Kubernetes labs

Table of Contents

1. LAB : Learning Kubernetes Cluster Installation

Getting started → https://kubernetes.io/docs/setup/

1.1 Kubectl and minikube installation

Check Virtual Box is installed: Outils système → Oracle VM VirtualBox
User = usera / password = usera

Command line kubectl to install

```
[usera@lx-1-3 ~]$ curl -LO https://storage.googleapis.com/kubernetes-release/release/v1.24.6/bin/linux/amd64/kubectl
[usera@lx-1-3 ~]$ chmod +x kubectl
[usera@lx-1-3 ~]$ sudo mv kubectl /bin/
```

Add bash completion

```
[usera@lx-7-1 ~]$ sudo yum install bash-completion
[usera@lx-7-1 ~]$ echo 'source <(kubectl completion bash)' >>~/.bashrc
[usera@lx-7-1 ~]$ source <(kubectl completion bash)
```

Follow instructions: https://kubernetes.io/docs/tasks/tools/install-minikube/ in order to install minikube
Start minikube with **at least 4GB de RAM** with driver virtual box

```
[usera@lx-1-3 ~]$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-latest.x86_64.rpm
[usera@lx-1-3 ~]$ sudo rpm -ivh minikube-latest.x86_64.rpm
[usera@lx-1-3 ~]$ minikube start --kubernetes-version=v1.24.6 --memory=4g --driver=virtualbox
```

<<< Be patient during preloaded-images download >>>

```
Préparation...                          ################################# [100%]
Mise à jour / installation...
   1:minikube-1.15.0-0                   ################################# [100%]
[droinpy@oc5004167418 ALTRAN]$ minikube start --memory 4096 --driver=virtualbox
😄  minikube v1.15.0 sur Redhat 7.8
🆕  Kubernetes 1.19.4 is now available. If you would like to upgrade, specify: --kubernetes-version=v1.19.4
✨  Utilisation du pilote virtualbox basé sur le profil existant
💿  Downloading VM boot image ...
    > minikube-v1.15.0.iso.sha256: 65 B / 65 B [-------------] 100.00% ? p/s 0s
    > minikube-v1.15.0.iso: 181.00 MiB / 181.00 MiB  100.00% 2.21 MiB p/s 1m22s
👍  Démarrage du noeud de plan de contrôle minikube dans le cluster minikube
🔄  Restarting existing virtualbox VM for "minikube" ...
🐳  Préparation de Kubernetes v1.19.2 sur Docker 19.03.12...
🔎  Verifying Kubernetes components...
🌟  Enabled addons: default-storageclass, storage-provisioner
🏄  Done! kubectl is now configured to use "minikube" cluster and "" namespace by default
```

Crtl+c if blocked “Enabled addons: default-storageclass, storage-provisioner”

1.2 Explore your Kubernetes cluster

Check Kubernetes version

[usera@lx-1-3 ~]$ **kubectl version**

WARNING: This version information is deprecated and will be replaced with the output from kubectl version --short. Use --output=yaml|json to get the full version.

Client Version: version.Info{Major:"1", Minor:"24", GitVersion:"v1.24.6", GitCommit:"b39bf148cd654599a52e867485c02c4f9d28b312", GitTreeState:"clean", BuildDate:"2022-09-21T13:19:24Z", GoVersion:"go1.18.6", Compiler:"gc", Platform:"windows/amd64"}

Kustomize Version: v4.5.4

Server Version: version.Info{Major:"1", Minor:"24", GitVersion:"v1.24.6", GitCommit:"b39bf148cd654599a52e867485c02c4f9d28b312", GitTreeState:"clean", BuildDate:"2022-09-21T13:12:04Z", GoVersion:"go1.18.6", Compiler:"gc", Platform:"linux/amd64"}

Get nodes

[usera@lx-1-3 ~]$ **kubectl get nodes**

NAME	STATUS	ROLES	AGE	VERSION
minikube	Ready	control-plane	54s	v1.24.6

You cluster is mono-node.

Your Kubernetes cluster is ready to be used.

2. LAB : YAML file

Source → https://kubernetes.io/docs/tasks/access-application-cluster/web-ui-dashboard/
Source → https://github.com/kubernetes/dashboard/blob/master/docs/user/access-control/creating-sample-user.md

2.1 Have a look on Yaml file

Have a look on https://raw.githubusercontent.com/kubernetes/dashboard/v2.6.1/aio/deploy/recommended.yaml

2.2 Kubernetes dashboard

Deploy Kubernetes Dashboard v2.6.1

```
[usera@lx-1-3 ~]$ kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.6.1/aio/deploy/recommended.yaml
namespace/kubernetes-dashboard created
serviceaccount/kubernetes-dashboard created
service/kubernetes-dashboard created
secret/kubernetes-dashboard-certs created
secret/kubernetes-dashboard-csrf created
secret/kubernetes-dashboard-key-holder created
configmap/kubernetes-dashboard-settings created
role.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrole.rbac.authorization.k8s.io/kubernetes-dashboard created
rolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
deployment.apps/kubernetes-dashboard created
service/dashboard-metrics-scraper created
deployment.apps/dashboard-metrics-scraper created
```

Creating a Service Account
We are creating Service Account with name admin-user in namespace kubernetes-dashboard first.

```
[usera@lx-1-3 ~]$ vim sa-admin.yaml
apiVersion: v1
kind: ServiceAccount
```

```
metadata:
  name: admin-user
  namespace: kubernetes-dashboard
[usera@lx-1-3 ~]$ kubectl apply -f sa-admin.yaml
serviceaccount/admin-user created
```

Tip: How to see 2 spaces for indentation with vi

```
[usera@lx-1-3 ~]$ vi sa-admin.yaml
: set list
apiVersion:.v1
kind:.ServiceAccount
metadata:
..name:.admin-user
..namespace:.kubernetes-dashboard

: set listchars=space:.
: set nolist
```

Creating a ClusterRoleBinding

```
[usera@lx-1-3 ~]$ cat <<EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: admin-user
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin-user
  namespace: kubernetes-dashboard
EOF
clusterrolebinding.rbac.authorization.k8s.io/admin-user created
```

Getting a Bearer Token

Now we need to find token we can use to log in. Execute following command:

```
[usera@lx-1-3 ~]$ kubectl -n kubernetes-dashboard create token admin-user
```

```
eyJhb[...}16XjD-qKxKg
```

Copy/backup token value

Launch Kubernetes proxy

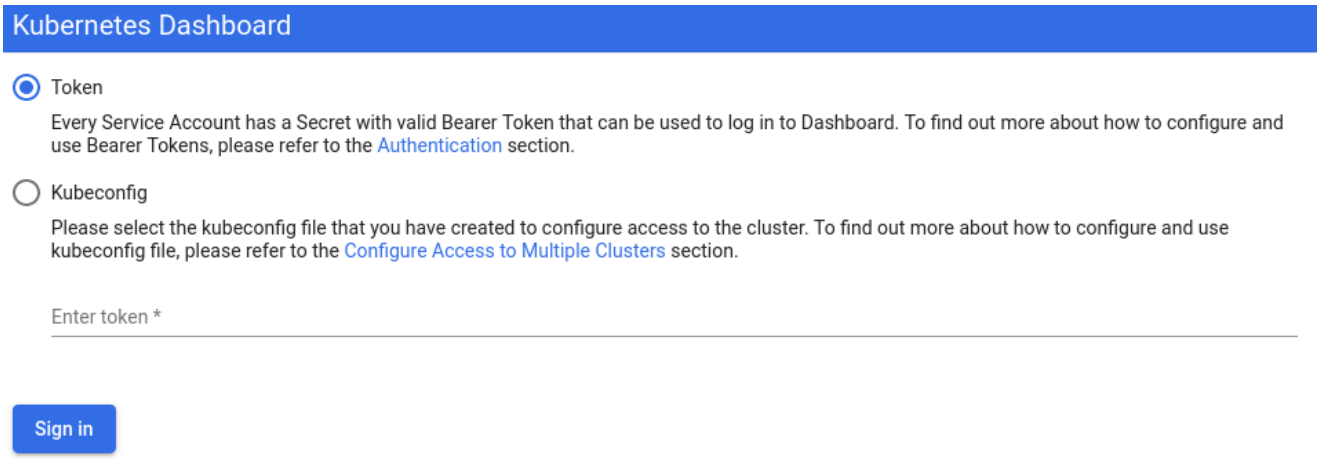
It creates a proxy or application-level gateway between localhost and the Kubernetes API server. It also allows to serve static content over specified HTTP path.

The command kubectl proxy runs kubectl in a mode where it acts as a reverse proxy. It handles locating the apiserver and authenticating.

```
[usera@lx-1-3 ~]$ kubectl proxy
Starting to serve on 127.0.0.1:8001
```

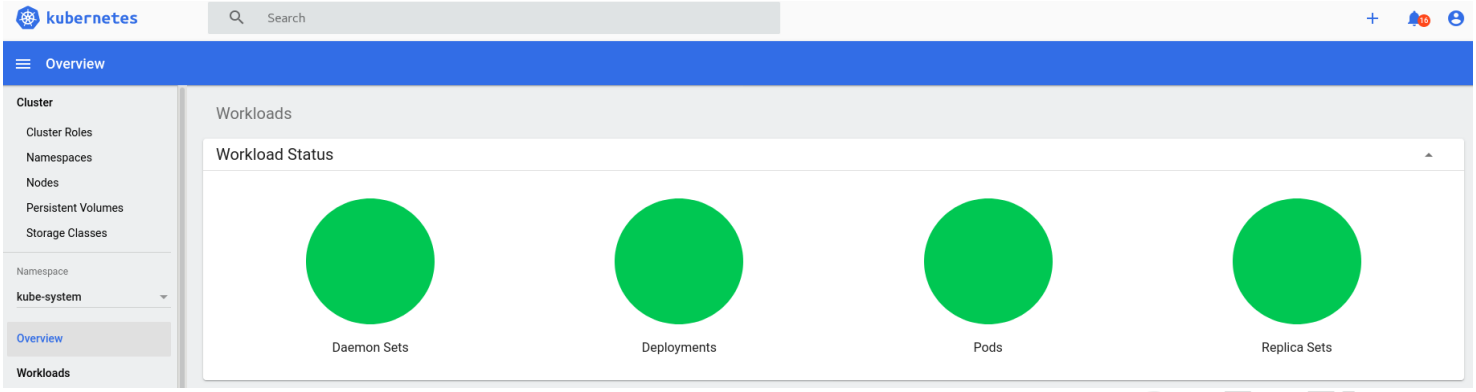
Kubectl proxy will make Dashboard available at http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/.

and this shall open up page as shown below:



Select token authentication and paste token and sign in

Navigate to kube-system namespace



Ctrl+C to stop kubectl proxy

3. LAB : Kubernetes Architecture

3.1 Explore your Kubernetes cluster

List API resources

```
[usera@lx-1-3 ~]$ kubectl api-resources
NAME                    SHORTNAMES    APIGROUP    NAMESPACED   KIND
[…]
pods                    po                                true       Pod
```

Use shortnames …

List all pods

```
[usera@lx-1-3 ~]$ kubectl get po --all-namespaces
NAMESPACE     NAME                          READY   STATUS    RESTARTS   AGE
kube-system   coredns-6955765f44-bdkqr          1/1     Running   0          18m
kube-system   etcd-minikube                     1/1     Running   0          18m
kube-system   kube-apiserver-minikube           1/1     Running   0          18m
kube-system   kube-controller-manager-minikube  1/1     Running   0          18m
kube-system   kube-proxy-rjs7f                  1/1     Running   0          18m
kube-system   kube-scheduler-minikube           1/1     Running   0          18m
kube-system   storage-provisioner               1/1     Running   0          19m
```

Do you retrieve Kubernetes master components : etcd apiserver controller-manager scheduler ?

4. LAB : Deploy echo-server App

Source → https://kubernetes.io/docs/tutorials/hello-minikube/

4.1 Run first your first Kubernetes deployment

Deploy echo-server with container image echo-server
Example

```
> kubectl create deployment echo-server --image=k8s.gcr.io/echoserver:1.4
deployment.apps/echo-server created
> kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
echo-server          1/1     1            1           166m
> kubectl get pods -o wide
NAME                      READY   STATUS    RESTARTS   AGE   IP          NODE       NOMINATED
NODE   READINESS GATES
echo-server-5f445fd9b7-nlddm   1/1     Running   0          28s   172.17.0.5   minikube   <none>
<none>
```

Remove your POD (identify pod name) and see what happens

```
> kubectl delete po echo-server-5f445fd9b7-nlddm
pod "echo-server-65cdcdb74b-f6slc" deleted
> kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
echo-server          1/1     1            1           167m
> kubectl get pods -o wide
NAME                      READY   STATUS    RESTARTS   AGE   IP          NODE       NOMINATED
NODE   READINESS GATES
echo-server-5f445fd9b7-hlmqv   1/1     Running   0          21s   172.17.0.5   minikube   <none>
<none>
```

→ Why is pod running ?

4.2 Access from Outside the cluster

Expose echo-server application

```
> kubectl expose deployment echo-server --type=NodePort --port=8080
service/echo-server exposed
> kubectl get pod,svc
NAME                          READY   STATUS    RESTARTS   AGE
```

```
pod/echo-server-5f445fd9b7-hlmqv   1/1     Running   0          96s

NAME                 TYPE       CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
service/echo-server   NodePort    10.98.125.247   <none>       8080:31818/TCP   18s
service/kubernetes    ClusterIP   10.96.0.1       <none>       443/TCP          8d
```

Test is on http://<any IP of node within cluster:<port number is randomly generated and it can be different for you>
Reminder NodePort range is from **30 000 to 32 767**

Retrieve IP from node

```
> kubectl get nodes -o wide
NAME       STATUS   ROLES                  AGE    VERSION   INTERNAL-IP     EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION   CONTAINER-RUNTIME
minikube   Ready    control-plane,master   5h4m   v1.22.3   192.168.59.100   <none>       Buildroot 2021.02.4   4.19.202         docker://20.10.8
```

Example here: http://192.168.59.100:31818

4.3 Scale up the deployment

Scale the deployment up to two replicas

```
 > kubectl scale deployment echo-server --replicas=2
deployment.apps/echo-server scaled
> kubectl get po -o wide
NAME                          READY   STATUS    RESTARTS   AGE   IP          NODE       NOMINATED NODE   READINESS GATES
echo-server-5f445fd9b7-hlmqv   1/1     Running   0          4m    172.17.0.5   minikube   <none>           <none>
echo-server-5f445fd9b7-zrvcc   1/1     Running   0          10s   172.17.0.6   minikube   <none>           <none>
```

5. LAB : Persistent Volume (PV)

Source → https://kubernetes.io/docs/tasks/configure-pod-container/configure-persistent-volume-storage/

5.1 Content preparation

Create a directory and add content on 1 node of cluster

```
[usera@lx-1-3 ~]$ minikube ssh

$ sudo su -
# mkdir -p /tmp/data
# echo 'Hello from Kubernetes storage' > /tmp/data/index.html
# cat /tmp/data/index.html
Hello from Kubernetes storage
# logout
$ logout
```

5.2 Persistent Volume creation

Create pv file – **vi pv-volume.yaml**

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: task-pv-volume
  labels:
    type: local
spec:
  storageClassName: manual
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: /tmp/data
```

Create pv and check

```
> kubectl create -f pv-volume.yaml
```

```
persistentvolume/task-pv-volume created
> kubectl get pv --all-namespaces
NAME            CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAIM
STORAGECLASS   REASON   AGE
task-pv-volume   10Gi       RWO           Retain           Available           manual                 23s
```

→ What do you notice ?

5.3 Bound PVC to PV

Create new file - **vi pv-claim.yaml**

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: task-pv-claim
  namespace: default
spec:
  storageClassName: manual
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 3Gi
```

Create pvc and check

```
> kubectl create -f pv-claim.yaml
persistentvolumeclaim/task-pv-claim created
> kubectl get pvc --all-namespaces
NAMESPACE   NAME          STATUS   VOLUME          CAPACITY   ACCESS MODES
STORAGECLASS   AGE
default     task-pv-claim   Bound    task-pv-volume   10Gi       RWO           manual         4m
```

5.4 Launch Nginx pod using storage

Create file – **vi pv-pod.yaml**

```
apiVersion: v1
kind: Pod
metadata:
```

```
  name: task-pv-pod
  namespace: default
spec:
  volumes:
    - name: task-pv-storage
      persistentVolumeClaim:
        claimName: task-pv-claim
  containers:
    - name: task-pv-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - mountPath: "/usr/share/nginx/html"
          name: task-pv-storage
```

Create pod and go to inside container and check if you see index.html

```
> kubectl create -f pv-pod.yaml
pod/task-pv-pod created
> kubectl get pod task-pv-pod
NAME         READY   STATUS    RESTARTS   AGE
task-pv-pod   1/1     Running   0          6m45s
> kubectl exec -it task-pv-pod -- /bin/bash
root@task-pv-pod:/# ls -latr /usr/share/nginx/html/
total 12
drwxr-xr-x 3 root root 4096 Aug 14 00:36 ..
-rw-r--r-- 1 root root   30 Aug 22 18:27 index.html
drwxr-xr-x 2 root root 4096 Aug 22 18:27 .
root@task-pv-pod:/# apt update
root@task-pv-pod:/# curl http://localhost/
Hello from Kubernetes storage
root@task-pv-pod:/# exit
exit
```

Delete pod pvc and pv

```
> kubectl get po task-pv-pod
NAME         READY   STATUS    RESTARTS   AGE
task-pv-pod   1/1     Running   0          79s
```

```
> kubectl delete po task-pv-pod
pod "task-pv-pod" deleted
> kubectl get po task-pv-pod
Error from server (NotFound): pods "task-pv-pod" not found
> kubectl delete pvc task-pv-claim
persistentvolumeclaim "task-pv-claim" deleted
> kubectl delete pv task-pv-volume
persistentvolume "task-pv-volume" deleted
```

Question → Why is pod deleted ?

