kubernetes

+

HELM

Introduction

Quick overview of k8s and the CLI
&
Why should you use Helm

Quick Disclaimer

This is a very short introduction into the kubernetes and helm ecosystem since the topic is very complex.

I highly encourage to dig deeper after this presentation by checking the official documentation.

Also if you have any questions just feel free to interrupt me.

kubernetes

> “ Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications. ”

What is kubernetes?

Keywords:

- **Automating deployments**
- **Scaling**
- **Containerized applications**

Glossary

High Level:

- Control Plane
- Node/Agent
- Controller/Operator
- API Resources

Low Level:

- Orchestration(API,Manager...)
- Kubelet
- Kube Proxy
- Container Runtime

Automating deployments

This can be easiley done via the Kubernetes Objects.

Options:

- Pod
- Deployment
- Daemon Set
- Stateful Set

Scaling

What do you do when you need to handle load?

Option 1

Scale vertically. This means adding more available RAM/CPU to the pod.

Option 2

Scale horizontally. This means adding more instances of the same service and load balancing between them.

Solution

In Kubernetes we can do both.

Vertically

- Requests & Limits

Horizontally

- Replicas & Services

Containerized applications

I hope you are familiar with Docker or better said containerd.

Why you ask?

Each pod in kubernetes runs one or more **container images**.

The underling container engine can be of any kind as long as it implements the **Container Runtime Interface** (CRI).

DNS

Every cluster needs one.

- Pods & Services have DNS entries
- It is scoped per namespace
- Queries are usually expanded via `/etc/resolv.conf`
- Can be changed in any shape or form you imagine
  eg. {service-name}.{namespace}.{svc|pod}.{cluster-domain}

Deployment

```yaml
apiVersion: apps/v1
kind: Deployment
...
spec:
  replicas: 3
  ...
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

Service

```yaml
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  type: ClusterIP # Other options NodePort/LoadBalancer
  # clusterIP: None for Headless Service
  selector:
    app.kubernetes.io/name: MyApp
  ports:
    - protocol: TCP
      port: 80
      targetPort: 9376
```

Ingress

```
apiVersion: networking.k8s.io/v1
kind: Ingress
...
spec:
  rules:
  - host: "*.foo.com"
    http:
      paths:
      - pathType: Prefix
        path: "/"
        backend:
          service:
            name: my-service
            port:
              number: 80
```

Kubeconfig

- Manages access to kubernetes clusters (plural)
- Usually you can find it in ~/.kube/config
- One file by default

Azure

```
az aks get-credentials --resource-group myResourceGroup --name myAKSCluste
```

K3s

```
export KUBECONFIG=/etc/rancher/k3s/k3s.yaml
```

Example

```
...
clusters:
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://192.168.1.12:6443
  name: homestack
contexts:
- context:
    cluster: homestack
    namespace: default
    user: default
  name: homestack
current-context: homestack
...
```

HELM

> “ **The package manager for Kubernetes.** ”

Package Manager?

- More like a templating engine with added stuff
- Very useful if you want to add versioning
- Adds an easy way to deploy/teardown a complex service
- Interacts with kubernetes
- Provides a ".Values" variable
- Hub for sharing charts artifacthub.io
- CLI tool, nothing more nothing less

More Details

- A deployable unit in helm is called a "Chart"
- Written in Go with the `text/template` package
- A release is a deployed chart on k8s
- Can upgrade/install/rollback releases
- It supports also repositories of charts

```
NAME            NAMESPACE       REVISION        UPDATED                                 STATUS          CHART                   APP VERSION
homestack-dns   default         1               2023-02-16 17:11:27.513353937 +0000 UTC deployed        coredns-0.1.0           1.0.0
kleilobby       default         2               2023-02-14 15:23:26.310650121 +0000 UTC deployed        klei-lobby-0.1.0        1.0.0
```

Links

- Helm Docs
- Kubernetes Docs
- Kubectl Docs
- Go template package
- Recommended labels
- Well-Known Labels, Annotations and Taints
- QoS for Pods

DEMO