Name: Patel Shivam S.
Enroll no: 21162101019
Sem: 7 Batch: 71 Branch: CBA

Sub: CS

Practical 10

AIM: **You are working for a company that uses IBM Cloud to store critical data in a Cloudant database. Your team has developed an API-based application that performs CRUD operations on the Cloudant database, and this application is now ready for deployment in a Kubernetes environment.**

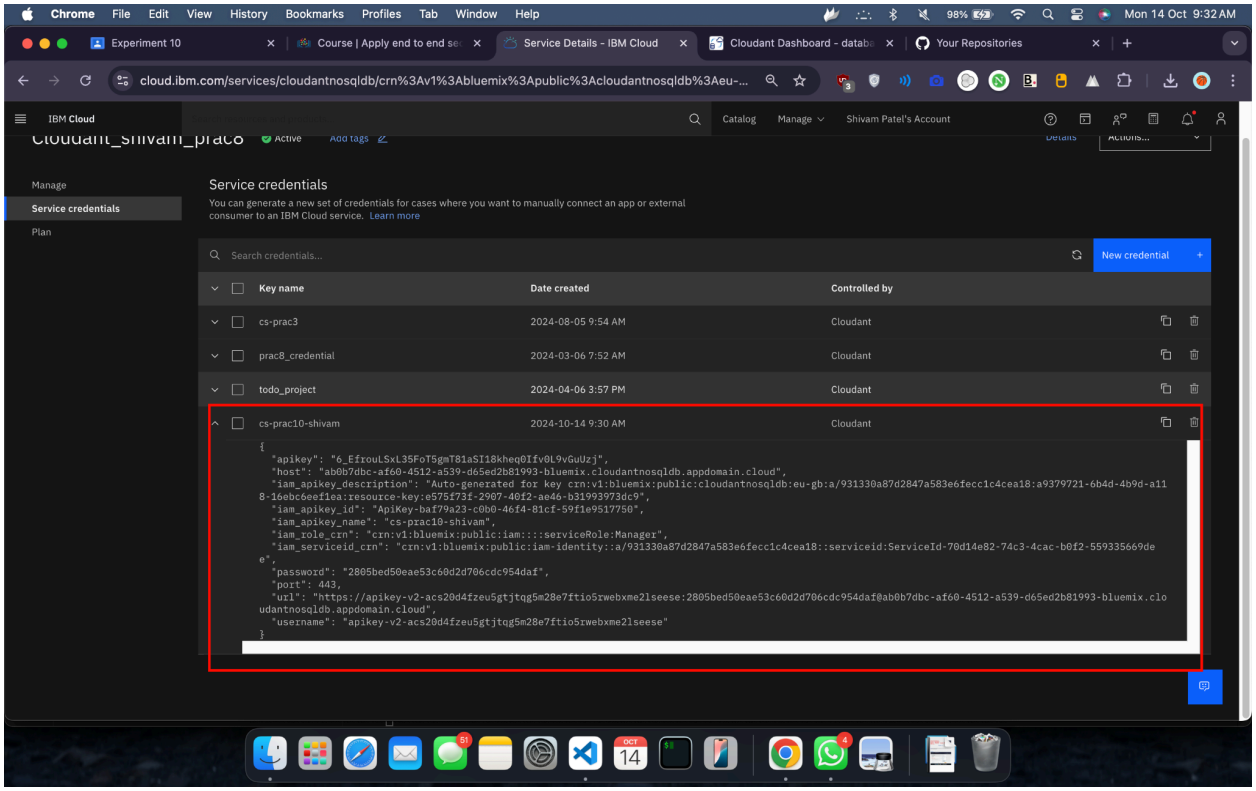
As part of the security team, your task is to ensure that the application adheres to security best practices, including limiting network traffic for the pods.

Task:

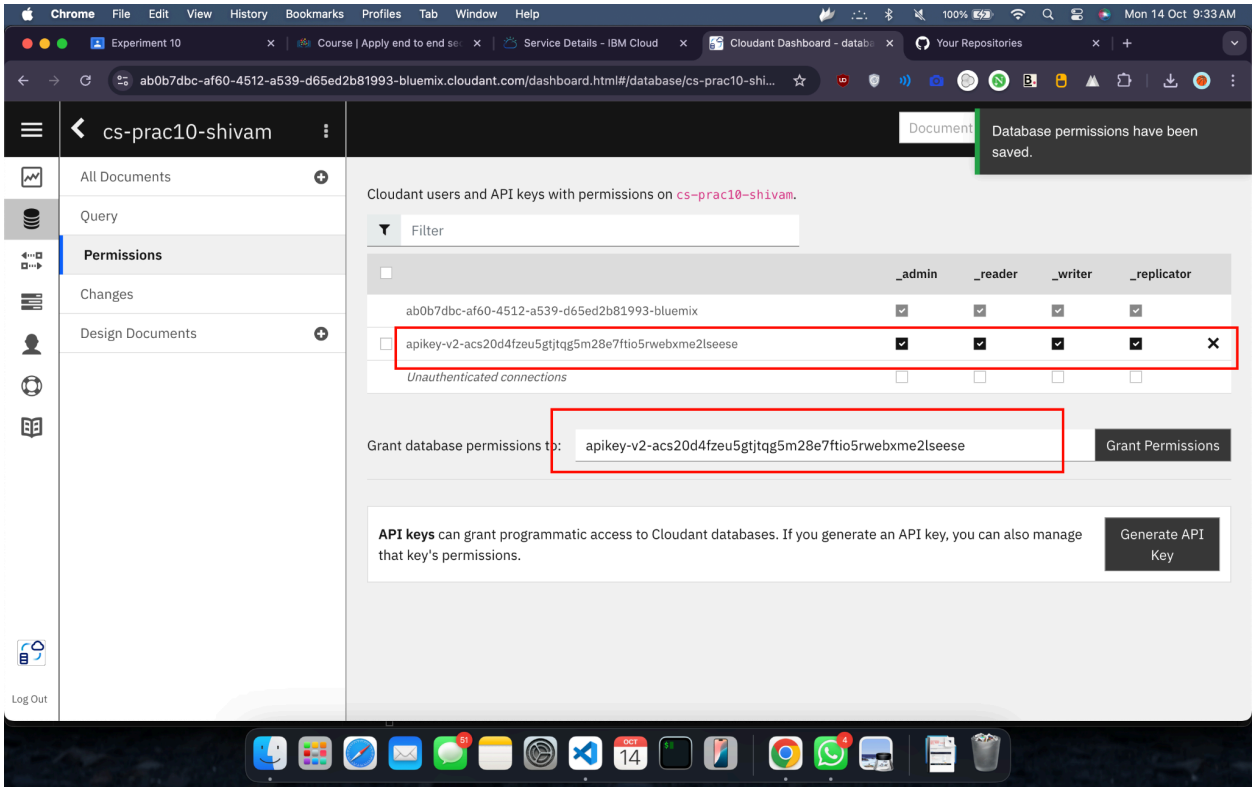
Deploy the existing API-based application on a Kubernetes cluster.
Configure a network policy that blocks all egress traffic from the pod.

SOLUTION:

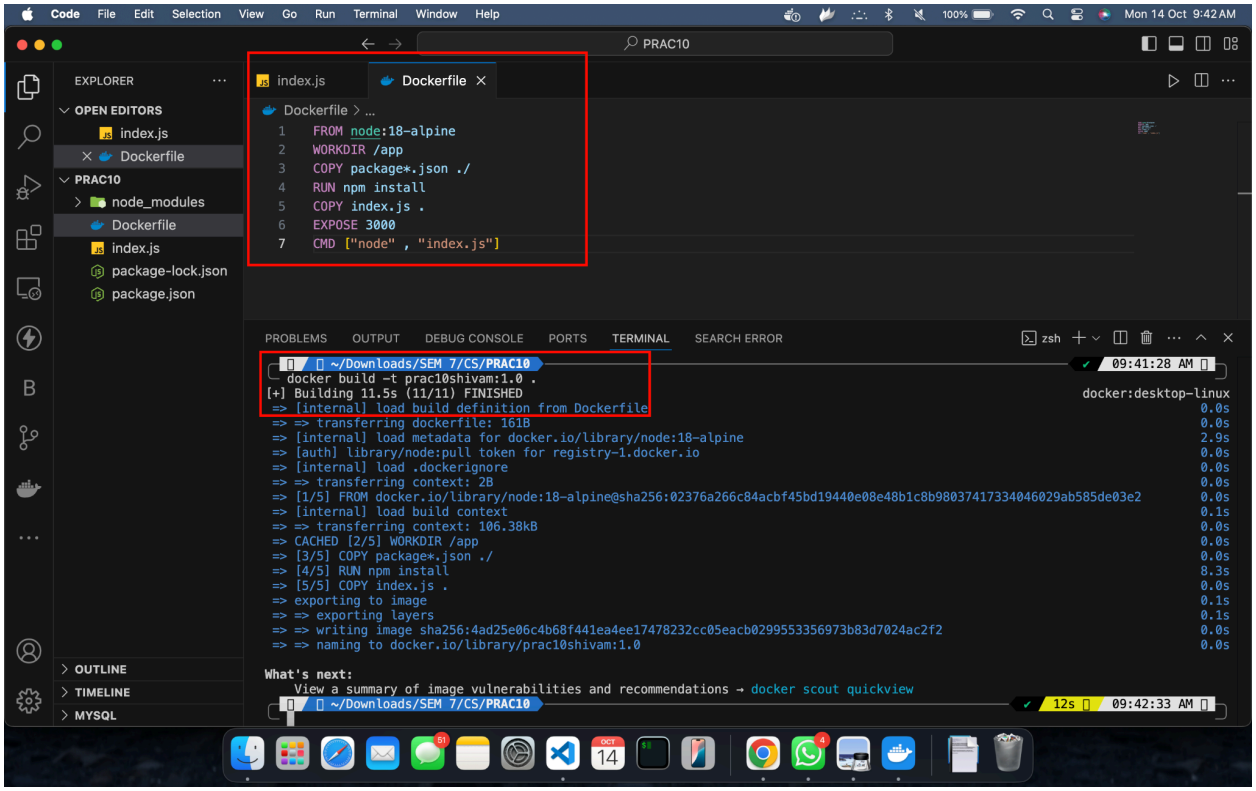
➢ Login to ibmcloud and generate service credentials of cloudant.