5.5 Dynamic provisioning

Storage Class exists – called standard and use hostpath as provisioner

```
[usera@lx-1-3 ~]$  kubectl get sc
NAME            PROVISIONER                RECLAIMPOLICY   VOLUMEBINDINGMODE
ALLOWVOLUMEEXPANSION   AGE
standard (default)   k8s.io/minikube-hostpath   Delete          Immediate           false                  9d
```

Copy previous pvc file into pv-claim-sc-standard

```
[usera@lx-1-3 ~]$  cp -p pv-claim.yaml pv-claim-sc-standard.yaml
```

Edit and change storage class name - **vi pv-claim-sc-standard.yaml**

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: task-pv-claim
  namespace: default
spec:
  storageClassName: standard
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 3Gi
```

Storage Class exists – called standard and use hostpath as provisioner

```
[usera@lx-1-3 ~]$ kubectl create -f pv-claim-sc-standard.yaml
```

```
persistentvolumeclaim/task-pv-claim created
[usera@lx-1-3 ~]$ kubectl get pv,pvc
NAME                                       CAPACITY   ACCESS MODES   RECLAIM POLICY
STATUS   CLAIM                 STORAGECLASS   REASON   AGE
persistentvolume/pvc-72c9897a-8cf7-4d56-ac26-583d337655c6   3Gi        RWO            Delete
Bound    default/task-pv-claim   standard               5m51s

NAME                            STATUS   VOLUME                                     CAPACITY   ACCESS
MODES   STORAGECLASS   AGE
persistentvolumeclaim/task-pv-claim   Bound    pvc-72c9897a-8cf7-4d56-ac26-583d337655c6   3Gi
RWO            standard       5m51s
```

Create pod and go to inside container and check if you see index.html

```
[usera@lx-1-3 ~]$ kubectl create -f pv-pod.yaml
pod/task-pv-pod created
[usera@lx-1-3 ~]$ kubectl get pod task-pv-pod
NAME         READY   STATUS    RESTARTS   AGE
task-pv-pod   1/1     Running   0          6m45s
[usera@lx-1-3 ~]$ kubectl exec -it task-pv-pod -- /bin/bash
root@task-pv-pod:/# ls -latr /usr/share/nginx/html/
total 12
drwxr-xr-x 3 root root 4096 Aug 14 00:36 ..
drwxr-xr-x 2 root root 4096 Aug 22 18:27 .
exit
```

Why is it empty ?
Retrieve pv name equal to pvc bound

```
[usera@lx-1-3 ~]$ kubectl get pv,pvc
NAME                                       CAPACITY   ACCESS MODES   RECLAIM POLICY
STATUS   CLAIM                 STORAGECLASS   REASON   AGE
persistentvolume/pvc-72c9897a-8cf7-4d56-ac26-583d337655c6   3Gi        RWO            Delete
Bound    default/task-pv-claim   standard               5m51s

NAME                            STATUS   VOLUME                                     CAPACITY   ACCESS
MODES   STORAGECLASS   AGE
persistentvolumeclaim/task-pv-claim   Bound    pvc-72c9897a-8cf7-4d56-ac26-583d337655c6   3Gi
RWO            standard       5m51s
[usera@lx-1-3 ~]$  kubectl get pv pvc-72c9897a-8cf7-4d56-ac26-583d337655c6  -o yaml | grep
hostpath
```

```
    pv.kubernetes.io/provisioned-by: k8s.io/minikube-hostpath
    path: /tmp/hostpath-provisioner/default/task-pv-claim
```

Create index.hml in /tmp/hostpath-provisioner/default/task-pv-claim

```
[usera@lx-1-3 ~]$ minikube ssh

$ sudo su -
# echo 'Hello from Kubernetes storage' >
/tmp/hostpath-provisioner/default/task-pv-claim/index.html
# cat /tmp/hostpath-provisioner/default/task-pv-claim/index.html
Hello from Kubernetes storage
# logout
$ logout
```

Check now if it is ok

```
[usera@lx-1-3 ~]$ kubectl exec -it task-pv-pod -- /bin/bash
root@task-pv-pod:/# ls -latr /usr/share/nginx/html/
total 12
drwxr-xr-x 3 root root 4096 Nov  5 18:21 ..
drwxrwxrwx 2 root root 4096 Nov 15 15:07 .
-rw-r--r-- 1 root root   30 Nov 15 15:07 index.html
root@task-pv-pod:/# apt update
root@task-pv-pod:/# curl http://localhost/
Hello from Kubernetes storage
root@task-pv-pod:/# exit
exit
```

Clean up: delete pod pvc

```
[usera@lx-1-3 ~]$ kubectl delete po task-pv-pod
pod "task-pv-pod" deleted
[usera@lx-1-3 ~]$ kubectl delete pvc task-pv-claim
persistentvolumeclaim "task-pv-claim" deleted
[usera@lx-1-3 ~]$ kubectl get pv
No resources found
```

Question → Why is pv deleted ?

6. LAB : Example Voting App

Source → https://github.com/dockersamples/example-voting-app

6.1 Deploy it

Go to K8specifications directory

```
[usera@lx-1-3 ~]$ git clone https://github.com/dockersamples/example-voting-app
[usera@lx-1-3 ~]$ cd example-voting-app/k8s-specifications/
[usera@lx-1-3 k8s-specifications]$ ls -altr
total 44
-rw-rw-r--  1 usera usera  317 Feb 18 17:51 worker-deployment.yaml
-rw-rw-r--  1 usera usera  216 Feb 18 17:51 vote-service.yaml
-rw-rw-r--  1 usera usera  369 Feb 18 17:51 vote-deployment.yaml
-rw-rw-r--  1 usera usera  221 Feb 18 17:51 result-service.yaml
-rw-rw-r--  1 usera usera  383 Feb 18 17:51 result-deployment.yaml
-rw-rw-r--  1 usera usera  203 Feb 18 17:51 redis-service.yaml
-rw-rw-r--  1 usera usera  492 Feb 18 17:51 redis-deployment.yaml
-rw-rw-r--  1 usera usera  191 Feb 18 17:51 db-service.yaml
-rw-rw-r--  1 usera usera  634 Feb 18 17:51 db-deployment.yaml
```

Add “ **namespace: vote**” in yaml file

```
[usera@lx]$ for i in $(ls *.yaml); do sed -i '0,/name: /{s/name:.*/&\n  namespace: vote/}' $i; done
```

Create namespace vote and deploy it

```
[usera@lx-1-3 k8s-specifications]$ kubectl create namespace vote
namespace/vote created
[usera@lx-1-3 k8s-specifications]$ kubectl create -f redis-deployment.yaml -f result-deployment.yaml -f vote-deployment.yaml -f worker-deployment.yaml -f db-deployment.yaml
deployment.apps/redis created
deployment.apps/result created
deployment.apps/vote created
deployment.apps/worker created
deployment.apps/db created
[usera@lx-1-3 k8s-specifications]$ kubectl create -f redis-service.yaml -f result-service.yaml -f vote-service.yaml -f db-service.yaml
service/redis created
service/result created
service/vote created
```

```
service/db created
```

Where is my application ?

```
[usera@lx-1-3 k8s-specifications]$ kubectl get po
NAME                          READY   STATUS    RESTARTS   AGE
echo-server-6799c4cf46-c7xfp   1/1     Running   0          154m
[usera@lx-1-3 k8s-specifications]$ kubectl -n vote get po
NAME                      READY   STATUS              RESTARTS   AGE
db-57c4fd6875-txf6b       1/1     Running             0          2m16s
redis-5cff845b56-bhqts    1/1     Running             0          2m16s
result-59977485df-khbpr   1/1     Running             0          2m16s
vote-6c79f79647-msc7v     1/1     Running             0          2m16s
worker-64cb5879d9-ftfg9   0/1     ContainerCreating   0          2m16s
```

6.2 Kubens

Source → https://github.com/ahmetb/kubectx

It allows to change into namespace

Linux : just copy kubens and kubectx scripts into your path

```
[usera@lx-1-3 k8s-specifications]$ wget
https://github.com/ahmetb/kubectx/releases/download/v0.9.1/kubens
[ usera@lx-1-3 k8s-specifications]$ sudo mv kubens /bin/kubens
[ usera@lx-1-3 k8s-specifications]$ sudo chmod +x /bin/kubens
```

Mac : via homebrew

I can now change namespace to vote

```
[usera@lx-1-3 k8s-specifications]$ kubens vote
Context "minikube" modified.
Active namespace is "vote".
```

Or I don’t use kubens but kubectl config

```
[usera@lx-7-11 k8s-specifications]$ kubectl config get-contexts
CURRENT   NAME       CLUSTER    AUTHINFO   NAMESPACE
*         minikube   minikube   minikube   default
[usera@lx-7-11 k8s-specifications]$ kubectl config set-context --current --namespace=vote
```

Check deployment state

```
[usera@lx-1-3 k8s-specifications]$ kubectl get po,svc
```

```
NAME                          READY   STATUS             RESTARTS   AGE
pod/db-7dc869cdcd-dcp4b       0/1     CrashLoopBackOff    6         9m10s
pod/redis-97698dc95-7d7rz     1/1     Running             0         9m10s
pod/result-75c97ddf5d-d7tlj   1/1     Running             0         9m10s
pod/vote-598c68df78-7cthx     1/1     Running             0         9m10s
pod/worker-5764d777cd-2lpwm   0/1     ContainerCreating   0         9m10s

NAME            TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)         AGE
service/db      ClusterIP   10.97.30.206    <none>        5432/TCP        9m10s
service/redis   ClusterIP   10.103.251.108  <none>        6379/TCP        9m10s
service/result  NodePort    10.111.68.55    <none>        5001:31001/TCP  9m10s
service/vote    NodePort    10.99.176.90    <none>        5000:31000/TCP  9m10s
```

Retrieve IP master

```
[usera@lx-1-3 k8s-specifications]$ kubectl get nodes -o wide
NAME      STATUS   ROLES    AGE   VERSION   INTERNAL-IP     EXTERNAL-IP   OS-IMAGE
KERNEL-VERSION   CONTAINER-RUNTIME
minikube   Ready    master   9d    v1.19.2   192.168.59.100   <none>        Buildroot 2020.02.6
4.19.114         docker://19.3.12
```

Test application http://192.168.59.100:31000 and http://192.168.59.100:31001
Does application work properly ?

7. LAB: Expose an app

Source → https://www.ovh.com/blog/getting-external-traffic-into-kubernetes-clusterip-nodeport-loadbalancer-and-ingress/

7.1 Proxy and NodePort

Kube proxy – see Kubernetes Dashboard – Chapter 1.5
NodePort – see Kubens – voting app – Chapter 2.2

7.2 Ingress Installation

Warning about lab resources : worker nodes >=4GB
Source → https://kubernetes.github.io/ingress-nginx/deploy/#minikube

Enable ingress addons

```
[usera@lx-1-3 k8s-specifications]$ minikube addons list
[usera@lx-1-3 k8s-specifications]$ minikube addons enable ingress
🔎  Verifying ingress addon...
🌟  The 'ingress' addon is enabled
[usera@lx-1-3 k8s-specifications]$ kubectl -n ingress-nginx get po | grep ingress
ingress-nginx-admission-create-7jspv        0/1     Completed   0          5m47s
ingress-nginx-admission-patch-mj8qn         0/1     Completed   0          5m47s
ingress-nginx-controller-558664778f-t2md6   1/1     Running     0          5m48s
```

7.3 Ingress Configuration

Add new file into k8s-specifications: **vote-ing.yaml**

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: vote-ingress
  namespace: vote
spec:
  rules:
  - host: vote.local.com
    http:
      paths:
      - path: /
```

```
      pathType: Prefix
      backend:
        service:
          name: vote
          port:
            number: 5000
```

Note: https://stackoverflow.com/questions/64125048/get-error-unknown-field-servicename-in-io-k8s-api-networking-v1-ingressbacken

Check Ingress Resource

```
[usera@lx-1-3 k8s-specifications]$ kubectl create -f vote-ing.yaml
ingress.networking.k8s.io/vote-ingress created
[usera@lx-1-3 k8s-specifications]$ kubectl get ing
NAME          CLASS    HOSTS           ADDRESS         PORTS   AGE
vote-ingress  <none>   vote.local.com  localhost  80     80s
```

Simulate DNS entre host:vote.local.com <-> IP of host.docker.internal
Edit /etc/hosts and add DNS entry as follows (depending on your env)

```
[usera@lx-1-3 k8s-specifications]$ sudo vim /etc/hosts
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

# minikuke
192.168.59.100 vote.local.com
```

Check if http://vote.local.com is now linked to vote service

8. LAB : Configmap / secret

Source → https://schoolofdevops.github.io/ultimate-kubernetes-bootcamp/9-vote-configmaps_and_secrets/

8.1 ConfigMap

Add new file – **vote-cm.yaml** onto k8s-specifications directory

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: vote
  namespace: vote
data:
  OPTION_A: Visa
  OPTION_B: Mastercard
```

Create it

```
[usera@lx-1-3 k8s-specifications]$ kubectl create -f vote-cm.yaml
configmap/vote created
```

See your config map

```
[usera@lx-1-3 k8s-specifications]$ kubectl describe cm vote
Name:         vote
Namespace:    vote
Labels:       <none>
Annotations:  <none>
Data
====
OPTION_A:
----
Visa
OPTION_B:
----
Mastercard
Events:  <none>
```

Go to ‘vote’ namespace and check if voting app is still alive since chapter 3

```
[usera@lx-1-3 k8s-specifications]$ kubens vote
Context "minikube" modified.
Active namespace is "vote".
[usera@lx-1-3 k8s-specifications]$ kubectl get po,svc,cm
NAME                          READY   STATUS    RESTARTS   AGE
pod/db-57c4fd6875-txf6b       1/1     Running   0          30h
pod/redis-5cff845b56-bhqts    1/1     Running   0          30h
pod/result-59977485df-khbpr   1/1     Running   0          30h
pod/vote-6c79f79647-msc7v     1/1     Running   0          30h
pod/worker-64cb5879d9-ftfg9   1/1     Running   0          30h

NAME            TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
service/db      ClusterIP   10.101.249.96  <none>        5432/TCP         30h
service/redis   ClusterIP   10.100.99.238  <none>        6379/TCP         30h
service/result  NodePort    10.98.60.250   <none>        5001:31001/TCP   30h
service/vote    NodePort    10.107.4.77    <none>        5000:31000/TCP   30h

NAME            DATA   AGE
configmap/vote  2      4m38s
```

Check if http://vote.local.com is still works
Question; → why is application is still alive with cats and dogs ?

Vote deployment must be updated accordingly.
Modify vote-deployment.yaml and following block (be careful with **indentation!**)

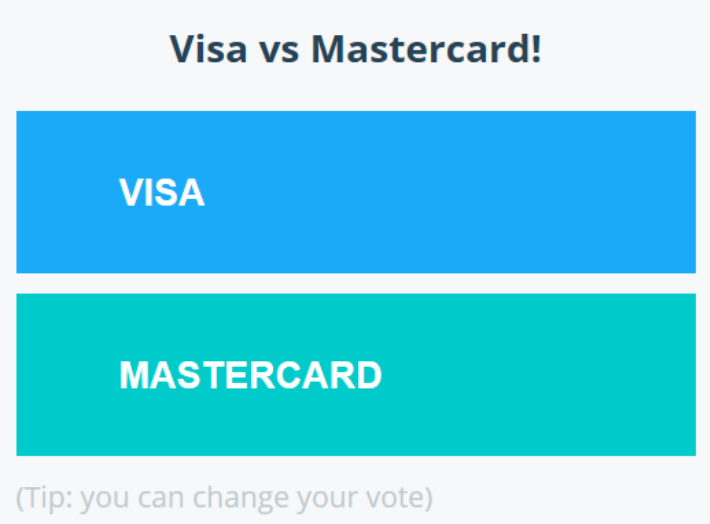
```
    envFrom:
     - configMapRef:
        name: vote
```

```
17      spec:
18        containers:
19        - image: dockersamples/examplevotingapp_vote:before
20          name: vote
21          ports:
22          - containerPort: 80
23            name: vote
24          envFrom:
25            - configMapRef:
26                name: vote
27
```

Apply the modification done in deployment

```
[usera@lx-1-3 k8s-specifications]$ kubectl apply -f vote-deployment.yaml
Warning: kubectl apply should be used on resource created by either kubectl create --save-config or kubectl apply
deployment.apps/vote configured
```

Check if http://vote.local.com is still works
And now … you see “Visa” and “Mastercard”



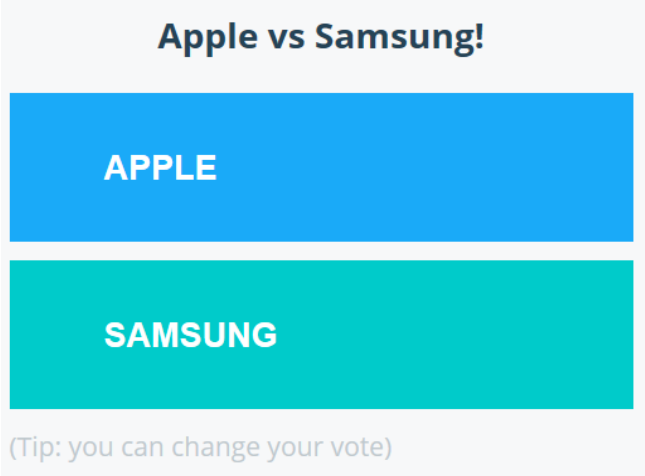
Edit vote-cm.yaml with new values Apple and Samsung
Save it and Kubernetes apply on it

```
[usera@lx-1-3 k8s-specifications]$ kubectl apply -f vote-cm.yaml
Warning: kubectl apply should be used on resource created by either kubectl create --save-config or kubectl apply
configmap/vote configured
```

Delete vote pod

```
[usera@lx-1-3 k8s-specifications]$ kubectl get po | grep vote
NAME                READY   STATUS    RESTARTS   AGE
vote-557ff6dd8-l5ghm      1/1     Running   0          32m
[usera@lx-1-3 k8s-specifications]$ kubectl delete po vote-557ff6dd8-l5ghm
pod "vote-557ff6dd8-l5ghm" deleted
P[usera@lx-1-3 k8s-specifications]$ kubectl get po
NAME                    READY   STATUS    RESTARTS   AGE
vote-557ff6dd8-9bst4      1/1     Running   0          36s
```

Check if http://vote.local.com is still works
And now … you see “Apple” and “Samsung”



Note: Automatic Updation of deployments on ConfigMap Updates
Currently, updating configMap does not ensure a new rollout of a deployment. What this means is even after updading configMaps, pods will not immediately reflect the changes.
There is a feature request for this https://github.com/kubernetes/kubernetes/issues/22368

Currently, this can be done by using immutable configMaps.
Create a configMaps and apply it with deployment.
To update, create a new configMaps and do not update the previous one. Treat it as immutable.
Update deployment spec to use the new version of the configMaps. This will ensure immediate update.

Note: reloader project addresses to this point: https://github.com/stakater/Reloader

8.2 Secret

We will add https support for local.com domain.

Create self-signed certificate

