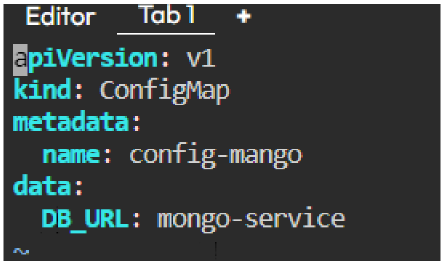
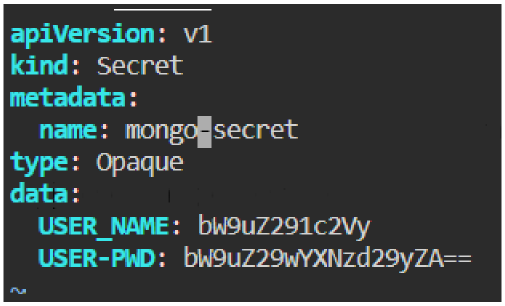
1- ConfigMap



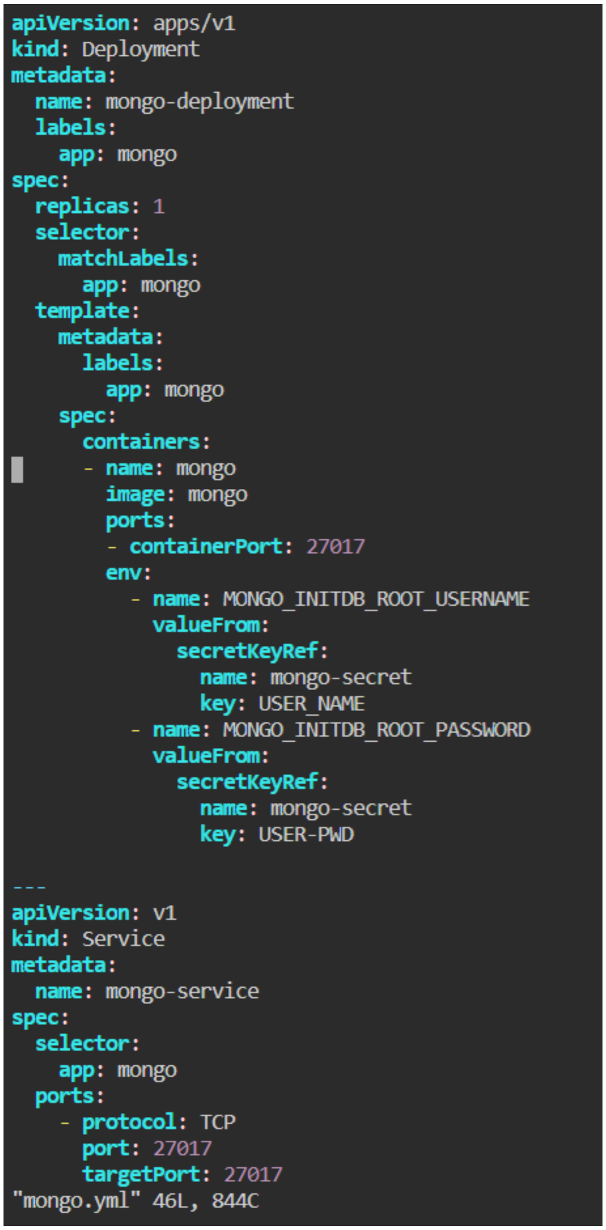
```
apiVersion: v1
kind: ConfigMap
metadata:
  name: config-mango
data:
  DB_URL: mongo-service
~
```

