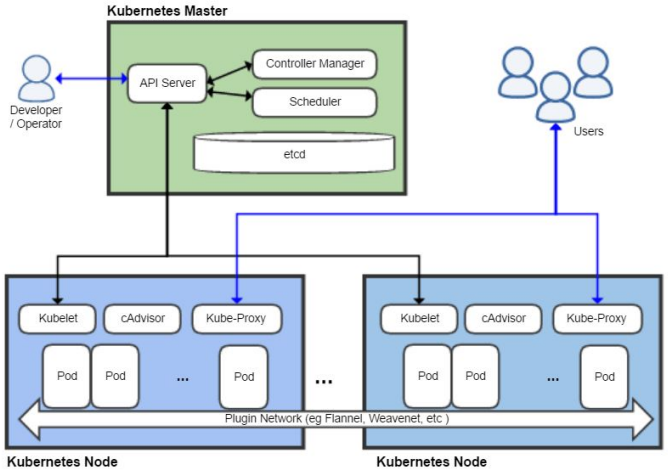
k8s overview

What is k8s

Kubernetes (k8s) is an open source solution for the orchestration of containerized applications. It's for automating apps deployment, scaling, and coordination in a cluster environment. It support docker containerization technology. k8s mainly is written on Go.

K8s components:

- on master node
- on worker node



Difference from Docker Swarm and why to use

- k8s supports much more possibilities for industry orchestration
- k8s allows more fine tuning for apps scaling and updating, data replication and fault tolerance
- detailed cluster provisioning
- k8s supports large nodes clusters
- detailed (self) monitoring
- more secure deployments
- supports GitOps
- supports helm (apps packages)
- a lot of third-party solutions integrations

A big difference between k8s and docker swarm in container runtime: *containerd* vs *dockerd*.

k8s cluster deployment

1. local and single node: minikube, microk8s
2. local, not local and multi nodes: k3s (less kube components)
3. k8s as Saas (cloud cluster): AWS EKS or GKE
4. On-premise (bare-metal) solution: deploy via kubespay, Deckhouse or manually (install *kubeadm*, *kubelet*, *containerd…*)

k8s basic concepts

- **pod**: single and main k8s unit, abstraction over containers (multiple containers in one pod are allowed)
- **service** (network and rules how pods communicate between each other)
- **deployment** (how to deploy pods, how to set pods replicas, how to update/rollback deployments)
- **configMap**: mount persistant raw data to pods
- **secrets**: operate with sensitive data in deployments
- **persistent volume**: write/read data during app work on the fly

k8s kind

```
apiVersion: v1
kind: Secret
metadata:
  name: {{ template "kube-prometheus-stack.fullname" . }}-prometheus
  namespace: {{ template "kube-prometheus-stack.namespace" . }}
  labels:
    app: {{ template "kube-prometheus-stack.name" . }}-prometheus
    app.kubernetes.io/component: prometheus
{{ include "kube-prometheus-stack.labels" . | indent 4 }}
data:
  object-storage-configs.yaml: {{ toYaml .Values.prometheus.prometheusSpec.thanos.objectStorageConfig.secret | b64enc | quote }}
{{- end }}
{{- end }}
```

```
apiVersion: v1
kind: Service
metadata:
  name: {{ template "kube-prometheus-stack.fullname" . }}-prometheus
  namespace: {{ template "kube-prometheus-stack.namespace" . }}
  labels:
    app: {{ template "kube-prometheus-stack.name" . }}-prometheus
    self-monitor: {{ .Values.prometheus.serviceMonitor.selfMonitor | quote }}
{{ include "kube-prometheus-stack.labels" . | indent 4 }}
{{- if .Values.prometheus.service.labels }}
{{ toYaml .Values.prometheus.service.labels | indent 4 }}
{{- end }}
{{- if .Values.prometheus.service.annotations }}
  annotations:
{{ toYaml .Values.prometheus.service.annotations | indent 4 }}
{{- end }}
spec:
{{- if .Values.prometheus.service.clusterIP }}
  clusterIP: {{ .Values.prometheus.service.clusterIP }}
{{- end }}
{{- if .Values.prometheus.service.topologyKeys }}
```