```
[usera@lx-1-3 k8s-specifications]$ openssl genrsa -out example.key 2048
Generating RSA private key, 2048 bit long modulus
...........+++
.....................................................+++
e is 65537 (0x10001)
[usera@lx-1-3 k8s-specifications]$ openssl req -new -key example.key -out example.csr -subj
"/C=FR/L=SOPHIA/O=Example/OU=IT/CN=*.local.com"
```

```
[usera@lx-1-3 k8s-specifications]$ openssl x509 -req -days 366 -in example.csr -signkey example.key -out example.crt
Signature ok
subject=/C=FR/L=SOPHIA/O=Example/OU=IT/CN=*.local.com
Getting Private key
```

Create TLS secret within ‘vote’ namespace

```
[usera@lx-1-3 k8s-specifications]$ ls -altr example.*
-rw-rw-r--. 1 droinpy droinpy 1679 20 févr. 11:50 example.key
-rw-rw-r--. 1 droinpy droinpy  976 20 févr. 11:51 example.csr
-rw-rw-r--. 1 droinpy droinpy 1147 20 févr. 11:52 example.crt
[usera@lx-1-3 k8s-specifications]$ kubens vote
Context "minikube" modified.
Active namespace is "vote".
[usera@lx-1-3 k8s-specifications]$ kubectl create secret tls ing-local-com --key example.key --cert example.crt
secret/ing-local-com created
```

Update ingress rule accordingly with correct indentation

```
        service:
          name: vote
          port:
            number: 5000
 tls:
 - hosts:
   - vote.local.com
   secretName: ing-local-com
```

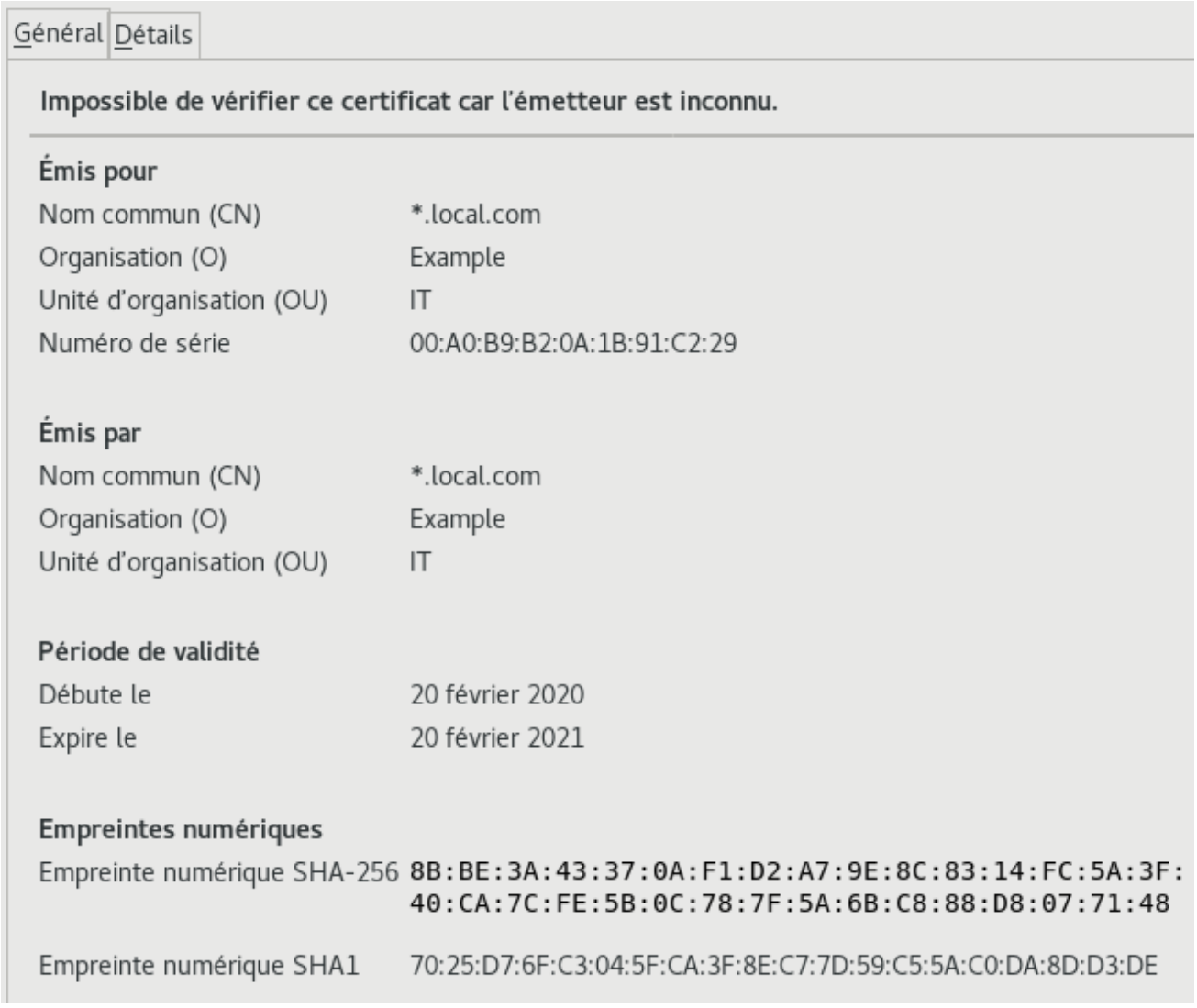
```
  rules:
  - host: vote.local.com
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: vote
            port:
              number: 5000
  tls:
  - hosts:
    - vote.local.com
    secretName: ing-local-com
```

Apply the modification

```
[usera@lx-1-3 k8s-specifications]$ kubectl apply -f vote-ing.yaml
ingress.networking.k8s.io/vote-ingress configured
```

Check if https://vote.local.com answers now with new certificate

PS: be careful with firefox protection:
Disable protection.



9. LAB: Metering

Source → https://github.com/kubernetes-sigs/metrics-server

9.1 Enable metering addons

Enable metrics-server addons

```
[usera@lx-1-3 kubernetes]$ kubectl top node
error: Metrics API not available
[usera@lx-1-3 k8s-specifications]$ minikube addons list
[usera@lx-1-3 k8s-specifications]$ minikube addons enable metrics-server
🔎  The 'metrics-server' addon is enabled
[usera@lx-1-3 k8s-specifications]$ kubectl get po --all-namespaces|grep metrics-server
kube-system          metrics-server-d9b576748-gvjcs            1/1     Running    0          70s
```

9.2 Test with TOP

Run Kubernetes top

```
[usera@lx-1-3 kubernetes]$ kubectl top node
NAME       CPU(cores)   CPU%   MEMORY(bytes)   MEMORY%
minikube   1203m        60%    1541Mi          42%
[usera@lx-1-3 kubernetes]$ kubectl top pod --all-namespaces
NAMESPACE       NAME                                       CPU(cores)   MEMORY(bytes)
ingress-nginx   nginx-ingress-controller-777868f4c5-szw8t   4m           97Mi
kube-system     coredns-6955765f44-bdkqr                   2m           6Mi
kube-system     coredns-6955765f44-qnbld                   2m           6Mi
kube-system     etcd-minikube                              12m          31Mi
kube-system     kube-apiserver-minikube                    22m          247Mi
kube-system     kube-controller-manager-minikube           7m           34Mi
kube-system     kube-proxy-rjs7f                           1m           12Mi
kube-system     kube-scheduler-minikube                    2m           10Mi
kube-system     kubernetes-dashboard-7c54d59f66-d7wsl      1m           10Mi
kube-system     metrics-server-77c968df9d-td4dm            1m           11Mi
kube-system     storage-provisioner                        1m           13Mi
vote            db-8c6dbd86f-6v46n                         190m         35Mi
vote            redis-97698dc95-7d7rz                      125m         3Mi
vote            result-75c97ddf5d-d7tlj                    1m           35Mi
vote            vote-74bc6dff6-w7hl8                       1m           60Mi
```

```
vote          worker-5764d777cd-2lpwm                 884m        33Mi
```

9.3 Voting App preparation

Go to ‘vote’ namespace and check if voting app is still alive since chapter 3

```
[usera@lx-1-3 k8s-specifications]$ kubens vote
Context "minikube" modified.
Active namespace is "vote".
[usera@lx-1-3 k8s-specifications]$ kubectl get po,svc,cm
NAME                          READY   STATUS    RESTARTS   AGE
pod/db-57c4fd6875-txf6b       1/1     Running   0          30h
pod/redis-5cff845b56-bhqts    1/1     Running   0          30h
pod/result-59977485df-khbpr   1/1     Running   0          30h
pod/vote-6c79f79647-msc7v     1/1     Running   0          30h
pod/worker-64cb5879d9-ftfg9   1/1     Running   0          30h
NAME            TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
service/db      ClusterIP   10.101.249.96  <none>        5432/TCP         30h
service/redis   ClusterIP   10.100.99.238  <none>        6379/TCP         30h
service/result  NodePort    10.98.60.250   <none>        5001:31001/TCP   30h
service/vote    NodePort    10.107.4.77    <none>        5000:31000/TCP   30h
NAME            DATA   AGE
configmap/vote  2      4m38s
```

Check if http://vote.local.com is still works

9.4 Add resources limit

Edit vote-deployment.yaml onto k8s-specifications directory and add following block as follows

```
      resources:
        limits:
          cpu: "200m"
          memory: "250Mi"
        requests:
          cpu: "100m"
          memory: "50Mi"
```

```
k8s-specifications > ! vote-deployment.yaml > {} spec >
31                path: /
32                port: 80
33            initialDelaySeconds: 5
34            periodSeconds: 3
35        envFrom:
36            - configMapRef:
37                name: vote
38        resources:
39            limits:
40                cpu: "200m"
41                memory: "250Mi"
42            requests:
43                cpu: "100m"
44                memory: "50Mi"
45
```

Apply it

```
[usera@lx-1-3 k8s-specifications]$ kubectl apply -f vote-deployment.yaml
deployment.apps/vote configured
```

Vote pod must restart

Check if vote is running on http://vote.local.com

10. LAB : LimitRange and ResourceQuota

Source → https://schoolofdevops.github.io/ultimate-kubernetes-bootcamp/cluster-administration/#defining-quotas

10.1 LimitRange

If your namespace has a resource quota, it is helpful to have a default value in place for CPU limit. Here are two of the restrictions that a resource quota imposes on a namespace:

Every Container that runs in the namespace must have its own CPU limit.

The total amount of CPU used by all Containers in the namespace must not exceed a specified limit.

If a Container does not specify its own CPU limit, it is given the default limit, and then it can be allowed to run in a namespace that is restricted by a quota.

Create into k8s-specifications/**vote-limitrange.yaml**

```
apiVersion: v1
kind: LimitRange
metadata:
  name: vote-limits
  namespace: vote
spec:
  limits:
  - default:
      cpu: 500m
      memory: 1Gi
    defaultRequest:
      cpu: 100m
      memory: 200Mi
    type: Container
```

Create it

```
[usera@lx-1-3 k8s-specifications]$ kubectl create -f vote-limitrange.yaml
limitrange/vote-limits created
```

Delete result pod

```
[usera@lx-1-3 k8s-specifications]$ kubectl  get po
NAME                    READY   STATUS    RESTARTS   AGE
db-5454f7cb64-2q8ds       1/1     Running   2          2d8h
redis-554668f9bf-trhxk    1/1     Running   6          12d
```

```
result-6fd78dc9b8-s7vv6   1/1     Running   2          2d8h
vote-54478b5f5f-5z44x     1/1     Running   0          24m
worker-785dc75fbf-pz5c6   1/1     Running   12         11d
[usera@lx-1-3 k8s-specifications]$ kubectl  describe po result-6fd78dc9b8-s7vv6 | grep -A 5
Limits:
[usera@lx-1-3 k8s-specifications]$ kubectl delete po result-6fd78dc9b8-s7vv6
pod "result-6fd78dc9b8-s7vv6" deleted
[usera@lx-1-3 k8s-specifications]$ kubectl get po
NAME                      READY   STATUS    RESTARTS   AGE
db-5454f7cb64-2q8ds       1/1     Running   2          2d8h
redis-554668f9bf-trhxk    1/1     Running   6          12d
result-6fd78dc9b8-tbglv   1/1     Running   0          39s
vote-54478b5f5f-5z44x     1/1     Running   0          25m
worker-785dc75fbf-pz5c6   1/1     Running   12         11d
[usera@lx-1-3 k8s-specifications]$ kubectl describe po result-6fd78dc9b8-tbglv| grep -A 5 Limits:
    Limits:
      cpu:     500m
      memory:  1Gi
    Requests:
      cpu:        100m
      memory:     200Mi
```

10.2 Quota

A resource quota, defined by a ResourceQuota object, provides constraints that limit aggregate resource consumption per namespace. It can limit the quantity of objects that can be created in a namespace by type, as well as the total amount of compute resources that may be consumed by resources in that project.

Create into k8s-specifications/**vote-quota.yaml**

```
apiVersion: v1
kind: ResourceQuota
metadata:
  name: vote-quota
  namespace: vote
spec:
  hard:
    limits.cpu: "4"
    limits.memory: 16Gi
```

Create it

```
> kubectl create -f vote-quota.yaml
limitrange/vote-quota created
```

Describe it

```
> kubectl describe quota
Name:          vote-quota
Namespace:     vote
Resource       Used    Hard
--------       ----    ----
limits.cpu     700m    4
limits.memory  1274Mi  16Gi
```

11. LAB : Liveness and readiness

Source → https://schoolofdevops.github.io/ultimate-kubernetes-bootcamp/pods-health-probes/

11.1 Preparation

Go to 'vote' namespace and check if voting app is still alive since chapter 3

```
[usera@lx-1-3 k8s-specifications]$ kubens vote
Context "minikube" modified.
Active namespace is "vote".
[usera@lx-1-3 k8s-specifications]$ kubectl get po,svc,cm
NAME                          READY   STATUS    RESTARTS   AGE
pod/db-57c4fd6875-txf6b       1/1     Running   0          30h
pod/redis-5cff845b56-bhqts    1/1     Running   0          30h
pod/result-59977485df-khbpr   1/1     Running   0          30h
pod/vote-6c79f79647-msc7v     1/1     Running   0          30h
pod/worker-64cb5879d9-ftfg9   1/1     Running   0          30h

NAME            TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
service/db      ClusterIP   10.101.249.96  <none>        5432/TCP         30h
service/redis   ClusterIP   10.100.99.238  <none>        6379/TCP         30h
service/result  NodePort    10.98.60.250   <none>        5001:31001/TCP   30h
service/vote    NodePort    10.107.4.77    <none>        5000:31000/TCP   30h

NAME            DATA   AGE
configmap/vote  2      4m38s
```

Check if http://vote.local.com still works

11.2 Liveness

Liveness probe checks the status of the pod(whether it is running or not). If livenessProbe fails, then the pod is subjected to its restart policy. The default state of livenessProbe is *Success*.
Let us add liveness probe to our *frontend* deployment. The following probe will check whether it is able to *access the port or not*.

Add following block (be careful about indentation) **vote-deployment.yaml**

```
    livenessProbe:
      tcpSocket:
        port: 80
      initialDelaySeconds: 5
      periodSeconds: 5
```

```
17      spec:
18        containers:
19        - image: dockersamples/examplevotingapp_vote:before
20          name: vote
21          ports:
22          - containerPort: 80
23            name: vote
24          livenessProbe:
25            tcpSocket:
26              port: 80
27            initialDelaySeconds: 5
28            periodSeconds: 5
```

Apply it

```
[usera@lx-1-3 k8s-specifications]$ kubectl apply -f vote-deployment.yaml
deployment.apps/vote configured
```

Check pod status

```
[usera@lx-1-3 k8s-specifications]$ kubectl get po,svc
NAME                          READY   STATUS    RESTARTS   AGE
pod/db-8c6dbd86f-6v46n        1/1     Running   3          3d20h
pod/redis-97698dc95-7d7rz     1/1     Running   3          3d21h
pod/result-75c97ddf5d-d7tlj   1/1     Running   3          3d21h
pod/vote-74bc6dff6-cdbtp      1/1     Running   0          99s
pod/worker-5764d777cd-2lpwm   1/1     Running   9          3d21h

NAME            TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
service/db      ClusterIP   10.97.30.206     <none>        5432/TCP         3d21h
service/redis   ClusterIP   10.103.251.108   <none>        6379/TCP         3d21h
service/result  NodePort    10.111.68.55     <none>        5001:31001/TCP   3d21h
service/vote    NodePort    10.99.176.90     <none>        5000:31000/TCP   3d21h
```

Re-do the above instructions but with liveness probe having wrong port on purposes

What is the consequence ?