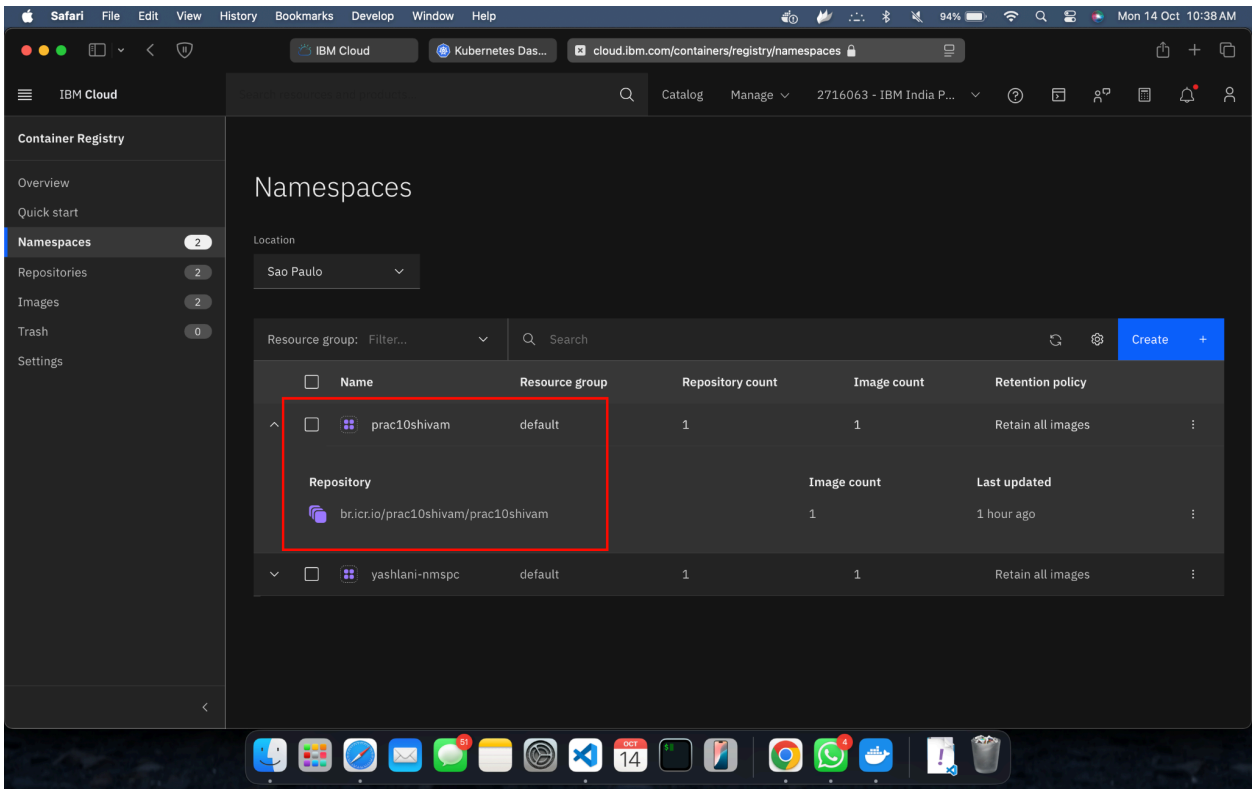


Give the permission

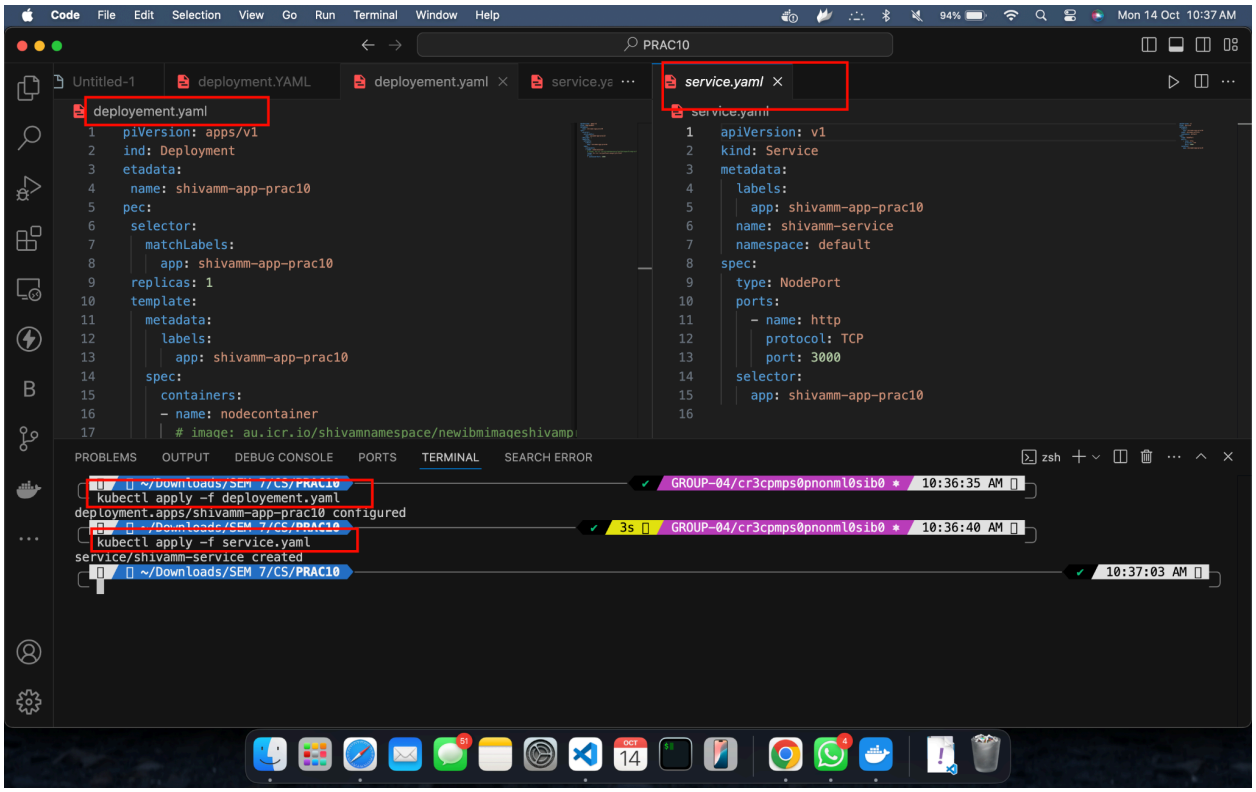


