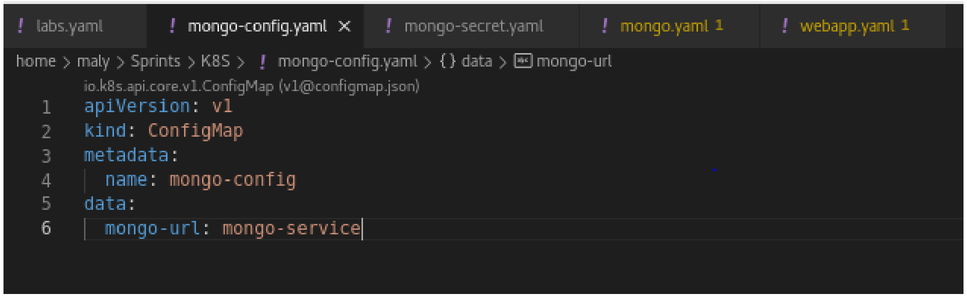
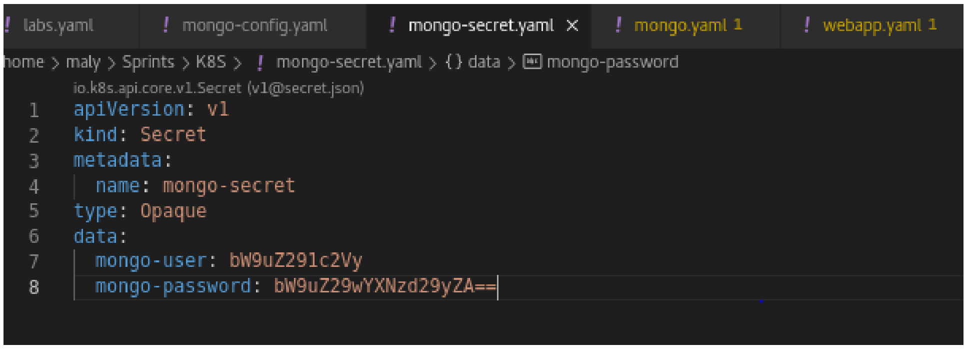
LAB3

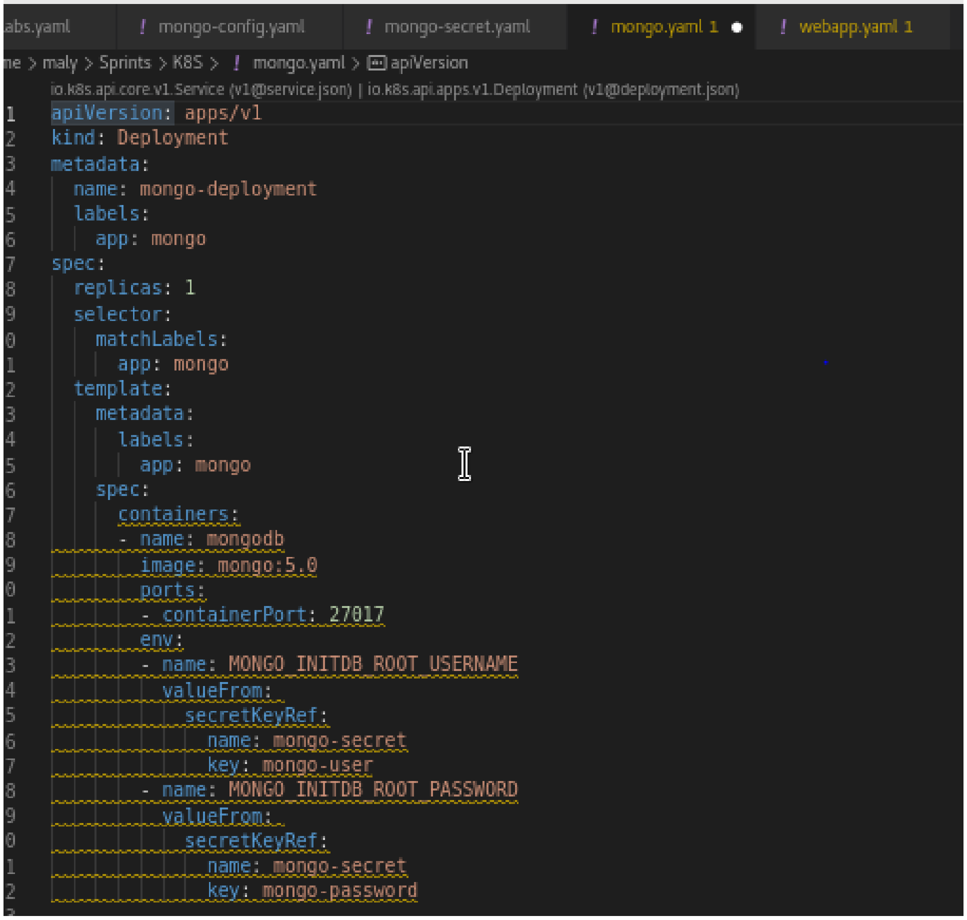
1....4) DEMO-mini project


! labs.yaml
! mongo-config.yaml ×
! mongo-secret.yaml
! mongo.yaml 1
! webapp.yaml 1
home > maly > Sprints > K8S > ! mongo-config.yaml > {} data > mongo-url
io.k8s.api.core.v1.ConfigMap (v1@configmap.json)