```
[usera@lx-1-3 k8s-specifications]$ kubectl get po,svc
NAME                          READY   STATUS             RESTARTS   AGE
pod/db-8c6dbd86f-6v46n        1/1     Running            3          3d20h
pod/redis-97698dc95-7d7rz     1/1     Running            3          3d21h
pod/result-75c97ddf5d-d7tlj   1/1     Running            3          3d21h
pod/vote-56fb94cbbc-44nv2     0/1     CrashLoopBackOff   4          105s
pod/worker-5764d777cd-2lpwm   1/1     Running            9          3d21h

NAME            TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
service/db      ClusterIP   10.97.30.206     <none>        5432/TCP         3d21h
service/redis   ClusterIP   10.103.251.108   <none>        6379/TCP         3d21h
service/result  NodePort    10.111.68.55     <none>        5001:31001/TCP   3d21h
service/vote    NodePort    10.99.176.90     <none>        5000:31000/TCP   3d21h
[usera@lx-1-3 k8s-specifications]$ kubectl describe po vote-56fb94cbbc-44nv2
Name:               vote-56fb94cbbc-44nv2
[…]
Normal   Created    22m (x4 over 23m)     kubelet, docker-desktop  Created container vote
  Normal   Started    22m (x4 over 23m)     kubelet, docker-desktop  Started container vote
  Warning  Unhealthy  22m (x10 over 23m)    kubelet, docker-desktop  Liveness probe failed: dial tcp
10.1.0.88:81: connect: connection refused
  Warning  BackOff    3m31s (x81 over 21m)  kubelet, docker-desktop  Back-off restarting failed
container
```

Fix the situation.

11.3 Readiness

Readiness probe checks whether your application is ready to serve the requests. When the readiness probe fails, the pod's IP is removed from the end point list of the service. The default state of readinessProbe is *Success*.
Readiness probe is configured just like liveness probe. But this time we will use *httpGet request*.

Add following block (be careful about indentation) **vote-deployment.yaml**

```
      readinessProbe:
        httpGet:
          path: /
          port: 80
        initialDelaySeconds: 5
```

```
periodSeconds: 3
```

```yaml
17      spec:
18        containers:
19        - image: dockersamples/examplevotingapp_vote:before
20          name: vote
21          ports:
22          - containerPort: 80
23            name: vote
24          livenessProbe:
25            tcpSocket:
26              port: 80
27            initialDelaySeconds: 5
28            periodSeconds: 5
29          readinessProbe:
30            httpGet:
31              path: /
32              port: 80
33            initialDelaySeconds: 5
34            periodSeconds: 3
```

Apply it

```
[usera@lx-1-3 k8s-specifications]$ kubectl apply -f vote-deployment.yaml
deployment.apps/vote configured
```

Check pod status

```
[usera@lx-1-3 k8s-specifications]$ kubectl get po
NAME                    READY   STATUS    RESTARTS   AGE
db-57c4fd6875-txf6b     1/1     Running   1          2d1h
redis-5cff845b56-bhqts  1/1     Running   1          2d1h
result-59977485df-khbpr 1/1     Running   1          2d1h
vote-78787cd64-2nbxz    1/1     Running   0          23s
worker-64cb5879d9-ftfg9 1/1     Running   2          2d1h
[usera@lx-1-3 k8s-specifications]$ kubectl describe po vote-78787cd64-2nbxz
Name:             vote-78787cd64-2nbxz
Namespace:        vote
  Ready:          True
  Restart Count:  0
  Liveness:       tcp-socket :80 delay=5s timeout=1s period=5s #success=1 #failure=3
  Readiness:      http-get http://:80/ delay=5s timeout=1s period=3s #success=1 #failure=3
```

12. LAB : Rolling Update

Source → https://kubernetes.io/docs/tutorials/kubernetes-basics/update/update-intro/

12.1 Preparation

Run nginx deployment in default namespace

```
[usera@lx-1-3 kubernetes-training]$ kubens default
Context "minikube" modified.
Active namespace is "default".
[usera@lx-1-3 kubernetes-training]$ kubectl create deployment rolling-update-nginx --image=nginx:1.9.1
deployment.apps/rolling-update-nginx created
[usera@lx-1-3 kubernetes-training]$ kubectl scale --replicas=5 deploy rolling-update-nginx
deployment.apps/rolling-update-nginx scaled
```

And see default rolling update strategy type

```
[usera@lx-1-3 kubernetes-training]$ kubectl get deploy rolling-update-nginx -o yaml | grep -A 4 " strategy:"
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
```

And see default rolling update strategy type

```
[usera@lx-1-3 kubernetes-training]$ kubectl get po
NAME                                    READY   STATUS    RESTARTS   AGE
rolling-update-nginx-687b6db999-fmq6d   1/1     Running   0          69s
rolling-update-nginx-687b6db999-jzf9n   1/1     Running   0          69s
rolling-update-nginx-687b6db999-mqkht   1/1     Running   0          69s
rolling-update-nginx-687b6db999-s5df7   1/1     Running   0          69s
rolling-update-nginx-687b6db999-tmpk6   1/1     Running   0          69s
[usera@lx-1-3 kubernetes-training]$ kubectl get rs
NAME                                    DESIRED   CURRENT   READY   AGE
rolling-update-nginx-687b6db999   5         5         0       4s
[usera@lx-1-3 kubernetes-training]$ kubectl rollout status deployment rolling-update-nginx
deployment "rolling-update-nginx" successfully rolled out
```

12.2 Apply an update

Change image version

```
[usera@lx-1-3 kubernetes-training]$ kubectl set image deployment rolling-update-nginx
nginx=nginx:1.7.9
deployment.extensions/rolling-update-nginx image updated
[usera@lx-1-3 kubernetes-training]$ kubectl get po
NAME                                READY   STATUS              RESTARTS   AGE
rolling-update-nginx-687b6db999-fmq6d   1/1     Running             0          9m
rolling-update-nginx-687b6db999-jzf9n   1/1     Running             0          9m
rolling-update-nginx-687b6db999-mqkht   1/1     Running             0          9m
rolling-update-nginx-687b6db999-s5df7   1/1     Running             0          9m
rolling-update-nginx-6d875b959c-5td4m   0/1     ContainerCreating   0          24s
rolling-update-nginx-6d875b959c-blchb   0/1     ContainerCreating   0          24s
rolling-update-nginx-6d875b959c-jcm5b   0/1     ContainerCreating   0          24s
```

And see default rolling update strategy type

```
[usera@lx-1-3 kubernetes-training]$ kubectl get po
NAME                                READY   STATUS    RESTARTS   AGE
rolling-update-nginx-6d875b959c-5td4m   1/1     Running   0          4m5s
rolling-update-nginx-6d875b959c-7jmq8   1/1     Running   0          3m31s
rolling-update-nginx-6d875b959c-blchb   1/1     Running   0          4m5s
rolling-update-nginx-6d875b959c-fllgj   1/1     Running   0          3m28s
rolling-update-nginx-6d875b959c-jcm5b   1/1     Running   0          4m5s
[usera@lx-1-3 kubernetes-training]$ kubectl get rs
NAME                          DESIRED   CURRENT   READY   AGE
rolling-update-nginx-687b6db999   0         0         0       71s
rolling-update-nginx-6d875b959c   5         5         5       18s
[usera@lx-1-3 kubernetes-training]$ kubectl exec  rolling-update-nginx-6d875b959c-jcm5b -- nginx
-v
nginx version: nginx/1.7.9
[usera@lx-1-3 kubernetes-training]$ kubectl rollout status deployment rolling-update-nginx
deployment "rolling-update-nginx" successfully rolled out
[usera@lx-1-3 kubernetes-training]$ kubectl rollout history deployment rolling-update-nginx
deployment.extensions/rolling-update-nginx
REVISION  CHANGE-CAUSE
1         <none>
2         <none>
```

12.3 Undo

Change image version

```
[usera@lx-1-3 kubernetes-training]$ kubectl rollout history deployment rolling-update-nginx --revision 2
deployment.extensions/rolling-update-nginx with revision #2
Pod Template:
Labels:       pod-template-hash=6d875b959c
run=rolling-update-nginx
Containers:
rolling-update-nginx:
Image:      nginx:1.7.9
Port:       <none>
Host Port:  <none>
Environment:        <none>
Mounts:     <none>
Volumes:      <none>
[usera@lx-1-3 kubernetes-training]$ kubectl rollout undo deployment rolling-update-nginx --to-revision=1
deployment.extensions/rolling-update-nginx rolled back
[usera@lx-1-3 kubernetes-training]$ kubectl get po
NAME                                READY   STATUS    RESTARTS   AGE
rolling-update-nginx-687b6db999-6pr9c   1/1     Running   0          3m50s
rolling-update-nginx-687b6db999-8jnn5   1/1     Running   0          3m48s
rolling-update-nginx-687b6db999-chszw   1/1     Running   0          3m50s
rolling-update-nginx-687b6db999-k9jrg   1/1     Running   0          3m50s
rolling-update-nginx-687b6db999-pq58z   1/1     Running   0          3m48s
P[usera@lx-1-3 kubernetes-training]$ kubectl get rs
NAME                          DESIRED   CURRENT   READY   AGE
hello-node-78cd77d68f          2         2         2       3d4h
rolling-update-nginx-687b6db999   5         5         5       6m21s
rolling-update-nginx-6d875b959c   0         0         0       5m28s
[usera@lx-1-3 kubernetes-training]$ kubectl exec  rolling-update-nginx-687b6db999-6pr9c -- nginx -v
nginx version: nginx/1.9.1
```

12.4 Cleaning

Delete deployment

```
[usera@lx-1-3 kubernetes-training]$ kubectl delete deployment rolling-update-nginx
deployment.extensions "rolling-update-nginx" deleted
```

13. LAB : Docker Image Creation

Source → https://minikube.sigs.k8s.io/docs/handbook/registry/

13.1 Registry

Enable ingress addons

```
[usera@lx-1-3 ~]$ minikube addons list
[usera@lx-1-3 ~]$ minikube addons enable registry
🔎  Verifying registry addon...
🌟  The 'registry' addon is enabled
[usera@lx-1-3 ~]$ kubectl -n kube-system get po | grep registry
registry-79jdt                          1/1     Running     0          2m2s
registry-proxy-2fllf                    1/1     Running     0          2m2s
```

13.2 Develop Docker Image

Create an empty directory and go to it

```
[usera@lx-1-3 kubernetes-training]$ minikube ssh
$ mkdir myimageweb
$ cd myimageweb
```

create vi **Dockerfile** file

```
# Dockerfile
# Indicate base image.
FROM centos
# Metadata indicating an image maintainer.
LABEL maintainer="py@contoso.com"
# End of life of centos8
# https://stackoverflow.com/questions/70926799/centos-through-vm-no-urls-in-mirrorlist
RUN sed -i 's/mirrorlist/#mirrorlist/g' /etc/yum.repos.d/CentOS-Linux-* && \
    sed -i 's|#baseurl=http://mirror.centos.org|baseurl=http://vault.centos.org|g' /etc/yum.repos.d/CentOS-Linux-*
# Use yum to install httpd
RUN yum install -y httpd
# Creates an HTML file and adds content to this file.
RUN echo "Hello World - Dockerfile" > /var/www/html/index.html
```

```
# Indicate port exposition
EXPOSE 80
# Sets a command or process that will run each time a container is run from the new image.
ENTRYPOINT /usr/sbin/httpd -D FOREGROUND
```

Build Docker image

```
$ docker build -t myimageapache:v2 .
Sending build context to Docker daemon
[…]
```

Create a container from built image but using another port for external

```
$ docker run -d --name myweb -p 81:80 myimageapache:v2
sha256:39696ccbb1637055bde536e9b89596a4e9b04e2dbf578b945b5da317b2ef5663
```

Test it on http://localhost:81

```
$ curl http://localhost:81
Hello World - Dockerfile
```

You can now push it into local registry running in minikube

```
$ docker tag myimageapache:v2 localhost:5000/myimageapache:v2
$ docker push localhost:5000/myimageapache:v2
The push refers to repository [localhost:5000/myimageapache]
ce6f84b4264c: Pushed
d03b0ead7206: Pushed
291f6e44771a: Pushed
v2: digest: sha256:b05fe9956697f0694da0d457299a973503d250fe699f7ec1a980ad7efdecd7dc size: 948
$ exit
```

13.3 Deploy previous image into Kubernetes

Create a deployment using myimageapache Docker image

```
[usera@lx-1-3 ~]$ kubens default
[usera@lx-1-3 ~]$ kubectl create deployment myweb --image=localhost:5000/myimageapache:v2
deployment.apps/myweb created
[usera@lx-1-3 ~]$ kubectl expose deployment myweb --port 80 --type NodePort
```

```
service/myweb exposed
[usera@lx-1-3 ~]$ kubectl get po,svc
NAME                                READY   STATUS    RESTARTS   AGE
pod/echo-server-6799c4cf46-c7xfp    1/1     Running   0          5h
pod/myweb-595457d495-gf8xt          1/1     Running   0          108s

NAME                 TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
service/echo-server  NodePort    10.98.125.247   <none>        8080:31818/TCP   23h
service/kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP          9d
service/myweb        NodePort    10.100.226.36   <none>        80:30955/TCP     3s
usera@lx-1-3 ~]$ curl http://192.168.59.100:30955
Hello World - Dockerfile
```

14. LAB: Sidecar

Source → https://kubernetes.io/docs/tasks/access-application-cluster/communicate-containers-same-pod-shared-volume/

14.1 Create a pod that runs two containers

Use yaml file from https://gist.github.com/matthewpalmer/047738f3b3804a5e91d08909ce7024a9 but remove version for nginx container.

Copy/paste the content into **pod2containers.yaml** onto kubernetes-training directory via **gedit**

```
# Example YAML configuration for the sidecar pattern.
# It defines a main application container which writes
# the current date to a log file every five seconds.
# The sidecar container is nginx serving that log file.
# (In practice, your sidecar is likely to be a log collection
# container that uploads to external storage.)
# To run:
#   kubectl apply -f pod.yaml
# Once the pod is running:
#
#   (Connect to the sidecar pod)
#   kubectl exec pod-with-sidecar -c sidecar-container -it bash
#
#   (Install curl on the sidecar)
#   apt-get update && apt-get install curl
#
#   (Access the log file via the sidecar)
#   curl 'http://localhost:80/app.txt'

apiVersion: v1
kind: Pod
metadata:
  name: pod-with-sidecar
spec:
  # Create a volume called 'shared-logs' that the
  # app and sidecar share.
  volumes:
  - name: shared-logs
    emptyDir: {}
```

```
# In the sidecar pattern, there is a main application
# container and a sidecar container.
containers:
# Main application container
- name: app-container
  # Simple application: write the current date
  # to the log file every five seconds
  image: alpine # alpine is a simple Linux OS image
  command: ["/bin/sh"]
  args: ["-c", "while true; do date >> /var/log/app.txt; sleep 5;done"]
  # Mount the pod's shared log file into the app
  # container. The app writes logs here.
  volumeMounts:
  - name: shared-logs
    mountPath: /var/log

# Sidecar container
- name: sidecar-container
  # Simple sidecar: display log files using nginx.
  # In reality, this sidecar would be a custom image
  # that uploads logs to a third-party or storage service.
  image: nginx
  ports:
    - containerPort: 80
  # Mount the pod's shared log file into the sidecar
  # container. In this case, nginx will serve the files
  # in this directory.
  volumeMounts:
  - name: shared-logs
    mountPath: /usr/share/nginx/html # nginx-specific mount path
```

Save and apply it into ‘default’ namespace

```
[usera@lx-1-3 kubernetes-training]$ kubens default
Context "minikube" modified.
Active namespace is "default".
[usera@lx-1-3 kubernetes-training]$ kubectl apply -f pod2containers.yaml
pod/pod-with-sidecar created
```

14.2 Test it

See the number of containers in pod-with-sidecar

```
[usera@lx-1-3 kubernetes-training]$ kubectl get po
NAME              READY   STATUS    RESTARTS   AGE
pod-with-sidecar   2/2     Running   0          28s
```

Go to sidecar-container and check if it communicates between application-container and sidecar-container

```
[usera@lx-1-3 kubernetes-training]$ kubectl exec pod-with-sidecar -c sidecar-container -it -- bash
root@pod-with-sidecar:/# apt-get update && apt-get install curl -y
Get:1 http://security-cdn.debian.org/debian-security buster/updates InRelease [65.4 kB]
Running hooks in /etc/ca-certificates/update.d...
done.
root@pod-with-sidecar:/# curl 'http://localhost:80/app.txt'
Sun Dec  1 14:10:33 UTC 2019
Sun Dec  1 14:18:29 UTC 2019
root@pod-with-sidecar:/# exit
[usera@lx-1-3 kubernetes-training]$
```

Question: → what do you see ?

Clean-up

```
[usera@lx-1-3 kubernetes-training]$ kubectl delete po pod-with-sidecar
pod "pod-with-sidecar" deleted
```

15. LAB : Kubernetes Cluster Installation multi-nodes

Getting started → https://kubernetes.io/docs/setup/
Source → https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/
Note : Dockershim has been removed from the Kubernetes project as of release 1.24

15.1 OS installation

Hardware configuration
Master hardware configuration 2CPU and 4GB – disk 40GB no swap – **Ubuntu 20.04.5** LTS
Node hardware configuration 1CPU and 2GB – disk 40GB no swap – **Ubuntu 20.04.5** LTS
OS configuration
Keyboard french Default installation and select OpenSSH server installation
Password: usera/usera and root/root

Before starting, check that minikube has 192.168.59.100 as IP address:

