Application Complète

MongoDB and Mongo-Express

L'objectif de ce workshop est de vous présenter le déploiement de bout en bout une application complète constitué de deux conteneurs MongoDB (Base de données NoSQL)et MongoExpress(Client Mongo).

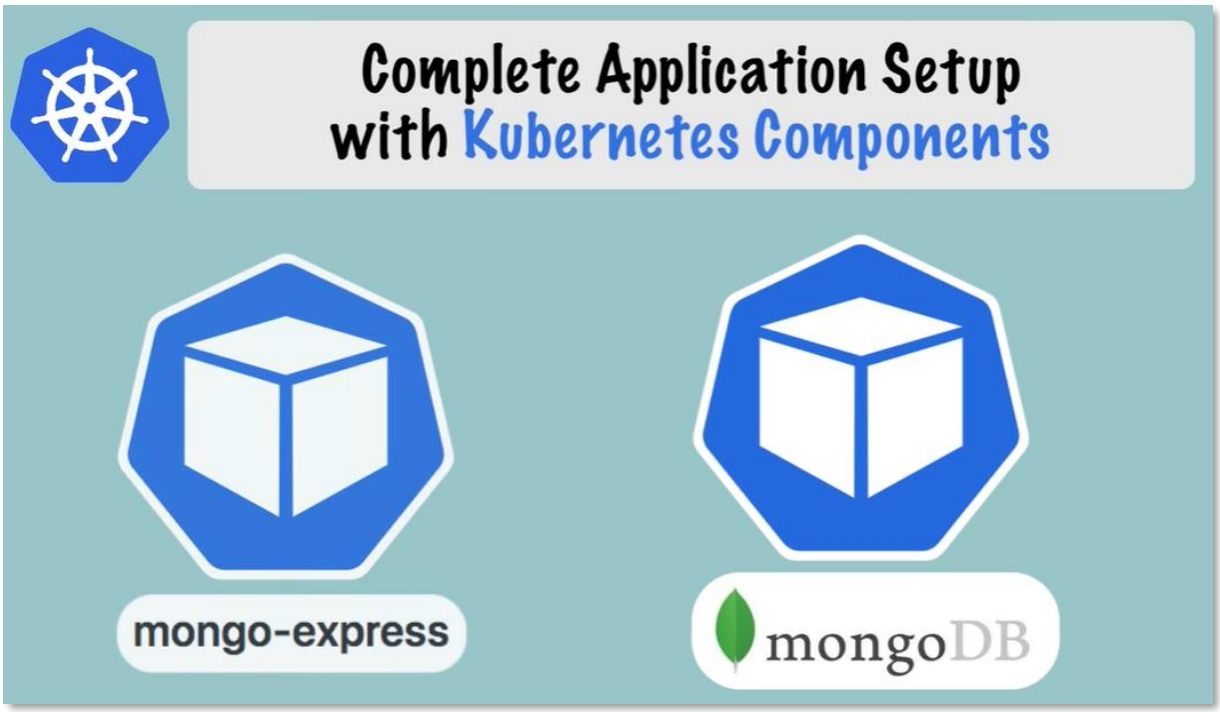
Les objets qu'on va manipuler sont :