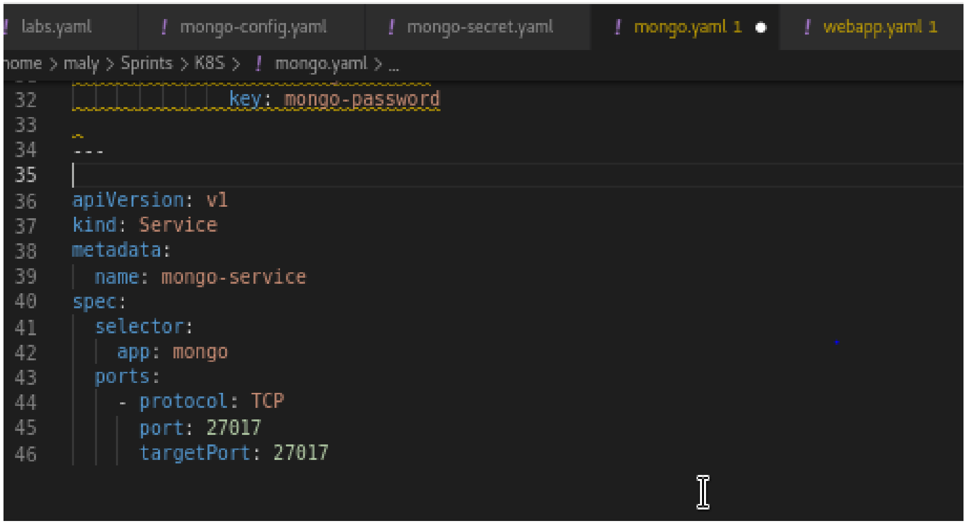
```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: mongo-config
data:
  mongo-url: mongo-service
```


! labs.yaml
! mongo-config.yaml
! mongo-secret.yaml ×
! mongo.yaml 1
! webapp.yaml 1
home > maly > Sprints > K8S > ! mongo-secret.yaml > {} data > mongo-password
io.k8s.api.core.v1.Secret (v1@secret.json)

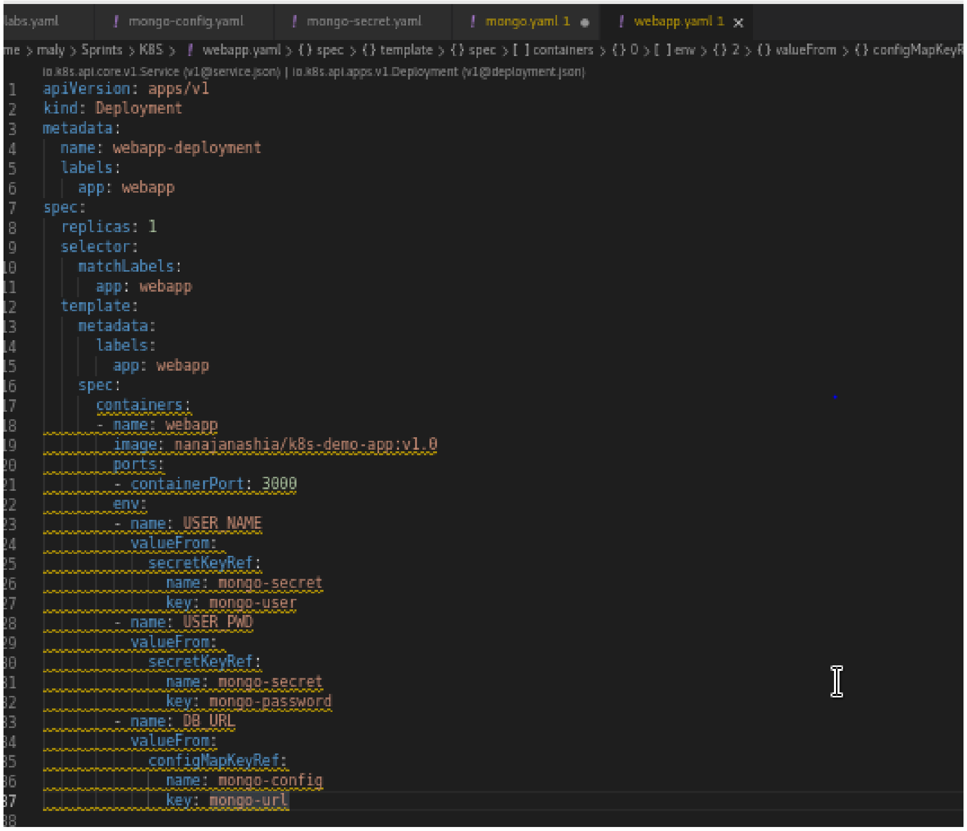
```yaml
apiVersion: v1
kind: Secret
metadata:
  name: mongo-secret
type: Opaque
data:
  mongo-user: bW9uZ291c2Vy
  mongo-password: bW9uZ29wYXNzd29yZA==
```


abs.yaml
mongo-config.yaml
mongo-secret.yaml
mongo.yaml 1
webapp.yaml 1
me > maly > Sprints > K8S > mongo.yaml > apiVersion
io.k8s.api.core.v1.Service (v1@service.json) | io.k8s.api.apps.v1.Deployment (v1@deployment.json)

