**Docker commands**

**Createcontainer docker run –dit centos**

**Start docker start container id**

**Stop docker stop containerid**

**Remove docker rm –f c.id**

**See running containers docker ps**

**Running n exec docker ps –a**

**Login to container docker exec-it c.id /bin/bash**

**Check dockerversion docker –version**

**Container utilisation docker stats**

**Crete imagefrom running con docker commit c.id dockerid/name**

**Port docker run –dit –p:80:80 httpd**

**Giving own name docker run –dit –-name swamy httpd**

**Checkimages dockerimages**

**Remove image docker rmi imagename**

**Push docker push imageid**

**Pull docker pull imageid**

**Check container docker inspect c.id{imagename}**

**Crete own image using dockerfile docker build –t swamy . {docker build .}**

**Create volume docker volume create volume1**

**See volumes docker volume ls**

**Inspect vol docker volume inspect**

**Remove volume docker volume rm volumename**

**All netwrks docker networl ls**

**Create network docker network create networkswmy**

**Delete docker networl rm networkname**

**Inspect docker network inspect networkid**

**Howmny with this image docker history httpd**

**Official images docker search imagename**

**Redy on ports docker logs container id**

**Keeplooking on logs docker logs –f c.id**

**Docker pause c.id**

**docker unpause c.id**

**docker kill c.id**

**all stopped containers deletedocker container prune**

**diskusage in docker docker system df**

**events docker system events**

**display sys wide info docker system info**

**unused images con networks docker prune –a**

**container size docker ps –a --size**

**[ ]# docker run --name=knote --network=knote –p3000:3000 –e Mongo\_URL=mongodb://mongo:27017/dev knote**

**[ ]# kubectl create deployment devhttpd1 --image=httpd –o yaml –dry-run=client > devhttpd.yaml**

**[ ]# kubectl expose deployment devhttpd1 --type=LoadBalancer –port=81 –target-port=80 - o yaml --dryrun=client > devhttpd service.yaml**

**K8s commands**

**Kubectl kubecontroller**

**Pod creation kubectl create deployment dep.name**

**View pods kubectl get pods**

**Fulldetails kubectl get pods –o wide**

**Delete pod kubectl delete pod podname**

**Loginto pod kubectl exec –it podname -- /bin/bash**

**Logs kubectl logs podname**

**Since 1hour kubectl logs –since = 1h podname**

**All details kubectl get all**

**Sorting pods lastfew min kubectl get events –sort- by=.metadata.creationTimestamp**

**Tailing of pods kubectl logs –f podname**

**Yamlfile kubectl get pods podname > .yamlfile**

**All pods in namespaces kubectl get pods –all-namespaces**

**Kubectl get services –all-namespaces –sort –by =.metadata.name**

**Pod huge information kubectl describe pod podname**

**Deployment**

**Create deployment kubectl create deployment devhttpd –image=httpd**

**Login deployment u canot login deployment**

**View deployments kubectl get deployments**

**Delete kubectl delete deployment d.name**

**Creating replicas kubectl scale deployment dep.name –replicas=3**

**Service**

**Create service kubectl expose deployment devhttpd1 --type=LoadBalancer -- port=81 --target-port=80**

**Create service from yaml file ::kubectl apply –f yamlfile**

**Cluster informations kubectl cluster –info**

**Delete cluster kubectl delete cluster –name ----.com …s3bucket yes**

**Upgrading app in k8s kubectl set image deployment name –image=d.name:0.02release**

**[ ]# kubectl create deployment devhttpd1 --image=httpd –o yaml –dry-run=client > devhttpd.yaml**

**[ ]# kubectl expose deployment devhttpd1 --type=LoadBalancer –port=81 –target-port=80 - o yaml --dryrun=client > devhttpd service.yaml**

**Node cpu ,mem kubectl top node**

**Pod cpu,mem kubectl top pod**

**Shortcuts**

**Events—ev**

**Replicasets—rs**

**Namespaces—ns**

**Nodes—no**

**Pods—po**

**deployments = deploy**

**componentstatuses = cs**

**configmaps = cm**

**endpoints = ep**

**events = ev**

**limitranges = limits**

**namespaces = ns**

**nodes = no**

**persistentvolumeclaims = pvc**

**persistentvolumes = pv**

**pods = po**

**replicationcontrollers = rc**

**resourcequotas = quota**

**serviceaccounts = sa**

**services = svc**

**customresourcedefinitions = crd, crds**

**daemonsets = ds**

**deployments = deploy**

**replicasets = rs**

**statefulsets = sts**

**horizontalpodautoscalers = hpa**

**cronjobs = cj**

**certificiaterequests = cr, crs**

**certificates = cert, certs**

**certificatesigningrequests = csr**

**ingresses = ing**

**networkpolicies = netpol**

**podsecuritypolicies = psp**

**replicasets = rs**

**scheduledscalers = ss**

**priorityclasses = pc**

**storageclasses = sc**

**see yaml for existed : kubectl get pods or deployments that name –o yaml**

**healthcheks : kubectl get componentstatuses**

**kubectl get events –sort-by=.metadata.creationtimestamp**

**replicaset kubectl explain replicaset**

**kubectl set image deployment dep.name heelo-world-rest-api hello-world-rest-api = in28minutes/dockerimage:latest**

***in linux vedios***

***cat auth.log | cut –d “ ” –f 11-----------gives input only 11 line***

***cat auth.log | cut –d “” –f 11 | sort | uniq |wc –l***

***for using above we need hugefile***

***ex. Cd/var/log,,,,,cd /var/log/fail2ban.log***