Un déployment est un concept de plus haut niveau qui gére les replicasets et fournit des mises à jour déclaratives aux pods ainsi que de nombreuses autres fonctionnalités utiles. Pour notre Tp, nous allons constituer certaines ressources à savoir :

Pour constituer nos ressources nous allons créer plusieurs fichier à savoir :

Un deployment prestashop avec l'image bitnami/prestashop :1.7

Un deployment MariaDB avec l'image bitnami/mariadb :10.1

Pour ce faire nous allons créer des fichiers de déploiements tels que :

Et le fichier de deploiement de prestashop au format yaml

```
yannick@debian-k8s-client:~/prestashop$ cat prestashop_dep.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
        name: prestashop-example-secret-configmap
spec:
  replicas: 1
  selector:
        matchLabels:
              app: prestashop
  template:
        metadata:
              labels:
                app: prestashop
        spec:
          volumes:
            - name: prestashop-data
              persistentVolumeClaim:
                  claimName: presta-pv-claim
          containers:
                - name: prestashop
                  image: bitnami/prestashop:latest
                  volumeMounts:
                        - mountPath: /bitnami
                          name: prestashop-data
                  ports:
                      - containerPort: 80
                      - containerPort: 443
                  env:
                      - name: PRESTASHOP_FIRST_NAME
                        valueFrom:
                                configMapKeyRef:
                                        name: site-info
                                        key: prestashop_first_name

                     - name: PRESTASHOP_LAST_NAME
                       valueFrom:
                                configMapKeyRef:
                                        name: site-info
                                        key: prestashop_last_name
                      - name: ALLOW_EMPTY_PASSWORD
```

Fichier Edition Affichage Terminal Onglets Aide

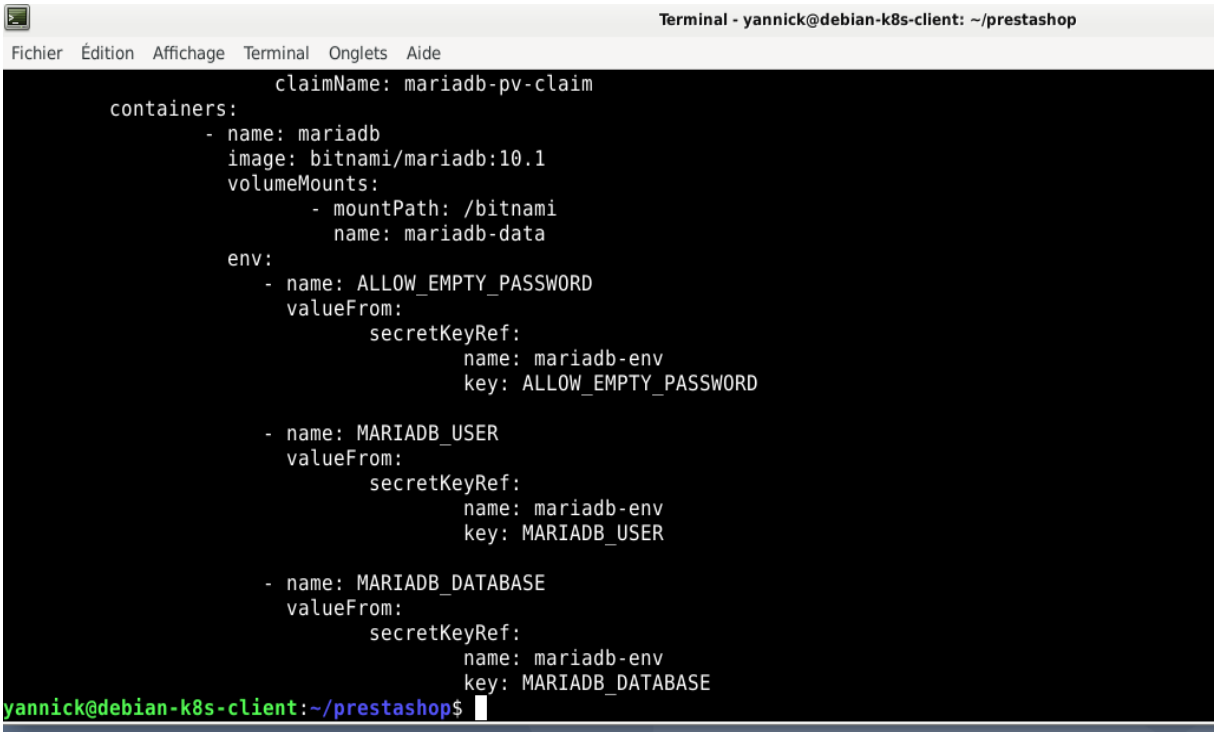
```
              persistentVolumeClaim:
                       claimName: presta-pv-claim
          containers:
                  - name: prestashop
                    image: bitnami/prestashop:latest
                    volumeMounts:
                           - mountPath: /bitnami
                             name: prestashop-data
                    ports:
                        - containerPort: 80
                        - containerPort: 443
                    env:
                        - name: PRESTASHOP_FIRST_NAME
                          valueFrom:
                                  configMapKeyRef:
                                          name: site-info
                                          key: prestashop_first_name

                       - name: PRESTASHOP_LAST_NAME
                         valueFrom:
                                  configMapKeyRef:
                                          name: site-info
                                          key: prestashop_last_name
                       - name: ALLOW_EMPTY_PASSWORD
                         valueFrom:
                                 secretKeyRef:
                                        name: presta-env
                                        key: ALLOW_EMPTY_PASSWORD

                       - name: PRETASHOP_DATABASE_USER
                         valueFrom:
                                 secretKeyRef:
                                        name: presta-env
                                        key: PRESTASHOP_DATABASE_USER

                       - name: PRETASHOP_DATABASE_NAME
                         valueFrom:
                                 secretKeyRef:
                                        name: presta-env
                                        key: PRESTASHOP_DATABASE_NAME
yannick@debian-k8s-client:~/prestashop$
```