2- secret



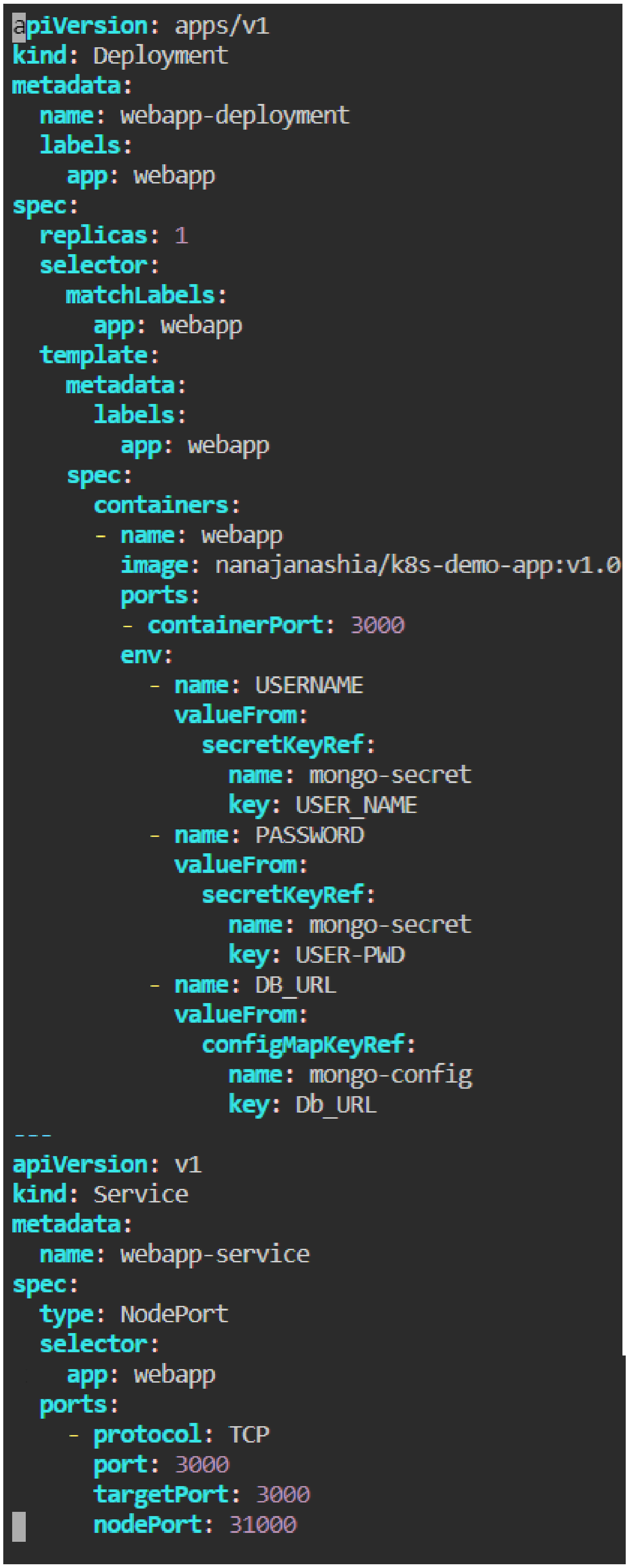
```
apiVersion: v1
kind: Secret
metadata:
  name: mongo-secret
type: Opaque
data:
  USER_NAME: bW9uZ291c2Vy
  USER-PWD: bW9uZ29wYXNzd29yZA==
~
```

3- MongoDB Deployment



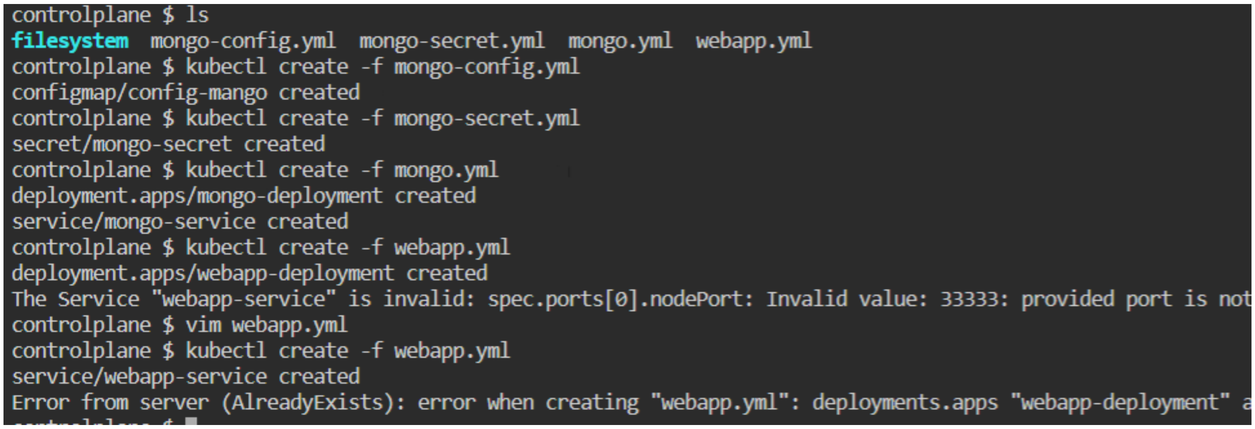
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mongo-deployment
  labels:
    app: mongo
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mongo
  template:
    metadata:
      labels:
        app: mongo
    spec:
      containers:
      - name: mongo
        image: mongo
        ports:
        - containerPort: 27017
        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: USER-PWD