1-Deploiment

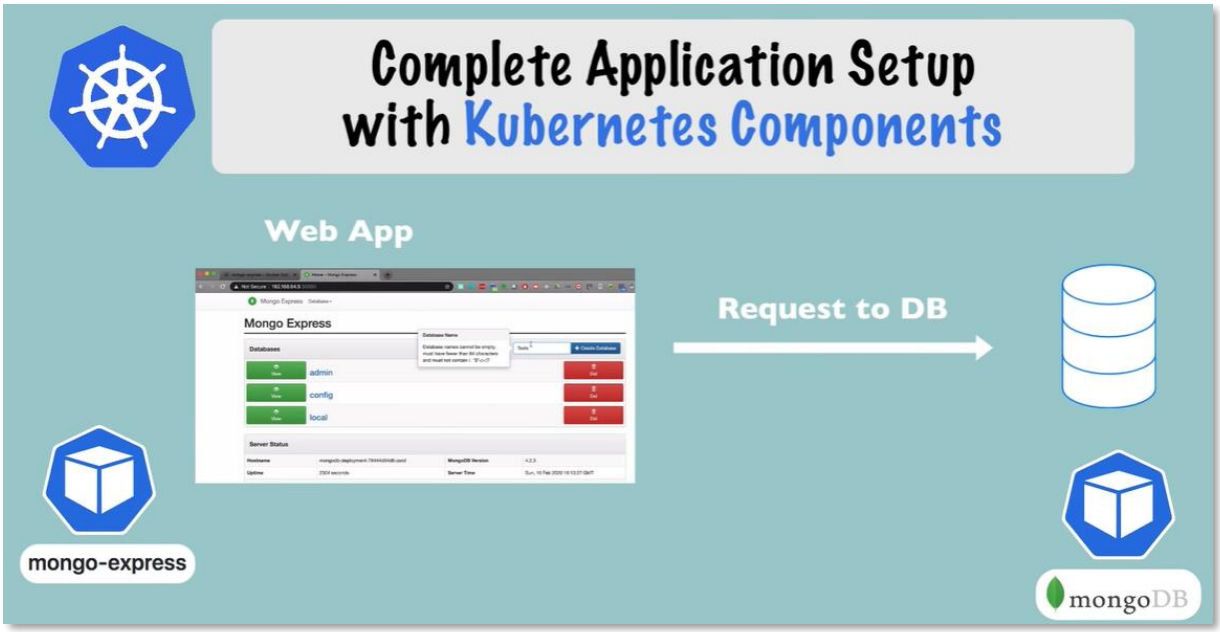
2-Service

3-ConfigMap

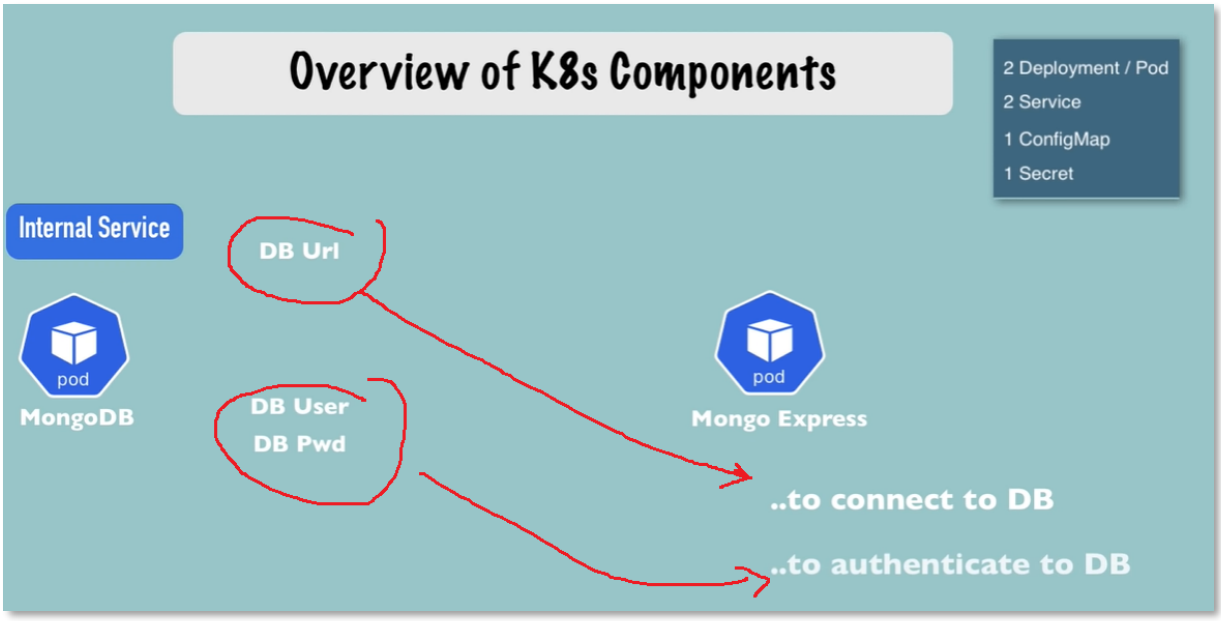
4-Secret



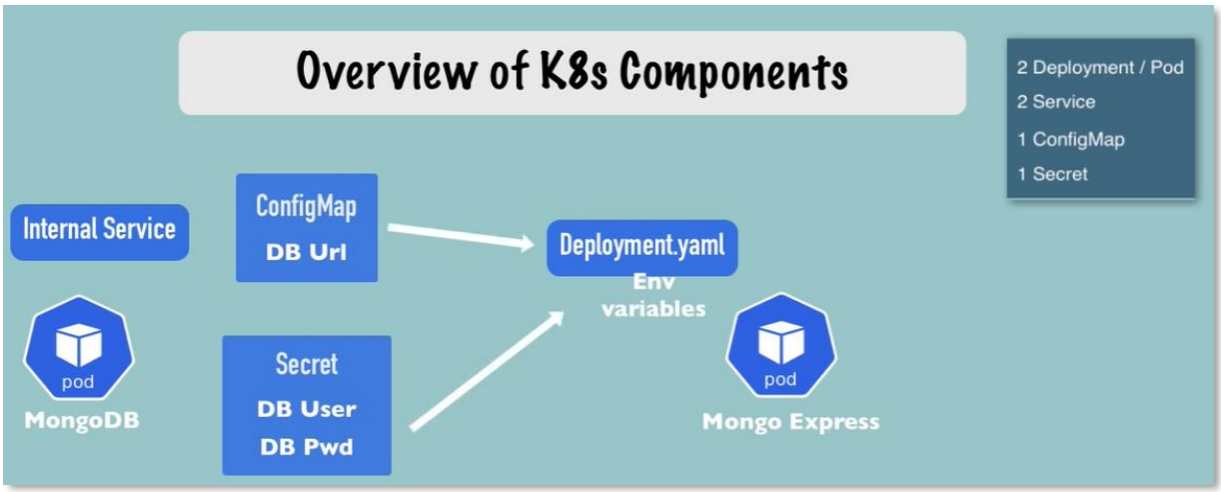
Le schéma générale de l'application va être comme suit :



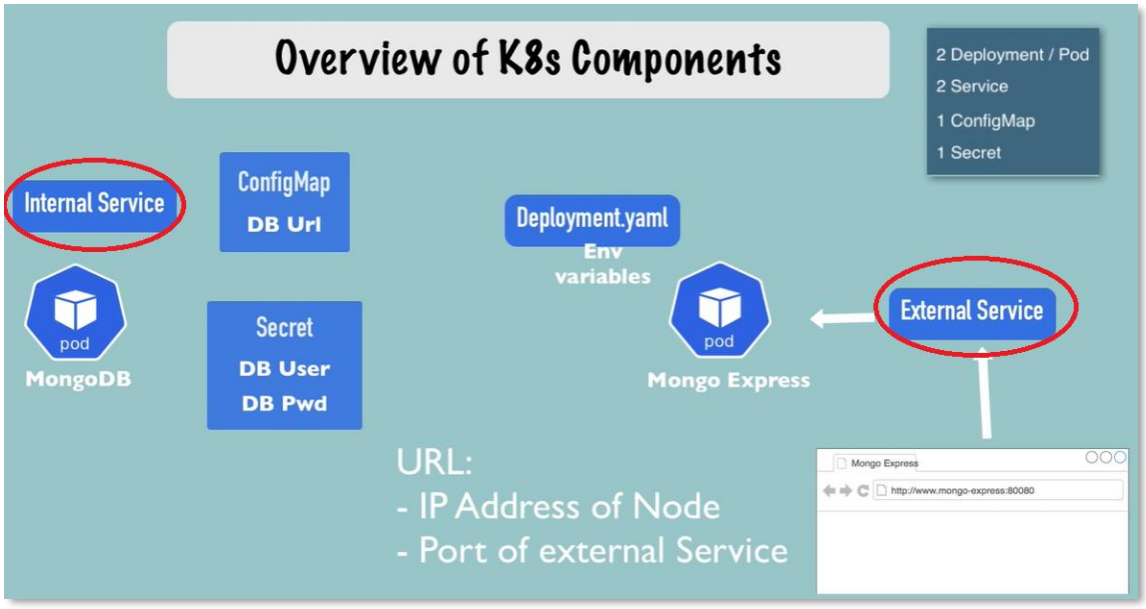
Les services internes n'acceptes pas des requests externes, du coup on est obliger de passer par des services externes qu'on appel des LoadBalancer

