systemctl restart docker ← check docker services

Date
Page

- **Minikube:**
Minikube is one of the easiest, most flexible and popular methods to run all-in-one or a multi-node local kubernetes cluster directly on our local workstations.

- **Requirements to run Minikube on our Local Workstation:**
- Kubectl is a binary used to access and manage any kubernetes cluster.
- Type-2 hypervisor or Container Runtime.

- **Start kubernetes cluster:**
minikube start:
Continue as root, use --force.

It will automatically install kubernetes.

- **Check the status:** Kubectl ← Mange the Cluster.
Minikube status

- **Stop Minikube:**
minikube stop

- **Delete Minikube:**
minikube delete

Dashboard:
minikube dashboard.

Getting just the dashboard URL
minikube dashboard --url.

- Basic Kubectl Commands :

kubectl version.

To get namespaces :
Kubectl get namespace.

Details for Specific Namespace :
kubectl get all --namespace default.

To check status of nodes :
kubectl get nodes.
kubectl get nodes --namespace kube-system
To check status of pods :
kubectl get pods.

To check status of services
kubectl get services.

To check status of deployments.
kubectl get deployments.

To check status of Replicaset :
kubectl get replicasets.

Kubectl get all ← check clusters.

Deployment : manage.
↓
Services / Replicaset : manage
↓
Pod
↓
Container.

To check logs :
Kubectl logs <pod_name>

- Details of node :

Kubectl describe node <node_name>
Kubectl describe node <node_name> - namespace <namespace_name>.

Details of Deployment :
Kubectl describe deployment <deployment_name>

Details of services :
Kubectl describe services <services_name>

Details of replicasets :
Kubectl describe replicasets <replicasets_name>

Details of pod :
Kubectl describe pod <pod_name>.

- To switch the Namespace:
  Kubectl config set-context --current --namespace=<namespace_name>

- To interact with Kubernetes:
Kubectl version
Kubectl cluster-info
Kubectl cluster-info dump
Kubectl cluster-info dump | jq.

⊙ Create a new namespace:
Kubectl create namespace dai ← namespace_name.
Kubectl get namespace.

- Delete Namespace:
Kubectl delete namespace <namespace_name>

⊙ Create a Deployment:
Using Kubectl to create a Deployment

a) Deploy an app:
We need to provide the deployment name and app image location (include the full repository url for image hosted outside Docker hub).

Kubectl create deployment <deploy_name> --image = httpd.latest.

-- interact with shell of

kubectl get deployments
kubectl get deployments -o wide

NOTE:
Pods that are running inside kubernetes are running on a private, isolated network.

kubectl proxy

curl http://localhost:8001/version.

⊙ Exploring your App:

kubectl describe pods
kubectl logs <pod_name>
kubectl logs <deployment_name> -c <container_name>

kubectl exec <pod-name> --env
kubectl exec <pod_name> --bash
kubectl exec -it lab1-56d594bcb4-6w662 --bash

kubectl exec -it <deployment_name> -c <container-name> --bash

Create service:

kubectl get svc/service
kubectl describe svc

- Accessing Apps:

Minikube type of service in Kubernetes:

1. NodePort:
Is the way to get external traffic directly to your service.

2. Loadbalancer:
Loadbalancer service is the way to expose a service to the internet, each service gets its own IP Address.

Kubectl get services

kubectl expose deployment /<deploy-name> --type = "Nodeport" --port 80

Kubectl describe services /<deploy_name>

kebectl port-forward <pod> 80:80

minikube service lab2 -n dai
~~depl~~ service_name. namespace_name.

Make ~~deployment~~ namespace → COAC
Make 2 deployments → 1. DBDA → httpd. latest
2. DITISS → httpd: latest

Expose service publically.

⊙ Create replica

kubectl create deployment lab3 --replicas=5 --image=nginx

Scale:

If the deployment named httpd's current replicasets is 3 ~~size is 2~~, Scale httpd to 5.

kubectl scale --current-replicas=3 --replicas=5 deployment/lab3

- Performing a Rolling update:

Note: To update the image of the application to version 2, use the set image command, followed by the deployment name and the new image version:

kubectl set image deployments/<deploy_name> <image_name>=repo_name/image:v2

kubectl get pods

kubectl describe services/<deploy-name>

confirm the update by running.

kubectl rollout status

- To roll back the deployment to your last working version:

kubectl rollout undo deployments/<deploy_name>

files - objective

- Rolling updates:

Rollout:

deployment name.

```
# kubectl set image deployments/test1 httpd=httpd:2.
# kubectl get deployments -o wide.
# kubectl rollout status deployments/test1.
```

Rollback:

```
# kubectl rollout undo deployments/test1
# kubectl rollout history deployments/test1
# kubectl rollout undo deployments/test1 --to-
  revision=2
```

- Create a busybox pod that echoes 'hello world' and then exits, but have the pod deleted automatically when it's completed.

```
# kubectl run <image_name> -it --image=busybox
  ϕ -- echo ϕ "hello world"
```

Use bash → to open terminal

```
# kubectl logs busybox1
```