---
apiVersion: v1
kind: Service
metadata:
  name: mongo-service
spec:
  selector:
    app: mongo
  ports:
    - protocol: TCP
      port: 27017
      targetPort: 27017
"mongo.yml" 46L, 844C
```

4- webapp dep



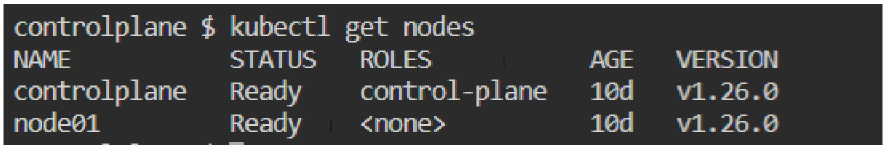
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: webapp-deployment
  labels:
    app: webapp
spec:
  replicas: 1
  selector:
    matchLabels:
      app: webapp
  template:
    metadata:
      labels:
        app: webapp
    spec:
      containers:
      - name: webapp
        image: nanajanashia/k8s-demo-app:v1.0
        ports:
        - containerPort: 3000
        env:
          - name: USERNAME
            valueFrom:
              secretKeyRef:
                name: mongo-secret
                key: USER_NAME
          - name: PASSWORD
            valueFrom:
              secretKeyRef:
                name: mongo-secret
                key: USER-PWD
          - name: DB_URL
            valueFrom:
              configMapKeyRef:
                name: mongo-config
                key: Db_URL
---
apiVersion: v1
kind: Service
metadata:
  name: webapp-service
spec:
  type: NodePort
  selector:
    app: webapp
  ports:
    - protocol: TCP
      port: 3000
      targetPort: 3000
      nodePort: 31000
```

applying deployments successfully



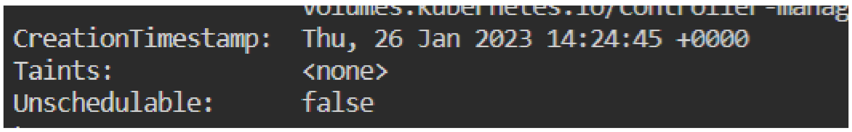
```
controlplane $ ls
filesystem  mongo-config.yml  mongo-secret.yml  mongo.yml  webapp.yml
controlplane $ kubectl create -f mongo-config.yml
configmap/config-mango created
controlplane $ kubectl create -f mongo-secret.yml
secret/mongo-secret created
controlplane $ kubectl create -f mongo.yml
deployment.apps/mongo-deployment created
service/mongo-service created
controlplane $ kubectl create -f webapp.yml
deployment.apps/webapp-deployment created
The Service "webapp-service" is invalid: spec.ports[0].nodePort: Invalid value: 33333: provided port is not
controlplane $ vim webapp.yml
controlplane $ kubectl create -f webapp.yml
service/webapp-service created
Error from server (AlreadyExists): error when creating "webapp.yml": deployments.apps "webapp-deployment" a
```

8- there are two nodes on the system



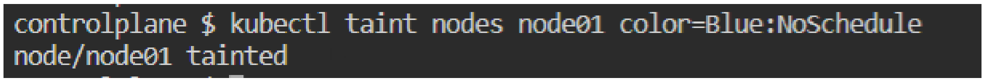
```
controlplane $ kubectl get nodes
NAME           STATUS   ROLES           AGE   VERSION
controlplane   Ready    control-plane   10d   v1.26.0
node01         Ready    <none>          10d   v1.26.0
```

9- there is no taints



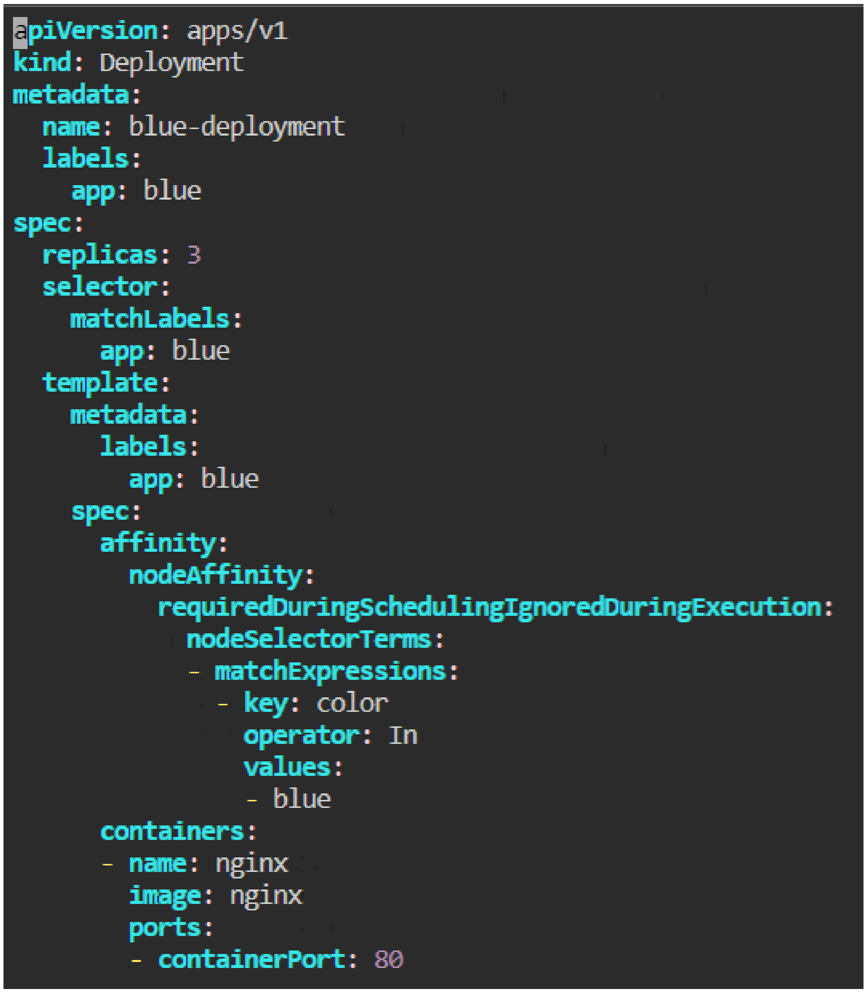
```
                    volumes.kubernetes.io/controller-manag
CreationTimestamp:  Thu, 26 Jan 2023 14:24:45 +0000
Taints:             <none>
Unschedulable:      false
```