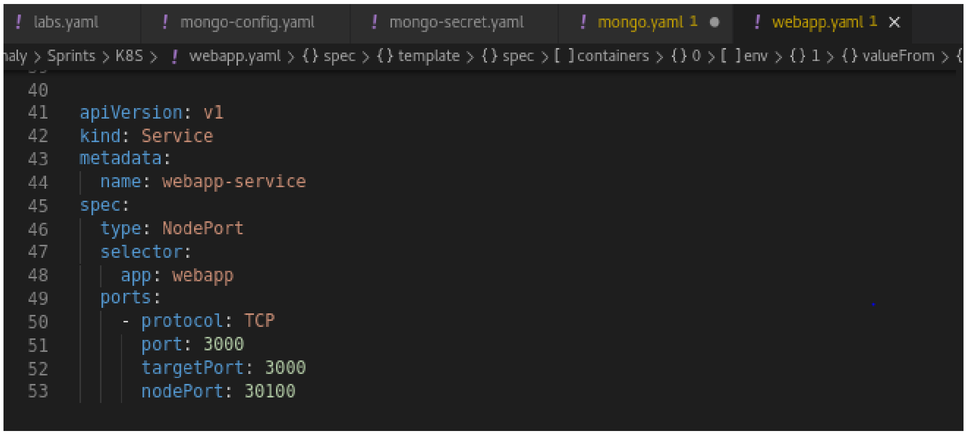
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mongo-deployment
  labels:
    app: mongo
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mongo
  template:
    metadata:
      labels:
        app: mongo
    spec:
      containers:
      - name: mongodb
        image: mongo:5.0
        ports:
        - containerPort: 27017
        env:
        - name: MONGO_INITDB_ROOT_USERNAME
          valueFrom:
            secretKeyRef:
              name: mongo-secret
              key: mongo-user
        - name: MONGO_INITDB_ROOT_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mongo-secret
              key: mongo-password
```



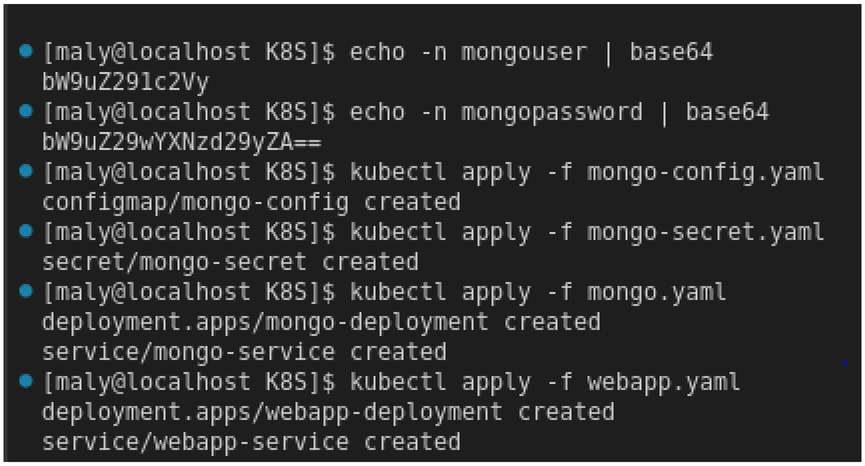
```yaml
              key: mongo-password

---

apiVersion: v1
kind: Service
metadata:
  name: mongo-service
spec:
  selector:
    app: mongo
  ports:
    - protocol: TCP
      port: 27017
      targetPort: 27017
```


labs.yaml
mongo-config.yaml
mongo-secret.yaml
mongo.yaml 1
webapp.yaml 1
me › maly › Sprints › K8S › webapp.yaml › { } spec › { } template › { } spec › [ ] containers › { } 0 › [ ] env › { } 2 › { } valueFrom › { } configMapKeyR
io.k8s.api.core.v1.Service (v1@service.json) | io.k8s.api.apps.v1.Deployment (v1@deployment.json)

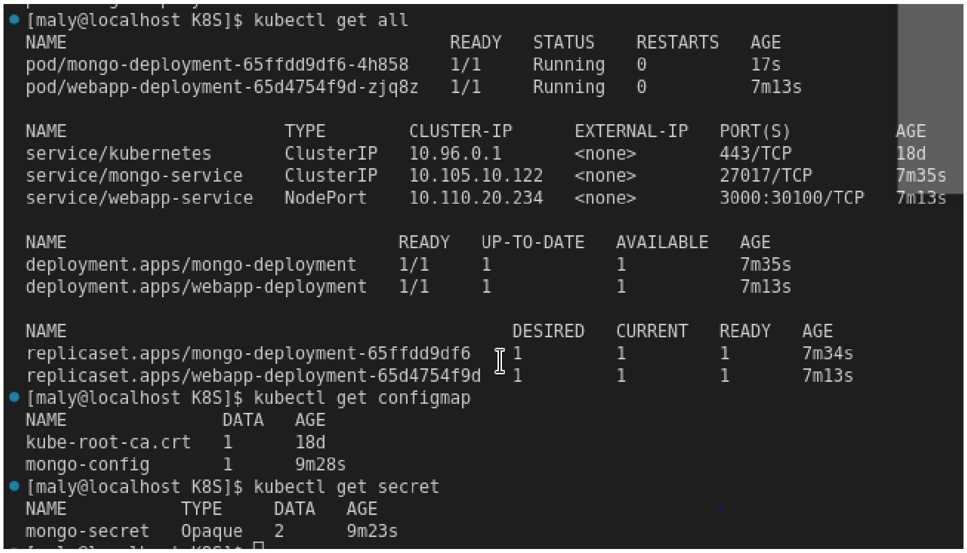
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: webapp-deployment
  labels:
    app: webapp
spec:
  replicas: 1
  selector:
    matchLabels:
      app: webapp
  template:
    metadata:
      labels:
        app: webapp
    spec:
      containers:
      - name: webapp
        image: nanajanashia/k8s-demo-app:v1.0
        ports:
        - containerPort: 3000
        env:
        - name: USER_NAME
          valueFrom:
            secretKeyRef:
              name: mongo-secret
              key: mongo-user
        - name: USER_PWD
          valueFrom:
            secretKeyRef:
              name: mongo-secret
              key: mongo-password
        - name: DB_URL
          valueFrom:
            configMapKeyRef:
              name: mongo-config
              key: mongo-url
```