```
$ minikube ip
192.168.59.100
```

Update **/etc/hosts**

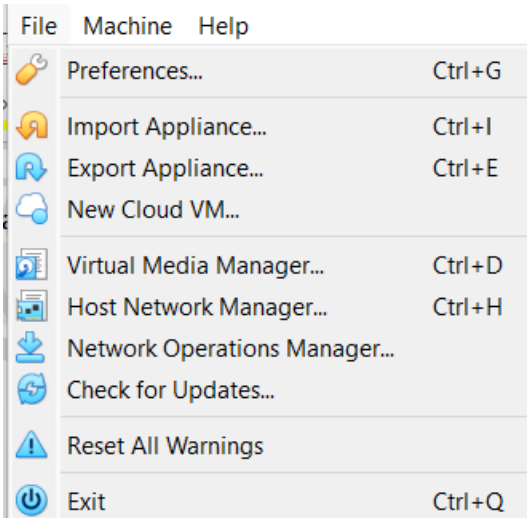
```
#
192.168.59.101 k8smaster k8smaster.lab.example.com
192.168.59.102 k8snode1 k8snode1.lab.example.com
192.168.59.103 k8snode2 k8snode2.lab.example.com
```

Wait for teacher who will create following directory and upload 3 ova files for k8smaster, k8ssnode1 and k8snode2

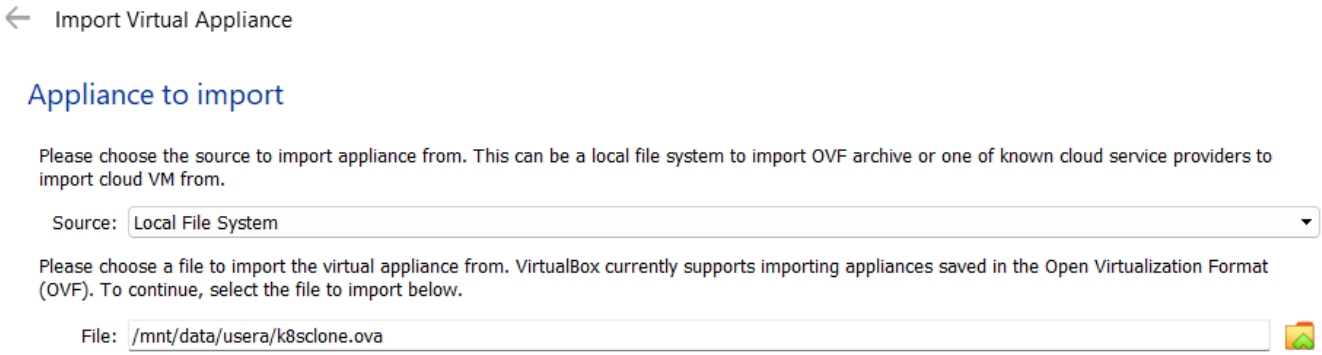
```
$ sudo mkdir /mnt/data/usera
$ sudo chown usera:usera /mnt/data/usera
```

15.2 Import k8smaster, k8snode1 and k8snode2

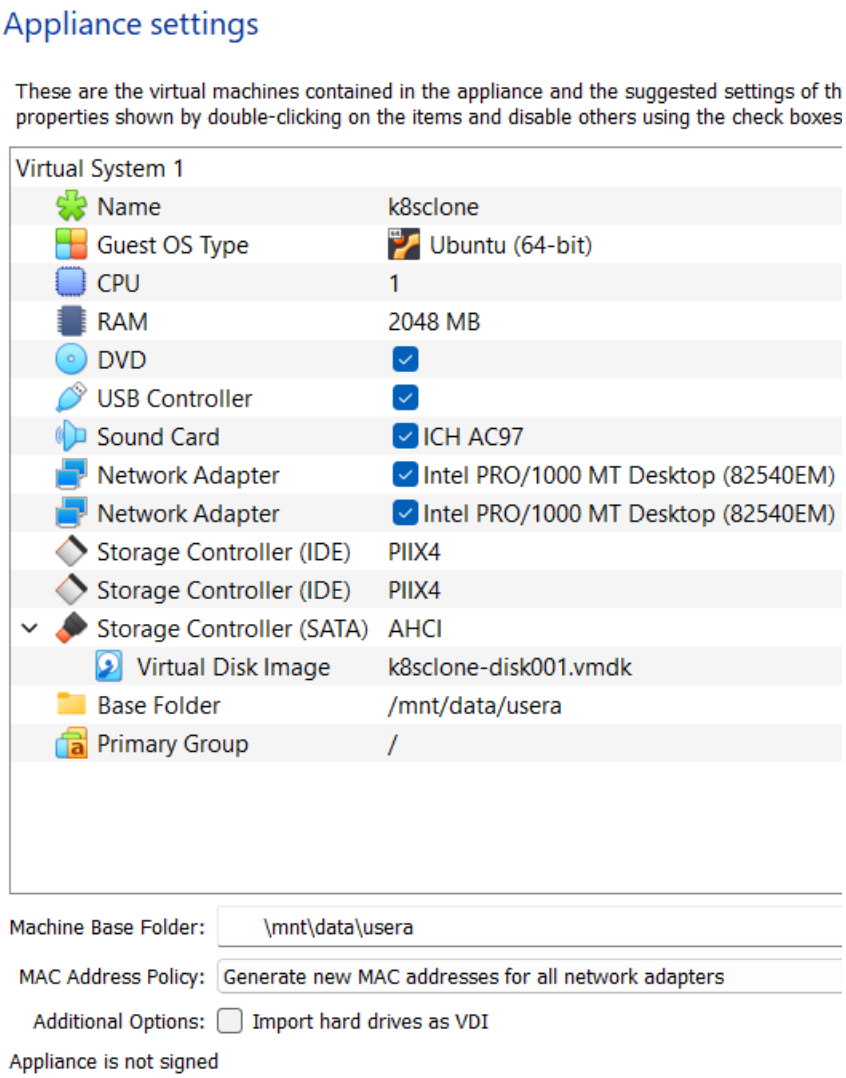
Open Oracle Virtual Box
File → Import appliance



Import k8sclone.ova



Modify base folder = **/mnt/data/usera**
Change MAC Address Policy **Generate new MAC Address for all network address**
Uncheck Additional Options: **Import hard drivers as VDI**



Repeat operation for k8snode1 and k8snode2
Power on 3 machines.

15.3 Kubernetes Cluster Initialization

Go to k8smaster
Initialize kubernetes master k8smaster **as root** **and** **single line**

```
root@k8smaster:~# kubeadm init --apiserver-advertise-address=192.168.59.101  --pod-network-cidr=172.16.0.0/16
[...]
To start using your cluster, you need to run the following as a regular user:

  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 192.168.59.101:6443 --token bnlndz.vuf1fxelmwdogjf9 \
    --discovery-token-ca-cert-hash
sha256:a4287b41fa72067ce0f9028648e8283853836949102699a546dbbccdefc8fcfd
root@k8smaster:~# logout
```

Switch to standard user (example : usera)

```
usera@k8smaster:~$ mkdir -p $HOME/.kube
usera@k8smaster:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
usera@k8smaster:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
usera@k8smaster:~$ kubectl get nodes
NAME        STATUS     ROLES    AGE   VERSION
k8smaster   NotReady   control-plane,master   20s   v1.23.1
```

Container Network Interface (CNI) installation
Source → https://docs.projectcalico.org/getting-started/kubernetes/quickstart

```
usera@k8smaster:~$ kubectl create -f
https://raw.githubusercontent.com/projectcalico/calico/v3.24.1/manifests/tigera-operator.yaml
namespace/tigera-operator created
customresourcedefinition.apiextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/caliconodestatuses.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworksets.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipreservations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org
created
customresourcedefinition.apiextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networksets.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/apiservers.operator.tigera.io created
customresourcedefinition.apiextensions.k8s.io/imagesets.operator.tigera.io created
customresourcedefinition.apiextensions.k8s.io/installations.operator.tigera.io created
customresourcedefinition.apiextensions.k8s.io/tigerastatuses.operator.tigera.io created
serviceaccount/tigera-operator created
clusterrole.rbac.authorization.k8s.io/tigera-operator created
clusterrolebinding.rbac.authorization.k8s.io/tigera-operator created
deployment.apps/tigera-operator created
usera@k8smaster:~$  wget
https://raw.githubusercontent.com/projectcalico/calico/v3.24.1/manifests/custom-resources.yaml
usera@k8smaster:~$ vim custom-resources.yaml <-- modify 192.168 by 172.16
  calicoNetwork:
    # Note: The ipPools section cannot be modified post-install.
    ipPools:
    - blockSize: 26
      cidr: 172.16.0.0/16
      encapsulation: VXLANCrossSubnet
      natOutgoing: Enabled
```

```
    nodeSelector: all()
usera@k8smaster:~$ kubectl create -f custom-resources.yaml
installation.operator.tigera.io/default created
apiserver.operator.tigera.io/default created
```

→ master must be ready

```
usera@k8smaster:~$ kubectl get po -A
NAMESPACE        NAME                                       READY   STATUS    RESTARTS   AGE
calico-apiserver   calico-apiserver-66c76fb49-cdshf         1/1     Running   0          58s
calico-apiserver   calico-apiserver-66c76fb49-vgj4x         1/1     Running   0          58s
calico-system      calico-kube-controllers-588575d68-m8j8c  1/1     Running   0          2m10s
calico-system      calico-node-f4kh2                        1/1     Running   0          2m10s
calico-system      calico-typha-5d4c4f958f-859mc            1/1     Running   0          2m10s
kube-system        coredns-78fcd69978-pd57z                 1/1     Running   0          16m
kube-system        coredns-78fcd69978-t8n2r                 1/1     Running   0          16m
kube-system        etcd-k8smaster111                        1/1     Running   0          16m
kube-system        kube-apiserver-k8smaster111              1/1     Running   0          16m
kube-system        kube-controller-manager-k8smaster111     1/1     Running   0          16m
kube-system        kube-proxy-sbp5f                         1/1     Running   0          16m
kube-system        kube-scheduler-k8smaster111              1/1     Running   0          16m
tigera-operator    tigera-operator-b78466769-56qnw          1/1     Running   0          3m40s

usera@k8smaster:~$ kubectl get nodes
NAME          STATUS   ROLES                  AGE   VERSION
k8smaster111   Ready    control-plane,master   17m   v1.22.4
```

15.4 Grow your cluster

Join Kubernetes worker nodes on k8snode1 and k8snode2 **as root**

Type our **own** token seen during kubeadm init

If you forget to write down the token, you can retrieve it by the command running on master

```
usera@k8smaster:~$ kubeadm token create --print-join-command
```

kubeadm join + --apiserver-advertise-address 192.168.59.102 --node-name k8snode1

See the following example

```
root@k8snode1:~# kubeadm join 192.168.59.101:6443 --token bnlndz.vuf1fxelmwdogjf9 --discovery-token-ca-cert-hash sha256:a4287b41fa72067ce0f9028648e8283853836949102699a546dbbccdefc8fcfd --apiserver-advertise-address 192.168.59.102 --node-name k8snode1
```

```
W1117 11:36:41.157886    6106 join.go:377] [preflight] WARNING: --control-plane is also required
when passing control-plane related flags such as [certificate-key, apiserver-advertise-address, apiserver-
bind-port]
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -
oyaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
root@k8snode2:~# kubeadm join 192.168.59.101:6443 --token bnlndz.vuf1fxelmwdogjf9 --
discovery-token-ca-cert-hash
sha256:a4287b41fa72067ce0f9028648e8283853836949102699a546dbbccdefc8fcfd --apiserver-
advertise-address 192.168.59.103 --node-name k8snode2
```

15.5 Access Kubernetes cluster from workstation

Config kubeconfig into workstation

```
[usera@lx-1-3 kubernetes-training]$ scp -p usera@k8smaster:~/.kube/config ~/config-kubeadm
[usera@lx-1-3 kubernetes-training]$ export KUBECONFIG=~/config-kubeadm
[usera@lx-1-3 kubernetes-training]$ kubectl get nodes
NAME        STATUS   ROLES                  AGE     VERSION
k8smaster   Ready    control-plane,master   34h     v1.23.1
k8snode1    Ready    <none>                 4h1m    v1.23.1
k8snode2    Ready    <none>                 3h59m   v1.23.1
```

16. LAB : Example Voting App

Source → https://github.com/dockersamples/example-voting-app

16.1 Deploy it

Create namespace vote

```
[usera@lx-1-3 ~]$ kubectl create namespace vote
namespace/vote created
```

Go to K8specifications directory and apply

```
[usera@lx-1-3 ~]$ cd example-voting-app/k8s-specifications/
[usera@lx-1-3 k8s-specifications]$ kubectl create -f .
deployment.apps/db created
service/db created
deployment.apps/redis created
service/redis created
deployment.apps/result created
service/result created
configmap/vote created
deployment.apps/vote created
ingress.networking.k8s.io/vote-ingress created
limitrange/vote-limits created
resourcequota/vote-quota created
service/vote created
deployment.apps/worker created
```

Where is my application ?

```
[usera@lx-1-3 k8s-specifications]$ kubectl get po
NAME                    READY   STATUS    RESTARTS   AGE
echo-server-6799c4cf46-c7xfp   1/1     Running   0          154m
[usera@lx-1-3 k8s-specifications]$ kubectl -n vote get po
NAME                    READY   STATUS              RESTARTS   AGE
db-57c4fd6875-txf6b     1/1     Running             0          2m16s
redis-5cff845b56-bhqts  1/1     Running             0          2m16s
result-59977485df-khbpr 1/1     Running             0          2m16s
vote-6c79f79647-msc7v   1/1     Running             0          2m16s
worker-64cb5879d9-ftfg9 0/1     ContainerCreating   0          2m16s
```

I can now change namespace to vote

```
[usera@lx-1-3 k8s-specifications]$ kubens vote
Context "kubernetes-admin@kubernetes" modified.
Active namespace is "vote".
```

Check deployment state

```
[usera@lx-1-3 k8s-specifications]$ kubectl -n vote get po,svc,ing -o wide
NAME                          READY   STATUS    RESTARTS   AGE    IP              NODE
NOMINATED NODE   READINESS GATES
pod/db-786c4bb6d6-vmbhq       1/1     Running   0          5m4s   172.16.249.1    k8snode1   <none>
<none>
pod/redis-67db9bd79b-dxjm8    1/1     Running   0          5m4s   172.16.249.3    k8snode1   <none>
<none>
pod/result-86d8966d87-j6cwj   1/1     Running   0          5m4s   172.16.185.194  k8snode2   <none>
<none>
pod/vote-676f78d64-m67b7      1/1     Running   0          5m4s   172.16.185.195  k8snode2   <none>
<none>
pod/worker-7cbf9df499-xbfhz   1/1     Running   0          5m4s   172.16.249.2    k8snode1   <none>
<none>

NAME           TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE    SELECTOR
service/db     ClusterIP   10.99.45.67     <none>        5432/TCP         5m4s   app=db
service/redis  ClusterIP   10.108.223.215  <none>        6379/TCP         5m4s   app=redis
service/result NodePort    10.107.203.180  <none>        5001:31001/TCP   5m4s   app=result
service/vote   NodePort    10.97.144.99    <none>        5000:31000/TCP   5m4s   app=vote

NAME                                CLASS    HOSTS           ADDRESS   PORTS     AGE
ingress.extensions/vote-ingress     <none>   vote.local.com            80, 443   5m4s
```

Retrieve IP master and worker nodes

```
[usera@lx-1-3 k8s-specifications]$ kubectl get nodes -o wide
NAME       STATUS   ROLES    AGE    VERSION   INTERNAL-IP     EXTERNAL-IP   OS-IMAGE
KERNEL-VERSION       CONTAINER-RUNTIME
k8smaster  Ready    master   161m   v1.19.4   192.168.59.101  <none>        Ubuntu 18.04.5 LTS
4.15.0-124-generic   docker://19.3.11
k8snode1   Ready    <none>   25m    v1.19.4   192.168.59.102  <none>        Ubuntu 18.04.5 LTS
4.15.0-124-generic   docker://19.3.11
k8snode2   Ready    <none>   25m    v1.19.4   192.168.59.103  <none>        Ubuntu 18.04.5 LTS
4.15.0-124-generic   docker://19.3.11
```

Test application http://192.168.59.101:31000 and http://192.168.59.101:31001
or http://192.168.59.102:31000 and http://192.168.59.102:31001
or http://192.168.59.103:31000 and http://192.168.59.103:31001

Why does it work on any ip ?

17. LAB: NGNIX Ingress deployment using Helm

Source → Source → https://github.com/kubernetes/ingress-nginx

17.1 Helm installation

Get Helm binary

```
[usera@lx-1-3 ~]$ wget https://get.helm.sh/helm-v3.4.1-linux-amd64.tar.gz
[usera@lx-1-3 ~]$ tar -xvf helm-v3.4.1-linux-amd64.tar.gz
linux-amd64/
linux-amd64/LICENSE
linux-amd64/README.md
linux-amd64/helm
[usera@lx-1-3 ~]$ sudo cp -p linux-amd64/helm /bin/
[usera@lx-1-3 ~]$ helm version
version.BuildInfo{Version:"v3.4.1", GitCommit:"c4e74854886b2efe3321e185578e6db9be0a6e29",
GitTreeState:"clean", GoVersion:"go1.14.11"}
```

17.2 Ingress Installation

Helm repository installation

```
[usera@lx-1-3 ~]$ helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
"ingress-nginx" has been added to your repositories
[usera@lx-1-3 ~]$ helm repo update
```

Create ingress-controller namespace and deploy Ingress Controller (ic)

```
[usera@lx-1-3 ~]$ kubectl create ns ingress-controller
namespace/ingress-controller created
[usera@lx-1-3 ~]$ helm install ic ingress-nginx/ingress-nginx --set
controller.hostNetwork=true,controller.service.type="",controller.kind=DaemonSet --namespace
ingress-controller
NAME: ic
LAST DEPLOYED: Fri Nov 20 10:10:15 2020
NAMESPACE: ingress-controller
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
The ingress-nginx controller has been installed.
```

An example Ingress that makes use of the controller:

```
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  annotations:
    kubernetes.io/ingress.class: nginx
  name: example
  namespace: foo
spec:
  rules:
    - host: www.example.com
      http:
        paths:
          - backend:
              serviceName: exampleService
              servicePort: 80
            path: /
  # This section is only required if TLS is to be enabled for the Ingress
  tls:
      - hosts:
          - www.example.com
        secretName: example-tls
```

If TLS is enabled for the Ingress, a Secret containing the certificate and key must also be provided:

```
apiVersion: v1
kind: Secret
metadata:
  name: example-tls
  namespace: foo
data:
  tls.crt: <base64 encoded cert>
  tls.key: <base64 encoded key>
type: kubernetes.io/tls
```

Check deployment

[usera@lx-1-3 k8s-specifications]$ **kubectl -n ingress-controller get po -o wide**

```
NAME                                READY   STATUS    RESTARTS   AGE   IP              NODE
NOMINATED NODE   READINESS GATES
ic-ingress-nginx-controller-d7d4s   1/1     Running   0          17m   192.168.59.102   k8snode1   <none>
<none>
ic-ingress-nginx-controller-qb4f7   1/1     Running   0          17m   192.168.59.103   k8snode2   <none>
<none>
```

Question → why do we have 2 pods ?

17.3 Ingress Configuration

Vote Ingress Resource has been already deployed previous

```
ingress.networking.k8s.io/vote-ingress created
```

However, this configuration must be updated since kubernetes v1.22

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: vote-ingress
  namespace: vote
spec:
  ingressClassName: nginx
  rules:
  - host: vote.local.com
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: vote
            port:
              number: 5000
```

