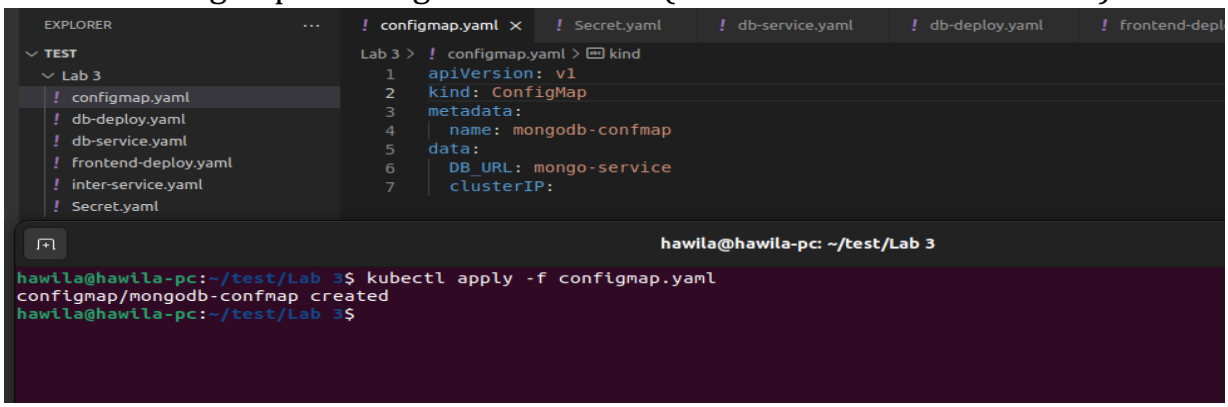
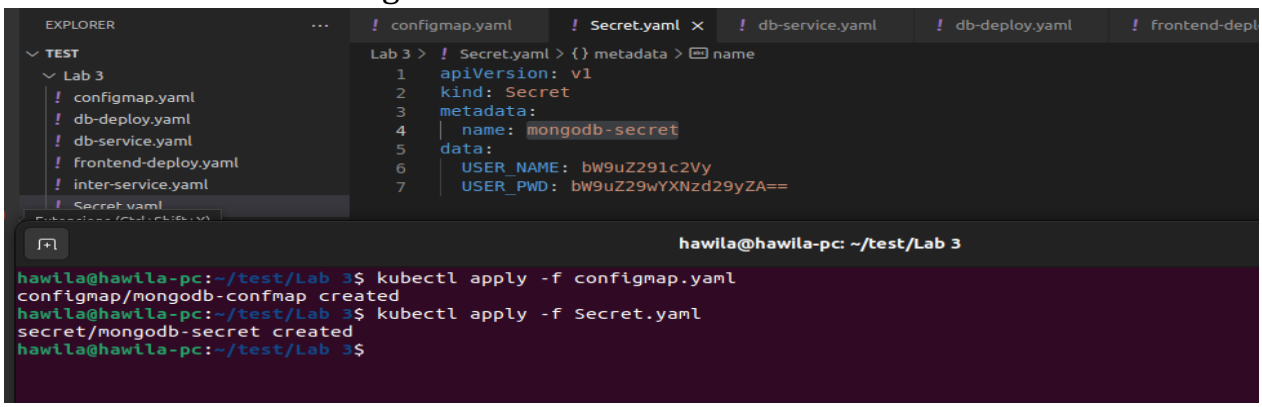
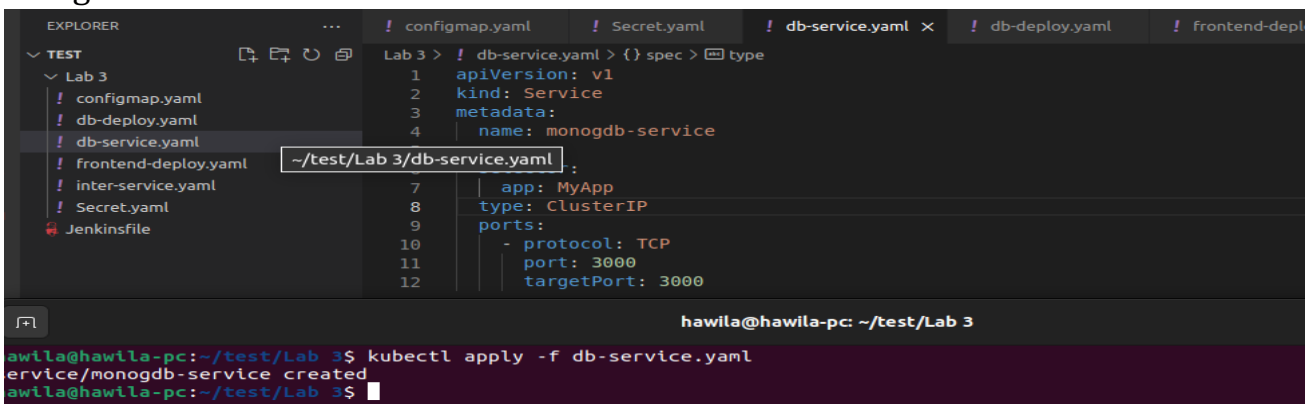
1 Create ConfgMap or MongoDB EndPoint. (The MondoDB sevice name)