labs.yaml
mongo-config.yaml
mongo-secret.yaml
mongo.yaml 1
webapp.yaml 1
naly › Sprints › K8S › webapp.yaml › { } spec › { } template › { } spec › [ ] containers › { } 0 › [ ] env › { } 1 › { } valueFrom › {

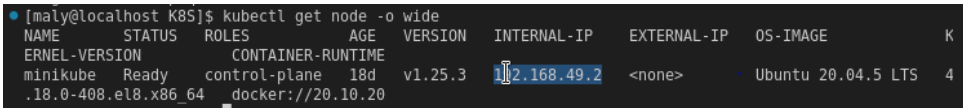
```yaml

apiVersion: v1
kind: Service
metadata:
  name: webapp-service
spec:
  type: NodePort
  selector:
    app: webapp
  ports:
    - protocol: TCP
      port: 3000
      targetPort: 3000
      nodePort: 30100
```



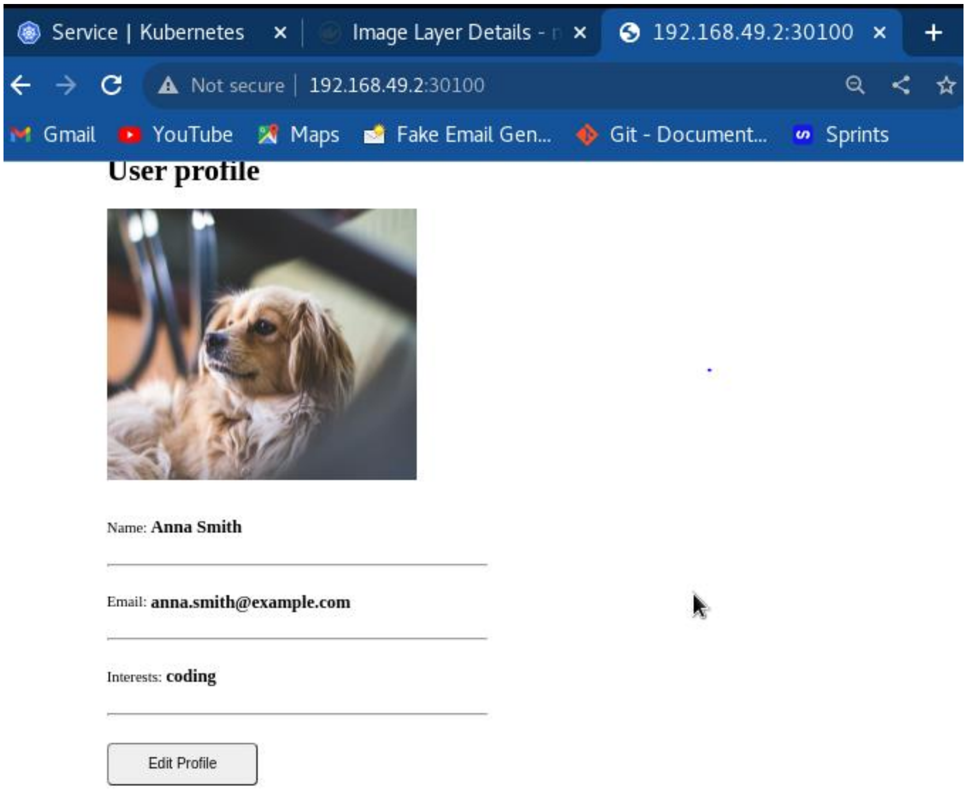
```
[maly@localhost K8S]$ echo -n mongouser | base64
bW9uZ291c2Vy
[maly@localhost K8S]$ echo -n mongopassword | base64
bW9uZ29wYXNzd29yZA==
[maly@localhost K8S]$ kubectl apply -f mongo-config.yaml
configmap/mongo-config created
[maly@localhost K8S]$ kubectl apply -f mongo-secret.yaml
secret/mongo-secret created
[maly@localhost K8S]$ kubectl apply -f mongo.yaml
deployment.apps/mongo-deployment created
service/mongo-service created
[maly@localhost K8S]$ kubectl apply -f webapp.yaml
deployment.apps/webapp-deployment created
service/webapp-service created
```



```
[maly@localhost K8S]$ kubectl get all
NAME                                    READY   STATUS    RESTARTS   AGE
pod/mongo-deployment-65ffdd9df6-4h858   1/1     Running   0          17s
pod/webapp-deployment-65d4754f9d-zjq8z  1/1     Running   0          7m13s

NAME                    TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
service/kubernetes      ClusterIP   10.96.0.1       <none>        443/TCP          18d
service/mongo-service   ClusterIP   10.105.10.122   <none>        27017/TCP        7m35s
service/webapp-service  NodePort    10.110.20.234   <none>        3000:30100/TCP   7m13s

NAME                               READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/mongo-deployment   1/1     1            1           7m35s
deployment.apps/webapp-deployment  1/1     1            1           7m13s

NAME                                          DESIRED   CURRENT   READY   AGE
replicaset.apps/mongo-deployment-65ffdd9df6   1         1         1       7m34s
replicaset.apps/webapp-deployment-65d4754f9d  1         1         1       7m13s
[maly@localhost K8S]$ kubectl get configmap
NAME               DATA   AGE
kube-root-ca.crt   1      18d
mongo-config       1      9m28s
[maly@localhost K8S]$ kubectl get secret
NAME           TYPE     DATA   AGE
mongo-secret   Opaque   2      9m23s
```



```
[maly@localhost K8S]$ kubectl get node -o wide
NAME       STATUS   ROLES           AGE   VERSION   INTERNAL-IP    EXTERNAL-IP   OS-IMAGE             K
ERNEL-VERSION          CONTAINER-RUNTIME
minikube   Ready    control-plane   18d   v1.25.3   192.168.49.2   <none>        Ubuntu 20.04.5 LTS   4
.18.0-408.el8.x86_64   docker://20.10.20
```