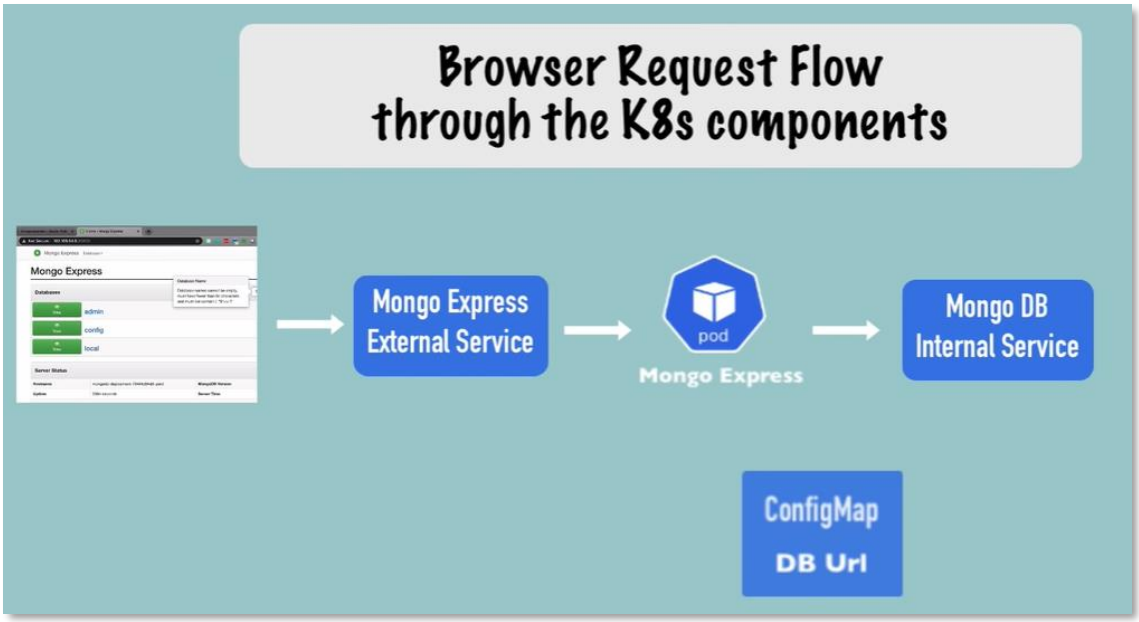


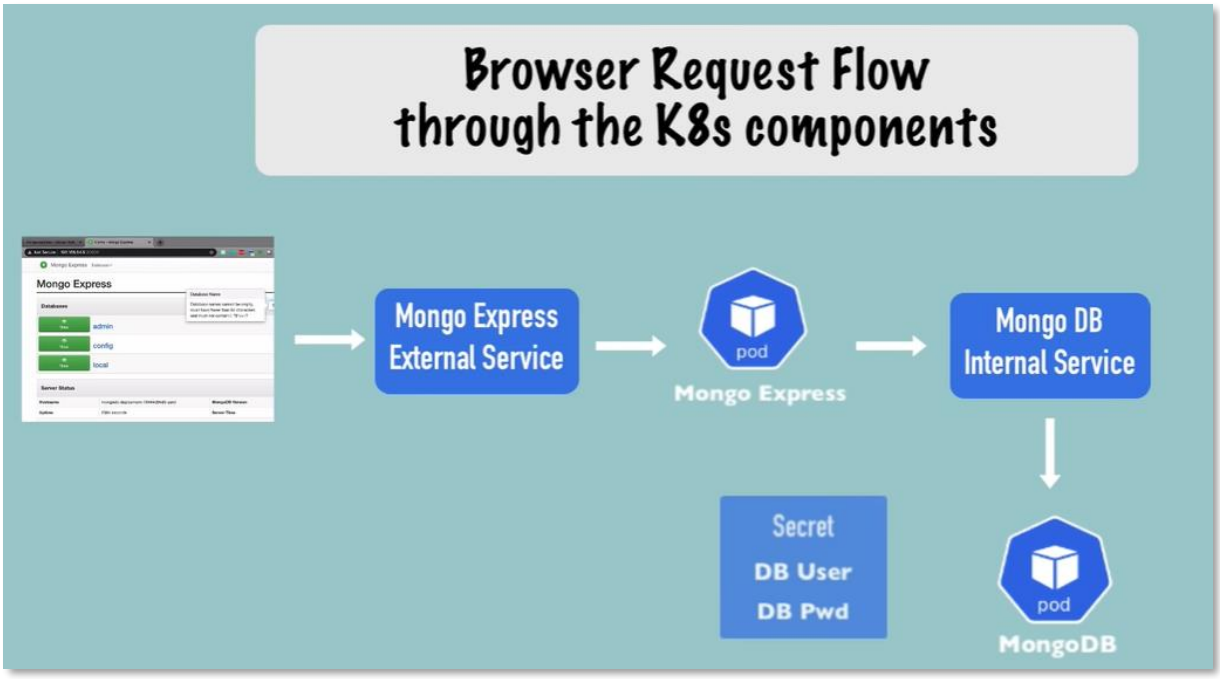
Pour stocker des informations (comme les variables d'environnement docker), on fait appel à des composants Kubernetes à savoir les ConfigMap et les Secret à travers les fichiers de configurations.

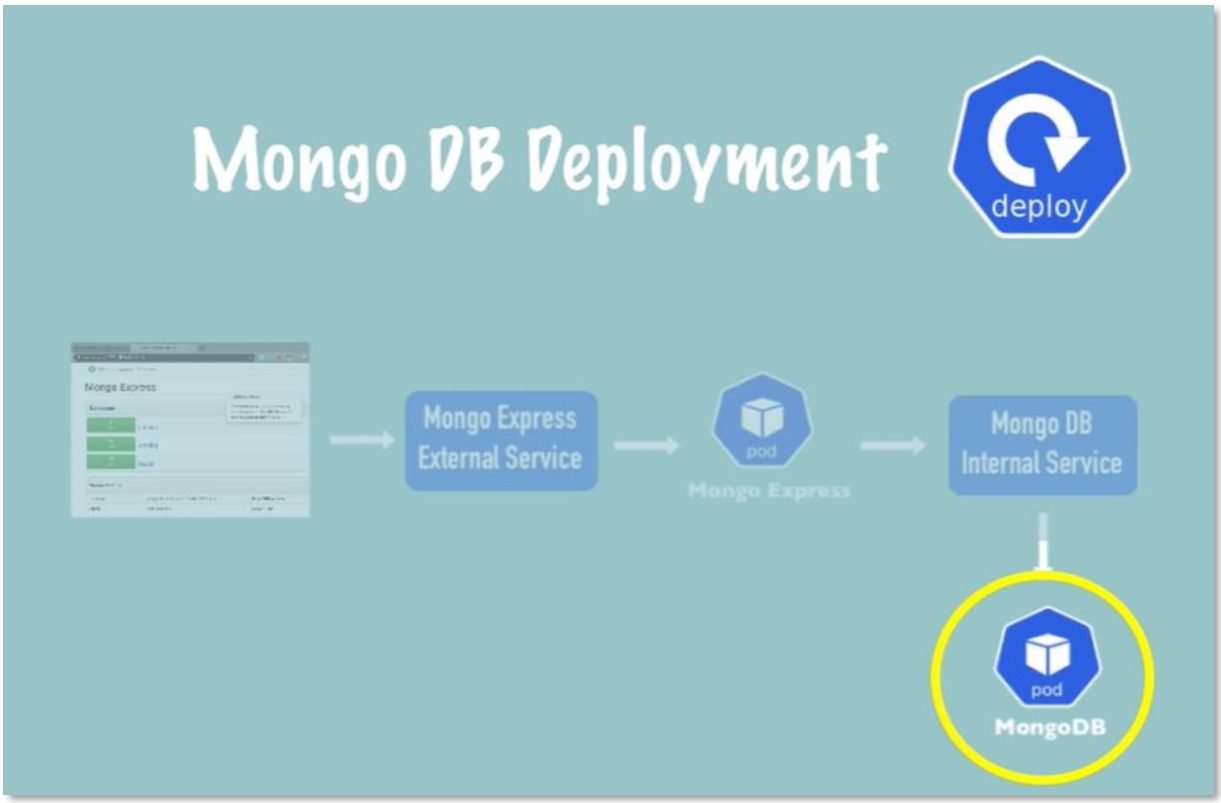


Enfin on a besoin de créer un service externe pour accéder au composant Mongo Express

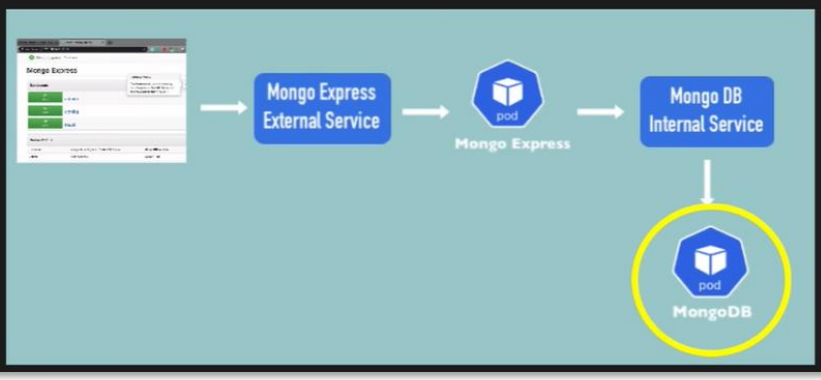





Browser Request Flow
through the K8s components
Mongo Express
Mongo Express
External Service
pod
Mongo Express
Mongo DB
Internal Service
Secret
DB User
DB Pwd
pod
MongoDB