2 Create A secret or MongoDB User & PWD



3 Create MongoDB Deployment Applicaton with Internal service (ClusterIp)
Mongo DB needs



Lab 3 > db-deploy.yaml > {} spec > {} template > {} spec > [ ] containers > {} 0 > image

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mongodb-deployment
  labels:
    app: mongo-app
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mongo-app
  template:
    metadata:
      labels:
        app: mongo-app
    spec:
      containers:
      - name: mongo-app
        image: mongo:5
        envFrom:
          - secretRef:
              name: mongodb-secret
        env:
         - name: MONGO_INITDB_ROOT_USERNAME
           value: root
         - name: MONGO_INITDB_ROOT_PASSWORD
           value: example
```

hawila@hawila-pc: ~/test/Lab 3

```
hawila@hawila-pc:~/test/Lab 3$ kubectl apply -f db-deploy.yaml
deployment.apps/mongodb-deployment created
hawila@hawila-pc:~/test/Lab 3$
```

4 Create webApp Deployment(FrontEnd(with external service) and it needs to access MongoDb,

Lab 3 > inter-service.yaml > {} spec > [ ] ports > {} 0 > # nodePort

```yaml
apiVersion: v1
kind: Service
metadata:
  name: frontend-service
spec:
  selector:
    app: frontend
  type: NodePort
  ports:
    - protocol: TCP
      port: 3000
      targetPort: 3000
      nodePort: 32000
```

hawila@hawila-pc: ~/test/Lab 3

```
hawila@hawila-pc:~/test/Lab 3$ kubectl apply -f inter-service.yaml
service/frontend-service created
hawila@hawila-pc:~/test/Lab 3$
```

Lab 3 > frontend-deploy.yaml > {} spec > {} template > {} spec > [ ] containers > {} 0 > [ ] envFrom

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: frontend
  labels:
    app: frontend
spec:
  replicas: 1
  selector:
    matchLabels:
      app: frontend
  template:
    metadata:
      labels:
        app: frontend
    spec:
      containers:
      - name: frontend-app
        image: nanajanashia/k8s-demo-app:v1.0
        envFrom:
          - secretRef:
              name: mongodb-secret
          - configMapRef:
              name: mongodb-confmap
```

hawila@hawila-pc: ~/test/Lab 3

```
hawila@hawila-pc:~/test/Lab 3$ kubectl apply -f frontend-deploy.yaml
deployment.apps/frontend created
hawila@hawila-pc:~/test/Lab 3$
```

```
hawila@hawila-pc: ~
hawila@hawila-pc:~$ kubectl get pods
NAME                                  READY   STATUS    RESTARTS   AGE
frontend-67dcddf4f9-dwkzc             1/1     Running   0          5h29m
mongodb-deployment-558db968d8-kmlwh   1/1     Running   0          5h50m
hawila@hawila-pc:~$ kubectl get deployment
NAME                 READY   UP-TO-DATE   AVAILABLE   AGE
frontend             1/1     1            1           5h45m
mongodb-deployment   1/1     1            1           5h50m
hawila@hawila-pc:~$ kubectl get service
NAME                 TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
frontend-service     NodePort    10.104.65.39    <none>        3000:32000/TCP   5h45m
kubernetes           ClusterIP   10.96.0.1       <none>        443/TCP          31h
monogdb-service      ClusterIP   10.104.2.244    <none>        3000/TCP         5h55m
hawila@hawila-pc:~$
```

```
hawila@hawila-pc: ~
        document.getElementById('input-name').value = document.getElementById('name').textContent;
        document.getElementById('input-email').value = document.getElementById('email').textContent;
        document.getElementById('input-interests').value = document.getElementById('interests').textContent;

        cont.style.display = 'none';
        contEdit.style.display = 'block';
    }
</script>
<body>
    <div class='container' id='container'>
        <h1>User profile</h1>
        <img src='profile-picture' alt="user-profile">
        <span>Name: </span><h3 id='name'>Anna Smith</h3>
        <hr />
        <span>Email: </span><h3 id='email'>anna.smith@example.com</h3>
        <hr />
        <span>Interests: </span><h3 id='interests'>coding</h3>
        <hr />
        <button class='button' onclick="updateProfile()">Edit Profile</button>
    </div>
    <div class='container' id='container-edit'>
        <h1>User profile</h1>
        <img src='profile-picture' alt="user-profile">
        <span>Name: </span><label for='input-name'></label><input type="text" id='input-name' value='Anna Smith' />
        <hr />
        <span>Email: </span><label for='input-email'></label><input type="email" id='input-email' value='anna.smith@example.com' />
        <hr />
        <span>Interests: </span><label for='input-interests'></label><input type="text" id='input-interests' value='coding' />
        <hr />
        <button class='button' onclick="handleUpdateProfileRequest()">Update Profile</button>
    </div>
</body>
hawila@hawila-pc:~$
```