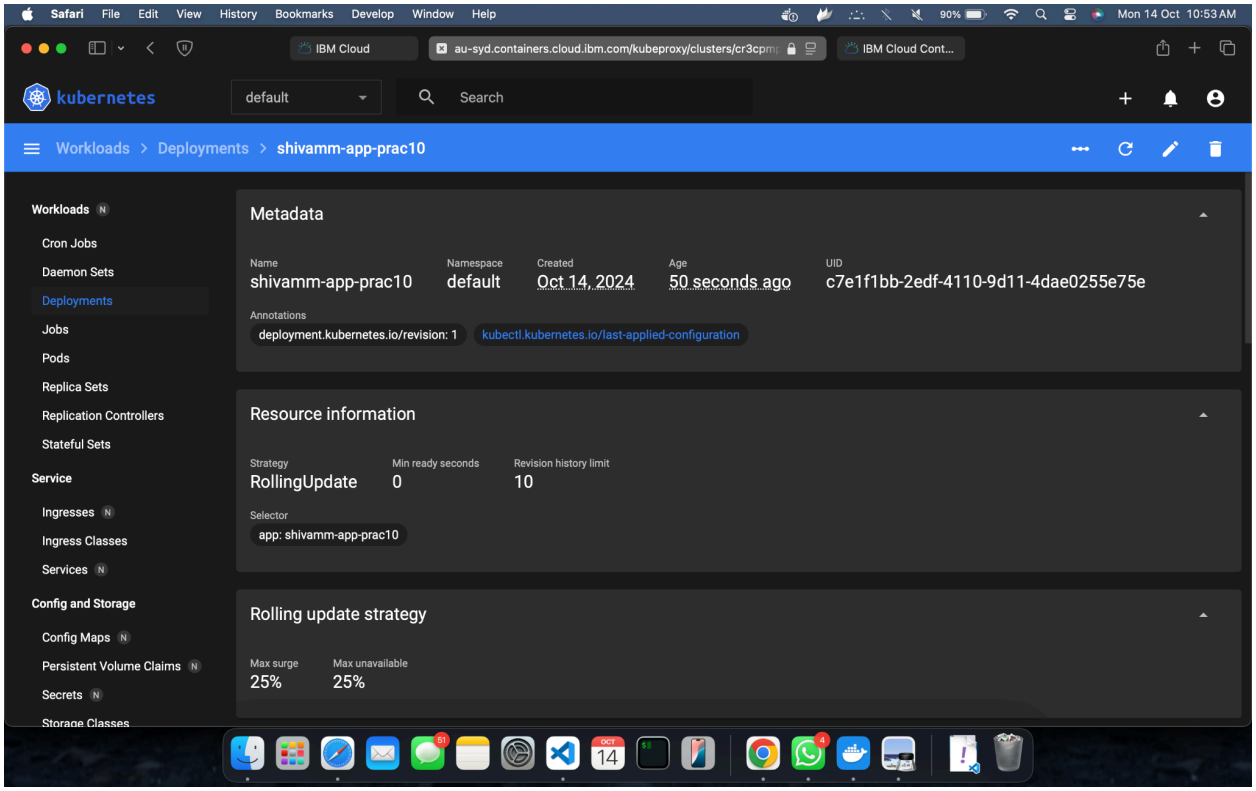
push docker image to ibm container registry