Delete Ingress Resources (IR) and re-create with above yaml file

```
[usera@lx-6-1 k8s-specifications]$ kubectl delete -f vote-ing.yaml
ingress.networking.k8s.io "vote-ingress" deleted
[usera@lx-6-1 k8s-specifications]$ kubectl apply -f vote-ing.yaml
ingress.networking.k8s.io/vote-ingress created
```

Edit /etc/hosts and add DNS entry as follows (depending on your env)

```
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

192.168.59.101 k8smaster k8smaster.lab.example.com
192.168.59.102 k8snode1 k8snode1.lab.example.com vote.local.com
192.168.59.103 k8snode2 k8snode2.lab.example.com vote.local.com
```

Check if http://vote.local.com is now linked to vote service

18. LAB: Metering

Source → https://github.com/kubernetes-sigs/metrics-server

18.1 Set metrics

Download components.yaml

```
[usera@lx-1-3 ~]$ wget https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml
```

Fix error due to https://github.com/kubernetes-sigs/metrics-server/issues/167

E1117 13:09:29.758612 1 server.go:132] unable to fully scrape metrics: [unable to fully scrape metrics from node k8snode2: unable to fetch metrics from node k8snode2: Get "https://192.168.59.103:10250/stats/summary?only_cpu_and_memory=true": x509: cannot validate certificate for 192.168.59.103 because it doesn't contain any IP SANs, unable to fully scrape metrics from node k8snode1: unable to fetch metrics from node k8snode1: Get "https://192.168.59.102:10250/stats/summary?only_cpu_and_memory=true": x509: cannot validate certificate for 192.168.59.102 because it doesn't contain any IP SANs, unable to fully scrape metrics from node k8smaster: unable to fetch metrics from node k8smaster: Get "https://192.168.59.101:10250/stats/summary?only_cpu_and_memory=true": x509: cannot validate certificate for 192.168.59.101 because it doesn't contain any IP SANs]

Edit **components.yaml** and add following line (in red bold)

```
    containers:
    - args:
      - --cert-dir=/tmp
      - --kubelet-insecure-tls
      - --secure-port=4443
```

Deploy it

```
[usera@lx-1-3 ~]$ kubectl apply -f components.yaml
clusterrole.rbac.authorization.k8s.io/system:aggregated-metrics-reader created
clusterrolebinding.rbac.authorization.k8s.io/metrics-server:system:auth-delegator created
rolebinding.rbac.authorization.k8s.io/metrics-server-auth-reader created
apiservice.apiregistration.k8s.io/v1beta1.metrics.k8s.io created
serviceaccount/metrics-server created
deployment.apps/metrics-server created
service/metrics-server created
clusterrole.rbac.authorization.k8s.io/system:metrics-server created
```

```
clusterrolebinding.rbac.authorization.k8s.io/system:metrics-server created
```

18.2 Test with TOP

Run Kubernetes top

```
[usera@lx-1-3 ~]$ kubectl top node
NAME       CPU(cores)   CPU%   MEMORY(bytes)   MEMORY%
minikube   1203m        60%    1541Mi          42%
[usera@lx-1-3 ~]$ kubectl top pod --all-namespaces
NAMESPACE       NAME                                        CPU(cores)   MEMORY(bytes)
ingress-nginx   nginx-ingress-controller-777868f4c5-szw8t   4m           97Mi
kube-system     coredns-6955765f44-bdkqr                    2m           6Mi
kube-system     coredns-6955765f44-qnbld                    2m           6Mi
kube-system     etcd-minikube                               12m          31Mi
kube-system     kube-apiserver-minikube                     22m          247Mi
kube-system     kube-controller-manager-minikube            7m           34Mi
kube-system     kube-proxy-rjs7f                            1m           12Mi
kube-system     kube-scheduler-minikube                     2m           10Mi
kube-system     kubernetes-dashboard-7c54d59f66-d7wsl       1m           10Mi
kube-system     metrics-server-77c968df9d-td4dm             1m           11Mi
kube-system     storage-provisioner                         1m           13Mi
vote            db-8c6dbd86f-6v46n                          190m         35Mi
vote            redis-97698dc95-7d7rz                       125m         3Mi
vote            result-75c97ddf5d-d7tlj                     1m           35Mi
vote            vote-74bc6dff6-w7hl8                        1m           60Mi
vote            worker-5764d777cd-2lpwm                     884m         33Mi
```

19. LAB: HPA

Source →
https://schoolofdevops.github.io/ultimate-kubernetes-bootcamp/10_kubernetes_autoscaling/#kubernetes-horizonntal-pod-autoscaling

19.1 Voting App preparation

Go to 'vote' namespace and check if voting app is still alive since chapter 3

```
[usera@lx-1-3 k8s-specifications]$ kubens vote
Context "kubernetes-admin@kubernetes" modified.
Active namespace is "vote".
[usera@lx-1-3 k8s-specifications]$ kubectl get po,svc,cm
NAME                          READY   STATUS    RESTARTS   AGE
pod/db-57c4fd6875-txf6b       1/1     Running   0          30h
pod/redis-5cff845b56-bhqts    1/1     Running   0          30h
pod/result-59977485df-khbpr   1/1     Running   0          30h
pod/vote-6c79f79647-msc7v     1/1     Running   0          30h
pod/worker-64cb5879d9-ftfg9   1/1     Running   0          30h
NAME            TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
service/db      ClusterIP   10.101.249.96  <none>        5432/TCP         30h
service/redis   ClusterIP   10.100.99.238  <none>        6379/TCP         30h
service/result  NodePort    10.98.60.250   <none>        5001:31001/TCP   30h
service/vote    NodePort    10.107.4.77    <none>        5000:31000/TCP   30h
NAME            DATA   AGE
configmap/vote  2      4m38s
```

Check if http://vote.local.com is still works

19.2 Check resources limit is set

Check you have already had following block into vote-deployment.yaml onto k8s-specifications directory

```
      resources:
        limits:
          cpu: "200m"
          memory: "250Mi"
```

```
        requests:
          cpu: "100m"
          memory: "50Mi"
```

```
k8s-specifications > ! vote-deployment.yaml > {} spec >
31                path: /
32                port: 80
33              initialDelaySeconds: 5
34              periodSeconds: 3
35            envFrom:
36              - configMapRef:
37                  name: vote
38            resources:
39              limits:
40                cpu: "200m"
41                memory: "250Mi"
42              requests:
43                cpu: "100m"
44                memory: "50Mi"
45
```

Apply it

```
[usera@lx-1-3 k8s-specifications]$ kubectl apply -f vote-deployment.yaml
deployment.apps/vote configured
```

Vote pod must restart and check if vote is still running on http://vote.local.com

19.3 Add Horizontal Pod Autoscaler (HPA)

Add new file **vote-hpa.yaml** onto k8s-specifications

```
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: vote
spec:
  minReplicas: 1
  maxReplicas: 5
  targetCPUUtilizationPercentage: 50
  scaleTargetRef:
    apiVersion: apps/v1
```

```
kind: Deployment
name: vote
```

k8s-specifications > ! vote-hpa.yaml > {} spec > # targetCPUUtilizationPercentage

```yaml
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: vote
spec:
  minReplicas: 1
  maxReplicas: 15
  targetCPUUtilizationPercentage: 50
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: vote
```

Apply it

```
[usera@lx-1-3 k8s-specifications]$ kubectl apply -f vote-hpa.yaml
horizontalpodautoscaler.autoscaling/vote created
```

Check hpa status

```
[usera@lx-1-3 k8s-specifications]$ kubectl get hpa
NAME   REFERENCE       TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
vote   Deployment/vote   2%/50%    1         5         1          50s
```

Note – also possible by command line

```
[usera@lx-1-3 k8s-specifications]$ kubectl autoscale deployment vote --cpu-percent=50 --min=1 --max=5 -n vote
```

19.4 Heavy workload vote

Add new file **vote-loadtest.yaml** onto k8s-specifications

```
apiVersion: batch/v1
kind: Job
metadata:
```

```
  name: loadtest
spec:
  template:
    spec:
      containers:
      - name: siege
        image: schoolofdevops/loadtest:v1
        command: ["siege",  "--concurrent=5", "--benchmark", "--time=10m", "http://vote:5000"]
      restartPolicy: Never
  backoffLimit: 4
```

```
k8s-specifications > ! vote-loadtest.yaml > {} spec > {} template > {} spec > [ ] containers > {} 0 > [ ] command > abc 3
1   apiVersion: batch/v1
2   kind: Job
3   metadata:
4     name: loadtest
5   spec:
6     template:
7       spec:
8         containers:
9         - name: siege
10          image: schoolofdevops/loadtest:v1
11          command: ["siege",  "--concurrent=10", "--benchmark", "--time=10m", "http://vote:5000"]
12        restartPolicy: Never
13    backoffLimit: 4
```

Question → why do you enter uhttp://vote:5000 as value in loadtest job ?

Apply load test job

```
[usera@lx-1-3 k8s-specifications]$ kubectl apply -f vote-loadtest.yaml
job.batch/loadtest created
```

And check the number of PODs for 10 minutes

```
[usera@lx-1-3 k8s-specifications]$ watch kubectl get po
NAME                    READY   STATUS    RESTARTS   AGE
db-786c4bb6d6-vmbhq     1/1     Running   1          2d20h
loadtest-r4mhj          1/1     Running   0          2m36s
redis-67db9bd79b-dxjm8  1/1     Running   1          2d20h
result-86d8966d87-j6cwj 1/1     Running   2          2d20h
vote-676f78d64-bvqvq    1/1     Running   0          35s
vote-676f78d64-hg875    1/1     Running   0          35s
vote-676f78d64-m67b7    1/1     Running   1          2d20h
```

```
vote-676f78d64-nxzss      1/1     Running   0          20s
vote-676f78d64-srsd5      0/1     Running   0          35s
worker-7cbf9df499-xbfhz   1/1     Running   1          2d20h
```

What does it happen ?

Cleanup

```
[usera@lx-1-3 k8s-specifications]$ kubectl delete hpa vote
horizontalpodautoscaler.autoscaling "vote" deleted
```

20. LAB: RBAC Human User

Source → https://docs.bitnami.com/tutorials/configure-rbac-in-your-kubernetes-cluster/

20.1 Authentication

Create key for student and generate csr

```
[usera@lx-1-3 k8s-specifications]$ cd
[usera@lx-1-3 ~]$ mkdir LAB20-RBAC
[usera@lx-1-3 ~]$ cd LAB20-RBAC
[usera@lx-1-3 LAB20-RBAC]$ openssl genrsa -out student.key 2048
Generating RSA private key, 2048 bit long modulus
.......................+++
......+++
e is 65537 (0x10001)
[usera@lx-1-3 LAB20-RBAC]$ openssl req -new -key student.key -out student.csr -subj
"/CN=student/O=education"
```

<<<<<<<<<<<<< it depends on Kubernetes education cluster >>>>>>>>>>>>

Retrieve certificate and key CA cluster

For example, find below instructions depends on Kubernetes cluster.

```
# go to master and copy kube-ca certificate + key
root@k8smaster:~# cp -p /etc/kubernetes/pki/ca.* /tmp/
root@k8smaster:~# chmod 644 /tmp/ca.*
[usera@lx-1-3 LAB20-RBAC]$ scp -p usera@k8smaster:/tmp/ca* .
usera@k8smaster's password:
ca.crt                                    100% 1066     1.9MB/s   00:00
ca.key                                    100% 1679     3.0MB/s   00:00
```

<<<<<<<<<<<<< it depends on Kubernetes education cluster >>>>>>>>>>>>

Sign previous certificate with it

```
[usera@lx-1-3 LAB20-RBAC]$ openssl x509 -req -in student.csr -CA ca.crt -CAkey ca.key -
CAcreateserial -out student.crt -days 500
Signature ok
subject=/CN=student/O=education
Getting CA Private Key
```

Add credentials and context – cluster name = kubernetes

```
[usera@lx-1-3 LAB20-RBAC]$ export KUBECONFIG=~/config-kubeadm
```

```
[usera@lx-1-3 LAB20-RBAC]$ kubectl config set-credentials student
--client-certificate=/home/usera/LAB20-RBAC/student.crt --client-key=/home/usera/LAB20-
RBAC/student.key
User "student" set.
[droinpy@oc5004167418 LAB20-RBAC]$ kubectl config set-context student-context --
cluster=kubernetes --namespace=vote --user=student
Context "student-context" created.
[droinpy@oc5004167418 LAB20-RBAC]$ grep student ~/config-kubeadm
    user: student
  name: student-context
- name: student
    client-certificate: /home/usera/LAB20-RBAC/student.crt
    client-key: /home/usera/LAB20-RBAC/student.key
```

List pods

```
[droinpy@oc5004167418 LAB20-RBAC]$ kubectl --context=student-context get pods
Error from server (Forbidden): pods is forbidden: User "student" cannot list resource "pods" in API
group "" in the namespace "vote"
```

20.2 Authorization

User

Create role vim **role-student.yaml**

```
kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  namespace: vote
  name: role-student
rules:
- apiGroups: ["", "extensions", "apps"]
  resources: ["deployments", "replicasets", "pods"]
  verbs: ["get", "list", "watch", "create", "update", "patch", "delete"] # You can also use ["*"]
```

Create rolebinding **vim binding-student.yaml**

```
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: binding-student
```

```
  namespace: vote
subjects:
- kind: User
  name: student
  apiGroup: ""
roleRef:
  kind: Role
  name: role-student
  apiGroup: ""
```

Note:

```
Warning: rbac.authorization.k8s.io/v1beta1 Role is deprecated in v1.17+, unavailable in v1.22+; use rbac.authorization.k8s.io/v1 Role
Warning: rbac.authorization.k8s.io/v1beta1 RoleBinding is deprecated in v1.17+, unavailable in v1.22+; use rbac.authorization.k8s.io/v1 RoleBinding
```

Create role role-student.yaml and binding bin

```
[usera@lx-1-3 LAB20-RBAC]$ kubectl apply -f role-student.yaml
role.rbac.authorization.k8s.io/role-student created
[usera@lx-1-3 LAB20-RBAC]$ kubectl apply -f binding-student.yaml

rolebinding.rbac.authorization.k8s.io/binding-student created
[usera@lx-1-3 LAB20-RBAC]$ kubectl --context=student-context get pods
NAME                      READY   STATUS      RESTARTS   AGE
db-786c4bb6d6-vmbhq       1/1     Running     3          5d3h
loadtest-r4mhj            0/1     Completed   0          2d7h
redis-67db9bd79b-dxjm8    1/1     Running     3          5d3h
result-86d8966d87-j6cwj   1/1     Running     4          5d3h
vote-676f78d64-hg875      1/1     Running     2          2d7h
worker-7cbf9df499-xbfhz   1/1     Running     4          5d3h
```

Group/Team

Check that you are not able to list pods in ‘default’ namespace

```
[usera@lx-1-3 LAB20-RBAC]$ kubectl --context=student-context get pods -n default
Error from server (Forbidden): pods is forbidden: User "student" cannot list resource "pods" in API group "" in the namespace "default"
```

Create vim **role-read-default.yaml**

```
kind: Role
```

```
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  namespace: default
  name: role-read-default
rules:
- apiGroups: ["", "extensions", "apps"]
  resources: ["deployments", "replicasets", "pods"]
  verbs: ["get", "list", "watch" ] # You can also use ["*"]
```

Create rolebinding **vim binding-education.yaml**

```
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: binding-education
  namespace: default
subjects:
- kind: Group
  name: education
  apiGroup: ""
roleRef:
  kind: Role
  name: role-read-default
  apiGroup: ""
```

Create role-read-default role and bind it

```
[usera@lx-1-3 LAB20-RBAC]$ kubectl create -f role-read-default.yaml
role.rbac.authorization.k8s.io/role-read-default created
[usera@lx-1-3 LAB20-RBAC]$ kubectl create -f binding-education.yaml
rolebinding.rbac.authorization.k8s.io/binding-education created
```

Check now with your user student belonging to education team, you are able to list pods in ‘default’ namespace

```
[usera@lx-1-3 LAB20-RBAC]$ kubectl --context=student-context get pods -n default
No resources found in default namespace.
```

21. LAB: RBAC Service Account

Back to your Kubernetes dashboard application

21.1 Kubernetes dashboard

Switch to imagePullPolicy: Always to imagePullPolicy: IfNotPresent
Download recommended.yaml

```
[usera@lx-1-3 ~]$ wget https://raw.githubusercontent.com/kubernetes/dashboard/v2.6.1/aio/deploy/recommended.yaml
[usera@lx-1-3 ~]$ sed -i 's?imagePullPolicy: Always?imagePullPolicy: IfNotPresent?g' recommended.yaml
```

Deploy Kubernetes Dashboard v2.6.1

```
[usera@lx-1-3 ~]$ kubectl apply -f recommended.yaml
namespace/kubernetes-dashboard created
serviceaccount/kubernetes-dashboard created
service/kubernetes-dashboard created
secret/kubernetes-dashboard-certs created
secret/kubernetes-dashboard-csrf created
secret/kubernetes-dashboard-key-holder created
configmap/kubernetes-dashboard-settings created
role.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrole.rbac.authorization.k8s.io/kubernetes-dashboard created
rolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
deployment.apps/kubernetes-dashboard created
service/dashboard-metrics-scraper created
deployment.apps/dashboard-metrics-scraper created
```

Creating a Service Account
We are creating Service Account with name admin-user in namespace kubernetes-dashboard first.

```
[usera@lx-1-3 ~]$ kubectl create sa admin-user  -n kubernetes-dashboard
serviceaccount/admin-user created
```

Creating a ClusterRoleBinding

```
[usera@lx-1-3 ~]$ cat <<EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
```

```
metadata:
  name: admin-user
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin-user
  namespace: kubernetes-dashboard
EOF
clusterrolebinding.rbac.authorization.k8s.io/admin-user created
```

Getting a Bearer Token

Now we need to find token we can use to log in. Execute following command:

```
[usera@lx-1-3 ~]$ kubectl -n kubernetes-dashboard create token admin-user
eyJhb[...}16XjD-qKxKg
```

Copy/backup token value

Launch Kubernetes proxy

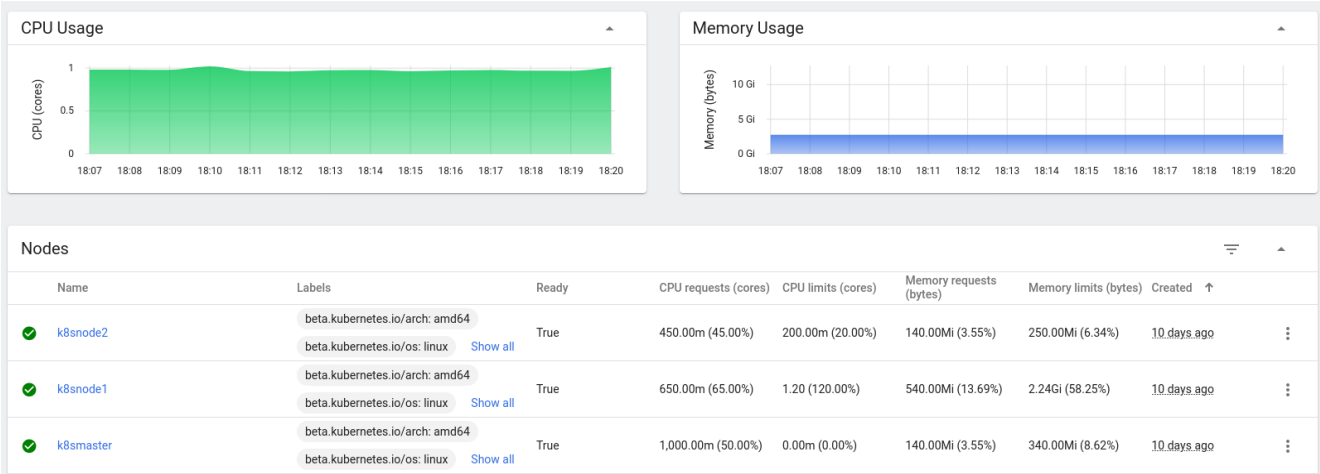
It creates a proxy or application-level gateway between localhost and the Kubernetes API server. It also allows to serve static content over specified HTTP path.