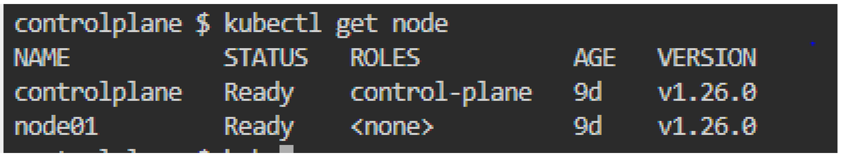
User profile

Name: **Anna Smith**

Email: **anna.smith@example.com**

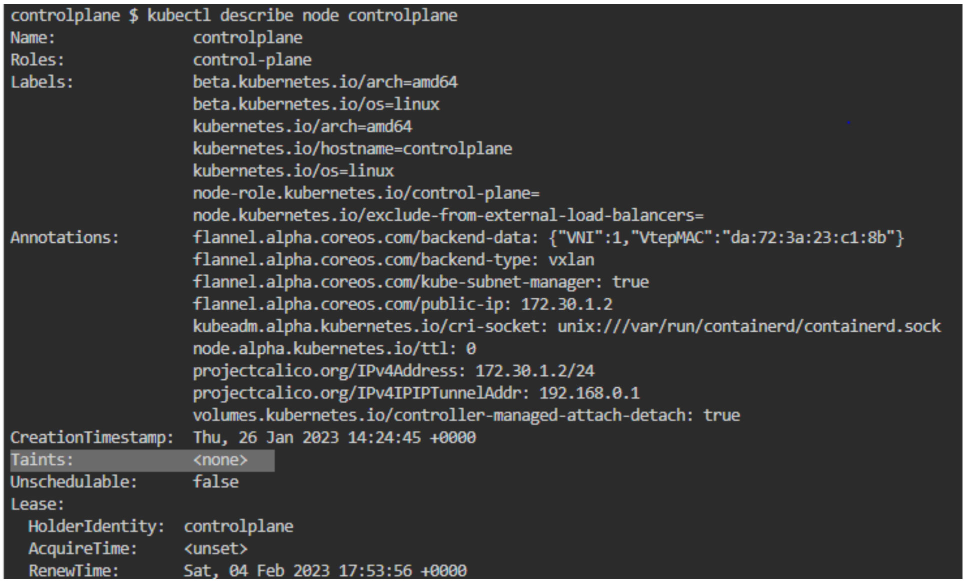
Interests: **coding**

8- How many Nodes exist on the system?



