

1. Create ConfigMap or MongoDB EndPoint. (The MongoDB service name)

```
! ConfigMap.yml > ...  
1  apiVersion: v1  
2  kind: ConfigMap  
3  metadata:  
4    name: mongodb-configmap  
5  data:  
6    DB_URL: mongo-service  
7    clusterIP_name: mongo-svc  
8
```

2. Create A secret or MongoDB User & PWD

```
! mongo-secret.yml > {} data  
1  apiVersion: v1  
2  kind: Secret  
3  metadata:  
4    name: mongo-secret  
5  data:  
6    USER_NAME: bW9uZ291c2Vy  
7    PASSWORD: bW9uZ29wYXNzd29yZA==  
8
```

3. Create MongoDB Deployment Application with Internal service (ClusterIp) Mongo DB needs username + password to operate

```
mongo-svc.yml > {} spec > [ ] ports > {} 0 > # nodePort
1  apiVersion: v1
2  kind: mongo-service
3  metadata:
4    name: mongo-svc
5  spec:
6    type: clusterIP
7    selector:
8      matchLabels:
9        app: mongo-db
10   ports:
11     - port: 80
12       targetPort: 80
13       nodePort: 30007
```

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: mongodb_deploy
5    matchLabels:
6      app: mongodb
7  spec:
8    replicas: 3
9    selector:
10     matchLabels:
11       app: mongodb_pod
12    template:
13     metadata:
14       labels:
15         app: mongodb_pod
16     spec:
17       containers:
18         - name: my-mongo-pod
19           image: mongo:5.0
20       env:
21         - name: MONGO_INITDB_ROOT_USERNAME
22           valueFrom:
23             secretKeyRef:
24               name: mongo-secret
25               key: USER_NAME
26         - name: MONGO_INITDB_ROOT_PASSWORD
27           valueFrom:
28             secretKeyRef:
```

```
spec:
  containers:
  - name: my-mongo-pod
    image: mongo:5.0
  env:
  - name: MONGO_INITDB_ROOT_USERNAME
    valueFrom:
      secretkeyRef:
        name: mongo-secret
        key: USER_NAME
  - name: MONGO_INITDB_ROOT_PASSWORD
    valueFrom:
      secretkeyRef:
        name: mongo-secret
        key: PASSWORD
  envFrom:
  - configMapRef:
      name: mongodb-configmap
```

4. Create webApp Deployment (FrontEnd( with external service) and it needs to access MongoDB, so it needs username + password + mongodb endpoint (mongodb service) container runs on 3000

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: frontend_deploy
5    matchLabels:
6      app: frontend
7  spec:
8    replicas: 3
9    selector:
10     matchLabels:
11       app: frontend_pod
12   template:
13     metadata:
14       labels:
15         app: frontend_pod
16     spec:
17       containers:
18       - name: my-frontend-pod
19         image: nanajanashia/k8s-demo-app:v1.0
20         env:
21         - name: MONGO_INITDB_ROOT_USERNAME
22           valueFrom:
23             secretKeyRef:
24               name: mongo-secret
25               key: USER_NAME
26         - name: MONGO_INITDB_ROOT_PASSWORD
27           valueFrom:
28             secretKeyRef:
29               name: mongo-secret
30               key: PASSWORD
31         envFrom:
32         configMapRef:
33         mongodb-configmap
```

```
! frontend_deploy.yml    ! NodPort-svc.yml x
! NodPort-svc.yml > {} spec > # NodePort
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: NodePort-svc
5  spec:
6    type: NodePort
7    ports:
8      - port: 3000
9      targetPort: 3000
10   NodePort: 30007
```

8- How many Nodes exist on the system?

```
ahmed@ahmed-IdeaPad-Gaming-3-15ARH05:~$ kubectl get nodes
NAME          STATUS    ROLES    AGE   VERSION
minikube      Ready     control-plane  7d3h  v1.26.1
```