Le fichier de deploiement de mariadb au format yaml



Nous allons appliqué la configuration au sein de notre cluster

On pourra ensuite taper dans un terminal la commande :

Kubectl apply –f nom du fichier

Kubectl apply –f mariadb_dep.yaml

```
yannick@debian-k8s-client:~/prestashop$ kubectl apply -f mariadb_dep.yaml
deployment.apps/mariadb-example-secret-configmap unchanged
yannick@debian-k8s-client:~/prestashop$
```

Kubectl apply –f prestashop_dep.yaml

```
yannick@debian-k8s-client:~/prestashop$ kubectl apply -f prestashop_dep.yaml
deployment.apps/prestashop-example-secret-configmap configured
yannick@debian-k8s-client:~/prestashop$
```

Apres avoir crée les deployments Prestashop et MariaDB on peut les voir avec la commande :

Kubectl get deployment

```
yannick@debian-k8s-client:~/prestashop/configmap$ kubectl get deploymen
NAME                                READY   UP-TO-DATE   AVAILABLE
mariadb-example-secret-configmap    0/1     1            0
prestashop-example-secret-configmap   0/1     1            0
yannick@debian-k8s-client:~/prestashop/configmap$
```

ConfigMap

Ce type de Ce type de volume est utilisé pour injecter des propriétés de paires clé-valeur dans des pods, telles que des informations de configuration d'application. Au lieu de définir des informations de configuration d'application au sein d'une image conteneur, vous pouvez les définir en tant que ressource Kubernetes pouvant être facilement mise à jour et appliquée à de nouvelles instances de pods au fur et à mesure de leur déploiement.

