**KUBERNETES**

https://kubernetes.io/docs/concepts/workloads/controllers/deployment/

Watch kubectl get all – информация о всех сущностях. Watch  проверяет каждые 2 секунды.

ETCD (port 2379) – NoSQL BD. Распределённое хранилище типа ключ-значение.

/etc — папкой для конфигов в unix системах и “d” — distributed.

>Put Name “Boris Johnson” -- создание записи

>Get name -- запрос значения.

**Для управления kube master c другой машины:**

ssh -r root@172.30.1.21:/root/.kube ~/

vim .kube/config

server: <https://172.30.1.21:6443/> -- машина мастер

**Автодополнение ввода для Kubectl**

sudo -s

kubectl completion bash > /etc/bash\_completion.d/kubectl

exit source /etc/bash\_completion.d/kubectl

BASH

source <(kubectl completion bash) *# настройка автодополнения в текущую сессию bash, предварительно должен быть установлен пакет bash-completion .*

echo "source <(kubectl completion bash)" >> ~/.bashrc *# добавление автодополнения autocomplete постоянно в командную оболочку bash.*

KUBECTL

sudo apt-get update

sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg

echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list

sudo apt-get update

sudo apt-get install -y kubectl

kubectl version –client

MINIKUBE - однокластерный для тестов.

До minikube должен быть установлен docker.

Установка

https://kubernetes.io - documents - tasks - Install tools

curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube\_latest\_amd64.deb

sudo dpkg -i minikube\_latest\_amd64.deb

minikube start

docker ps

minikube status

kubectl delete pods --all

kubectl delete services --all

kubectl get nodes - сколько всего нод, включая мастер, включая версии.

kubectl get nodes -owide - тоже, плюс инфа о нодах, включая ОС.

kubectl exec -it mynginx – bash – зайти в Pod.

KUBEADM

Поставить docker и kubeadm на все машины. Инициализация kubeadm на мастере.

РАЗВОРАЧИВАНИЕ POD

kubectl run nginx --image nginx - создаёт POD, развёртывая в нём контейнер docker.

kubectl get pods -owide - краткая информация о подах

kubectl describe pod nginx – полная информация о поде.

Создание POD

apiVersion: v1

kind: Pod

metadata:

name: nginx

labels:

app: nginx

tier: frondend

spec:

containers:

- name: nginx

image: nginx

- name: busybox

image: busybox

kubectl apply -f pod1.yml – создание POD. Apply и create равнозначны.

REPLICASET

kubectl create -f replicaset-definition.yml

kubectl get replicaset или get rs

kubectl relace -f replicaset-definition.yml - при измении количества реплик в файле.

kubectl delete replicaset имя-replicaset

kubectl scale --replicas=6 -f replicaset-definition.yml - не меняет реплики в файле.

при последующем обновлении соскейлится вниз.

kubectlscale rs myapp-replicaset --replicas=2

kubectl edit rs myapp-replicaset

replicaset.yml

apiVersion: apps/v1

kind: ReplicaSet

metadata:

name: myapp-replicaset

labels:

app: myapp

spec:

selector:

matchLabels:

app: myapp

replicas: 3

template:

metadata:

name: nginx2

labels:

app: myapp

spec:

containers:

- name: nginx

image: nginx

DEPLOYMENT

При создании deployment’а запускается rollut – snapshot, ревизия версий.

Быстрый способ: create deployment nginx --image=nginx

Kubectl create -f deployment.yml –record – рекомендуемый способ создания deployment c записью в историю.

Kubectl get deploy

kubectl apply -f deployment-definition.yml - заменяет файл с одной версией на новый.

kubectl set image deployment/myapp-deployment nginx=nginx:1.7.1 - обновлет без изменений в файле

kubectl set image deploy myapp-deployment nginx=nginx:1.18 --record

kubectl scale deployment myapp-deployment –replicas=2

kubectl edit deploy myapp-deployment –record – изменение версии с записью в history.

deployment.yml

apiVersion: apps/v1

kind: Deployment

metadata:

name: myapp-deployment

labels:

tier: frontend

app: nginx

spec:

selector:

matchLabels:

app: myapp

replicas: 4

template:

metadata:

name: nginx2

labels:

app: myapp

spec:

containers:

- name: nginx

image: nginx

ROLLOUT

kubectl rollout status deployment/myapp-deployment

kubectl rollout history deployment/myapp-deployment

kubectl rollout undo deployment/myapp-deployment

SERVICES

Node port 30000-32767

Port службы

Target port – порт пода.

kubectl get svc

kubectl create -f service.yml

minikube service myapp-service –url

sevise.yml

apiVersion: v1

kind: Service

metadata:

name: myapp-service

spec:

type: NodePort

ports:

- port: 80

targetPort: 80

nodePort: 30004

selector:

app: myapp

CLUSTER IP

Kubectl create -f -service-definition.yml

Kubectl get svc

service-definition.yml

apiVersion: v1

kind: Service

metadata:

name: back-end

spec:

type: ClusterIP

ports:

- port: 80

targetPort: 80

#nodePort: 30004

selector:

app: myapp

type: back-end