10- tainting node



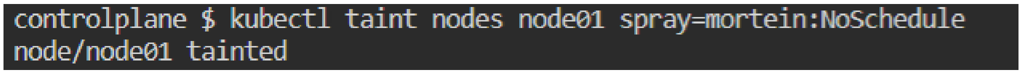
```
controlplane $ kubectl taint nodes node01 color=Blue:NoSchedule
node/node01 tainted
```

11-



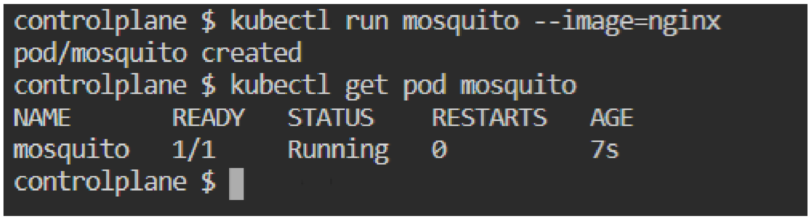
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: blue-deployment
  labels:
    app: blue
spec:
  replicas: 3
  selector:
    matchLabels:
      app: blue
  template:
    metadata:
      labels:
        app: blue
    spec:
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
            - matchExpressions:
              - key: color
                operator: In
                values:
                - blue
      containers:
      - name: nginx
        image: nginx
        ports:
        - containerPort: 80
```

12-



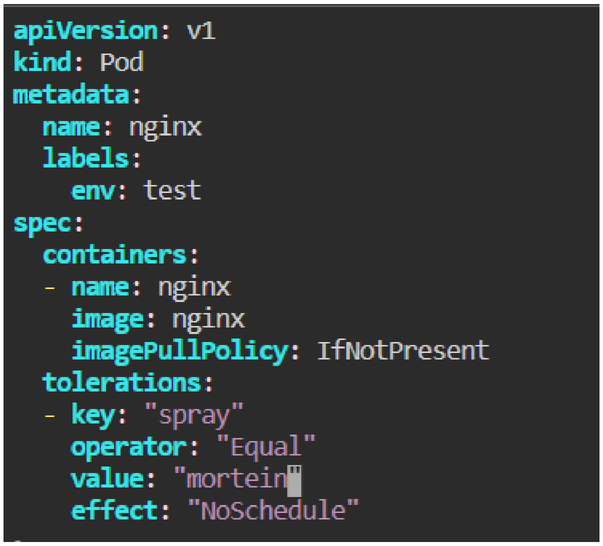
```
controlplane $ kubectl taint nodes node01 spray=mortein:NoSchedule
node/node01 tainted
```

13 &14- mosquito pod Is running

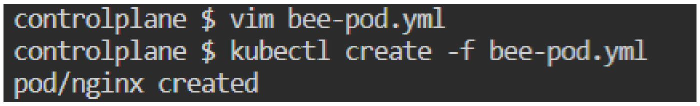


```
controlplane $ kubectl run mosquito --image=nginx
pod/mosquito created
controlplane $ kubectl get pod mosquito
NAME       READY   STATUS    RESTARTS   AGE
mosquito   1/1     Running   0          7s
controlplane $
```

15-



```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels:
    env: test
spec:
  containers:
  - name: nginx
    image: nginx
    imagePullPolicy: IfNotPresent
  tolerations:
  - key: "spray"
    operator: "Equal"
    value: "mortein"
    effect: "NoSchedule"
```



```
controlplane $ vim bee-pod.yml
controlplane $ kubectl create -f bee-pod.yml
pod/nginx created
```