The command kubectl proxy runs kubectl in a mode where it acts as a reverse proxy. It handles locating the apiserver and authenticating.

```
[usera@lx-1-3 ~]$ kubectl proxy
Starting to serve on 127.0.0.1:8001
```

Kubectl proxy will make Dashboard available at http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/.

and this shall open up page as shown below if you click on Cluster → Nodes (for example)



Navigate ...

Ctrl+C to stop kubectl proxy

22. LAB: Advanced scheduling

Source → https://schoolofdevops.github.io/ultimate-kubernetes-bootcamp/advanced_pod_scheduling/#advanced-pod-scheduling

22.1 Preparation

Scale deployment vote to 2

```
[usera@lx-1-3 k8s-specifications]$ kubectl scale --replicas=2 deploy vote
deployment.apps/vote scaled
[[usera@lx-1-3 k8s-specifications]$ kubectl get po -o wide|grep vote
vote-676f78d64-hg875      1/1     Running     2          2d7h   172.16.249.20    k8snode1   <none>
<none>
vote-676f78d64-ssw7z      1/1     Running     0          45s    172.16.185.208   k8snode2   <none>
<none>
```

22.2 Nodeselector

Patch vote deployment with node selector

```
[usera@lx-1-3 k8s-specifications]$ kubectl label node k8snode2 zone=vote
node/k8snode2 labeled
[usera@lx-1-3 k8s-specifications]$ kubectl patch deploy vote -p '{"spec":{"template":{"spec":
{"nodeSelector":{"zone":"vote"}}}}}'
deployment.apps/vote patched
[usera@lx-1-3 k8s-specifications]$ kubectl get po -o wide|grep vote
vote-5fccdc9c7-5zvl2      1/1     Running     0          79s    172.16.185.209   k8snode2   <none>
<none>
vote-5fccdc9c7-tr2rd      1/1     Running     0          64s    172.16.185.210   k8snode2   <none>
<none>
```

22.3 Preparation

Add scale=1 and remove nodeSelector block

```
[usera@lx-1-3 k8s-specifications]$ kubectl scale --replicas=1 deploy vote
[usera@lx-1-3 k8s-specifications]$ kubectl edit deploy vote
      nodeSelector:
        zone: vote
```

```
[usera@lx-1-3 k8s-specifications]$ kubectl get po -o wide|grep vote
vote-7875577b6-g8672      1/1     Running       0          31s    172.16.249.24    k8snode2   <none>
<none>
```

22.4 Anti-affinity

Add anti-affinity block

```
[usera@lx-1-3 k8s-specifications]$ kubectl edit deploy vote
      affinity:
        podAntiAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
          - labelSelector:
              matchExpressions:
              - key: app
                operator: In
                values:
                - vote
            topologyKey: kubernetes.io/hostname
[usera@lx-1-3 k8s-specifications]$ kubectl get po -o wide|grep vote
vote-7875577b6-g8672      1/1     Running       0          31s    172.16.249.24    k8snode2   <none>
<none>
vote-869f4d496d-trdbn     0/1     Terminating   0          31s    172.16.185.215   k8snode2   <none>
<none>
[usera@lx-1-3 k8s-specifications]$ kubectl scale --replicas=2 deploy vote
deployment.apps/vote scaled
[usera@lx-1-3 k8s-specifications]$ kubectl get po -o wide|grep vote
vote-7875577b6-g8672      1/1     Running     0          46s    172.16.249.24    k8snode1   <none>
<none>
vote-7875577b6-qg4dx      0/1     Running     0          2s     172.16.185.216   k8snode2   <none>
<none>
[usera@lx-1-3 k8s-specifications]$ kubectl scale --replicas=3 deploy vote
deployment.apps/vote scaled
[usera@lx-1-3 k8s-specifications]$ kubectl get po -o wide|grep vote
vote-7875577b6-cpht8      0/1     Pending     0          8s     <none>           <none>     <none>
<none>
vote-7875577b6-g8672      1/1     Running     0          10m    172.16.249.24    k8snode1   <none>
<none>
```

```
vote-7875577b6-qzvfz     1/1     Running     0          11s    172.16.185.217   k8snode2   <none>
<none>
```

Why do we have a pending pod ?

Cleanup

```
[usera@lx-1-3 k8s-specifications]$ kubectl label node k8snode2 zone-
node/k8snode2 unlabeled
[usera@lx-1-3 k8s-specifications]$ kubectl get nodes --show-labels
NAME        STATUS   ROLES    AGE    VERSION   LABELS
k8smaster   Ready    master   5d7h   v1.19.4
beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/arch=amd64,kubernetes.io/
hostname=k8smaster,kubernetes.io/os=linux,node-role.kubernetes.io/master=
k8snode1    Ready    <none>   5d5h   v1.19.4
beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/arch=amd64,kubernetes.io/
hostname=k8snode1,kubernetes.io/os=linux
k8snode2    Ready    <none>   5d5h   v1.19.4
beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/arch=amd64,kubernetes.io/
hostname=k8snode2,kubernetes.io/os=linux
[usera@lx-1-3 k8s-specifications]$ kubectl scale --replicas=2 deploy vote
deployment.apps/vote scaled
[usera@lx-1-3 k8s-specifications]$ kubectl scale --replicas=1 deploy vote
deployment.apps/vote edited
[usera@lx-1-3 k8s-specifications]$ kubectl get po -o wide|grep vote
d64-ssw7z     1/1     Running     0          45s    172.16.185.208   k8snode2   <none>           <none>
```

23. LAB: Drain and maintenance

Source → https://schoolofdevops.github.io/ultimate-kubernetes-bootcamp/cluster-administration/#drain-a-node

23.1 Preparation

Scale vote deployment to 2

```
[usera@lx-1-3 ~]$ kubectl scale --replicas=2 deploy vote
deployment.apps/vote scaled
[usera@lx-1-3 ~]$ kubectl get po -o wide|grep vote
vote-7875577b6-2ld5j     1/1     Running     0          41s    172.16.249.25    k8snode1   <none>
<none>
vote-7875577b6-kcwqp     1/1     Running     0          29s    172.16.185.219   k8snode2   <none>
<none>
[usera@lx-1-3 ~]$ kubectl get nodes
NAME        STATUS   ROLES    AGE    VERSION
k8smaster   Ready    master   5d7h   v1.19.4
k8snode1    Ready    <none>   5d5h   v1.19.4
k8snode2    Ready    <none>   5d5h   v1.19.4
```

23.2 Cordon

Cordon and drain k8snode2

```
[usera@lx-1-3 ~]$ kubectl cordon k8snode2
node/k8snode2 cordoned
[usera@lx-1-3 ~]$ kubectl get nodes
NAME        STATUS                     ROLES    AGE    VERSION
k8smaster   Ready                      master   5d7h   v1.19.4
k8snode1    Ready                      <none>   5d5h   v1.19.4
k8snode2    Ready,SchedulingDisabled   <none>   5d5h   v1.19.4
[usera@lx-1-3 ~]$ kubectl get po -o wide|grep vote
vote-7875577b6-2ld5j     1/1     Running     0          59s    172.16.249.25    k8snode1   <none>
<none>
vote-7875577b6-kcwqp     1/1     Running     0          47s    172.16.185.219   k8snode2   <none>
<none>
[usera@lx-1-3 ~]$ kubectl drain k8snode2
node/k8snode2 already cordoned
error: unable to drain node "k8snode2", aborting command...
```

```
There are pending nodes to be drained:
 k8snode2
cannot delete DaemonSet-managed Pods (use --ignore-daemonsets to ignore): ingress-controller/ic-ingress-nginx-controller-qb4f7, kube-system/calico-node-lms8t, kube-system/kube-proxy-wfpx7
cannot delete Pods with local storage (use --delete-local-data to override): kube-system/metrics-server-7bf66cc664-s6sm9
[usera@lx-1-3 ~]$ kubectl drain k8snode2 --ignore-daemonsets --delete-local-data
node/k8snode2 already cordoned
WARNING: ignoring DaemonSet-managed Pods: ingress-controller/ic-ingress-nginx-controller-qb4f7, kube-system/calico-node-lms8t, kube-system/kube-proxy-wfpx7
evicting pod vote/vote-7875577b6-kcwqp
evicting pod kube-system/metrics-server-7bf66cc664-s6sm9
evicting pod vote/result-86d8966d87-j6cwj
pod/metrics-server-7bf66cc664-s6sm9 evicted
pod/vote-7875577b6-kcwqp evicted
pod/result-86d8966d87-j6cwj evicted
node/k8snode2 evicted
[usera@lx-1-3 ~]$ kubectl get po -o wide|grep vote
vote-7875577b6-2ld5j      1/1     Running            0          2m12s   172.16.249.25   k8snode1   <none>
<none>
vote-7875577b6-wc6px      0/1     Pending            0          40s     <none>          <none>     <none>
<none>
[usera@lx-1-3 ~]$ kubectl get nodes
NAME        STATUS                     ROLES    AGE    VERSION
k8smaster   Ready                      master   5d7h   v1.19.4
k8snode1    Ready                      <none>   5d5h   v1.19.4
k8snode2    Ready,SchedulingDisabled   <none>   5d5h   v1.19.4
[usera@lx-1-3 ~]$ kubectl get po --all-namespaces -o wide|grep node2
ingress-controller   ic-ingress-nginx-controller-qb4f7        1/1     Running        3          2d9h
192.168.59.103   k8snode2    <none>          <none>
kube-system          calico-node-lms8t                        1/1     Running        1          2d4h    192.168.59.103
k8snode2    <none>          <none>
kube-system          kube-proxy-wfpx7                         1/1     Running        3          5d5h
192.168.59.103   k8snode2    <none>          <none>
```

23.3 Clean-up

Un-cordon

```
[usera@lx-1-3 ~]$ kubectl uncordon k8snode2
node/k8snode2 uncordoned
[usera@lx-1-3 ~]$ kubectl get nodes
NAME        STATUS   ROLES    AGE    VERSION
k8smaster   Ready    master   5d7h   v1.19.4
k8snode1    Ready    <none>   5d5h   v1.19.4
k8snode2    Ready    <none>   5d5h   v1.19.4
[[usera@lx-1-3 ~]$ kubectl get po --all-namespaces -o wide|grep node2
ingress-controller   ic-ingress-nginx-controller-qb4f7        1/1     Running     3          2d9h
192.168.59.103   k8snode2    <none>          <none>
kube-system          calico-node-lms8t                        1/1     Running     1          2d4h
192.168.59.103   k8snode2    <none>          <none>
kube-system          kube-proxy-wfpx7                         1/1     Running     3          5d5h
192.168.59.103   k8snode2    <none>          <none>
vote                 vote-7875577b6-wc6px                     0/1     Running     0          2m24s
172.16.185.220   k8snode2    <none>          <none>
```

Question → Why do we have 4 pods running on node2 only ?

THE END

24. APPENDIX: OS, Docker and kubeadm manual installation

24.1 OS Ubuntu installation

Boot https://ubuntu.com/download/server and follow Ubuntu installation instructions
Master hardware configuration 2CPU and 4GB – disk 40GB no swap – **Ubuntu 20.04.5** LTS
Node hardware configuration 1CPU and 2GB – disk 40GB no swap – **Ubuntu 20.04.5** LTS

```
$ sudo mkdir /mnt/data/usera
$ sudo chown usera:usera /mnt/data/usera
```

VM name = k8sclone
If Oracle Virtual Box, you create 2 networks :

- NAT for external connection
- Host-only for internal connection

Ubuntu installation
Update to the new installer
Keyboard french
Default installation and select OpenSSH server installation
usera/usera and root/root

VM configuration

Copy **calico_v3_24_1.tar** archive provided by teacher on k8sclone
Wait for him if this archive does not exist on your education machine
(see appendix Docker Hub download limit)

```
[usera@lx-6-1 ~]$ cd /mnt/data/usera
[usera@lx-6-1 ~]$ scp -p calico/calico_v3.24.1.tar 192.168.59.101:/home/usera/
```

Copy **docker-io.tar** archive provided by teacher on k8sclone
Wait for him if this archive does not exist on your education machine
(see appendix Docker Hub download limit)

```
[usera@lx-6-1 ~]$  scp -p docker.io.tar  192.168.59.101:/home/usera/
```

Copy **install_docker_kubeadm.sh** shell script provided by teacher on k8sclone
(see appendix for script details)

```
[usera@lx-6-1 ~]$ scp -p install_docker_kubeadm.sh 192.168.59.101:/home/usera/
```

Go to k8sclone

```
[usera@lx-6-1 ~]$ ssh 192.168.59.101
usera@192.168.59.101's password:
```

Add executable **install_docker_kubeadm.sh** shell script
(see appendix for script details)

```
[usera@k8snode ~]$ chmod +x install_docker_kubeadm.sh
```

Switch to root

```
sudo su -
```

Update **/etc/hosts**

```
#
192.168.59.101 k8smaster k8smaster.lab.example.com
192.168.59.102 k8snode1 k8snode1.lab.example.com
192.168.59.103 k8snode2 k8snode2.lab.example.com
```

Disable swap into **/etc/fstab**

```
# /swap.img none swap sw 0 0
```

Run swapoff -a

```
swapoff -a
```

At least 5GB for / (and 2 CPU for master)

```
lvresize --size +15G /dev/ubuntu-vg/ubuntu-lv
resize2fs /dev/mapper/ubuntu--vg-ubuntu--lv
```

Check enp0s8=192.168.59.101

```
ip a
```

--> STOP here if it is not the case and call teacher

Set hostname

```
hostnamectl  set-hostname k8sclone
```

Update **/etc/cloud/cloud.cfg** with correct hostname

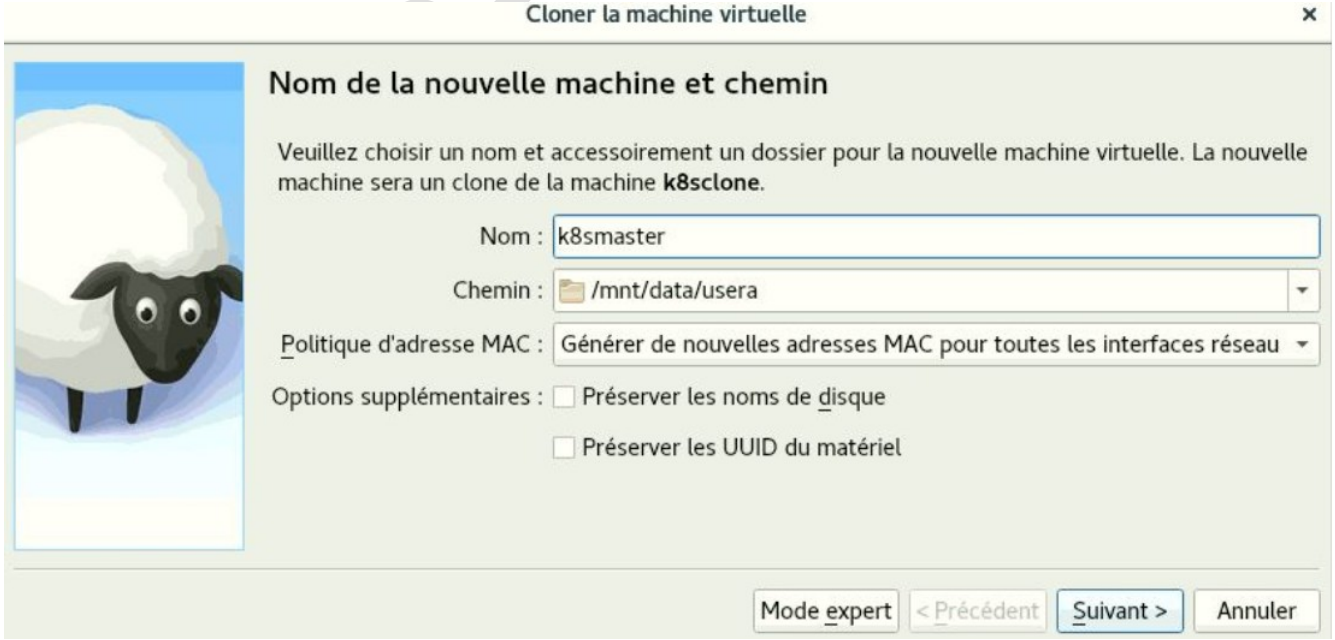
```
# This will cause the set+update hostname module to not operate (if true)
preserve_hostname: true
```

shutdown k8sclone for clone – **check twice that you are on k8sclone**

```
# shutdown now -h
```