Un ConfigMap ou se trouveront les informations concernant le site (ex : PRESTASHOP_FIRST_NAME , PRESTASHOP_LAST_NAME)

Pour notre configmap nous allons créer un fichier .yaml nommé presta_configmap qui va contenir un username et un password de type opaque

```
yannick@debian-k8s-client:~/prestashop/configmap$ cat settings.txt
PRETASHOP_FIRST_NAME=
PRESTASHOP_LAST_NAME=
yannick@debian-k8s-client:~/prestashop/configmap$
```

```
yannick@debian-k8s-client:~/prestashop$ cat presta_configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
       name: site-info
data:
        prestashop_first: djidjou
        prestashop_last_name: yannick
yannick@debian-k8s-client:~/prestashop$
```

Ensuite appliqué le fichier créé avec la commande : kubectl apply –f presta_configmap.yaml

```
yannick@debian-k8s-client:~/prestashop$ kubectl apply -f presta_configmap.yaml
configmap/site-info unchanged
yannick@debian-k8s-client:~/prestashop$
```

Les Secrets :

Ce volume est utilisé pour injecter des données sensibles dans les pods, telles que les mots de passe.
Les données sont encodées en base64 et sont stockés dans un volume *tmpfs*.
Le fonctionnement est semblable au ConfigMap.

```
yannick@debian-k8s-client:~/prestashop$ cat secret.yaml
apiVersion: v1
kind: Secret
metadata:
        name: personal-info
type: Opaque
data:
        username: eWFubmljaw==
        password: dG90bw==
yannick@debian-k8s-client:~/prestashop$
```

Un Secret ou se trouveront les informations confidentielles (ex : PRESTASHOP_DATABASE_NAME , PRESTASHOP_DATABASE_USER, PRESTASHOP_DATABASE_PASSWORD)

```
yannick@debian-k8s-client:~/prestashop$ cat presta_env.txt
PRESTASHOP_DATABASE_NAME=bitnami_prestashop
PRETASHOP_DATABASE_USER=bn_prestahsop
ALLOW_EMPTY_PASSWORD=yes
MARIADB_HOST=
MARIADB_PORT_NUMBER=3306
yannick@debian-k8s-client:~/prestashop$
```

```
secret.yaml   settings.txt
yannick@debian-k8s-client:~/prestashop/configmap$ cat secret.yaml
apiVersion: v1
kind: Secret
metadata:
  name: mysecret
type: Opaque
data:
  username: YWRtaW4=yannick
  password: MWYyZDF1MmU2N2Rm
yannick@debian-k8s-client:~/prestashop/configmap$
```

Plusieurs services, pour accéder à votre forum et à la base de données.

Nous avons créé des fichiers pour les services :

Le fichier mariadb_service.yaml

```
yannick@debian-k8s-client:~/prestashop$ cat mariadb_service.yaml
apiVersion: v1
kind: Service
metadata:
        name: mariadb-service
        labels:
            run: mariadb-service
spec:
   type: NodePort
   ports:
     - port: 3306
       targetPort: 3306
       protocol: TCP
   selector:
       run : mariadb
yannick@debian-k8s-client:~/prestashop$
```

Le fichier prestashop_service.yaml

Le fichier presta_pvc.yaml

```
yannick@debian-k8s-client:~/prestashop$ cat presta_pvc.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
      name: presta-pv-claim
      labels:
          type: local
spec:
   storageClassName: manual
   accessModes:
         - ReadWriteOnce
   resources:
         requests:
             storage: 1Gi

yannick@debian-k8s-client:~/prestashop$
```

Le fichier presta_pv.yaml

```
yannick@debian-k8s-client:~/prestashop$ cat presta_pv.yaml
apiVersion: v1
kind: PersistentVolume
metadata:
      name: presta-pv
      labels:
          type: local
spec:
   storageClassName: manual
   capacity:
         storage: 1G
   accessModes:
         - ReadWriteOnce
   hostPath:
        path: "/tmp/presta_data"

yannick@debian-k8s-client:~/prestashop$
```

Le fichier mariadb_pvc

```
yannick@debian-k8s-client:~/prestashop$ cat mariadb_pvc.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
      name: mariadb-pv-claim
      labels:
          type: local
spec:
   storageClassName: manual
   accessModes:
         - ReadWriteOnce
   resources:
         requests:
               storage: 1Gi

yannick@debian-k8s-client:~/prestashop$
```