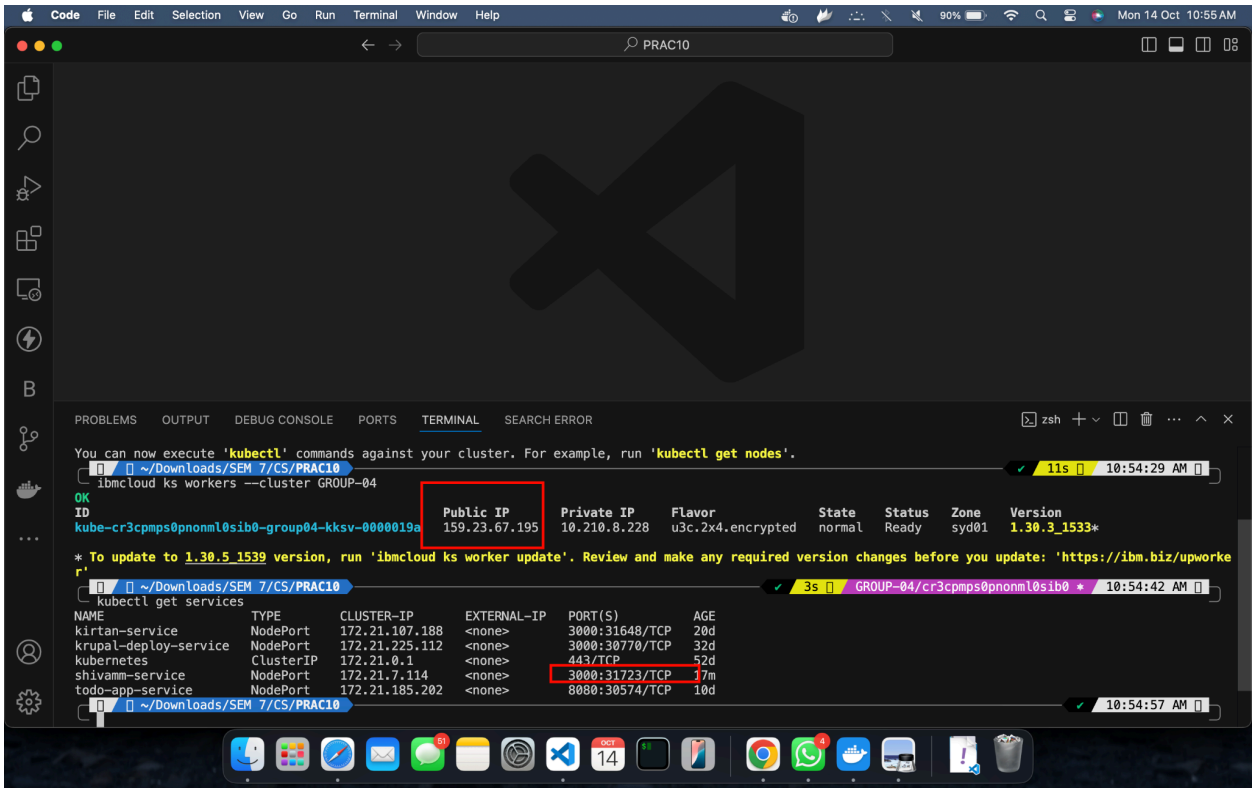


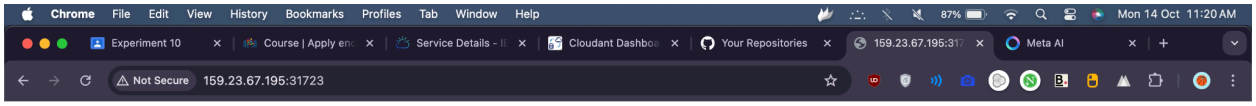
Create deployment and service files.





