

## GRADED EXERCISE - SUBJECT - 3H

## 1. Créer un namespace qui porte le nom exam-kub dédié à notre application

Définition : Un namespace est un cluster virtuelle dans un cluster physique Kubernetes. Il permet d'isoler notre application, et projet en fonction de nous besoin.

On va créer notre ns avec yaml:

- **nano 285893.yaml**

voici le fichier yaml pour notre namespace

**apiVersion: v1**

**kind: Namespace**

**metadata:**

```
k8s-client@k8s-client:~$ kubectl create -f285893.yaml  
namespace/285893 created
```

Lister les namespaces au sein de notre cluster :

```
k8s-client@k8s-client:~$ kubectl get ns
```

NAME	STATUS	AGE
298366	Active	58m
default	Active	14h
kube-node-lease	Active	14h
kube-public	Active	14h
kube-system	Active	14h

changer le namespace par défaut par cette commande :

```
kubectl config set-context $(kubectl config current-context) --namespace  
298366
```

## **2. volumes**

Avant de déployer prestashop on va créer d'abord les volumes

Pvc , pv données doivent être persistées, vous utiliserez les objets de type pv,  
pvc

**nano pv\_pvc.yaml :**

**apiVersion: v1**

**kind: PersistentVolume**

**metadata:**

**name: prestashop-pv**

**labels:**

**prestashop\_storage: 'true'**

**spec:**

**capacity:**

**storage: 5Gi**

**volumeMode: Filesystem**

**accessModes:**

**- ReadWriteOnce**

**persistentVolumeReclaimPolicy: Retain**

**storageClassName: ''**

**hostPath:**

**path: /data/prestashop**

**type: DirectoryOrCreate**

---

**apiVersion: v1**  
**kind: PersistentVolume**  
**metadata:**  
    **name: mariadb-pv**  
**labels:**  
    **mariadb\_storage: 'true'**  
**spec:**  
    **capacity:**  
        **storage: 5Gi**  
    **volumeMode: Filesystem**  
**accessModes:**  
    - **ReadWriteOnce**  
**persistentVolumeReclaimPolicy: Retain**  
**storageClassName: ''**  
**hostPath:**  
    **path: /data/mariadb**  
    **type: DirectoryOrCreate**

---

**apiVersion: v1**  
**kind: PersistentVolumeClaim**  
**metadata:**  
    **namespace: '285893'**  
**name: prestashop-pvc**  
**spec:**

**accessModes:**  
- **ReadWriteOnce**  
**volumeMode: Filesystem**  
**resources:**  
  **requests:**  
    **storage: 5Gi**  
**storageClassName: "**  
**selector:**  
**matchLabels:**  
  **prestashop\_storage: 'true'**

---

**apiVersion: v1**  
**kind: PersistentVolumeClaim**  
**metadata:**  
  **namespace: '285893'**  
**name: mariadb-pvc**  
**spec:**  
  **accessModes:**  
    - **ReadWriteOnce**  
  **volumeMode: Filesystem**  
  **resources:**  
    **requests:**  
      **storage: 5Gi**  
  **storageClassName: "**  
  **selector:**  
  **matchLabels:**  
    **mariadb\_storage: 'true'**

```
k8s-client@k8s-client:~$ kubectl create -f pv_pvc.yaml
persistentvolume/prestashop-pv created
persistentvolume/mariadb-pv created
persistentvolumeclaim/prestashop-pvc created
persistentvolumeclaim/mariadb-pvc created
```

### **3. secret :**

maintenant on va créer le fichier yaml pour Secret ou se trouveront les informations confidentielles ( ex : PRESTASHOP\_DATABASE\_NAME , PRESTASHOP\_DATABASE\_USER, PRESTASHOP\_DATABASE\_PASSWORD )

```
k8s-client@k8s-client:~$ nano secret_prestashop.yaml
```

```
apiVersion: v1
kind: Secret
metadata:
  namespace: '285893'
name: prestashop-secret
type: Opaque
stringData:
  prestashop_password: 'SUPINFOmsc1'
  mariadb_host: mariadb-service
  prestashop_database_name: prestashop_name
  prestashop_database_user: prestashop_user
  prestashop_database_password: 'SUPINFOmsc1'
```