Mongo DB Deployment
deploy
Mongo Express
External Service
pod
Mongo Express
Mongo DB
Internal Service
pod
MongoDB

```
[~]$ kubectl get all
NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)
  AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP
  10m
[~]$
```



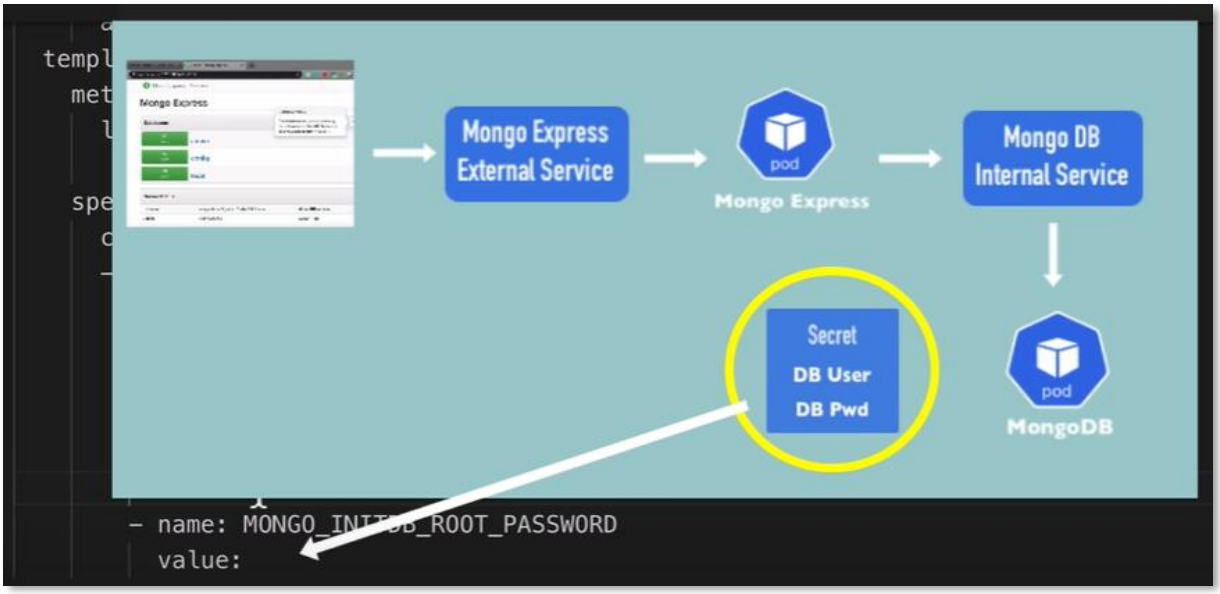

```
template:
  metadata:
    labels:
      app: mongodb
  spec:
    containers:
    - name: mongodb
      image: mongo
      ports:
      - containerPort: 27017
      env:
      - name: MONGO_INITDB_ROOT_USERNAME
        value:
      - name: MONGO_INITDB_ROOT_PASSWORD
        value:
```

Deployment Config File is checked into repository.

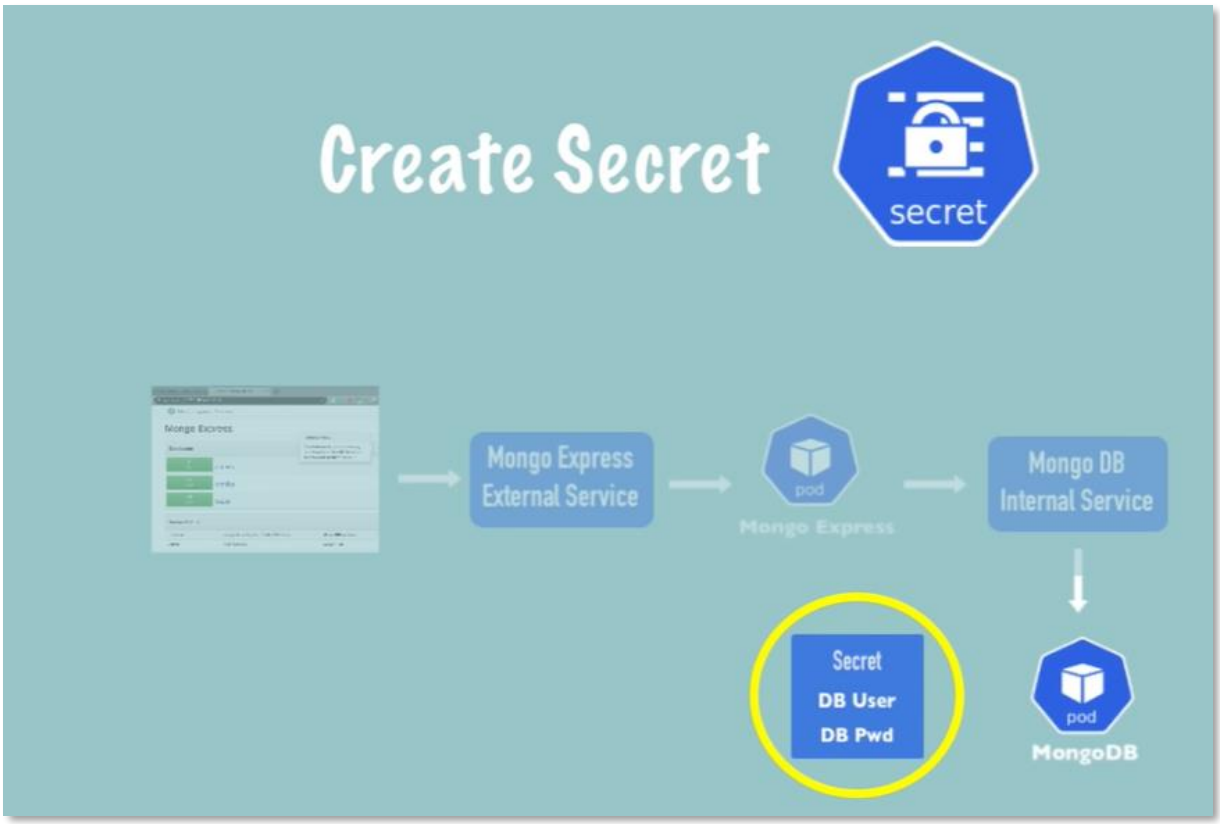
Username and Password should not go here!