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: {{ template "kube-prometheus-stack.fullname" . }}-grafana-datasource
  namespace: {{ template "kube-prometheus-stack-grafana.namespace" . }}
{{- if .Values.grafana.sidecar.datasources.annotations }}
  annotations:
    {{- toYaml .Values.grafana.sidecar.datasources.annotations | nindent 4 }}
{{- end }}
  labels:
    {{ $.Values.grafana.sidecar.datasources.label }}: {{ $.Values.grafana.sidecar.datasources.labelValue | quote }}
    app: {{ template "kube-prometheus-stack.name" $ }}-grafana
{{ include "kube-prometheus-stack.labels" $ | indent 4 }}
data:
  datasource.yaml: |-
    apiVersion: 1
{{- if .Values.grafana.deleteDatasources }}
    deleteDatasources:
{{ tpl (toYaml .Values.grafana.deleteDatasources | indent 6) . }}
{{- end }}
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ template "kube-prometheus-stack.operator.fullname" . }}
  namespace: {{ template "kube-prometheus-stack.namespace" . }}
  labels:
    {{- include "kube-prometheus-stack.prometheus-operator.labels" . | nindent 4 }}
{{- if .Values.prometheusOperator.labels }}
{{ toYaml .Values.prometheusOperator.labels | indent 4 }}
{{- end }}
{{- if .Values.prometheusOperator.annotations }}
  annotations:
{{ toYaml .Values.prometheusOperator.annotations | indent 4 }}
{{- end }}
spec:
  replicas: 1
  revisionHistoryLimit: {{ .Values.prometheusOperator.revisionHistoryLimit }}
  selector:
    matchLabels:
      app: {{ template "kube-prometheus-stack.name" . }}-operator
      release: {{ $.Release.Name | quote }}
  {{- with .Values.prometheusOperator.strategy }}
  strategy:
    {{- toYaml . | nindent 4 }}
  {{- end }}
  template:
    metadata:
      labels:
        {{- include "kube-prometheus-stack.prometheus-operator.labels" . | nindent 8 }}
```

kubectl

It’s main CLI tool to operate with k8s cluster. It requires ***kubecong***.

- kubectl get <resource>
- kubectl delete <resource>
- kubectl describe <resource>
- kubectl rollout restart <resource> –replicas=<n>
- kubectl logs <pod_name>
- kubectl top pod/node
- kubectl apply -f ./my-manifest.yaml
- kubectl patch pod <pod_name>
- kubectl cluster-info
- kubectl scale --replicas=3 -f foo.yaml
- kubectl exec my-pod –
- kubectl edit svc/<service_name>

kubectl

```
vasilii@vasilii-sora:~/.kube$ kubectl get pods
NAME                                                READY   STATUS    RESTARTS   AGE
prometheus-operator-grafana-d46b79dd9-2qd6l         3/3     Running   0          47d
prometheus-operator-kube-p-operator-7cc8f8f54d-b9xtz   1/1     Running   0          47d
prometheus-operator-kube-state-metrics-879d67f87-4k946   1/1     Running   0          7h35m
prometheus-prometheus-operator-kube-p-prometheus-0   2/2     Running   0          47d
vasilii@vasilii-sora:~/.kube$
```

```
vasilii@vasilii-sora:~/.kube$ kubectl logs prometheus-prometheus-operator-kube-p-prometheus-0 | head -n 10
ts=2024-12-25T09:49:39.421Z caller=main.go:617 level=info msg="Starting Prometheus Server" mode=server version="(version=2.52.0, branch=HEAD, revision=879d80922a227c37df502e7315fad8ceb10a986d)"
ts=2024-12-25T09:49:39.421Z caller=main.go:622 level=info build_context="(go=go1.22.3, platform=linux/amd64, user=root@1b4f4c206e41, date=20240508-21:56:43, tags=netgo,builtinassets,stringlabels)"
ts=2024-12-25T09:49:39.421Z caller=main.go:623 level=info host_details="(Linux 5.10.205-195.807.amzn2.x86_64 #1 SMP Tue Jan 16 18:28:59 UTC 2024 x86_64 prometheus-prometheus-operator-kube-p-prometheus-0 (none))"
```

```
vasilii@vasilii-sora:~/.kube$ kubectl top pods
NAME                                                CPU(cores)   MEMORY(bytes)
prometheus-operator-grafana-d46b79dd9-2qd6l         2m           291Mi
prometheus-operator-kube-p-operator-7cc8f8f54d-b9xtz   4m           39Mi
prometheus-operator-kube-state-metrics-879d67f87-4k946   3m           32Mi
prometheus-prometheus-operator-kube-p-prometheus-0   62m          1292Mi
```

```
vasilii@vasilii-sora:~/.kube$ kubectl exec -it prometheus-operator-grafana-d46b79dd9-2qd6l
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kub
prometheus-operator-grafana-d46b79dd9-2qd6l:/usr/share/grafana$ pwd
/usr/share/grafana
```

```
vasilii@vasilii-sora:~/.kube$ kubectl get svc prometheus-operator-grafana -o yaml
apiVersion: v1
kind: Service
metadata:
  annotations:
    meta.helm.sh/release-name: prometheus-operator
    meta.helm.sh/release-namespace: monitoring
  creationTimestamp: "2023-01-12T10:21:39Z"
  labels:
    app: grafana
    app.kubernetes.io/instance: prometheus-operator
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/name: grafana
    app.kubernetes.io/version: 10.4.1
    helm.sh/chart: grafana-7.3.11
  name: prometheus-operator-grafana
  namespace: monitoring
  resourceVersion: "671824271"
  uid: 451168ba-6b96-4a0d-bca4-0c3f9a0cd766
spec:
  clusterIP: 172.20.141.86
  clusterIPs:
  - 172.20.141.86
  internalTrafficPolicy: Cluster
  ipFamilies:
  - IPv4
  ipFamilyPolicy: SingleStack
  ports:
  - name: http-web
    port: 80
    protocol: TCP
    targetPort: 3000
  selector:
    app.kubernetes.io/instance: prometheus-operator
    app.kubernetes.io/name: grafana
  sessionAffinity: None
  type: ClusterIP
status:
  loadBalancer: {}
```