8- How many Nodes exist on the system?

```
Editor   Tab 1   +                                                    60 min
Initialising Kubernetes... done

controlplane $ kubectl get nodes
NAME           STATUS   ROLES           AGE     VERSION
controlplane   Ready    control-plane   6d10h   v1.26.0
node01         Ready    <none>          6d9h    v1.26.0
controlplane $
```

9- Do you see any taints on master ?

```
Editor   Tab 1   +                                                    56 min
controlplane $ kubectl describe node controlplane
Name:               controlplane
Roles:              control-plane
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=controlplane
                    kubernetes.io/os=linux
                    node-role.kubernetes.io/control-plane=
                    node.kubernetes.io/exclude-from-external-load-balancers=
Annotations:        flannel.alpha.coreos.com/backend-data: {"VNI":1,"VtepMAC":"da:72:3a:23:c1:8b"}
                    flannel.alpha.coreos.com/backend-type: vxlan
                    flannel.alpha.coreos.com/kube-subnet-manager: true
                    flannel.alpha.coreos.com/public-ip: 172.30.1.2
                    kubeadm.alpha.kubernetes.io/cri-socket: unix:///var/run/containerd/containerd.sock
                    node.alpha.kubernetes.io/ttl: 0
                    projectcalico.org/IPv4Address: 172.30.1.2/24
                    projectcalico.org/IPv4IPIPTunnelAddr: 192.168.0.1
                    volumes.kubernetes.io/controller-managed-attach-detach: true
CreationTimestamp:  Thu, 26 Jan 2023 14:24:45 +0000
Taints:             <none>
Unschedulable:      false
Lease:
  HolderIdentity:  controlplane
  AcquireTime:     <unset>
  RenewTime:       Thu, 02 Feb 2023 00:33:20 +0000
Conditions:
  Type                 Status  LastHeartbeatTime                 LastTransitionTime                Reason                       Message
  ----                 ------  -----------------                 ------------------                ------                       -------
  NetworkUnavailable   False   Thu, 02 Feb 2023 00:01:05 +0000   Thu, 02 Feb 2023 00:01:05 +0000   FlannelIsUp                  Flannel is r
unning on this node
  MemoryPressure       False   Thu, 02 Feb 2023 00:33:00 +0000   Thu, 26 Jan 2023 14:24:39 +0000   KubeletHasSufficientMemory   kubelet has
sufficient memory available
  DiskPressure         False   Thu, 02 Feb 2023 00:33:00 +0000   Thu, 26 Jan 2023 14:24:39 +0000   KubeletHasNoDiskPressure     kubelet has
```

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

```
Editor   Tab 1   +                                                    55 min
controlplane $ kubectl label node controlplane color=blue
node/controlplane labeled
controlplane $ kubectl describe node controlplane
Name:               controlplane
Roles:              control-plane
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    color=blue
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=controlplane
```