Leave it blank

On va référencer ces valeurs de l'objet Secret



On créer ensuite l'objet Secret



mongo.yaml | mongo-secret.yaml

```yaml
apiVersion: v1
kind: Secret
metadata:
    name: mongodb-secret
type: Opaque
data:
    mongo-root-username: dXNlcm5hbWU=
    mongo-root-password: cGFzc3dvcmQ=
```

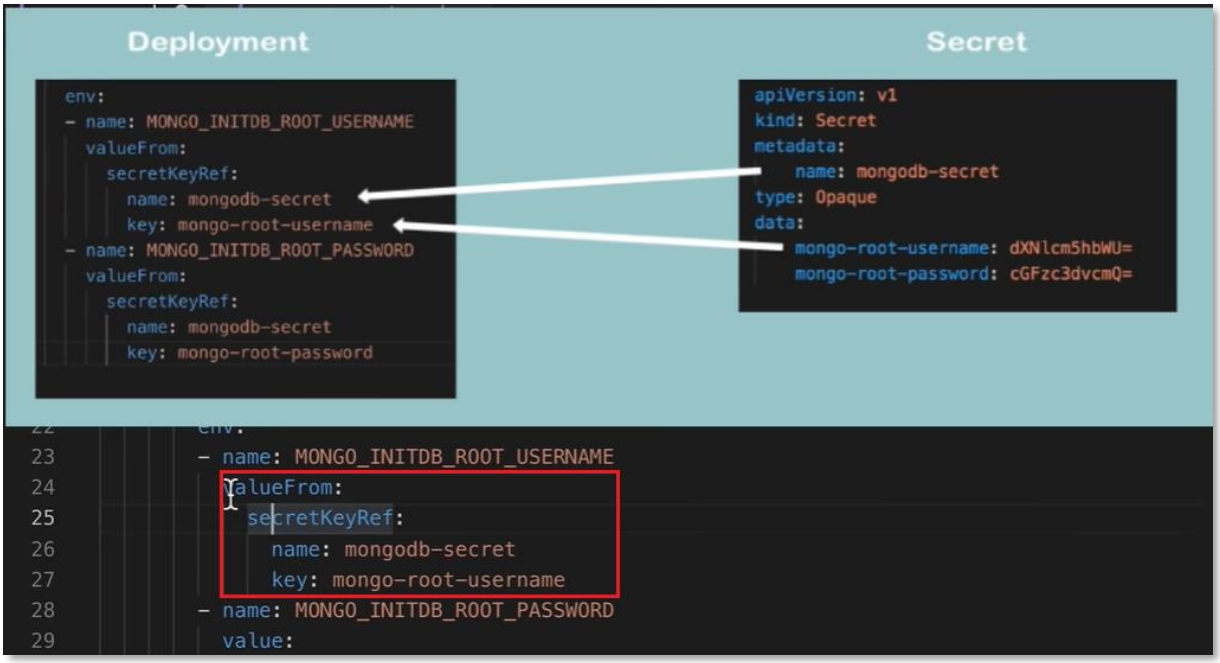
On va créer un secret ici(les user name et password doivent être crypté)

```
[~]$ kubectl get all
NAME                 TYPE
  AGE
service/kubernetes   ClusterIP
  10m
[~]$ echo -n 'username' | base64
dXNlcm5hbWU=
[~]$ echo -n 'password' | base64
cGFzc3dvcmQ=
```

L'ordre est très important, on va commencer par créer l'objet secret puis mongo-deploiment

```
[k8s-configuration]$ kubectl apply -f mongo-secret.yaml
secret/mongodb-secret created
```

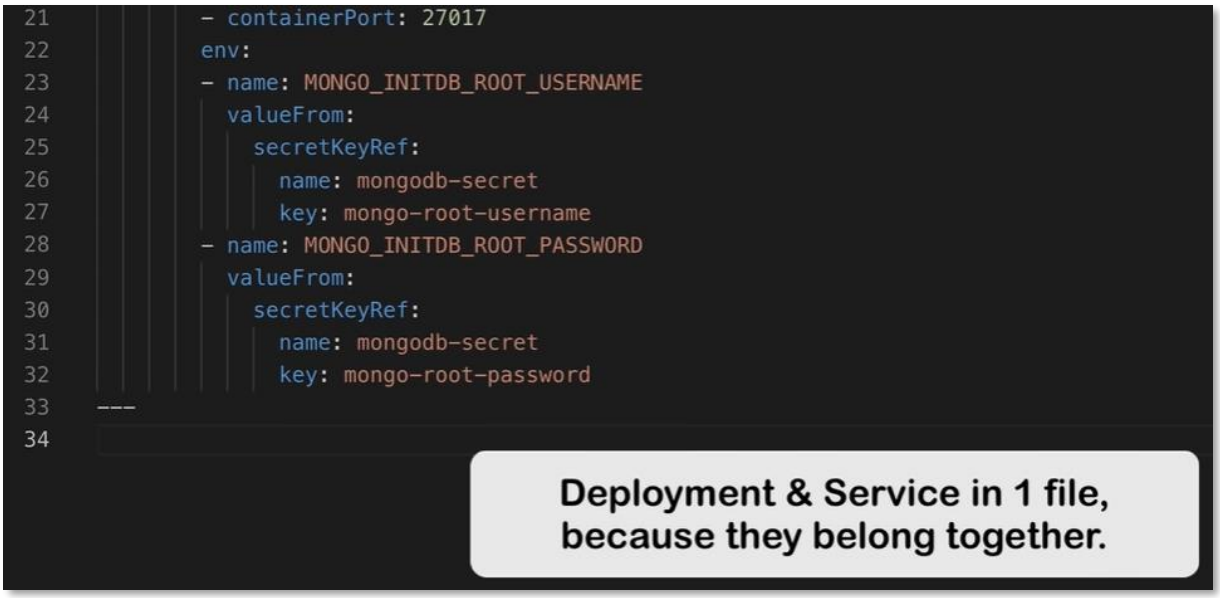
```
[k8s-configuration]$ kubectl get secret
NAME                  TYPE                                  DATA   AGE
default-token-4clzs   kubernetes.io/service-account-token   3      22m
mongodb-secret        Opaque                                2      68s
```



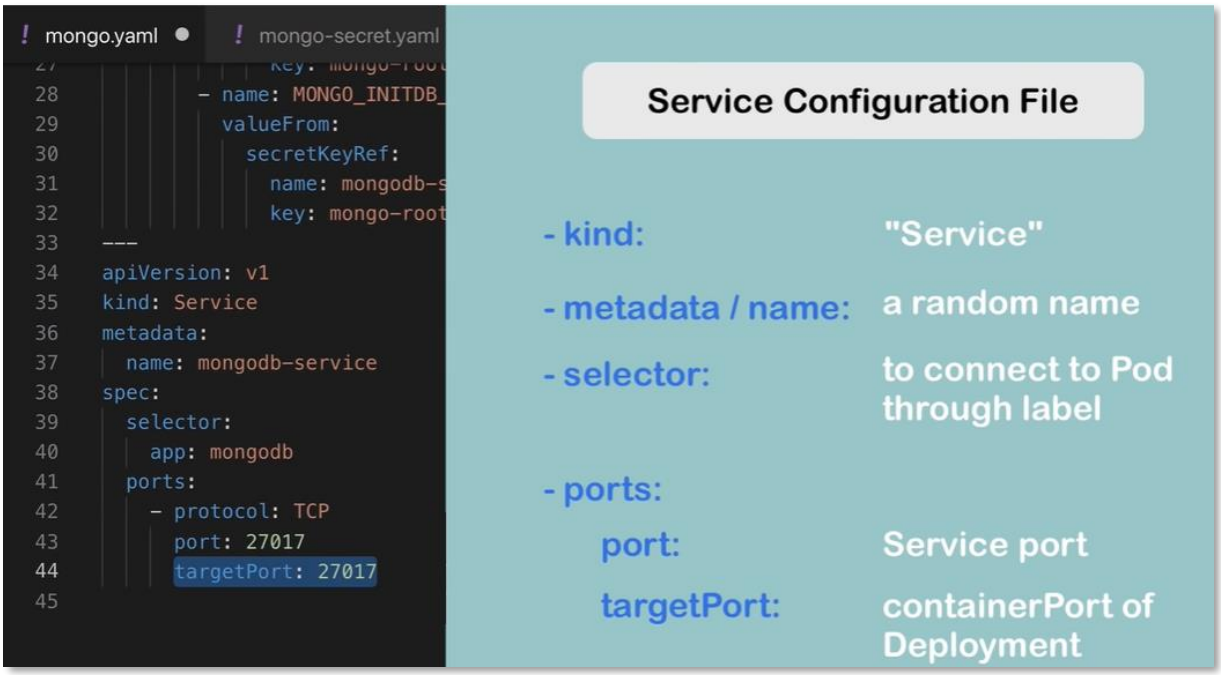

```
[k8s-configuration]$ kubectl get secret
NAME                  TYPE
default-token-4clzs   kubernetes.io/service-account-token
mongodb-secret        Opaque
[k8s-configuration]$ kubectl apply -f mongo.yaml
deployment.apps/mongodb-deployment created
```

