Kubernetes for beginners

Kubernetes Overview (S2)

Containers

- eg. Docker
- containerize applications
- isolated enviroments with own processes an down network etc.
- like VM's but share the same kernel
    - different linux distributions are possible to run
    - Containers don't include the OS
- Containerized applications on dockerhub
- Docker run commands to start containers
- Containers are instances of images

Container Orchestration

- Automatically deploying and managing containers
- docker swarm, kubernetes, mesos
- multiple instances easy to deploy as a service level

Kubernetes Architecture

- Nodes: physical or virtual machine kubernetes is running on
- Cluster: set of nodes running on a cluster
- Master: node with kubernetes configured as master
    - Watches over other (worker)nodes and balances loads handles crashes, …
- Kubernetes contains:
    - API Server (master)
    - etcd (master)
    - kubelet (worker)
    - Container Runtime (worker, eg. Docker)
    - controller (master)
    - scheduler (master)
- kubectl
    - command line tool for kubernetes

Setup Kubernetes (S3)

- local (minikube, kubeadm), cloud(aws, gcp, azure), play-with-k8s.com

Minikube

- preconfigured single node cluster

Kubeadm

- https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/
- setup multi-node cluster on separate machines
- install container runtime on all nodes
- install kubeadm tool on all nodes
- initialize master
- requires POD network
    - https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/create-cluster-kubeadm/#pod-network
    - For problems with DNS:
        - sudo nano /etc/NetworkManager/NetworkManager.conf
        - comment out #dns=dnsmasq
        - sudo service network-manager restart
        - kubectl -n kube-system delete pod -l k8s-app=kube-dns
- join worker nodes with command given by initializing master node

Google Cloud Platform

- go to kubernetes tab
    - click create cluster
    - connect cluster when it's initialized

https://labs.play-with-k8s.com/ → simple test enviroment for kubernetes

Kubernetes Concepts (S4)

POD's

- https://kubernetes.io/docs/concepts/workloads/pods/pod-overview/
- single instance of application with a docker container running incide
    - 1 container per POD usually
    - Or helper container within the same POD
- creating new pods to scale up/ or adding new nodes
- deploy pod with *kubectl run <application> --image <image-name>*
    - downloads image from dockerhub or private repo
- *kubectl get pods* to see all pods
- *kubectl describe pod <podname>* to see information of pod

YAML Introduction (S5)

- https://learn.getgrav.org/16/advanced/yaml

Kubernetes Concepts PODs, ReplicaSets, Deployments (S6)

- yamls as inputs for creation of objects
- *kubectl create -f …yml*

Replication Controllers and ReplicaSets

- controllers: monitor objects and respond
- Replication Controller →
    - ensures a defined number of instances run at all times
    - load balancing between pods across nodes
    - defined with yaml
    - *kubectl get replicationcontroller*
- Replica Sets
    - New way of replication controller
        - Api version is different
        - Requires selector with matchLabels to specify the pods
            - Monitors pods with the specified labels
        - *kubectl get replicaset*
        - scale replica set:
            - *kubectl replace –f <replica-definition file>*
            - *kubectl scale –replicas=<number> <replica-definition-file>*

Deployments

- allows for rolling updates, rollbacks, pause update, …
- deployment object over replicaset
- deployment-definition.yml similar to replicaset
- *kubectl get deployments*
- *kubectl get all*
- in productive mostly just creating deployment definition file

- new deplayoment triggers new rollout which creates new revision
    - creating also a new replica set
- *kubectl rollout status <deployment-name>*
- *kubectl rollout history <deployment-name>*
- deployment strategies:
    - 1. Recreate: destroy older versions and then deploy newer
    - 2. Rolling Update(default): Take down and bring up one by one
- Change deployment-definition yaml *and kubectl apply –f <deployment-name>*
- Rollback to go to previous revision
    - *kubectl rollout undo <deployment-name>*

Networking in Kubernetes (S7)

- every POD has its own IP address
    - down over local network in Node
- every Node has its own IP address
    - when cluster is setup Networking is not setup automatically
    - network has to be setup manual
        - use calico, flannel, weave-net, nsx … to do it
        - will create a network with routing techs

Services (S8)

- enable communication between components of application
- enable connectivity between applications/users

NodePort Service

- access from outside node done with service
- forward requests on Node Ports to Port on POD Port
- acts like virtual server inside the node (has own IP address)
- create with definition file
    - set ports in spec definition (targetPort, port, nodePort)
    - selectors to connect with specific POD
- *kubectl create <service-definition>*
- *kubectl get services*
- automatically selects POD for LoadBalancing
    - service spans across nodes

ClusterIp Service

- Creates virtual IP inside Cluster
- create with definition file
    - set ports in spec definition (targetPort, port)
    - selectors to connect with specific POD
- *kubectl create <service-definition>*
- *kubectl get services*

LoadBalancer Service

- Distribute load across servers
- Allows access from outside