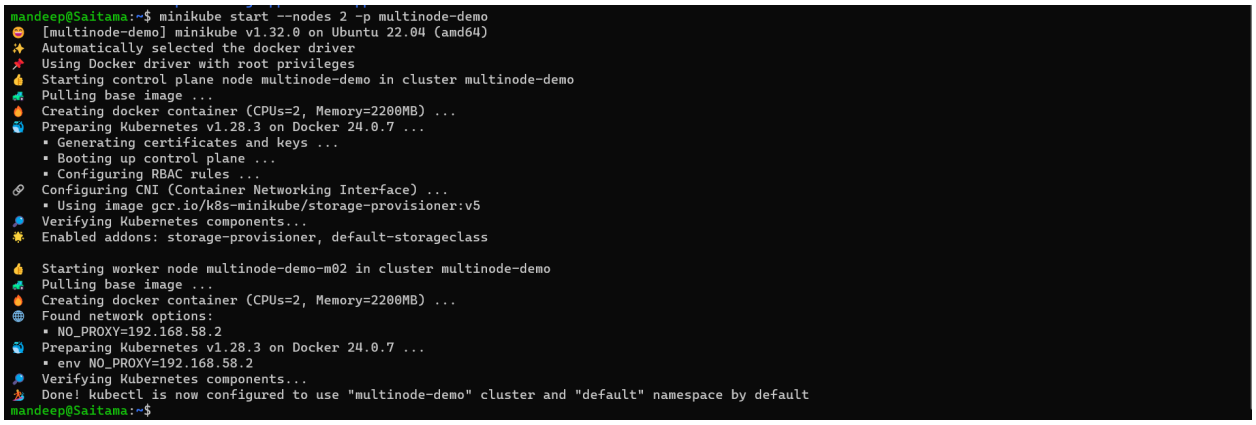
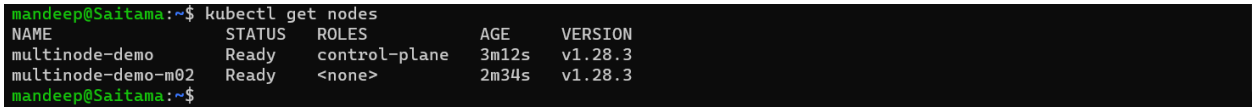
Week 2 lab 3
Mandeep Singh
Student ID 101495592

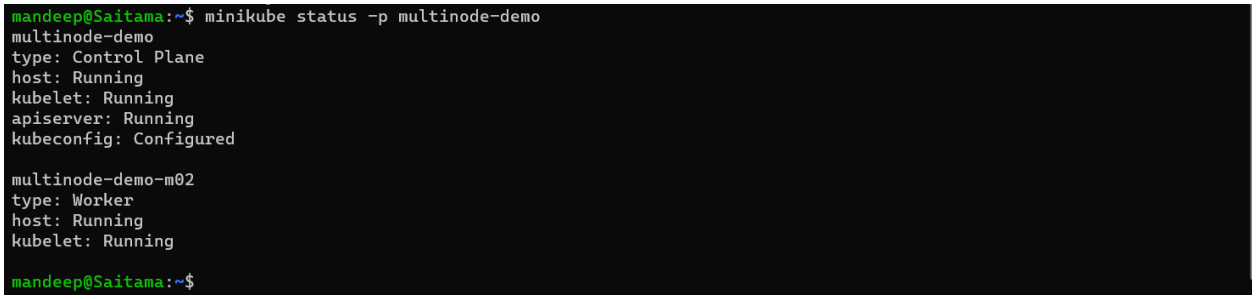
minikube start --nodes 2 -p multinode-demo