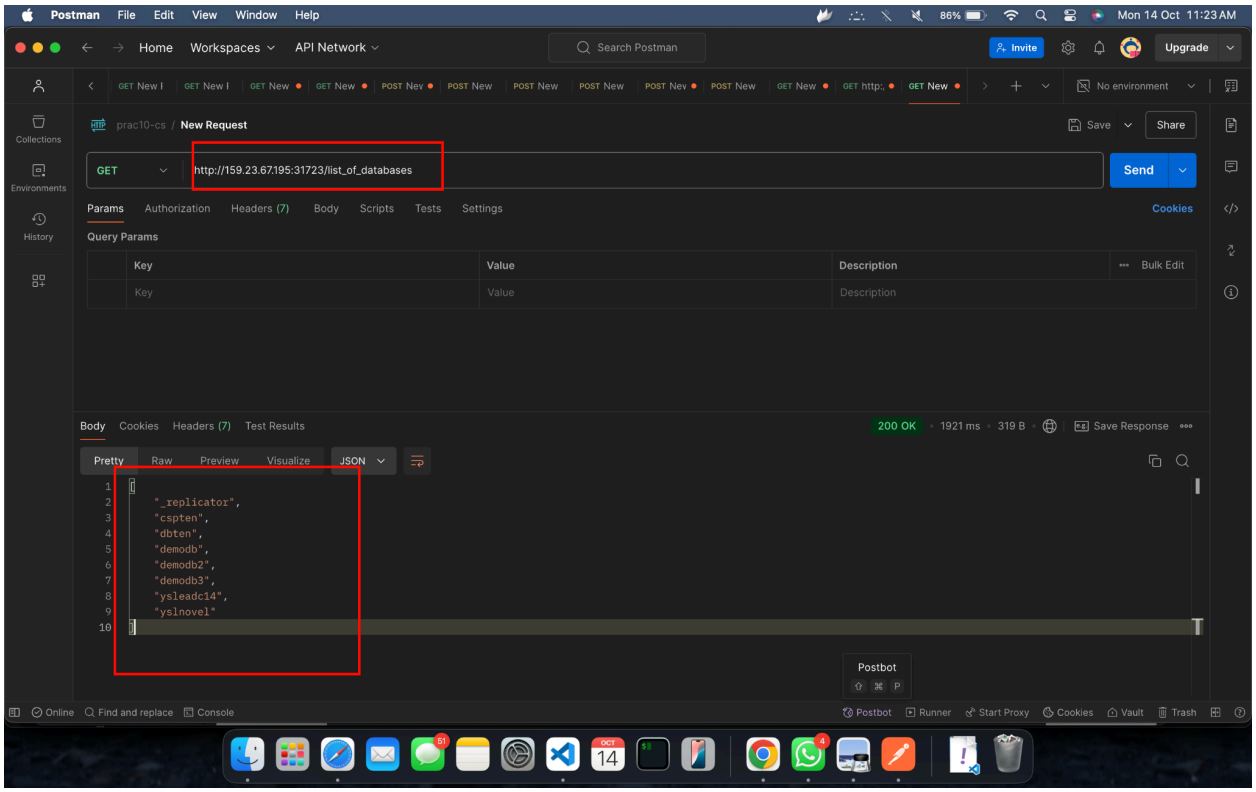
Get IP and port and check connectivity



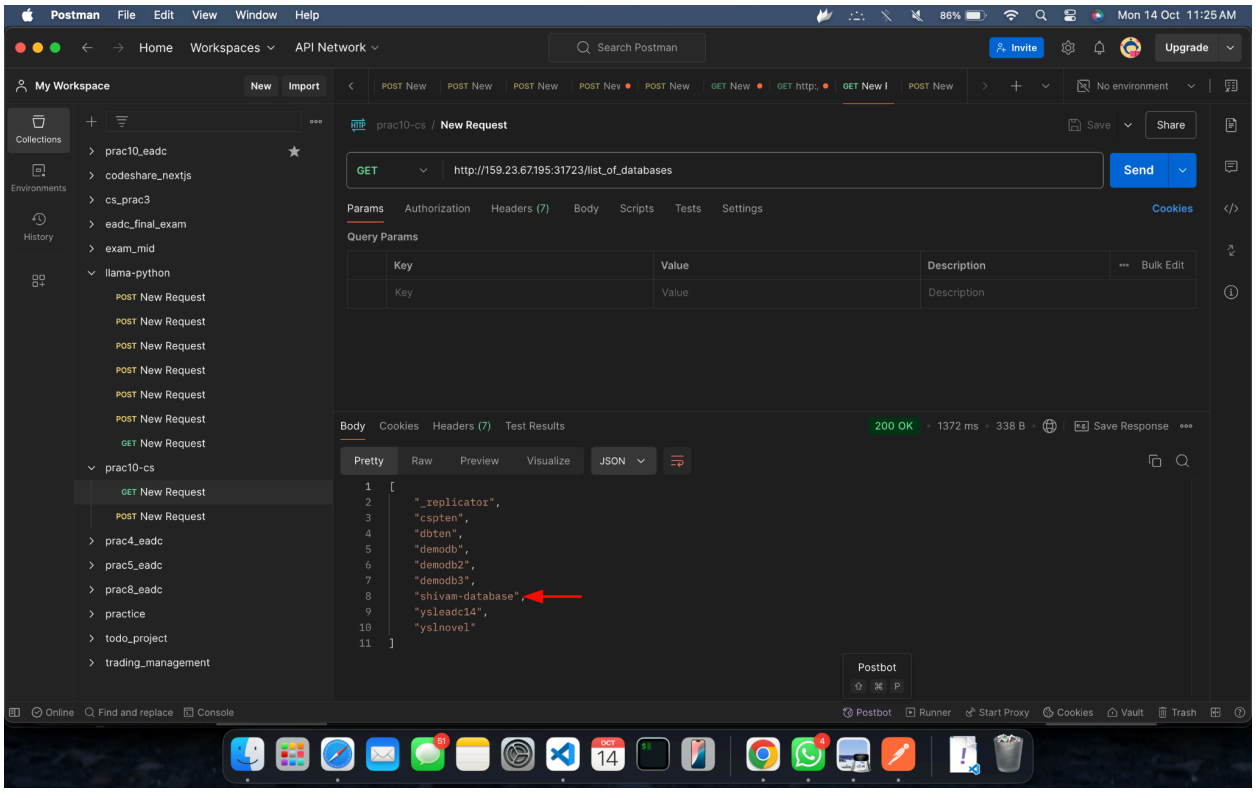


Welcome to cloudant database on IBM Cloud