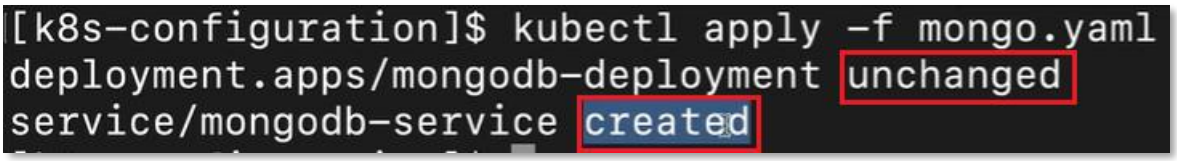
Etape suivante, création d'un service interne pour pouvoir joindre mongodb

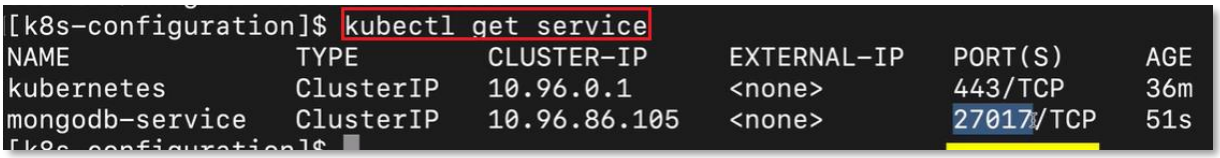
Mongo DB Internal Service
SVC
Mongo Express
External Service
pod
Mongo Express
Mongo DB
Internal Service
pod
MongoDB


- containerPort: 27017
env:
- name: MONGO_INITDB_ROOT_USERNAME
valueFrom:
secretKeyRef:
name: mongodb-secret
key: mongo-root-username
- name: MONGO_INITDB_ROOT_PASSWORD
valueFrom:
secretKeyRef:
name: mongodb-secret
key: mongo-root-password

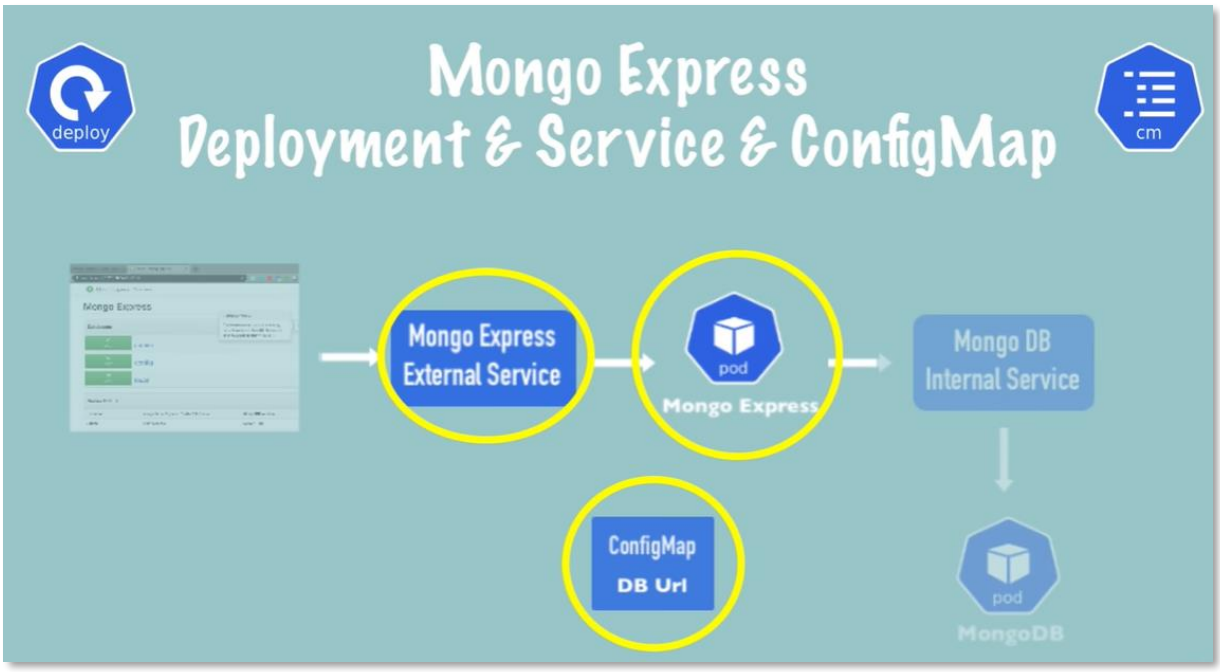
Deployment & Service in 1 file,
because they belong together.

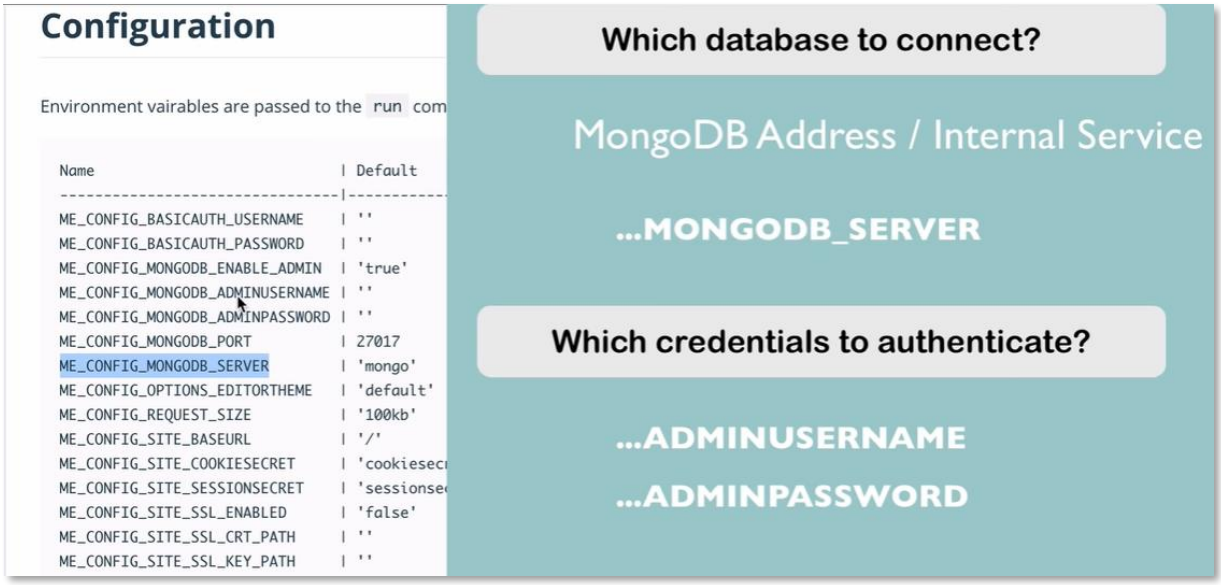

mongo.yaml
mongo-secret.yaml
28 - name: MONGO_INITDB_
29 valueFrom:
30 secretKeyRef:
31 name: mongodb-s
32 key: mongo-root
33 ---
34 apiVersion: v1
35 kind: Service
36 metadata:
37 name: mongodb-service
38 spec:
39 selector:
40 app: mongodb
41 ports:
42 - protocol: TCP
43 port: 27017
44 targetPort: 27017
45
Service Configuration File
- kind: "Service"
- metadata / name: a random name
- selector: to connect to Pod through label
- ports:
port: Service port
targetPort: containerPort of Deployment