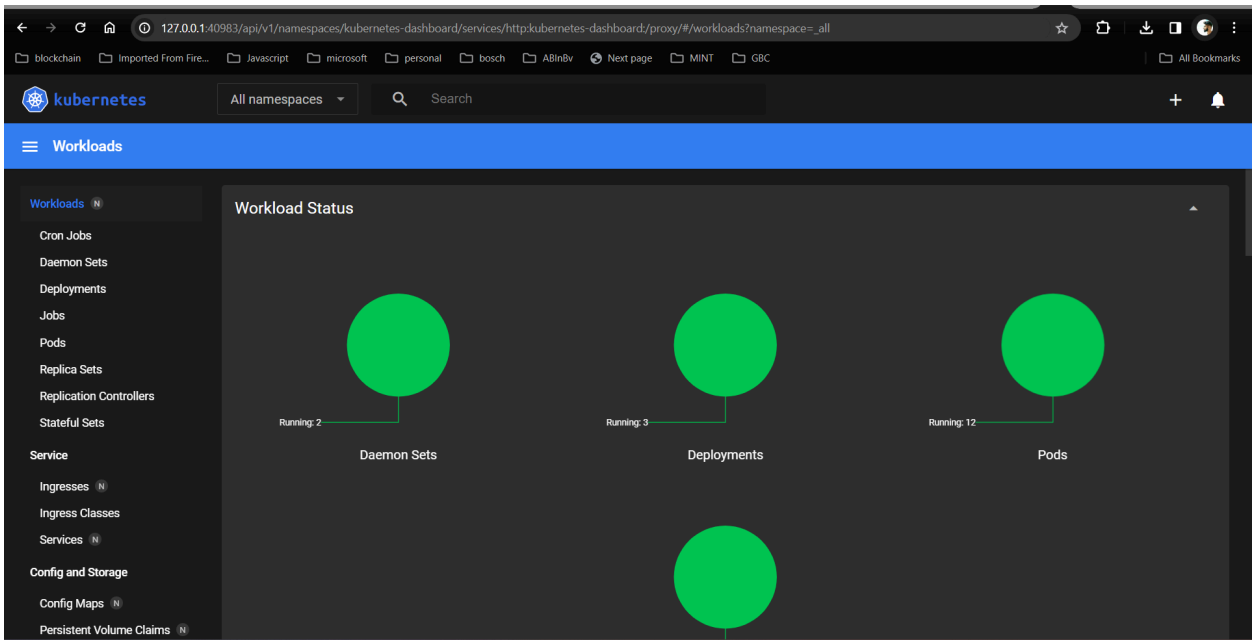
kubectl get nodes



minikube status -p multinode-demo



minikube dashboard -p multinode-demo



minikube stop -p multinode-demo
minikube delete --all

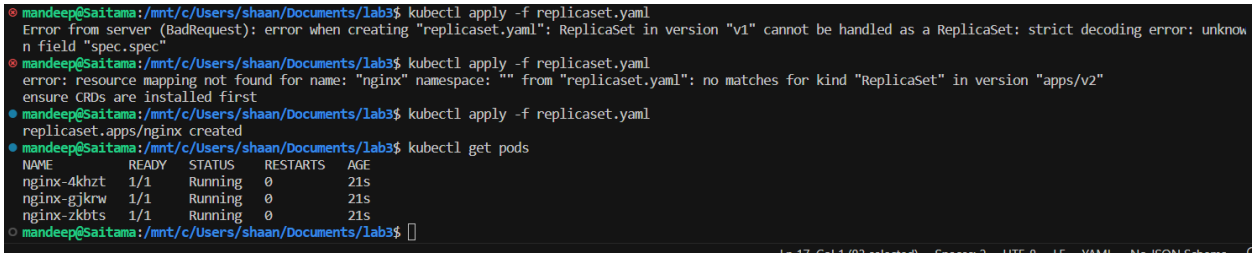
```
mandeep@Saitama:~$ minikube stop -p multinode-demo
✋  Stopping node "multinode-demo"  ...
🛑  Powering off "multinode-demo" via SSH ...
✋  Stopping node "multinode-demo-m02"  ...
🛑  Powering off "multinode-demo-m02" via SSH ...
🛑  2 nodes stopped.
mandeep@Saitama:~$ minikube delete --all
🔥  Deleting "minikube" in docker ...
🔥  Removing /home/mandeep/.minikube/machines/minikube ...
💀  Removed all traces of the "minikube" cluster.
🔥  Deleting "multinode-demo" in docker ...
🔥  Removing /home/mandeep/.minikube/machines/multinode-demo ...
🔥  Removing /home/mandeep/.minikube/machines/multinode-demo-m02 ...
💀  Removed all traces of the "multinode-demo" cluster.
🔥  Successfully deleted all profiles
mandeep@Saitama:~$
```