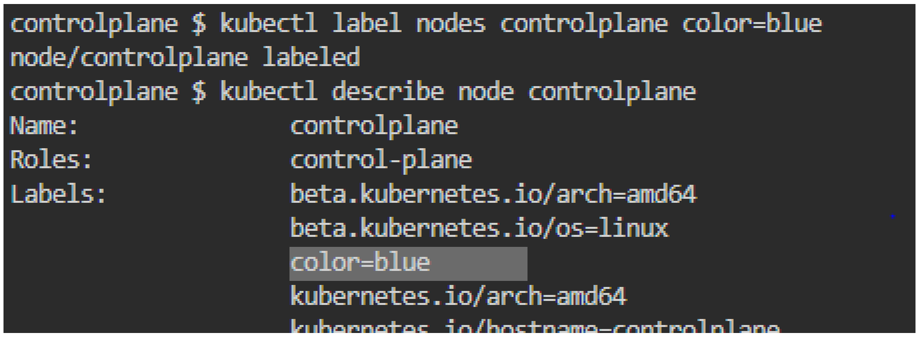
```
controlplane $ kubectl get node
NAME           STATUS   ROLES           AGE   VERSION
controlplane   Ready    control-plane   9d    v1.26.0
node01         Ready    <none>          9d    v1.26.0
```

9- Do you see any taints on master?



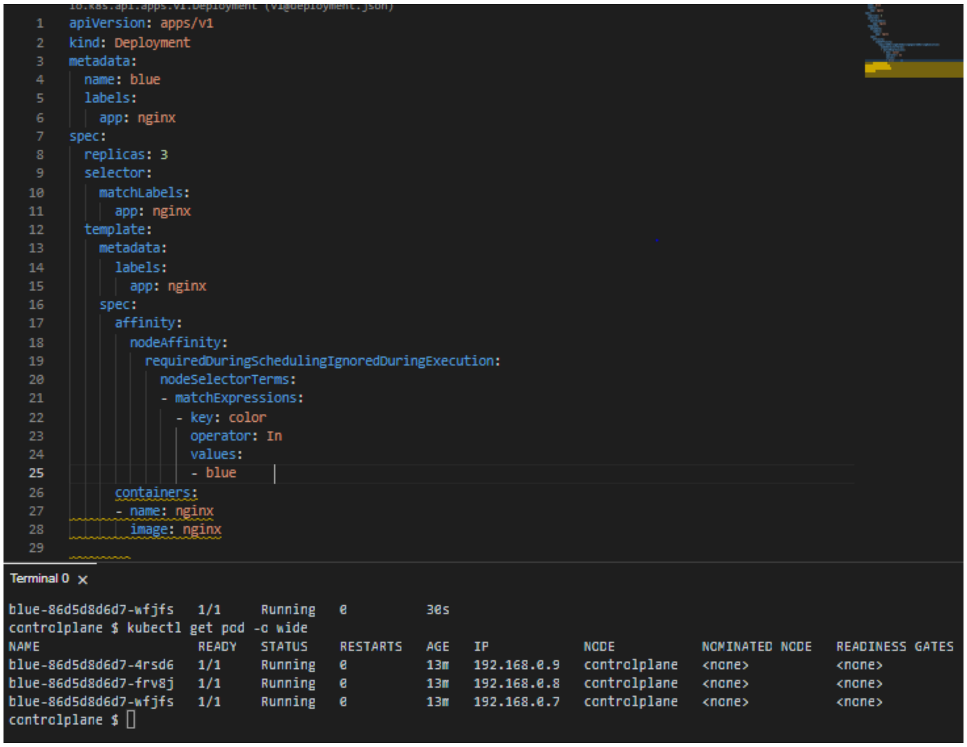
```
controlplane $ kubectl describe node controlplane
Name:               controlplane
Roles:              control-plane
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=controlplane
                    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:       Sat, 04 Feb 2023 17:53:56 +0000
```

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



```
controlplane $ kubectl label nodes 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
```

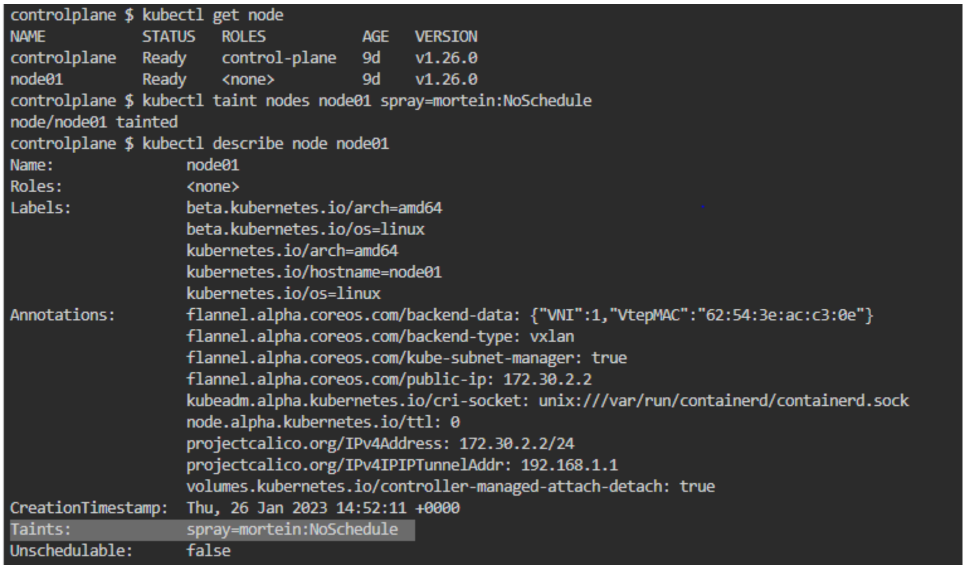
11- Create a new deployment named blue with the nginx image and 3 replicas Set Node Afnity to the deployment to place the pods on master only
NodeAfnity: requiredDuringSchedulingIgnoredDuringExecuton
Key: color values: blue



```
io.k8s.api.apps.v1.Deployment (v1@deployment.json)
apiVersion: apps/v1
kind: Deployment
metadata:
  name: blue
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
            - matchExpressions:
              - key: color
                operator: In
                values:
                - blue
      containers:
      - name: nginx
        image: nginx
```

