

## **KUBERNETES**

## Skill Tree: Color in the boxes and level up your skills

Use for individuals or as a group by picking a colour each and coloring in a part of the box. Everyone's journey is different and you can interpret the goals flexibly. The aim is to inspire you to learn and try new things. Not everything needs to be completed.

(set your own goal)

within Kubernetes

田

Develop your

own Operator

Implement GitOps

Configure storage

class

Upgrade the

cluster using kubeadm

Implement

**Custom Resource** 

Definitions (CRDs) and

Operators

Apply network

Policies to control

traffic between Pods

Perform rolling

updates and rollbacks

Secure sensitive

data with Secrets

Set up a

container runtime

e.g. Containerd, CRI-0

Ш



(set your own goal)

(set your own goal)

Contribute to

Kubernetes community

Use Kubernetes

on Bare Metal

Use Cluster API

Configure Pod

affinity and anti-affinity

Use advanced

networking solutions

like Istio or Calico

Use DaemonSets

to run Pods on all

nodes

Set resource

requests for Pods

Filter resources

using Selectors

Create a basic

Pod

Create a canary deployment

Implement

Advanced Network

**Policies** 

Set up Cluster-wide

Logging, Monitoring,

and Alerting

Perform cluster

performance tuning

ţţţ

Set up a multi-node

cluster with kubeadm

Implement

storage solutions with

CSI drivers

3

Configure Jobs to

run batch tasks

Set resource

limits for Pods

using a Service

Check pod

logs

(set your own goal)

Create a

blue-green deployment

Create your

own Helm chart

Configure High

Availability (HA) clusters

**Use Kubernetes** 

API directly

Install and

configure a

networking solution e.g. Calico, Flannel

Use Service

Mesh for advanced

networking and security

Create

StatefulSets for

stateful applications

Control Pod

scheduling tolerations

Apply Labels to

organize resources

Install and

configure kubectl

>\_

custom admission controller Optimize costs 1

> Use external storage solutions

Create your own

with Kubernetes

田 Implement custom admission controllers

Use Persistent Volume Provisioners

practices

Optimize resource usage

ţţţ

Deploy a Service mesh solution, e.g., Istio, Linkerd

Implement

Kustomize

K

Configure CronJobs

for scheduled tasks

Set up

Persistent Volumes and Claims

for external access e.g. NGINX, Traefik

N

Use kubectl to manage resources

>\_

Implement **Advanced Ingress** 

Configurations

health

+

Definition (CRD)

Set up monitoring

Set up a CI/CD

pipeline that deploys

in the cluster

Implement Kubernetes logging and tracing

(set your own goal)

Implement

cloud-native

observability practices

B

园 Monitor node

Apply Pod Security Admission

Use OPA for policies

Set up a cluster using kubeadm

Create your own Custom Resource

Use Helm for

package management

イ

HELM

for Pods Configure

Role-Based Access Control (RBAC)

Configure horizontal Pod autoscaling

Scale Pods within a deployment

using a Service

Create and

organize namespaces

Manage configuration data with ConfigMaps

999

Deploy a local

Kubernetes cluster e.g. Minikube, Kind

1 tile = 1 point

Icons by Icons8.com

**Total Score** 

**START HERE** Name: .

CC BY-NC-SA 4.0