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

```
Editor   Tab 1   +                                                                    46 min
apiVersion: apps/v1
kind: Deployment
metadata:
  name: blue
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
              - matchExpressions:
              - key: color
              operator: In
              values:
                - blue
~
~
~
"deploy.yaml" 30L, 589C
```

```
Editor   Tab 1   +                                                                    43 min
controlplane $ vim deploy.yaml
controlplane $ kubectl apply -f deploy.yaml
deployment.apps/blue created
controlplane $ kubectl get pods -o wide
NAME                    READY   STATUS    RESTARTS   AGE   IP            NODE           NOMINATED NODE   READINESS GATES
blue-86bf6f4d74-2qhqd   1/1     Running   0          55s   192.168.0.7   controlplane   <none>           <none>
blue-86bf6f4d74-rpcnp   1/1     Running   0          55s   192.168.0.8   controlplane   <none>           <none>
blue-86bf6f4d74-tbmgd   1/1     Running   0          55s   192.168.0.9   controlplane   <none>           <none>
controlplane $
```

12- node01 taint

```
Editor   Tab 1   +                                                                    36 min
controlplane $ kubectl taint node node01 spray=mortien:NoSchedule
node/node01 tainted
controlplane $ kubectl describe node01
error: the server doesn't have a resource type "node01"
controlplane $ kubectl describe node node01
Name:               node01
Roles:              <none>
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=node01
                    kubernetes.io/os=linux
Annotations:        flannel.alpha.coreos.com/backend-data: {"VNI":1,"VtepMAC":"62:54:3e:ac:c3:0e"}
                    flannel.alpha.coreos.com/backend-type: vxlan
                    flannel.alpha.coreos.com/kube-subnet-manager: true
                    flannel.alpha.coreos.com/public-ip: 172.30.2.2
                    kubeadm.alpha.kubernetes.io/cri-socket: unix:///var/run/containerd/containerd.sock
                    node.alpha.kubernetes.io/ttl: 0
                    projectcalico.org/IPv4Address: 172.30.2.2/24
                    projectcalico.org/IPv4IPIPTunnelAddr: 192.168.1.1
                    volumes.kubernetes.io/controller-managed-attach-detach: true
CreationTimestamp:  Thu, 26 Jan 2023 14:52:11 +0000
Taints:             spray=mortien:NoSchedule
Unschedulable:      false
Lease:
  HolderIdentity:  node01
  AcquireTime:     <unset>
  RenewTime:       Thu, 02 Feb 2023 00:53:17 +0000
Conditions:
  Type                 Status  LastHeartbeatTime                 LastTransitionTime                Reason                       Message
  ----                 ------  -----------------                 ------------------                ------                       -------
  NetworkUnavailable   False   Thu, 02 Feb 2023 00:01:04 +0000   Thu, 02 Feb 2023 00:01:04 +0000   FlannelIsUp                  Flannel is r
unning on this node
  MemoryPressure       False   Thu, 02 Feb 2023 00:51:13 +0000   Thu, 26 Jan 2023 14:52:11 +0000   KubeletHasSufficientMemory   kubelet has
```

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

```
Editor  Tab 1  +                                                    34 min
apiVersion: v1

kind: Pod

metadata:

  name: mosquito

spec:

  containers:

  - name: nginx

    image: nginx:1.14.2

    ports:

    - containerPort: 80
```

14- What is the state o the mosquito POD?

```
Editor  Tab 1  +                                                    33 min
controlplane $ vim pod.yaml
controlplane $ kubectl apply -f pod.yaml
pod/mosquito created
controlplane $ kubectl get pod mosquito
NAME       READY   STATUS    RESTARTS   AGE
mosquito   1/1     Running   0          40s
controlplane $
```

15- Create another pod named bee with the NGINX image

```
Editor  Tab 1  +                                                    27 min
apiVersion: v1
kind: Pod
metadata:
  name: bee
spec:
  containers:
  - name: nginx
    image: nginx:1.14.2
    ports:
    - containerPort: 80
  tolerations:
    - key: "spray"
      operator: "Exists"
      effect: "NoSchedule"
~
```

```
Editor  Tab 1  +                                                    26 min
controlplane $ kubectl apply -f pod.yaml
pod/bee created
controlplane $ kubectl get po -o wide
NAME                     READY   STATUS    RESTARTS   AGE     IP             NODE           NOMINATED NODE   READINESS GATES
bee                      1/1     Running   0          16s     192.168.1.3    node01         <none>           <none>
blue-86bf6f4d74-2qhqd    1/1     Running   0          17m     192.168.0.7    controlplane   <none>           <none>
blue-86bf6f4d74-rpcnp    1/1     Running   0          17m     192.168.0.8    controlplane   <none>           <none>
blue-86bf6f4d74-tbmgd    1/1     Running   0          17m     192.168.0.9    controlplane   <none>           <none>
mosquito                 1/1     Running   0          6m56s   192.168.0.10   controlplane   <none>           <none>
controlplane $
```