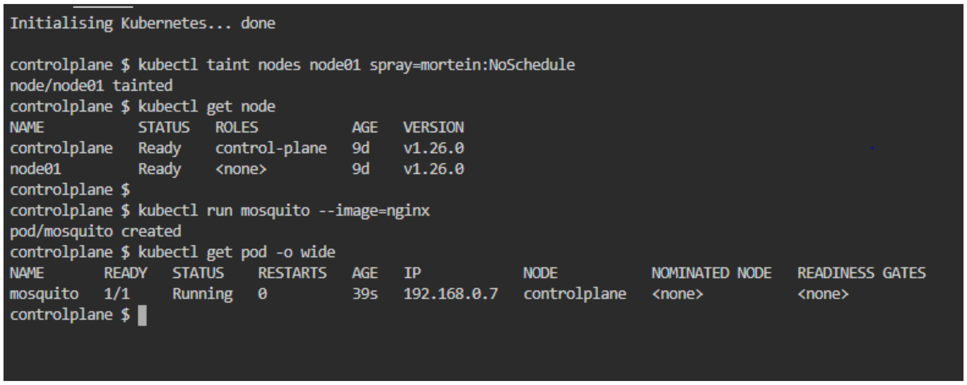
```
blue-86d5d8d6d7-wfjfs   1/1     Running   0          30s
controlplane $ kubectl get pod -o wide
NAME                    READY   STATUS    RESTARTS   AGE   IP            NODE           NOMINATED NODE   READINESS GATES
blue-86d5d8d6d7-4rsd6   1/1     Running   0          13m   192.168.0.9   controlplane   <none>           <none>
blue-86d5d8d6d7-frv8j   1/1     Running   0          13m   192.168.0.8   controlplane   <none>           <none>
blue-86d5d8d6d7-wfjfs   1/1     Running   0          13m   192.168.0.7   controlplane   <none>           <none>
controlplane $ 
```

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



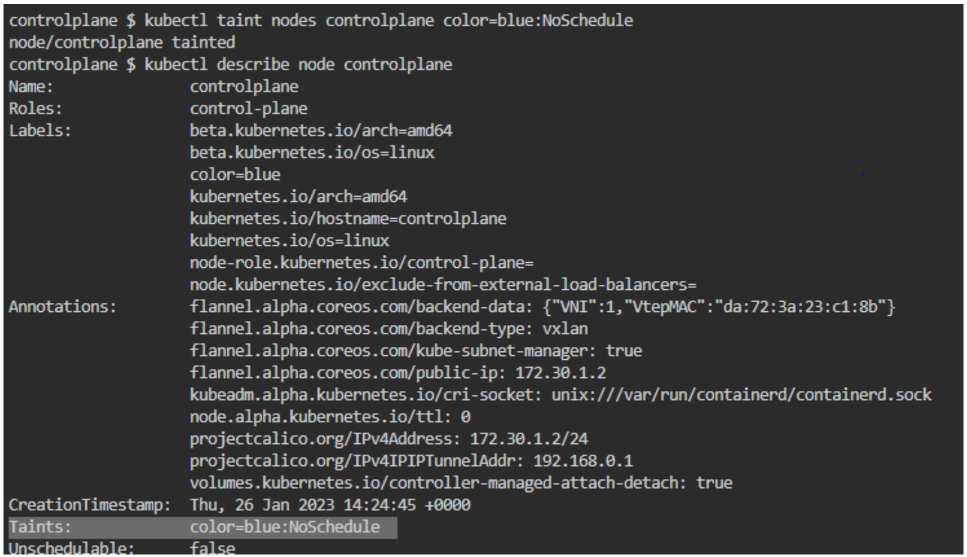
```
controlplane $ kubectl get node
NAME           STATUS   ROLES           AGE   VERSION
controlplane   Ready    control-plane   9d    v1.26.0
node01         Ready    <none>          9d    v1.26.0
controlplane $ kubectl taint nodes node01 spray=mortein:NoSchedule
node/node01 tainted
controlplane $ kubectl 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=mortein:NoSchedule
Unschedulable:      false
```

13- Create a new pod with the NGINX image, and Pod name as mosquito
14- What is the state of the mosquito POD?