**mariadb\_root\_password: 'SUPINFOmsc1'**

**k8s-client@k8s-client:~\$ kubectl create -f secret\_prestashop.yaml**  
**secret/prestashop-secret created**

#### **4. configMap**

ConfigMap ou se trouveront les informations concernant le site ( ex :  
PRESTASHOP\_FIRST\_NAME , PRESTASHOP\_LAST\_NAME )

**k8s-client@k8s-client:~\$ nano configMap\_prestashop.yaml**

**apiVersion: v1**

**kind: ConfigMap**

**metadata:**

**namespace: '285893'**

**name: prestashop-configmap**

**k8s-client@k8s-client:~\$ kubectl create -f configMap\_prestashop.yaml**  
**configmap/prestashop-configmap created**

## **5. deployment prestashop**

Maintenant on peut créer notre deployment

```
k8s-client@k8s-client:~$ nano deployment_prestashop.yaml
```

```
apiVersion: apps/v1
```

```
kind: Deployment
```

```
metadata:
```

```
namespace: '285893'
```

```
  name: prestashop-deployment
```

```
  labels:
```

```
    app: prestashop
```

```
spec:
```

```
  replicas: 1
```

```
  selector:
```

```
    matchLabels:
```

```
      app: prestashop
```

```
  template:
```

```
    metadata:
```

```
      labels:
```

```
        app: prestashop
```

```
    spec:
```

```
      initContainers:
```

```
        - name: init-prestashop
```

```
          image: busybox:1.28
```

```
          command:
```

```
            - chmod
```

- '777'
- '/bitnami/prestashop'

**volumeMounts:**

- **mountPath:** '/bitnami/prestashop'
- name:** mypd

**containers:**

- **name:** prestashop
- image:** bitnami/prestashop:1.7

**volumeMounts:**

- **mountPath:** '/bitnami/prestashop'
- name:** mypd

**ports:**

- **containerPort:** 80

**env:**

- **name:** PRESTASHOP\_FIRST\_NAME
- valueFrom:**
  - configMapKeyRef:**
    - name:** prestashop-configmap
    - key:** prestashop\_first\_name
- **name:** PRESTASHOP\_LAST\_NAME
- valueFrom:**
  - configMapKeyRef:**
    - name:** 'prestashop-configmap'
    - key:** prestashop\_last\_name
- **name:** PRESTASHOP\_HOST
- valueFrom:**
  - configMapKeyRef:**
    - name:** 'prestashop-configmap'
    - key:** prestashop\_host



**- name: PRESTASHOP\_PASSWORD**  
**valueFrom:**  
**secretKeyRef:**  
**name: 'prestashop-secret'**  
**key: prestashop\_password**

**- name: PRESTASHOP\_EMAIL**  
**valueFrom:**  
**secretKeyRef:**  
**name: 'prestashop-secret'**  
**key: prestashop\_email**

**- name: MARIADB\_HOST**  
**valueFrom:**  
**secretKeyRef:**  
**name: 'prestashop-secret'**  
**key: mariadb\_host**

**- name: PRESTASHOP\_DATABASE\_NAME**  
**valueFrom:**  
**secretKeyRef:**  
**name: 'prestashop-secret'**  
**key: prestashop\_database\_name**

**- name: PRESTASHOP\_DATABASE\_USER**  
**valueFrom:**  
**secretKeyRef:**  
**name: 'prestashop-secret'**  
**key: prestashop\_database\_user**

**- name: PRESTASHOP\_DATABASE\_PASSWORD**  
**valueFrom:**  
**secretKeyRef:**  
**name: 'prestashop-secret'**

```
    key: prestashop_database_password
volumes:
- name: mypd
  persistentVolumeClaim:
    claimName: prestashop-pvc
```

```
k8s-client@k8s-client:~$ kubectl create -f deployment_prestashop.yaml
deployment.apps/prestashop-deployment created
```