9- Do you see any taints on master?

```
ahmed@ahmed-IdeaPad-Gaming-3-15ARH05:~$ kubectl describe nodes minikube | grep Taint
Taints:                <none>
```

10- Apply a label color=blue to the master node

```
ahmed@ahmed-IdeaPad-Gaming-3-15ARH05:~$ kubectl taint nodes minikube color=blue:NoSchedule
node/minikube tainted
```

11- Create a new deployment named blue with the nginx image and 3 replicas

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: blue
5    Labels:
6      app: nginx
7  spec:
8    selector:
9      matchLabels:
10       app: nginx
11    replicas: 3
12    template:
13      metadata:
14        labels:
15          app: nginx
16      spec:
17        affinity:
18          nodeAffinity:
19            requiredDuringSchedulingIgnoredDuringExecution:
20              nodeSelectorTerms:
21                - matchExpressions:
22                  - key: color
23                    operator: In
24                    values:
25                      - blue
26      containers:
27        - name: nginx
28          image: nginx
```

```
nodeSelectorTerms
ahmed@ahmed-IdeaPad-Gaming-3-15ARH05:/media/ahmed/k/Ahmed/Sprints Tasks/K8s tasks/Ta
sk 3$ kubectl create -f blue.yml
deployment.apps/blue created
ahmed@ahmed-IdeaPad-Gaming-3-15ARH05:/media/ahmed/k/Ahmed/Sprints Tasks/K8s tasks/Ta
```

12- Create a taint on node01 with key of spray, value of mortein and effect of NoSchedule

```
ahmed@ahmed-IdeaPad-Gaming-3-15ARH05:/media/ahmed/k/Ahmed/Sprints Tasks/K8s tasks/Task 3$ kubectl taint node nod01 spray=mortein:NoSchedule
Error from server (NotFound): nodes "nod01" not found
ahmed@ahmed-IdeaPad-Gaming-3-15ARH05:/media/ahmed/k/Ahmed/Sprints Tasks/K8s tasks/Task 3$
```

13- Create a new pod with the NGINX image, and Pod name as mosquito

```
ahmed@ahmed-IdeaPad-Gaming-3-15ARH05:/media/ahmed/k/Ahmed/Sprints Tasks/K8s tasks/Task 3$ kubectl run mosquito --image=nginx
pod/mosquito created
ahmed@ahmed-IdeaPad-Gaming-3-15ARH05:/media/ahmed/k/Ahmed/Sprints Tasks/K8s tasks/Task 3$
```

14- What is the state of the mosquito POD?

```
ahmed@ahmed-IdeaPad-Gaming-3-15ARH05:/media/ahmed/k/Ahmed/Sprints Tasks/K8s tasks/Task 3$ kubectl get po mosquito
NAME        READY   STATUS    RESTARTS   AGE
mosquito    0/1     Pending   0           69s
ahmed@ahmed-IdeaPad-Gaming-3-15ARH05:/media/ahmed/k/Ahmed/Sprints Tasks/K8s tasks/Task 3$
```

15- Create another pod named bee with the NGINX image, which has a toleration set to

```
ahmed@ahmed-IdeaPad-Gaming-3-15ARH05:/media/ahmed/k/Ahmed/Sprints Tasks/K8s tasks/Task 3$ kubectl create -f bee.yml
pod/bee created
ahmed@ahmed-IdeaPad-Gaming-3-15ARH05:/media/ahmed/k/Ahmed/Sprints Tasks/K8s tasks/Task 3$
```



```
! bee.yml > {} spec > [ ] containers > {} 0 > image
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: bee
5  spec:
6    containers:
7      - name: nginx
8        image: nginx
9    tolerations:
10     - key: "spray"
11       operator: "Equal"
12       value: "mortein"
13       effect: "NoSchedule"
```

```
ahmed@ahmed-IdeaPad-Gaming-3-15ARH05:/media/ahmed/k/Ahmed/Sprints Tasks/K8s tasks/Task 3$ kubectl get po
```

NAME	READY	STATUS	RESTARTS	AGE
bee	0/1	Pending	0	87s