[k8s-configuration]$ kubectl apply -f mongo.yaml
deployment.apps/mongodb-deployment unchanged
service/mongodb-service created

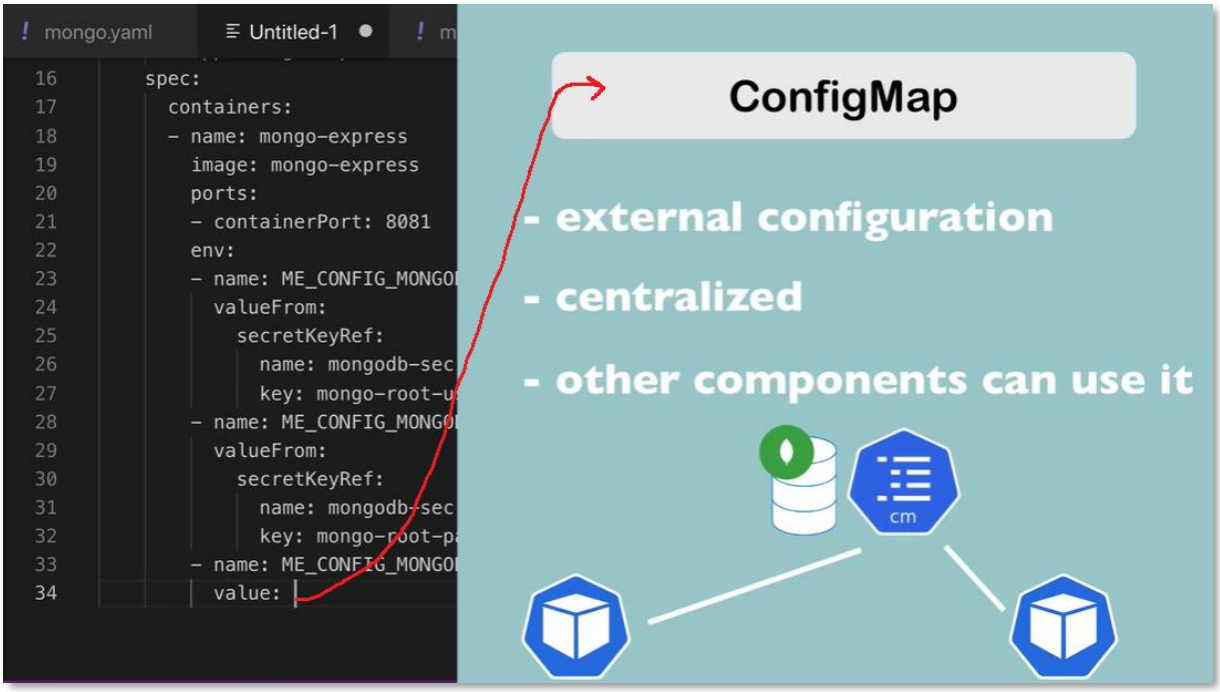

[k8s-configuration]$ kubectl get service
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 36m
mongodb-service ClusterIP 10.96.86.105 <none> 27017/TCP 51s

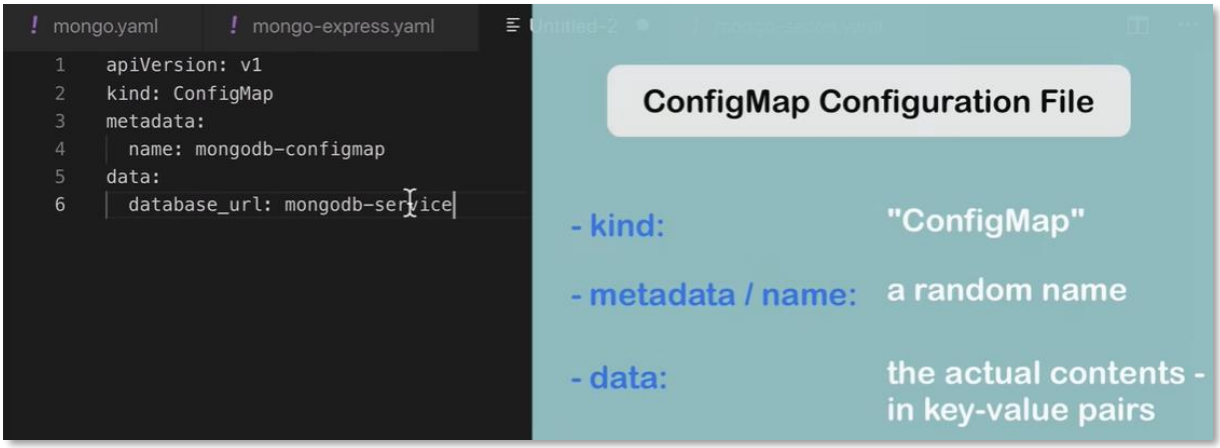
```
[k8s-configuration]$ kubectl describe service mongodb-service
Name:              mongodb-service
Namespace:         default
Labels:            <none>
Annotations:       kubectl.kubernetes.io/last-applied-configu
                     {"apiVersion":"v1","kind":"Service","meta
ns":{},"name":"mongodb-service","namespace":"default"},"spec"
:...
Selector:          app=mongodb
Type:              ClusterIP
IP:                10.96.86.105
Port:              <unset>  27017/TCP
TargetPort:        27017/TCP
Endpoints:         172.17.0.6:27017
Session Affinity:  None
```