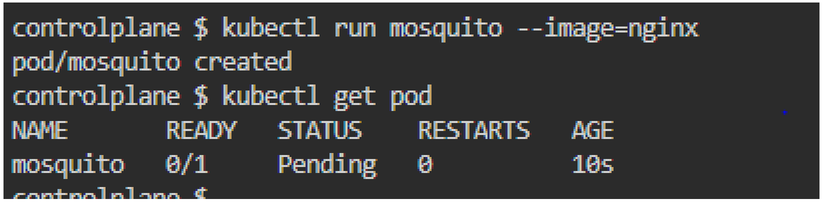


```
Initialising Kubernetes... done

controlplane $ kubectl taint nodes node01 spray=mortein:NoSchedule
node/node01 tainted
controlplane $ kubectl get node
NAME           STATUS   ROLES           AGE   VERSION
controlplane   Ready    control-plane   9d    v1.26.0
node01         Ready    <none>          9d    v1.26.0
controlplane $
controlplane $ kubectl run mosquito --image=nginx
pod/mosquito created
controlplane $ kubectl get pod -o wide
NAME       READY   STATUS    RESTARTS   AGE   IP            NODE           NOMINATED NODE   READINESS GATES
mosquito   1/1     Running   0          39s   192.168.0.7   controlplane   <none>           <none>
controlplane $
```

Another sol:

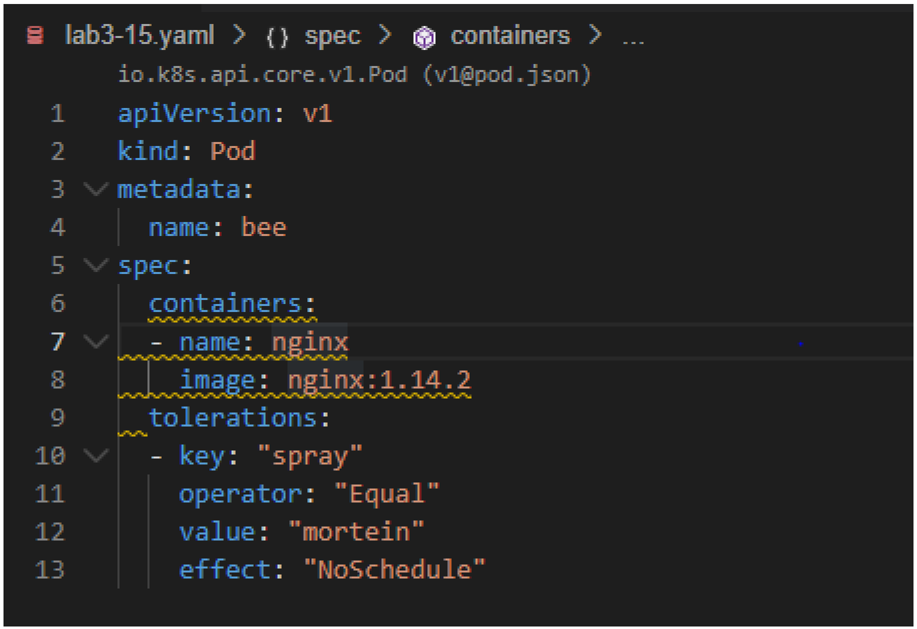


```
controlplane $ kubectl taint nodes controlplane color=blue:NoSchedule
node/controlplane tainted
controlplane $ kubectl describe node controlplane
Name:               controlplane
Roles:              control-plane
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    color=blue
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=controlplane
                    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:             color=blue:NoSchedule
Unschedulable:      false
```



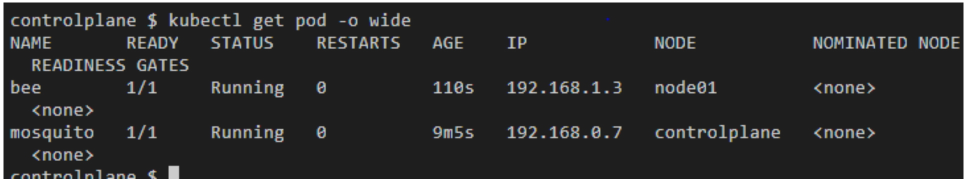
```
controlplane $ kubectl run mosquito --image=nginx
pod/mosquito created
controlplane $ kubectl get pod
NAME       READY   STATUS    RESTARTS   AGE
mosquito   0/1     Pending   0          10s
controlplane $
```

15- Create another pod named bee with the NGINX image, which has a toleraton set to the taint Mortein Image name: nginx Key: spray Value: mortein Efect: NoSchedule Status: Running



lab3-15.yaml > {} spec > containers > ...
io.k8s.api.core.v1.Pod (v1@pod.json)

```
apiVersion: v1
kind: Pod
metadata:
  name: bee
spec:
  containers:
  - name: nginx
    image: nginx:1.14.2
  tolerations:
  - key: "spray"
    operator: "Equal"
    value: "mortein"
    effect: "NoSchedule"
```



```
controlplane $ kubectl get pod -o wide
NAME       READY   STATUS    RESTARTS   AGE    IP            NODE           NOMINATED NODE
  READINESS GATES
bee        1/1     Running   0          110s   192.168.1.3   node01         <none>
  <none>
mosquito   1/1     Running   0          9m5s   192.168.0.7   controlplane   <none>
  <none>
controlplane $
```