## **6. deployment mariadb**

```
k8s-client@k8s-client:~$ nano deployment_mariadb.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  namespace: '285893'
  name: mariadb-deployment
  labels:
    app: mariadb
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mariadb
  template:
    metadata:
      labels:
        app: mariadb
    spec:
```

**initContainers:**

- **name: init-mariadb**  
**image: busybox:1.28**  
**command:**
  - **chmod**
  - **'777'**
  - **'/bitnami/mariadb'**

**volumeMounts:**

- **mountPath: '/bitnami/mariadb'**  
**name: mypd**

**containers:**

- **name: mariadb**  
**image: bitnami/mariadb:10.1**  
**volumeMounts:**
    - **mountPath: '/bitnami/mariadb'**  
**name: mypd**
- ports:**
- **containerPort: 3306**

**env:**

- **name: MARIADB\_USER**  
**valueFrom:**  
**secretKeyRef:**
  - name: prestashop-secret**
  - key: prestashop\_database\_user**
- **name: MARIADB\_PASSWORD**  
**valueFrom:**  
**secretKeyRef:**
  - name: prestashop-secret**
  - key: prestashop\_database\_password**

```

- name: MARIADB_DATABASE
  valueFrom:
    secretKeyRef:
      name: prestashop-secret
      key: prestashop_database_name
- name: MARIADB_ROOT_PASSWORD
  valueFrom:
    secretKeyRef:
      name: prestashop-secret
      key: mariadb_root_password
volumes:
- name: mypd
  persistentVolumeClaim:
    claimName: mariadb-pvc

```

```

k8s-client@k8s-client:~$ kubectl create -f deployment_mariadb.yaml
deployment.apps/mariadb-deployment created

```

## 7. creation des services

- prestashop :

```

k8s-client@k8s-client:~$ nano service_prestashop.yaml

```

```

apiVersion: v1
kind: Service
metadata:
  namespace: '285893'

```

```
name: prestashop-service
spec:
  type: NodePort
  selector:
    app: prestashop
  ports:
    - port: 8080
      nodePort: 30000
      name: http
```

```
k8s-client@k8s-client:~$ kubectl create -f service_prestashop.yaml
service/prestashop-service created
```

- [service mariadb](#)

```
k8s-client@k8s-client:~$ nano service_mariabd.yaml
```

```
apiVersion: v1
kind: Service
metadata:
  namespace: '285893'
name: mariadb-service
spec:
  type: ClusterIP
  selector:
    app: mariadb
  ports:
    - port: 3306
```

```
k8s-client@k8s-client:~$ kubectl create -f service_mariabd.yaml
service/mariadb-service created
```

un petit get all pour voir nos ressources

```
k8s-client@k8s-client:~$ kubectl get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/mariadb-deployment-84d9f4fcc8-mlvrl	1/1	Running	0	2m16s
pod/prestashop-deployment-6d978b6f8f-vrbxp	1/1	Running	0	2m51s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/mariadb-service	ClusterIP	10.104.255.140	<none>	3306/TCP	95s
service/prestashop-service	NodePort	10.111.104.133	<none>	8080:30000/TCP	116s

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/mariadb-deployment	1/1	1	1	2m16s
deployment.apps/prestashop-deployment	1/1	1	1	2m51s

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/mariadb-deployment-84d9f4fcc8	1	1	1	2m16s
replicaset.apps/prestashop-deployment-6d978b6f8f	1	1	1	2m51s

```
k8s-client@k8s-client:~$
```

```
k8s-client@k8s-client:~$ kubectl get nodes -o wide
```

NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXTERNAL-IP	OS-IMAGE	KERNEL-VERSION	CONTAINER-RUNTIME
k8s-1	Ready	master	15h	v1.16.3	192.168.48.141	<none>	Ubuntu 18.04.4 LTS	4.15.0-128-generic	docker://18.6.2
k8s-2	Ready	<none>	15h	v1.16.3	192.168.48.142	<none>	Ubuntu 18.04.4 LTS	4.15.0-128-generic	docker://18.6.2
k8s-3	Ready	<none>	15h	v1.16.3	192.168.48.143	<none>	Ubuntu 18.04.4 LTS	4.15.0-128-generic	docker://18.6.2