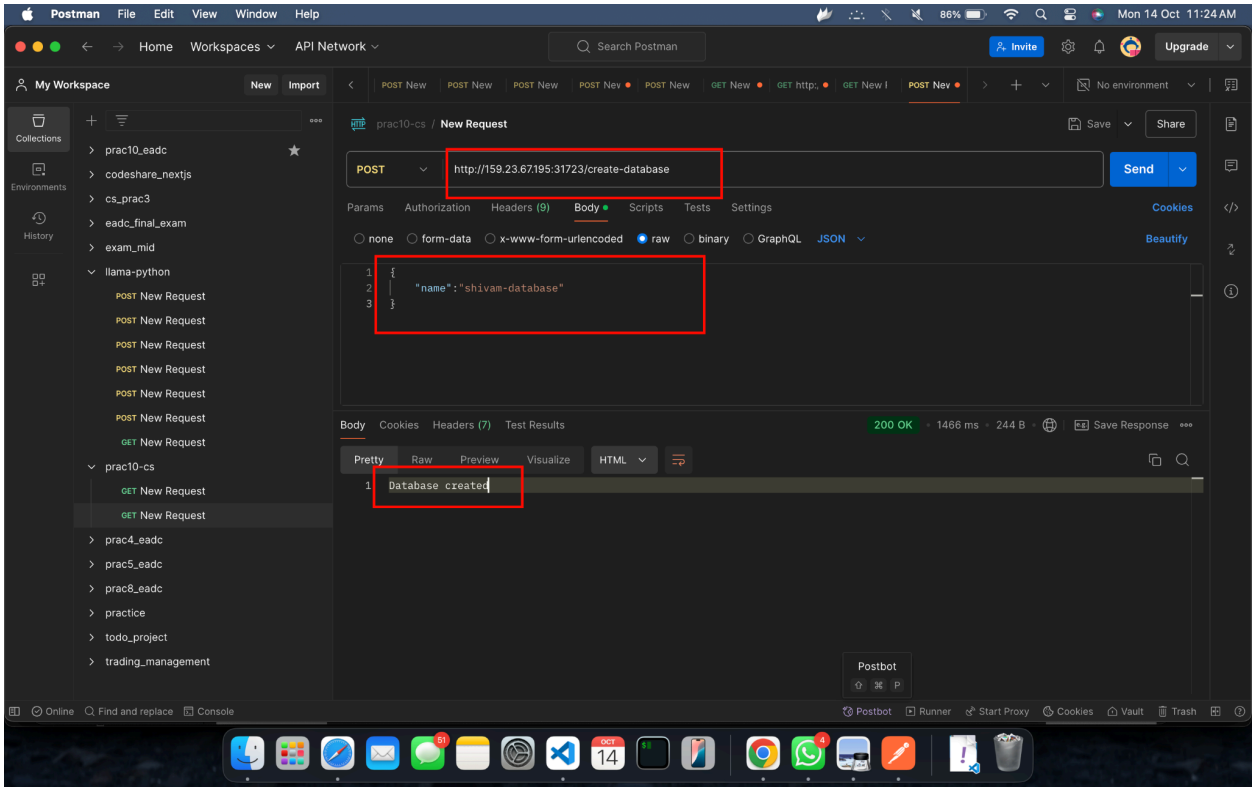
Check requests via Postman if the application accepts so

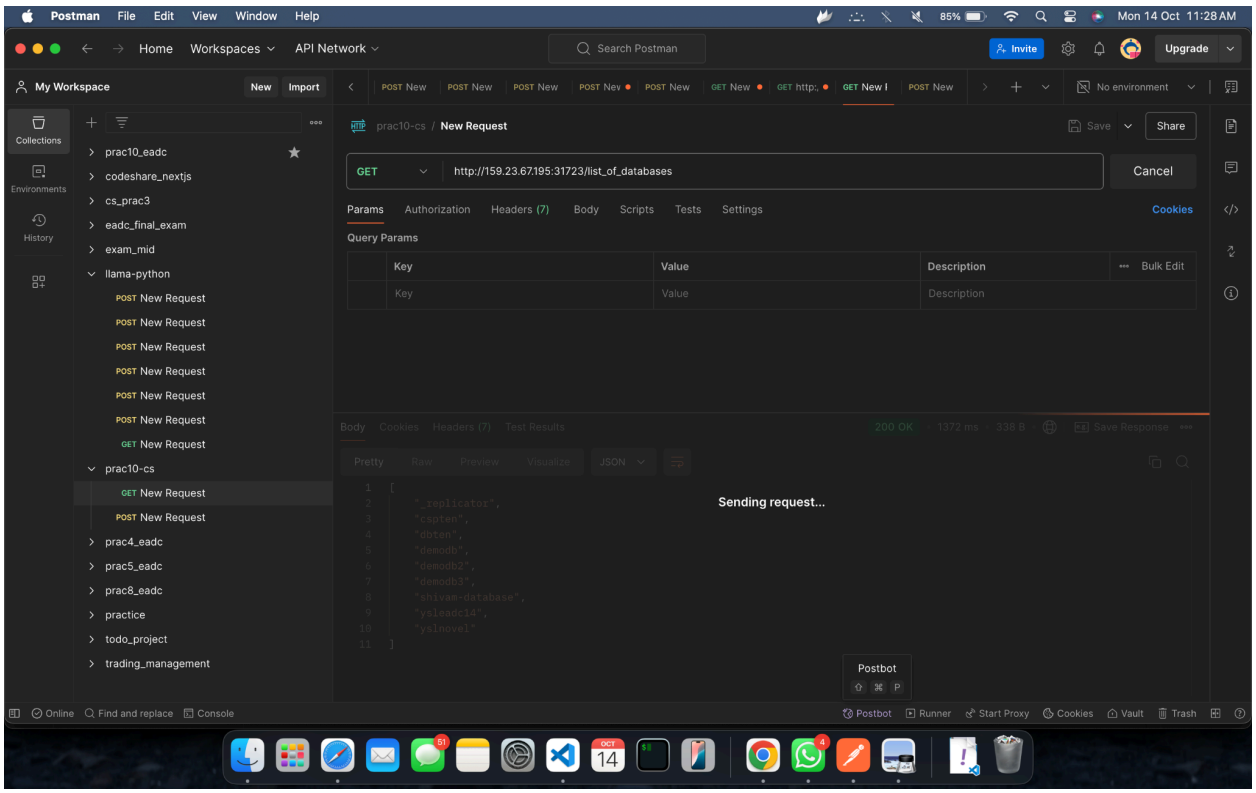


To see list of databases.