replicaset.yaml

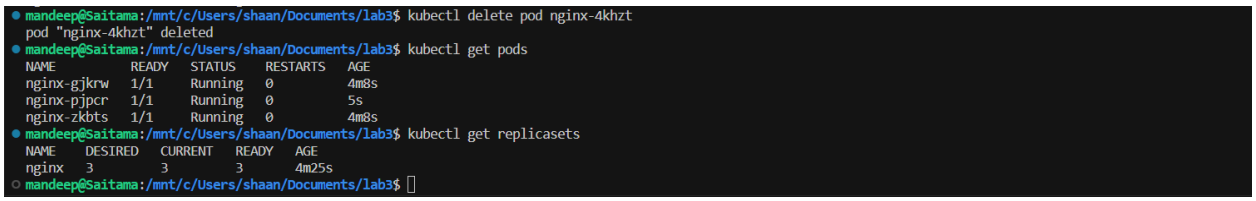
```yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: nginx
  labels:
    app: nginx
```

```
    tier: lb
spec:
  replicas: 3
  selector:
    matchLabels:
      tier: lb
  template:
    metadata:
      labels:
        tier: lb
    spec:
      containers:
        - name: nginx-replicaset
          image: nginx
```

kubectl apply -f replicaset.yaml
kubectl get pods



kubectl delete pod nginx-4khzt
kubectl get pods
kubectl get replicasets

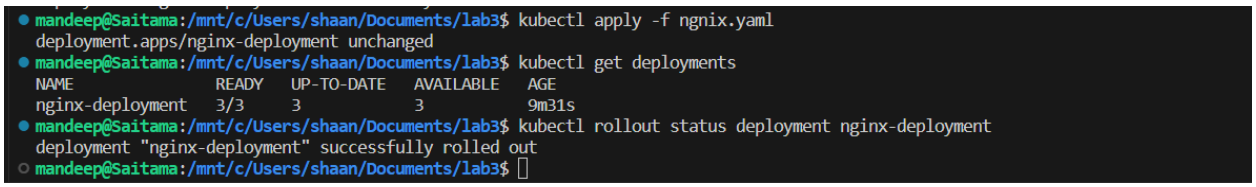


nginx.yaml

```
apiVersion: apps/v1
kind: Deployment
```

```yaml
metadata:
  name: nginx-deployment
  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
```

Kubectl apply -f nginx.yaml
kubectl get deployments
kubectl rollout status deployment nginx



stateful-nginx.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  ports:
    - port: 80
      name: web
```

```yaml
  clusterIP: None
  selector:
    app: nginx
---
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: web
spec:
  selector:
    matchLabels:
      app: nginx # has to match .spec.template.metadata.labels
  serviceName: "nginx"
  replicas: 3 # by default is 1
  template:
    metadata:
      labels:
        app: nginx # has to match .spec.selector.matchLabels
    spec:
      terminationGracePeriodSeconds: 10
      containers:
        - name: nginx
          image: k8s.gcr.io/nginx-slim:0.8
          ports:
            - containerPort: 80
              name: web
          volumeMounts:
            - name: www
              mountPath: /usr/share/nginx/html
  volumeClaimTemplates:
    - metadata:
        name: www
      spec:
        accessModes: ["ReadWriteOnce"]
        storageClassName: "my-storage-class"
        resources:
          requests:
            storage: 1Gi
```

kubectl apply -f stateful-nginx.yaml
kubectl get statefulsets

```
mandeep@Saitama:/mnt/c/Users/shaan/Documents/lab3$ kubectl apply -f stateful-nginx.yaml
service/nginx unchanged
statefulset.apps/web created
mandeep@Saitama:/mnt/c/Users/shaan/Documents/lab3$ kubectl get statefulsets
NAME   READY   AGE
web    0/3     23s
mandeep@Saitama:/mnt/c/Users/shaan/Documents/lab3$ kubectl get statefulsets
NAME   READY   AGE
web    0/3     33s
mandeep@Saitama:/mnt/c/Users/shaan/Documents/lab3$
```

daemonSet-nginx.yaml

```yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: nginx
spec:
  selector:
    matchLabels:
      name: nginx-lb
  template:
    metadata:
      labels:
        name: nginx-lb
    spec:
      containers:
        - name: nginx
          image: nginx
```

kubectl apply -f daemonSet-nginx.yaml
kubectl get daemonset

```
mandeep@Saitama:/mnt/c/Users/shaan/Documents/lab3$ kubectl apply -f daemonSet-nginx.yaml
daemonset.apps/nginx created
mandeep@Saitama:/mnt/c/Users/shaan/Documents/lab3$ kubectl get daemonset
NAME    DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
nginx   1         1         1       1            1           <none>          19m
mandeep@Saitama:/mnt/c/Users/shaan/Documents/lab3$
```

nginx-deployment-resource-limit.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
```

```
  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
          resources:
            limits:
              memory: "256Mi"
              cpu: "200m"
            requests:
              memory: "128Mi"
              cpu: "100m"
```

kubectl apply -f nginx-deployment-resource-limit.yaml
kubectl get deployments

```
mandeep@Saitama:/mnt/c/Users/shaan/Documents/lab3$ kubectl apply -f ng
inx-deployment-resource-limit.yaml
deployment.apps/nginx-deployment configured
mandeep@Saitama:/mnt/c/Users/shaan/Documents/lab3$ kubectl get deployments
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3     3            3           56m
mandeep@Saitama:/mnt/c/Users/shaan/Documents/lab3$
```

