

1. Create the mongo database for the backend tier and get the details of the created object.

```
apiVersion: v1
kind: Pod
metadata:
  name: mongo
  labels:
    type: backended
spec:
  containers:
  - name: dbserver
    image: mongo
    ports:
    - containerPort: 27017
```

```
vagrant@master-node:~/project$
vagrant@master-node:~/project$ nano backended.yml
vagrant@master-node:~/project$ kubectl create -f backended.yml
-bash: kubectl: command not found
vagrant@master-node:~/project$ kubectl create -f backended.yml
pod/mongo created
vagrant@master-node:~/project$ kubectl get pod -o wide
NAME      READY   STATUS    RESTARTS   AGE   IP            NODE           NOMINATED NODE   READINESS GATES
mongo     1/1     Running   0           24s   192.168.158.63 worker-node02   <none>           <none>
```

2. Create the service object for the mongo database and get the details of the object.

```

GNU nano 5.6.1
apiVersion: v1
kind: Service
metadata:
  name: mongo
spec:
  ports:
    - port: 27017
      targetPort: 27017
  selector:
    type: backened

```

```

vagrant@master-node:~/project$ nano dbservice.yml
vagrant@master-node:~/project$ kubectl create -f dbservice.yml
service/mongo created
vagrant@master-node:~/project$ kubectl get service -o wide

```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE	SELECTOR
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	4m50s	<none>
mongo	ClusterIP	10.110.160.66	<none>	27017/TCP	12s	type=backened

3. Create the node js application with the below yml file and get the details of the created object. I used the env variable for connecting the app to database.

```

GNU nano 5.6.1
apiVersion: v1
kind: Pod
metadata:
  name: app
  labels:
    type: frontend
spec:
  containers:
    - name: appserver
      image: devopsedu/employee
      ports:
        - containerPort: 8888
      env:
        - name: MONGO_URL
          value: mongodb://mongo:27017/mango
      imagePullPolicy: Always

```

```
vagrant@master-node:~/project$ nano frontend.yml
vagrant@master-node:~/project$ kubectl create -f frontend.yml
pod/app created
vagrant@master-node:~/project$ kubectl get pod -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED	NODE	READINESS	GATES
app	1/1	Running	0	9s	192.168.87.245	worker-node01	<none>		<none>	
mongo	1/1	Running	0	3m58s	192.168.158.63	worker-node02	<none>		<none>	

4. Create the service object for the node js application with below yml file and get the details of the object.

```
GNU nano 5.6.1
apiVersion: v1
kind: Service
metadata:
  name: appservice
spec:
  type: NodePort
  ports:
  - port: 8888
    targetPort: 8888
    nodePort: 30002
  selector:
    type: frontend
```

```
vagrant@master-node:~/project$ nano appservice.yml
vagrant@master-node:~/project$ kubectl create -f appservice.yml
service/appservice created
vagrant@master-node:~/project$ kubectl get service -o wide
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE	SELECTOR
appservice	NodePort	10.101.152.45	<none>	8888:30002/TCP	9s	type=frontend
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	7m50s	<none>
mongo	ClusterIP	10.110.160.66	<none>	27017/TCP	3m12s	type=backened

5. Get the details of the app and db pod. And also get the details of the node in which app and db pod is running. Take the IP address of that pod and try to access it.

```
vagrant@master-node:~/project$ kubectl get pod -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED	NODE	READINESS	GATES
app	1/1	Running	0	2m58s	192.168.87.245	worker-node01	<none>		<none>	
mongo	1/1	Running	0	6m47s	192.168.158.63	worker-node02	<none>		<none>	

```
vagrant@master-node:~/project$ kubectl get service -o wide
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE	SELECTOR
appservice	NodePort	10.101.152.45	<none>	8888:30002/TCP	101s	type=frontend
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	9m22s	<none>
mongo	ClusterIP	10.110.160.66	<none>	27017/TCP	4m44s	type=backened

```
vagrant@master-node:~/project$ kubectl get nodes -o wide
```

NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXTERNAL-IP	OS-IMAGE	KERNEL-VERSION	CONT
master-node	Ready	control-plane,master	7d18h	v1.23.0	10.0.0.10	<none>	Ubuntu 21.10	5.13.0-22-generic	cri-
worker-node01	Ready	worker	7d18h	v1.23.0	10.0.0.11	<none>	Ubuntu 21.10	5.13.0-22-generic	cri-
worker-node02	Ready	worker	7d18h	v1.23.0	10.0.0.12	<none>	Ubuntu 21.10	5.13.0-22-generic	cri-

6. Take the IP of the app server and try to access it. Add the employee in the application and get the details of the employee for the verification.

Not secure | http://10.0.0.12:30002/AddEmployee

on Web Servic... whizlab az104 portal.azure.com Instructor-Led Onli... AZ-400 outlook login edureka devops ansible

## Add New Employee

First Name :

Last Name :

ID :

Department :

Designation :

Salary :

Added employee with id 103215 into the employee collection.

Not secure | http://10.0.0.12:30002/GetEmployee

Service... whizlab az104 portal.azure.com Instructor-Led Onli... AZ-400 outlook login edureka devops ansible ad hoc

## Get Employee Information

Emp Id :

Found the employee with id 103215

First Name	Last Name	ID	Department	Designation	Salary
akshay	verma	103215	IT	Infra support	30000