

Glasgow University - Skills Workshop

Topic: Kubernetes & Helm

Date: 21. November 2019

Author: Nicola Mössner

Kubernetes

What is it?

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

Why do we use it at ResDiary?

- Containerisation, shorter deploy times, automated scaling/updates/rollback, built-in logging/metrics, etc.

Terminology

- Node: A node is a worker machine in Kubernetes. A node may be a VM or physical machine, depending on the cluster. Each node contains the services necessary to run pods and is managed by the master components
- Pod: Pods are the smallest deployable units of computing that can be created and managed in Kubernetes. A Pod is a group of one or more containers (such as Docker containers), with shared storage/network, and a specification for how to run the containers. A Pod models an application-specific "logical host"
- Service: An abstract way to expose an application running on a set of Pods as a network service. Kubernetes gives Pods their own IP addresses and a single DNS name for a set of Pods, and can load-balance across them.
- Deployment: A Deployment provides declarative updates for Pods. You describe a desired state in a Deployment, and the Deployment Controller changes the actual state to the desired state at a controlled rate.
- Kubectl: Kubectl is a command line interface for running commands against Kubernetes clusters

Helm

- Allows you to define all your resources in yaml files as a chart. Provides you further with functionality to install/update/delete that chart in the desired Kubernetes cluster and namespace.

```
# Create a new helm chart
helm create mychart
# Use helm to deploy a chart
helm install --name=mychart .\mychart\
# Use upgrade from now on
helm upgrade mychart .\mychart\
```

Demo

```
kubectl get all

# Install chart
helm install --name=mychart .\mychart\ # or
helm upgrade mychart .\mychart\

# Access apps through services
kubectl port-forward svc/my-nginx-service 30909:80 # in a new window
kubectl port-forward svc/todo-app-service 30908:80 # in a new window

# Delete pod, watch pods
kubectl get pods -o wide -w # in a new window
kubectl delete pod <POD_NAME>

# Scale deployment manually
kubectl scale deployment todo-app-deployment --replicas=0
kubectl scale deployment todo-app-deployment --replicas=4

# Autoscaler for php website (apache webserver)
kubectl run php-apache --image=k8s.gcr.io/hpa-example --requests=cpu=200m --
limits=cpu=500m --expose --port=80
kubectl autoscale deployment php-apache --cpu-percent=50 --min=1 --max=10
kubectl get hpa -w
kubectl exec -it busybox-pod sh # in a new window
    while true; do wget -q -O- http://php-apache.default.svc.cluster.local; done
```

The Future of Kubernetes

- Micro VMs
- Kubernetes as hypervisor
 - legacy apps can be migrated and run in containers as they are with little modification.

Learn more:

- Kata Containers - <https://katacontainers.io/>
- KubeVirt - <https://github.com/kubevirt/kubevirt>
- <https://www.infoq.com/news/2019/05/kubernetes-future/>

Further Resources

- Official tutorials (<https://kubernetes.io/docs/tutorials/>)
- Google Kubernetes Engine (<https://cloud.google.com/kubernetes-engine/>)
- Kubectl cheat sheet (<https://kubernetes.io/docs/reference/kubectl/cheatsheet/>)

Sources

- <https://kubernetes.io/docs/concepts/> (for Terminology)