nginx-deployment-healthcheck.yaml

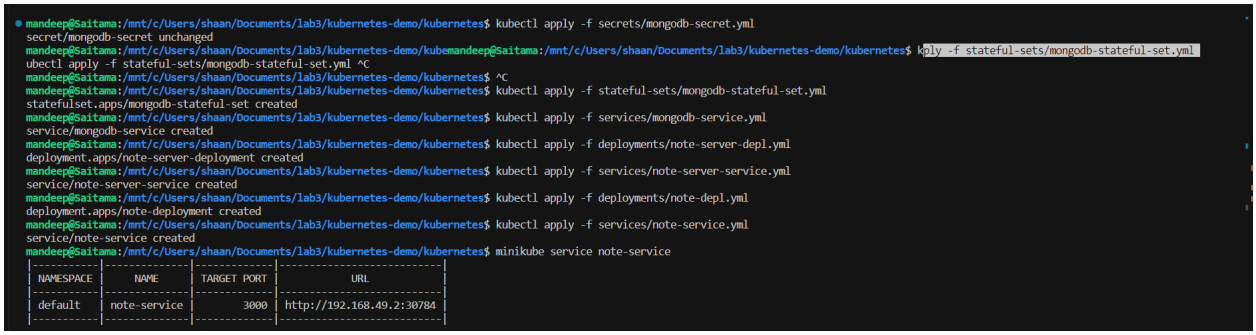
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
```

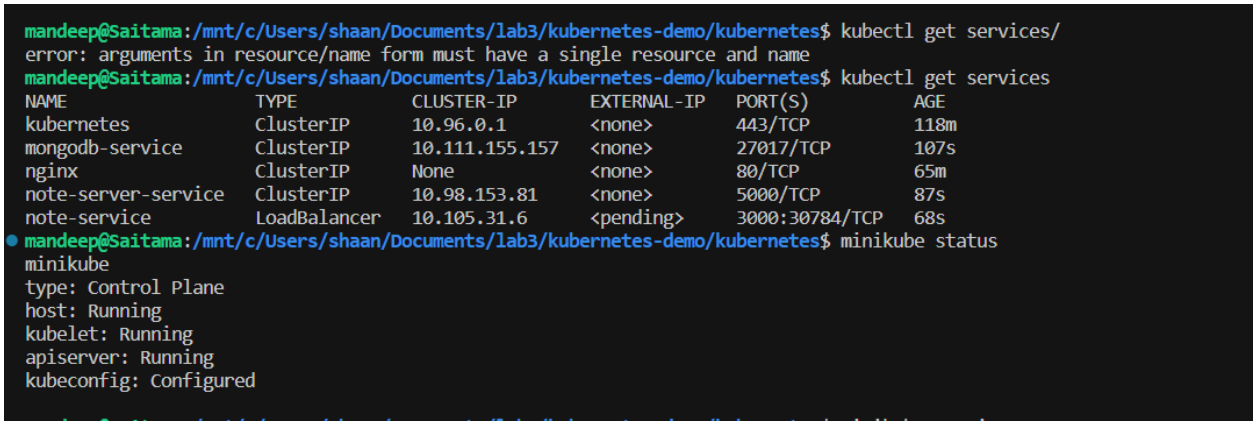
```
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.14.2
          ports:
            - containerPort: 80
          livenessProbe:
            httpGet:
              path: /
              port: 80
            initialDelaySeconds: 15
            periodSeconds: 10
          readinessProbe:
            httpGet:
              path: /
              port: 80
            initialDelaySeconds: 5
            periodSeconds: 5
```

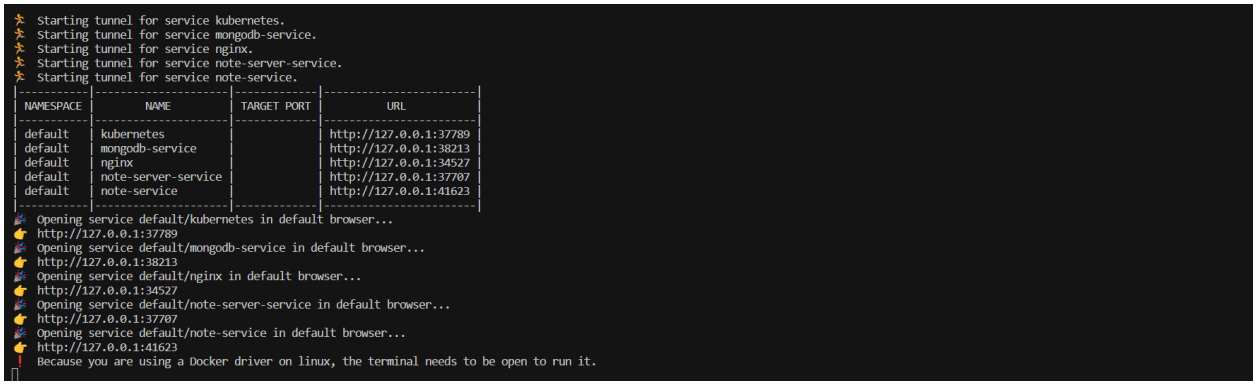
```
mandeep@Saitama:/mnt/c/Users/shaan/Documents/lab3$ kubectl apply -f nginx-deployment-healthcheck.yaml
deployment.apps/nginx-deployment configured
mandeep@Saitama:/mnt/c/Users/shaan/Documents/lab3$ kubectl get deployments
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3     1            3           74m
mandeep@Saitama:/mnt/c/Users/shaan/Documents/lab3$
```

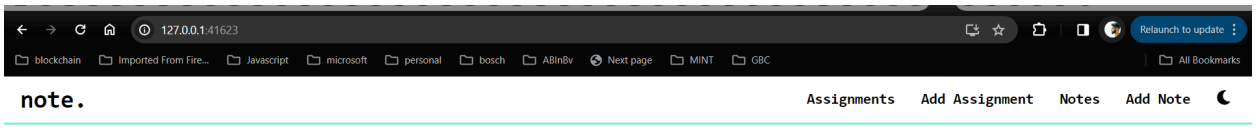
Running the full stack application
[Deploy MERN App To K8s (Minikube) (hashnode.dev)]

[GitHub - Ananya2001-an/kubernetes-demo: Learn how to deploy a MERN app to K8s](#)









Important Assignments

Delete one pod and check