```
k8s-client@k8s-client:~$
```

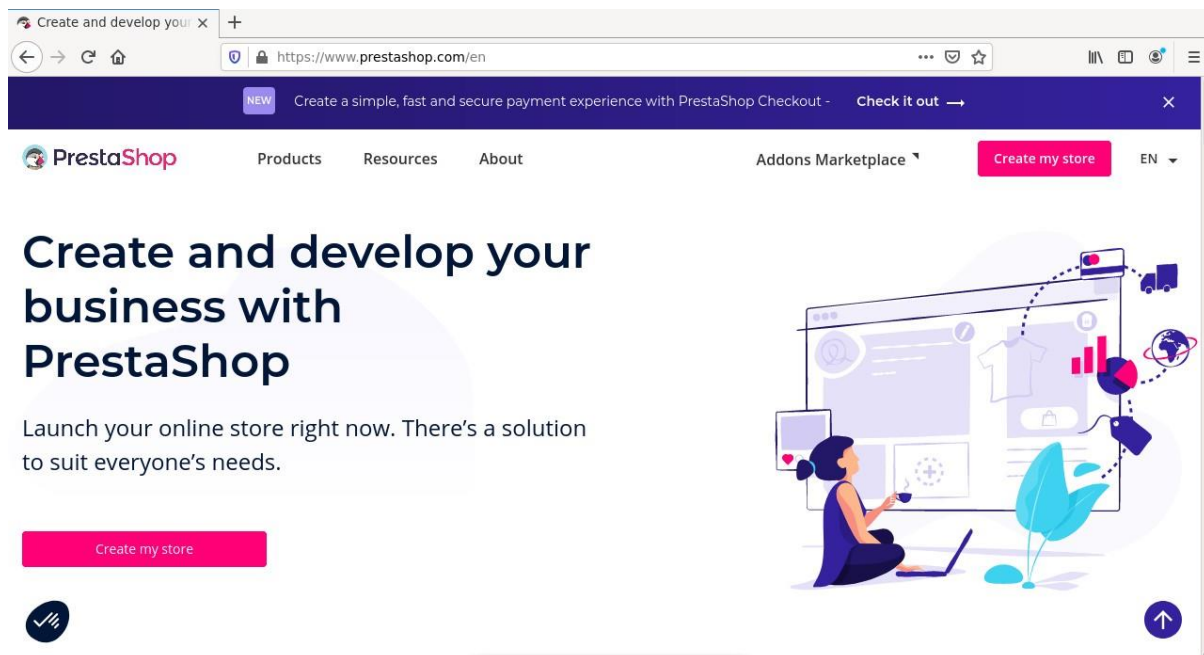
```
k8s-client@k8s-client:~$ kubectl get services -o wide
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE	SELECTOR
mariadb-service	ClusterIP	10.104.255.140	<none>	3306/TCP	5m43s	app=mariadb
prestashop-service	NodePort	10.111.104.133	<none>	8080:30000/TCP	6m4s	app=prestashop

```
k8s-client@k8s-client:~$
```

sur le navigateur notre application est a l'ecoute sur

**192.168.48.141:30000**



## 8. La Mise a jours des cluster

On doit tout d'abord commencer par mettre à jour notre nœud master k8s-1 , et par la suite mettre à jour les autres nœuds de travail un par un k8s-2 et k8s3.

