Kubernetes

Cluster - Group of physical or virtual servers wherein Kubernetes is installed

Node (master) - Physical or virtual server that controls the Kubernetes cluster

Node (worker) - Physical or virtual servers where workloads run in a given container technology

Pods - Group of containers and volumes which share the same network namespace

Labels - User defined Key: Value pair associated to Pods

Master - Control plane components which provide access point for admins to manage cluster workloads

Service - An abstraction which serves as a proxy for a group of Pods performing a service

Kubernetes Objects

The objects include: Workloads, Services, Config & Storage, Clusters & Metadata

Installation

- sudo curl -sSLf k0s.sh | sudo sh
- sudo k0s install controller
- sudo systemctl start k0scontroller
- sudo systemctl enable k0scontroller
- mkdir ~/Documents
- sudo cp /var/lib/k0s/pki/admin.conf ~/Documents/kubeconfig.cfg
- sudo chown $USER ~/Documents/kubeconfig.cfg
- export set KUBECONFIG=~/Documents/kubeconfig.cfg

Install kubectl

- k0s kubectl get nodes
- curl -LO "https://dl.k8s.io/release/$(curl -L -s >https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
- sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl

Start a single instance of a pod

kubectl run mywebserver --image=nginx

Create a resource from the command line:

kubectl create deployment myotherwebserver --image=nginx

Accessing the terminal via the Lens IDE and creating a resource via terminal:

kubectl create -f ./my-manifest.yaml

Example of YAML file

```
---
apiVersion: v1
kind: Pod
metadata:
  name: rss-site
  labels:
    app: web
spec:
  containers:
    - name: front-end
      image: nginx
      ports:
        - containerPort: 80
    - name: rss-reader
      image: nickchase/rss-php-nginx:v1
      ports:
        - containerPort: 88
```

Create or apply changes to a resource

kubectl apply -f ./my-manifest.yaml

Delete a resource via Lens

kubectl delete -f ./my-manifest.yaml

Scale a resource

kubectl scale --replicas=3 deployment.apps/myotherwebserver

or

kubectl scale --replicas=3 -f my-manifest.yaml

Connect to a running container

kubectl attach mywebserver -c mynginx -i

Run a command in a single container pod

kubectl exec mywebserver -- /home/user/myscript.sh

Delete a resource

kubectl delete pod/mywebserver

or

kubectl delete -f ./my-manifest.yaml

Viewing resources

View the cluster and client configuration

kubectl config view

List all resources in the default namespace

kubectl get services

List all resources in a specific namespace

kubectl get pods -n my-app

List all resources in all namespaces in wide format

kubectl get pods -o wide --all-namespaces

List all resources in json (or yaml) format

kubectl get pods -o json

Describe resource details

kubectl describe pods

kubectl describe pod mywebserver

Get documentation for a resource

kubectl explain pods

kubectl explain pod mywebserver

List of resources sorted by name

kubectl get services --sort-by=.metadata.name

List resources sorted by restart count

kubectl get pods --sort-by='.status.containerStatuses[0].restartCount'

Rolling update pods for resource

kubectl rolling-update echoserver -f my-manifest.yaml

Networking

Types of services

ClusterIP is the default ServiceType, ClusterIP services have a cluster-internal IP address, so they can only be reached by other cluster components.

NodePort enables you to create a service that's available from outside the cluster by exposing the service on the same port for every node. For example, the same service might be available on host1.example.com:32768, host2.example.com:32768, and host3.example.com:32768.

LoadBalancer requires coordination with your cloud provider's load balancer, which automatically routes requests to the service. For this reason, not all distributions of Kubernetes will support LoadBalancer services.

ExternalName is the most complex ServiceType, coordinating the service with your DNS server.