--- Full Clone

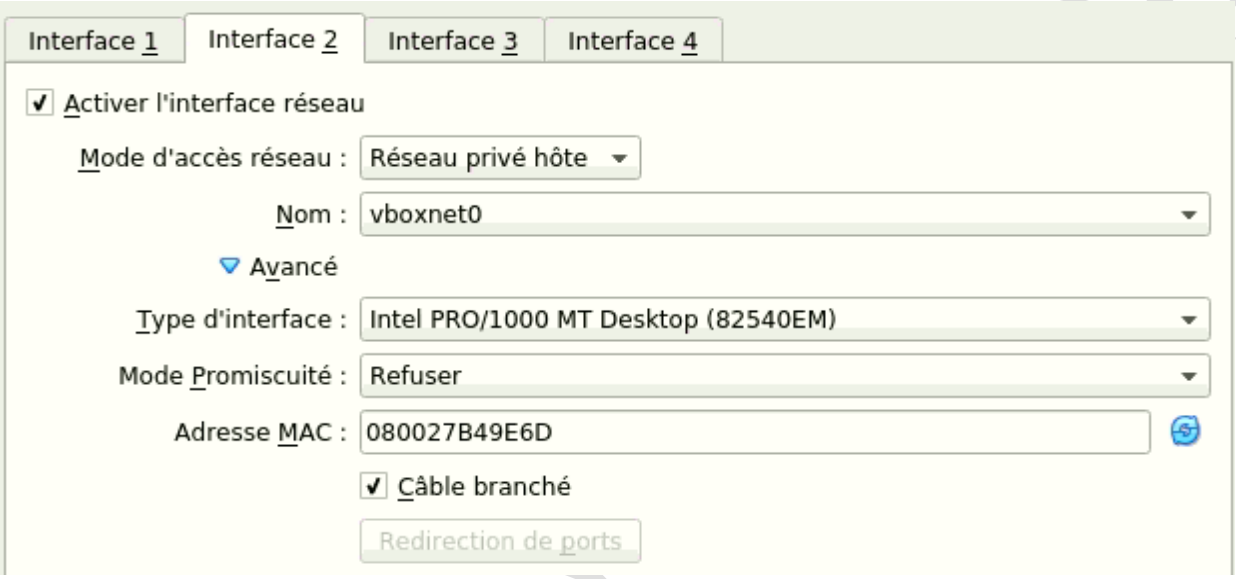
Back to lab desktop and we will clone them onto /mnt/data filesystems where 192GB are available



→ Select “**Generate new MAC addresses for all network adapters**”
→ Full clone

Clone into k8smaster k8snode1 and k8snode2
Keep k8sclone (for backup)

Check MAC addresses are different on enp0s8-interface 2 on k8smaster/k8snode1/k8snode2



Configure network on k8smaster, k8snode1 and k8snode2
k8smaster = 192.168.59.101/24
k8snode1 = 192.168.59.102/24
k8snode2 = 192.168.59.103/24

Warning:
Because k8smaster, k8snode1 and k8snode2 are cloned machine, all machines are same IP=192.168.59.101.

You power on k8smaster first (and k8snode1 and k8snode2 must be power off).
First login to k8smaster via 192.168.59.101
As root, update **/etc/netplan/00-installer-config.yaml** as follows:

```
network:
  ethernets:
    enp0s3:
      dhcp4: true
```

```
    enp0s8:
      addresses: [192.168.59.101/24]
      dhcp4: false
      dhcp6: false
  version: 2
```

Set hostname k8smaster

hostnamectl set-hostname k8smaster

Shutdown k8smaster

Then, you power on k8snode1 (and k8smaster and k8snode2 must be power off).
First login to k8snode1 via 192.168.59.101
As root, update **/etc/netplan/00-installer-config.yaml** - k8snode1

```
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      addresses: [192.168.59.102/24]
      dhcp4: false
      dhcp6: false
  version: 2
```

Set hostname k8snode1

hostnamectl set-hostname k8snode1

Shutdown k8snode1

Finally, you power on k8snode2 (and k8smaster and k8snode1 must be power off).
First login to k8snode2 via 192.168.59.101
As root, update **/etc/netplan/00-installer-config.yaml** - k8snode2

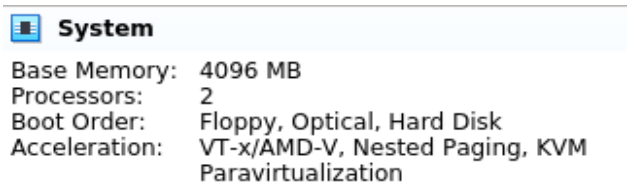
```
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      addresses: [192.168.59.103/24]
      dhcp4: false
      dhcp6: false
  version: 2
```

Set hostname k8snode2

```
hostnamectl  set-hostname k8snode2
```

Shutdown k8snode2

Increase the CPU from 1 to 2 for k8smaster



Power on k8smaster, k8snode1 and k8snode2
Check that k8smaster is reachable via 192.168.59.101
Check that k8snode1 is reachable via 192.168.59.102
Check that k8snode2 is reachable via 192.168.59.103

24.2 Script Installation

Script :

```
#!/bin/bash

sudo apt-get update && sudo apt-get install -y apt-transport-https ca-certificates curl software-properties-common gnupg2

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key --keyring /etc/apt/trusted.gpg.d/docker.gpg add -

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) \
stable"

sudo apt-get update && sudo apt-get install -y containerd.io=1.2.13-2 docker-ce=5:19.03.11~3-0~ubuntu-$(lsb_release -cs) docker-ce-cli=5:19.03.11~3-0~ubuntu-$(lsb_release -cs)

cat <<EOF | sudo tee /etc/docker/daemon.json
{
```

```
"exec-opts": ["native.cgroupdriver=systemd"],
"log-driver": "json-file",
"log-opts": {
"max-size": "100m"
},
"storage-driver": "overlay2"
}
EOF

# Create /etc/systemd/system/docker.service.d
sudo mkdir -p /etc/systemd/system/docker.service.d
# Restart Docker
sudo systemctl daemon-reload
sudo systemctl restart docker
sudo systemctl enable docker

sudo usermod -aG docker usera

sudo apt-get update && sudo apt-get install -y apt-transport-https curl
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
cat <<EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb https://apt.kubernetes.io/ kubernetes-xenial main
EOF
sudo apt-get update
sudo apt-get install -y kubectl=1.23.1-00 kubeadm=1.23.1-00 kubelet=1.23.1-00
```

24.3 Docker Installation

Source → https://kubernetes.io/docs/setup/production-environment/container-runtimes/#docker
Install Docker CE 19.0.3 on k8smaster, k8snode1 and k8snode2 – Copy paste below lines – run them as root:

```
# (Install Docker CE)
## Set up the repository:
### Install packages to allow apt to use a repository over HTTPS
sudo apt-get update && sudo apt-get install -y \
  apt-transport-https ca-certificates curl software-properties-common gnupg2
# Add Docker's official GPG key:
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key --keyring
/etc/apt/trusted.gpg.d/docker.gpg add -
```

```
# Add the Docker apt repository:
sudo add-apt-repository \
  "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) \
  stable"
# Install Docker CE
sudo apt-get update && sudo apt-get install -y \
  containerd.io \
  docker-ce \
  docker-ce-cli
# Set up the Docker daemon
cat <<EOF | sudo tee /etc/docker/daemon.json
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
# Create /etc/systemd/system/docker.service.d
sudo mkdir -p /etc/systemd/system/docker.service.d
# Restart Docker
sudo systemctl daemon-reload
sudo systemctl restart docker
sudo systemctl enable docker
```

Note:
If you copy/paste above block, you need to run third time.

24.4 Kubeadm Installation

Install kubeadm on k8smaster k8snode1 and k8snode2 as root

```
sudo apt-get update && sudo apt-get install -y apt-transport-https curl
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
cat <<EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb https://apt.kubernetes.io/ kubernetes-xenial main
```

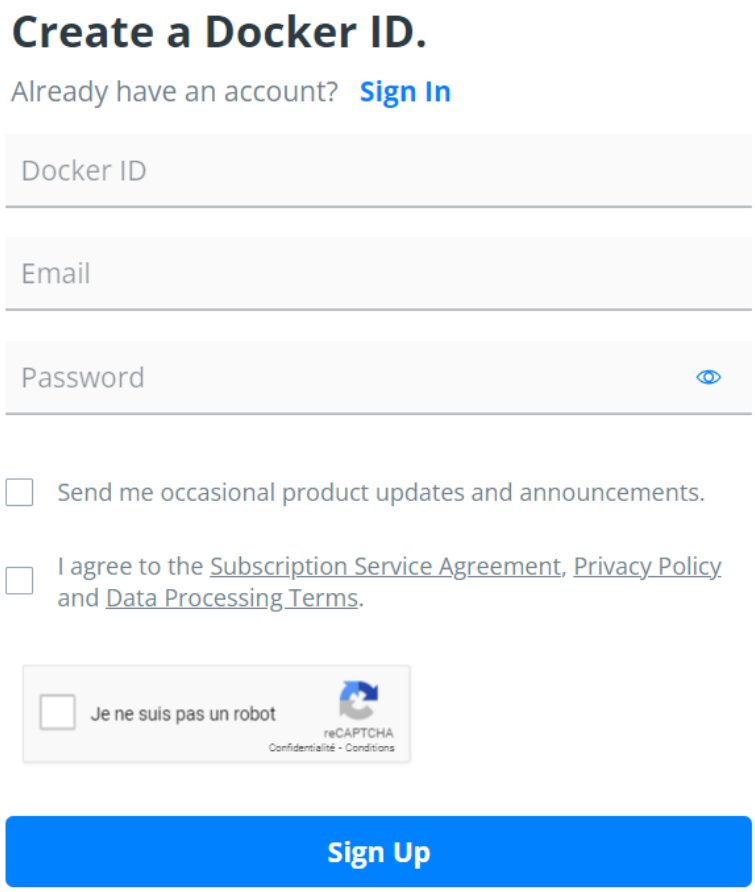
```
EOF
sudo apt-get update
sudo apt-get install -y kubectl=1.23.1-00 kubeadm=1.23.1-00 kubelet=1.23.1-00
```

24.5 Increase Docker limit rate

You may encounter **ErrImagePull** as follows:

```
[usera@lx-1-3 kubernetes-training]$ kubectl describe po <po name>
  Normal   Pulling      3m17s (x4 over 4m47s)  kubelet          Pulling image
"docker.io/calico/pod2daemon-flexvol:v3.21.1"
  Warning  Failed       3m16s (x4 over 4m46s)  kubelet          Failed to pull image
"docker.io/calico/pod2daemon-flexvol:v3.21.1": rpc error: code = Unknown desc = Error
response from daemon: toomanyrequests: You have reached your pull rate limit. You may
increase the limit by authenticating and upgrading: https://www.docker.com/increase-rate-limit
  Warning  Failed       3m16s (x4 over 4m46s)  kubelet          Error: ErrImagePull
```

You can increase Docker rate limit if you go to https://hub.docker.com/signup and create Docker ID



Source → https://kubernetes.io/docs/tasks/configure-pod-container/configure-service-account/

Method1 : create a secret + update deployment with imagePullSecrets

You create myprivatedocker secret type docker-registry in target namespace (here vote)

```
[usera@lx-6-1 ~]$ kubectl create secret docker-registry myprivatedocker
--docker-server=https://index.docker.io/v1/ --docker-username=<docker name id> --docker-
password=<docker password> --docker-email=<docker mail> -n vote
secret/myprivatedocker created
```

You add imagePullSecrets into default service account in target namespace (here vote)

```
[usera@lx-6-1 ~]$ kubectl -n vote patch serviceaccount default -p '{"imagePullSecrets":
[{"name": "myprivatedocker"}]}'
serviceaccount/default patched
```

You repeat 2 steps on other namespaces where you have ImagePullBackOff

If you have still ErrImagePullBackoff error, you force to pull the pod image with a delete

```
[...]
vote            result-86d8966d87-g6bjv            0/1     ImagePullBackOff   0          134m
vote            vote-6d4876585f-46trj              0/1     ImagePullBackOff   0          134m
vote            worker-7cbf9df499-zw9gr            0/1     ImagePullBackOff   0          134m
[usera@lx-6-1 ~]$ kubectl delete po result-86d8966d87-g6bjv vote-6d4876585f-46trj worker-
7cbf9df499-zw9gr -n vote
pod "result-86d8966d87-g6bjv" deleted
pod "vote-6d4876585f-46trj" deleted
pod "worker-7cbf9df499-zw9gr" deleted
[usera@lx-6-1 ~]$ kubectl get po -A
NAMESPACE          NAME                                       READY   STATUS            RESTARTS   AGE
calico-apiserver   calico-apiserver-66c76fb49-dhn5l           1/1     Running           0          3h15m
calico-apiserver   calico-apiserver-66c76fb49-vq596           1/1     Running           0          3h15m
calico-system      calico-kube-controllers-588575d68-kvw9c    1/1     Running           0          3h16m
calico-system      calico-node-gdn27                          1/1     Running           0          3h4m
calico-system      calico-node-xv489                          1/1     Running           0          3h6m
calico-system      calico-node-z69h5                          1/1     Running           0          3h16m
calico-system      calico-typha-769f7954b9-kpvtz              1/1     Running           0          3h16m
calico-system      calico-typha-769f7954b9-qj27s              1/1     Running           0          3h4m
ingress-controller ic-ingress-nginx-controller-6gjz5          1/1     Running           0          117m
ingress-controller ic-ingress-nginx-controller-hjmb4          1/1     Running           0          117m
kube-system        coredns-78fcd69978-r6c4t                   1/1     Running           0          3h38m
kube-system        coredns-78fcd69978-xl7jj                   1/1     Running           0          3h38m
```

```
kube-system      etcd-k8smaster                      1/1   Running          0        3h38m
kube-system      kube-apiserver-k8smaster            1/1   Running          0        3h38m
kube-system      kube-controller-manager-k8smaster   1/1   Running          0        3h38m
kube-system      kube-proxy-r29xf                    1/1   Running          0        3h38m
kube-system      kube-proxy-vv88q                    1/1   Running          0        3h6m
kube-system      kube-proxy-ztrk4                    1/1   Running          0        3h4m
kube-system      kube-scheduler-k8smaster            1/1   Running          0        3h38m
tigera-operator  tigera-operator-b78466769-wpzdd     1/1   Running          0        3h31m
vote             db-7fd7dd8c6d-xhnxs                 1/1   Running          0        27m
vote             redis-67db9bd79b-6vtx4              1/1   Running          0        26m
vote             result-86d8966d87-vlng8             1/1   Running          0        38s
vote             vote-6d4876585f-fcpsj               1/1   Running          0        38s
vote             worker-7cbf9df499-wjlg4             0/1   ContainerCreating  0      38s
```

Method 2 : docker login onto nodes

If the problem still persists, you can pull manually Docker images on k8snode1 and k8snode2

```
usera@k8snode1:~$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID,
head over to https://hub.docker.com to create one.
Username: formation123
Password:
WARNING! Your password will be stored unencrypted in /home/usera/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
usera@k8snode1:~$ docker pull calico/node:v3.21.1
v3.21.1: Pulling from calico/node
82af961e07d6: Pull complete
cc016e6e931b: Pull complete
Digest: sha256:a56f71d9f877c4bb75e2823c30c9bec3b1e29eae0ba97bc0dd01b25c69099cca
Status: Downloaded newer image for calico/node:v3.21.1
docker.io/calico/node:v3.21.1
usera@k8snode1:~$ docker pull calico/pod2daemon-flexvol:v3.21.1
v3.21.1: Pulling from calico/pod2daemon-flexvol
235a22629943: Pull complete
9e4a3266b915: Pull complete
d04283e8af03: Pull complete
aefc1645e78a: Pull complete
```

```
0314b6034116: Pull complete
eb9c819261ce: Pull complete
1b13f574f3ee: Pull complete
Digest: sha256:480ff7c9a9c981d29412f89bde9101d2b68edca3141a90f91e7e4f9cb8c93783
Status: Downloaded newer image for calico/pod2daemon-flexvol:v3.21.1
docker.io/calico/pod2daemon-flexvol:v3.21.1
usera@k8snode1:~$ docker pull calico/cni:v3.21.1
v3.21.1: Pulling from calico/cni
2c442e4957d9: Pull complete
8d453defe9ec: Pull complete
2e015015b133: Pull complete
69de73d4ec7f: Pull complete
66cda53ceeb3: Pull complete
Digest: sha256:0777ddd585fb9d5005190a2c3642f1a39ffaa3ee52e5f015f86870eb80479982
Status: Downloaded newer image for calico/cni:v3.21.1
docker.io/calico/cni:v3.21.1
usera@k8snode1:~$ docker pull calico/typha:v3.21.1
v3.21.1: Pulling from calico/typha
7c4c2450c98a: Pull complete
d53a28e2db71: Pull complete
0b8f79fea4f7: Pull complete
07471a4c95a6: Pull complete
3978f5329bbc: Pull complete
a1be522ef4fd: Pull complete
a475f3562df8: Pull complete
9b0e4d83908a: Pull complete
Digest: sha256:25903e3dc572b4af9eea0adec4147b4d943dc231bf7bc2ea83ff7c1b9240379c
Status: Downloaded newer image for calico/typha:v3.21.1
docker.io/calico/typha:v3.21.1
```

24.6 Do not update Calico operator version

You must keep Calico operator version 1.28.1

```
usera@k8smaster:~$ kubectl patch deployment.apps/tigera-operator -p '{"spec":{"template":
{"spec":{"containers":[{"name":"tigera-operator","env":
[{"name":"TIGERA_OPERATOR_INIT_IMAGE_VERSION","value": "v1.28.1"}]}]}}}}' -n
tigera-operator
```

```
deployment.apps/tigera-operator patched
usera@k8smaster:~$ kubectl patch deployment.apps/tigera-operator -p '{"spec":{"template":
{"spec":{"containers":[{"name":"tigera-operator","image":"quay.io/tigera/
operator:v1.28.1"}]}}}}' -n tigera-operator
deployment.apps/tigera-operator patched
```

to use calico version v1.24.1