8s/demo/pvc-single-replica-jiva.yaml -o pvc-single-replica-jiva.yaml
```

Modificamos el tamaño del volumen editando el archivo y cambiando “5Gi” por “100M”

Creamos la storage class y el volumen persistente con:

```
sudo kubectl apply -f pvc-single-replica-jiva.yaml
```

Podemos ver las storage class, volúmenes persistentes y sus claims con:

```
sudo kubectl get sc,pv,pvc
```

Los datos de los volúmenes estarían en el directorio `/var/openebs` en cada uno de los nodos de almacenamiento. Al usar minikube, los volúmenes estarán dentro del directorio `/var/lib/docker/volumes/minikube/_data/openebs/`.

Podemos ver el volumen recién creado haciendo:

```
sudo ls -l /var/lib/docker/volumes/minikube/_data/openebs/
```

Creando un Stateful Workload con OpenEBS

Para ver OpenEBS en funcionamiento vamos a desplegar un Stateful Workload que persista los volúmenes usando OpenEBS.

Primero vamos a descargar los manifiestos ejemplo desde el proyecto OpenEBS:

```
cd ~
git clone https://github.com/openebs/openebs.git
cd openebs/k8s/demo/crunchy-postgres
```

Vamos a cambiar la cantidad de réplicas del volumen editando el archivo `set.json` mediante:

```
nano set.json
```

Cambiar el valor de `storageClassName` de `openebs-jiva-default` a `openebs-standard`.

Analizar los manifiestos dentro del archivo enumerando y describiendo cada objeto.

Para desplegar ejecutar:

```
sudo kubectl create -f set-sa.json
sudo kubectl create clusterrolebinding permissive-binding \
  --clusterrole=cluster-admin \
```

```
--user=admin \  
--user=kubelet \  
--group=system:serviceaccounts
```

```
sudo kubectl create -f set-service.json  
sudo kubectl create -f set-primary-service.json  
sudo kubectl create -f set-replica-service.json  
sudo kubectl create -f set.json
```

Explique que hace cada uno de los comandos anteriores.

Para ver los volúmenes:

```
sudo kubectl get pvc  
sudo kubectl get pv
```

Para ver los servicios desplegados:

```
sudo kubectl get service
```

Recuerde eliminar el cluster con privilegios de root una vez terminada la práctica.