

Malmaal and Kubernetes

by Lucas Meurer

<https://malmaal.io>

Kubernetes Intro

Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation.

<https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>

- Basically a very complex way to run docker containers

Main Components

- pods
 - group of one or more containers
 - temporary
- deployments
 - declare the desired state for a pod
 - scale pods
 - roll out new versions and pod updates
- services
 - define a logical set of pods
 - exposes them to other pods or the outside world

A sample Deployment

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

A sample Service

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx
  ports:
    - name: http
      protocol: TCP
      port: 80
      targetPort: 80
```

`curl http://my-service` should now work inside the cluster

- will be load balanced between pods

Digital Ocean Kubernetes (DOKS)

- managed k8s cluster on digitalocean
- malmal previously ran on google cloud
 - that was too expensive
 - when doks had the cluster-autoscaler I finally switched

<https://www.digitalocean.com/products/kubernetes/>

Helm

- package manager for kubernetes
- a package is called "chart"
- automatically creates deployments, services, etc..
- charts can have dependencies so an app chart could have a dependency to the postgres chart

```
$ helm repo add bitnami https://charts.bitnami.com/bitnami
$ helm install my-rabbitmq bitnami/rabbitmq --values values.yaml
```

values.yaml

```
replicas: 1

resources:
  limits:
    cpu: 50m
    memory: 200Mi
```

Deployment?

- in the beginning I managed all the kubernetes resources myself
- very tedious and a lot of copy & paste and `kubectl apply -f`

GitOps!

- infrastructure as code
- git is the single source of truth for infrastructure
- changes are automatically applied when a branch is merged

<https://www.gitops.tech/>

flux

Flux is a tool for keeping Kubernetes clusters in sync with sources of configuration (like Git repositories), and automating updates to configuration when there is new code to deploy.

<https://toolkit.fluxcd.io/>

- dedicated infrastructure repo
- it "just works" (no complicated ci pipelines)
- can add things to the cluster from
 - other git repos
 - helm charts
 - kubernetes yaml files
- used to set up kubernetes tools like cert-manager

werf.io

Werf implements GitOps using helm templates

- werf
 - automatically builds and publishes docker images
 - applies the current helm chart
 - tracks the current deployment
 - allows quick rollbacks
 - works with GitLab CI, Github Actions and more
 - can easily spin up review environments

Lets have a look at some of malmals infrastructure!