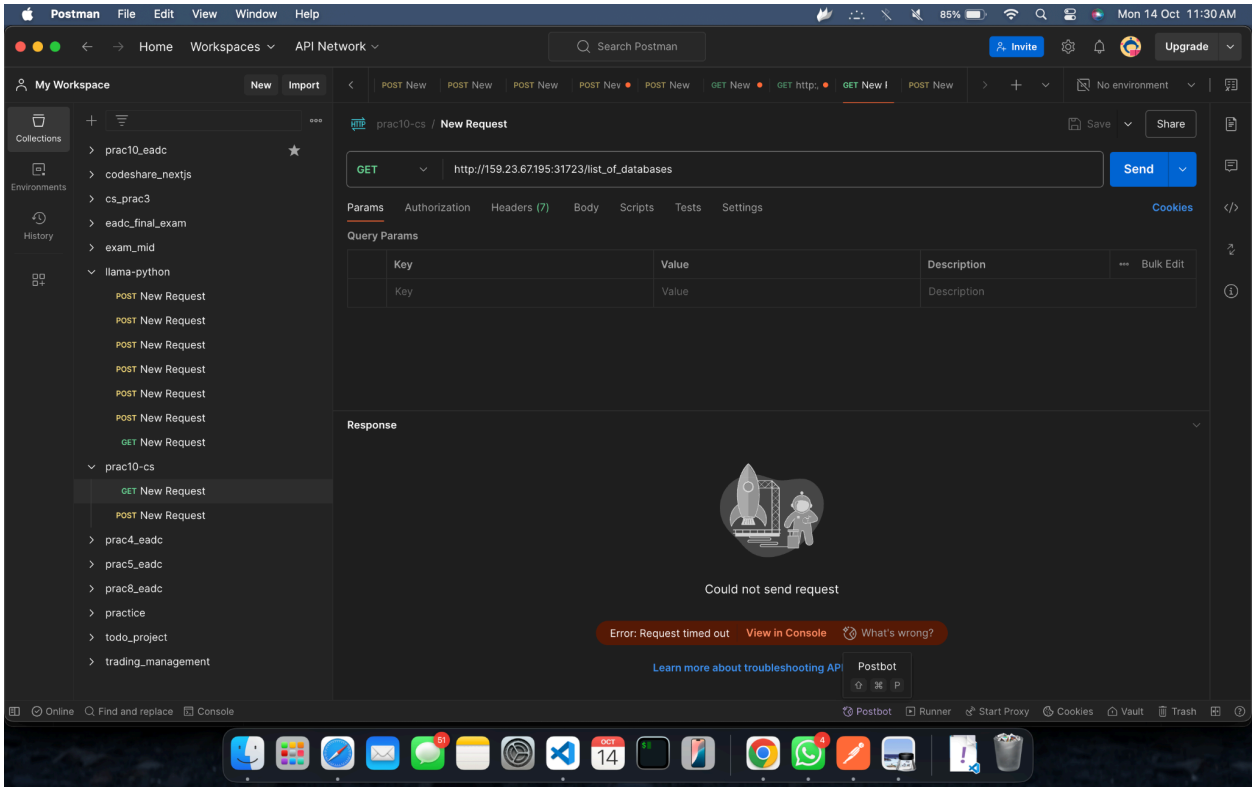


To create a database.





Create global network policy to deny both egress and ingress traffic.



GlobalNetworkPolicy.yaml:

```
apiVersion: projectcalico.org/v3
kind: GlobalNetworkPolicy
metadata:
 name: default-deny
spec:
 selector: projectcalico.org/namespace != 'kube-system'
 types:
   - Ingress
   - Egress
```

Dockerfile:

```
FROM node:18-alpine
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY index.js .
EXPOSE 3000
CMD ["node" , "index.js"]
```

Deployment.yaml:

```
apiVersion: apps/v1
kind: Deployment
metadata:
 name: shivamm-app-prac10
spec:
 selector:
   matchLabels:
     app: shivamm-app-prac10
 replicas: 1
 template:
   metadata:
     labels:
       app: shivamm-app-prac10
   spec:
     containers:
     - name: nodecontainer
```

```
      # image: au.icr.io/shivamnamespace/newibmimageshivamprac4:2.0
      image: br.icr.io/yashlani-nmspc/yslcldnt:1.0
      ports:
      - containerPort: 3000
```

service.yaml:

```
apiVersion: v1
kind: Service
metadata:
 labels:
   app: shivamm-app-prac10
 name: shivamm-service
 namespace: default
spec:
 type: NodePort
 ports:
   - name: http
     protocol: TCP
     port: 3000
 selector:
   app: shivamm-app-prac10
```