Dans mon cas, je souhaite upgrade mon master de la version 1.16 vers la version 1.17. Pour ce faire, nous entamerons une phase de recherche afin de trouver la dernière version stable de kubeadm et kubelet en v1.17. je vais utiliser l'outil (apt) afin de rechercher mes nouveaux paquets

**apt-get update && \**

**apt-cache policy kubeadm**

### Résultat :

```
kubeadm:

Installed: 1.16.0-00

Candidate: 1.17.0-00

...
```

### apt-cache policy kubelet

### Résultat :

```
kubelet:

Installed: 1.16.0-00

Candidate: 1.17.0-00

...
```

d'abord nous attaquer l'installation du nouveau paquet kubeadm :

### apt-get upgrade -y kubeadm=1.17.0-00

Ensuite, nous vérifions que le téléchargement fonctionne et possède la version attendue

### kubeadm version

### Résultat :

```
kubeadm version: &version.Info{Major:"1", mineur:"17",
GitVersion:"v1.17.0" ...}
```

À présent, nous allons interroger l'outil kubeadm

### kubeadm upgrade plan



## Résultat :

Components that must be upgraded manually after you have upgraded the control plane with 'kubeadm upgrade apply':

COMPONENT	CURRENT	AVAILABLE
Kubelet	4 x v1.16.0	v1.17.0

Upgrade to the latest stable version:

COMPONENT	CURRENT	AVAILABLE
API Server	v1.16.0	v1.17.0
Controller Manager	v1.16.0	v1.17.0
Scheduler	v1.16.0	v1.17.0
Kube Proxy	v1.16.0	v1.17.0
CoreDNS	1.6.2	1.6.5
Etcd	3.3.15	3.4.3-0

You can now apply the upgrade by executing the following command:

```
kubeadm upgrade apply v1.17.0
```

L'étape suivante consiste à **rendre notre nœud master unschedulable**:

**kubectl drain k8s-1 --ignore-daemonsets**

Enfin, nous allons appliquer les nouveaux changement grâce la commande suivante :

**kubeadm upgrade apply v1.17.0**

**Résultat :**

```
[upgrade/successful] SUCCESS! Your cluster was upgraded to "v1.17.0".  
Enjoy!
```

Nous devons par la suite installer la version stable de kubelet en v1.17.0 et redémarrer le service de manière à prendre en compte sa nouvelle version :

**apt-get upgrade -y kubelet=1.17.0-00 && \**

**systemctl restart kubelet**

Enfin, on n'oublie pas de rendre notre nœud à nouveau schedulable :

**kubectl uncordon master**

En lançant la commande ci-dessous, on peut remarquer que nos nœuds workers ne possèdent pas la même version que notre nœud master:

kubectl get nodes

**Résultat :**

NAME	STATUS	ROLES	AGE	VERSION
------	--------	-------	-----	---------

K8s-1	Ready	master	33m	v1.17.0
-------	-------	--------	-----	---------

2	Ready	<none>	32m	v1.16.0
---	-------	--------	-----	---------

	Ready	<none>	31m	v1.16.0
--	-------	--------	-----	---------

## Upgrade des worker k8s-2 et k8s-3

Premièrement, nous allons rendre notre nœud de travail unschedulable depuis notre nœud master:

```
kubectrl drain k8s-2 --ignore-daemonsets
```

Ensuite il faut être connecté sur un worker et installer les dernières versions stables de kubeadm et kubelet depuis notre nœud de travail :

```
ssh root@k8s2 "apt-get update && apt-get upgrade -y kubeadm=1.17.0-00  
kubelet=1.17.0-00"
```

Depuis notre master nous allons mettre à niveau notre worker :

```
kubeadm upgrade k8s-2 config --kubelet-version v1.17.0
```

Ensuite, il faut redémarrer le service kubelet :

```
ssh root@ "systemctl restart kubelet"
```

Enfin, on n'oublie pas de rendre notre nœud à nouveau schedulable :

```
kubectrl uncordon k8s-2
```

on fait la meme chose pour le node k8s-3

En vérifiant les nœuds disponibles de notre cluster

```
kubectrl get nodes
```

Résultat :

NAME	STATUS	ROLES	AGE	VERSION
K8s-1	Ready	master	55m	v1.17.0
K8s-2	Ready	<none>	53m	v1.17.0
K8s-3	Ready	<none>	51m	v1.17.0

## 9. Le lien github

<https://github.com/batachelydia/examen>