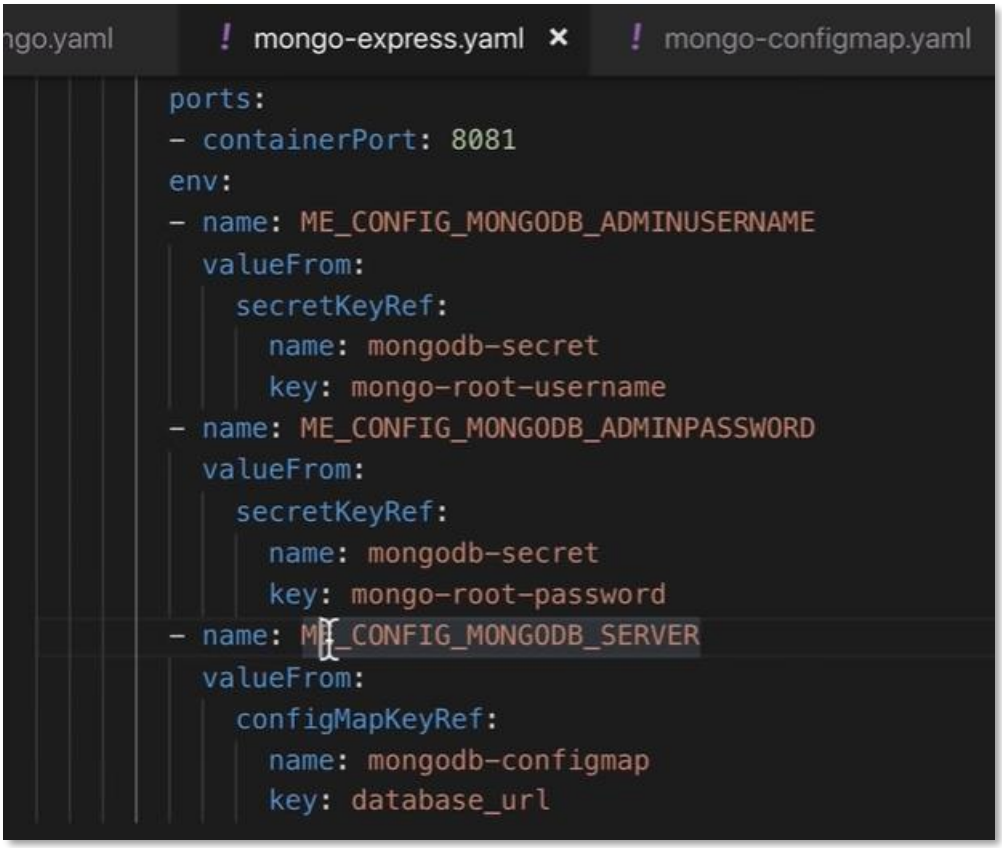




ConfigMap pour le stockage des Data centralisés




mongo.yaml
mongo-express.yaml
1 apiVersion: v1
2 kind: ConfigMap
3 metadata:
4 name: mongodb-configmap
5 data:
6 database_url: mongodb-service
ConfigMap Configuration File
- kind: "ConfigMap"
- metadata / name: a random name
- data: the actual contents - in key-value pairs


mongo-express.yaml
mongo-configmap.yaml
ports:
- containerPort: 8081
env:
- name: ME_CONFIG_MONGODB_ADMINUSERNAME
valueFrom:
secretKeyRef:
name: mongodb-secret
key: mongo-root-username
- name: ME_CONFIG_MONGODB_ADMINPASSWORD
valueFrom:
secretKeyRef:
name: mongodb-secret
key: mongo-root-password
- name: ME_CONFIG_MONGODB_SERVER
valueFrom:
configMapKeyRef:
name: mongodb-configmap
key: database_url

```
[k8s-configuration]$ kubectl apply -f mongo-configmap.yaml
configmap/mongodb-configmap created
[k8s-configuration]$ kubectl apply -f mongo-express.yaml
deployment.apps/mongo-express created
[k8s-configuration]$ kubectl get pod
NAME                                  READY   STATUS              RESTARTS
E
mongo-express-797845bd97-p9grr        0/1     ContainerCreating   0
mongodb-deployment-78444d94d6-zsrcl   1/1     Running             0
m
[k8s-configuration]$ kubectl get pod
NAME                                  READY   STATUS              RESTARTS
E
mongo-express-797845bd97-p9grr        0/1     ContainerCreating   0
s
mongodb-deployment-78444d94d6-zsrcl   1/1     Running             0
```

mongo.yaml | mongo-express.yaml

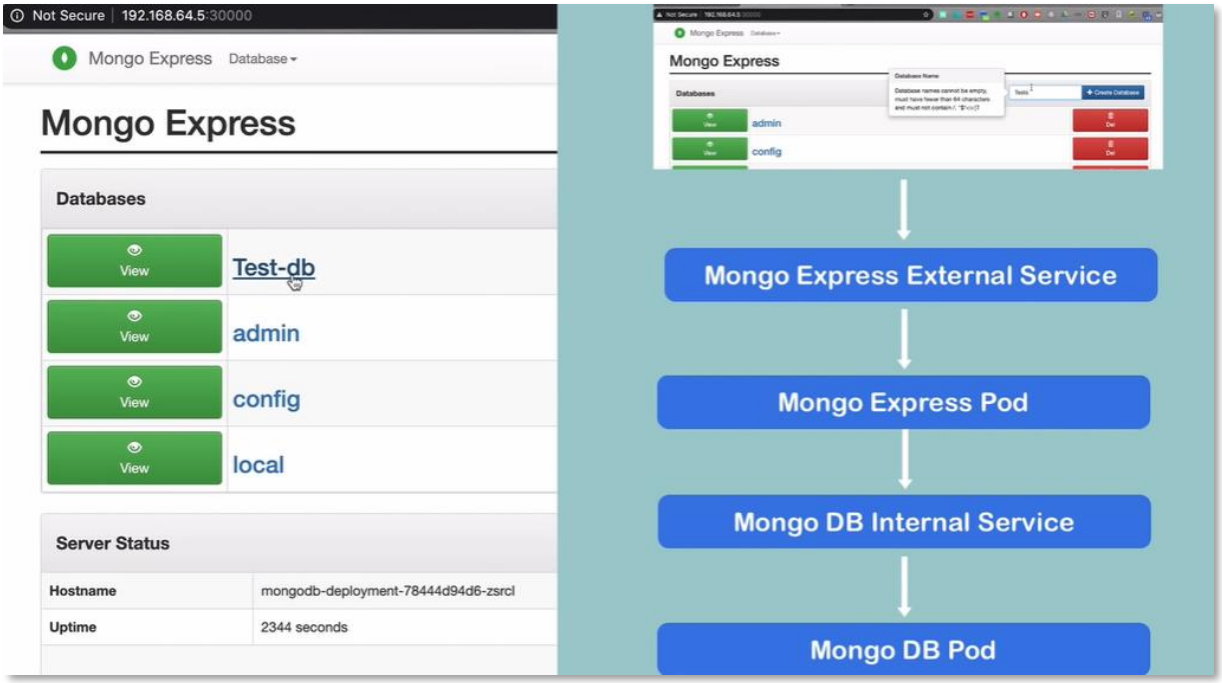
```
31                name: mongodb-secret
32                key: mongo-root-password
33          - name: ME_CONFIG_MONGODB_SERV
34            valueFrom:
35              configMapKeyRef:
36                name: mongodb-configmap
37                key: database_url
38  ---
39  apiVersion: v1
40  kind: Service
41  metadata:
42    name: mongo-express-service
43  spec:
44    selector:
45      app: mongo-express
46    type: LoadBalancer
47    ports:
48      - protocol: TCP
49        port: 8081
50        targetPort: 8081
51        nodePort: 30000
```

How to make it an External Service?

- type: "Loadbalancer"

..assigns service an external IP address and so accepts external requests

- nodePort: must be between 30000-32767


Not Secure | 192.168.64.5:30000
Mongo Express
Database
Mongo Express
Databases
View
Test-db
View
admin
View
config
View
local
Server Status
Hostname
mongodb-deployment-78444d94d6-zsrcl
Uptime
2344 seconds
Mongo Express External Service
Mongo Express Pod
Mongo DB Internal Service
Mongo DB Pod

Fin !