Le fichier mariadb_pv

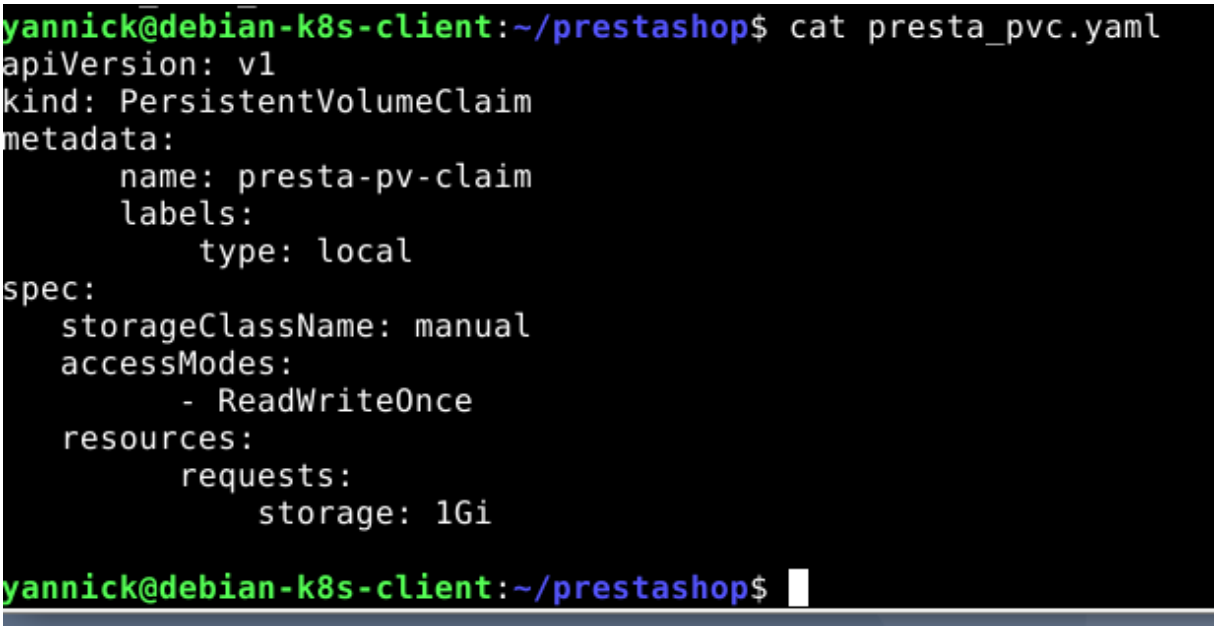
```
yannick@debian-k8s-client:~/prestashop$ cat mariadb_pv.yaml
apiVersion: v1
kind: PersistentVolume
metadata:
      name: mariadb-pv
      labels:
          type: local
spec:
   storageClassName: manual
   capacity:
         storage: 1G
   accessModes:
         - ReadWriteOnce
   hostPath:
        path: "/tmp/mariadb_data"

yannick@debian-k8s-client:~/prestashop$
```

```
yannick@debian-k8s-client:~/prestashop$ cat presta_env.txt
PRESTASHOP_DATABASE_NAME=bitnami_prestashop
PRETASHOP_DATABASE_USER=bn_prestahsop
ALLOW_EMPTY_PASSWORD=yes
MARIADB_HOST=
MARIADB_PORT_NUMBER=3306
yannick@debian-k8s-client:~/prestashop$
```

```
yannick@debian-k8s-client:~/prestashop$ cat presta_pv.yaml
apiVersion: v1
kind: PersistentVolume
metadata:
      name: presta-pv
      labels:
          type: local
spec:
   storageClassName: manual
   capacity:
         storage: 1G
   accessModes:
         - ReadWriteOnce
   hostPath:
        path: "/tmp/presta_data"

yannick@debian-k8s-client:~/prestashop$
```


yannick@debian-k8s-client:~/prestashop$ cat presta_pvc.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
name: presta-pv-claim
labels:
type: local
spec:
storageClassName: manual
accessModes:
- ReadWriteOnce
resources:
requests:
storage: 1Gi
yannick@debian-k8s-client:~/prestashop$