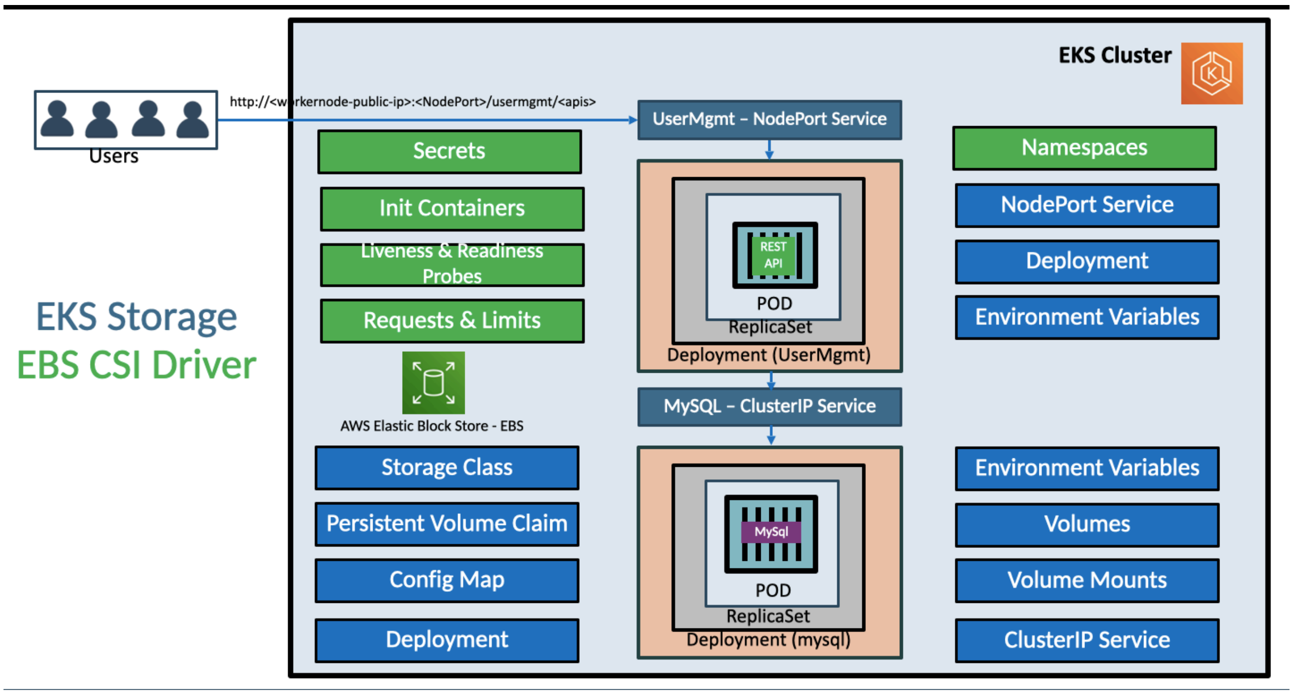
Eks Project - Usermanagement Microservice with Mysql database



- **We are going to use EBS CSI Driver and use EBS Volumes for persistence storage to MySQL Database**

Kubernetes Important Concepts for Application Deployments

Storage Class - for persistant volume
Persistent Volume Claim - to claim the volume
Config Map - config file for sql schema
Deployment, Environment Variables, Volumes, VolumeMounts
ClusterIP Service - for my sql server
Deployment, Environment Variables - define db data
NodePort Service - for user managemant service
Secrets - for db password
Init Containers - it will wait for my sql pod to comeup then connect to usermgnt Microservice
Liveness & Readiness Probes - to check the pod status
Requests & Limits - for the pod limits
Namespaces - for the isolated resource to create
Resource quota - how much resource can be utilised in the namespace

Create Cluster:

```
eksctl create cluster --name=eksdemo1 \
                      --region=us-east-1 \
                      --zones=us-east-1a,us-east-1b \
                      --without-nodegroup
```

```
eksctl utils associate-iam-oidc-provider \
    --region us-east-1 \
    --cluster eksdemo1 \
    —approve
```

Create Public Node Group:

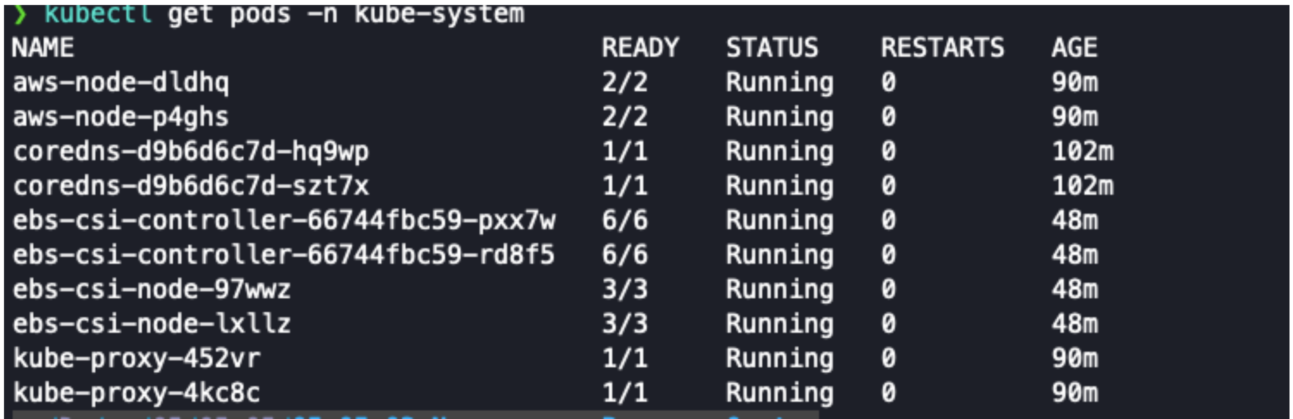
```
eksctl create nodegroup --cluster=eksdemo1 \
                       --region=us-east-1 \
                       --name=eksdemo1-ng-public1 \
                       --node-type=t3.medium \
                       --nodes=2 \
                       --nodes-min=2 \
                       --nodes-max=4 \
                       --node-volume-size=20 \
                       --ssh-access \
                       --ssh-public-key=kube-demo \
                       --managed \
                       --asg-access \
                       --external-dns-access \
                       --full-ecr-access \
                       --appmesh-access \
                       --alb-ingress-access
```

In node-group security group allow all traffic and save

To Install ebs csi driver
Create iam policy ec2 full access for ebs_csi_driver and attach to the node group iam role

Deploy EBS CSI Driver

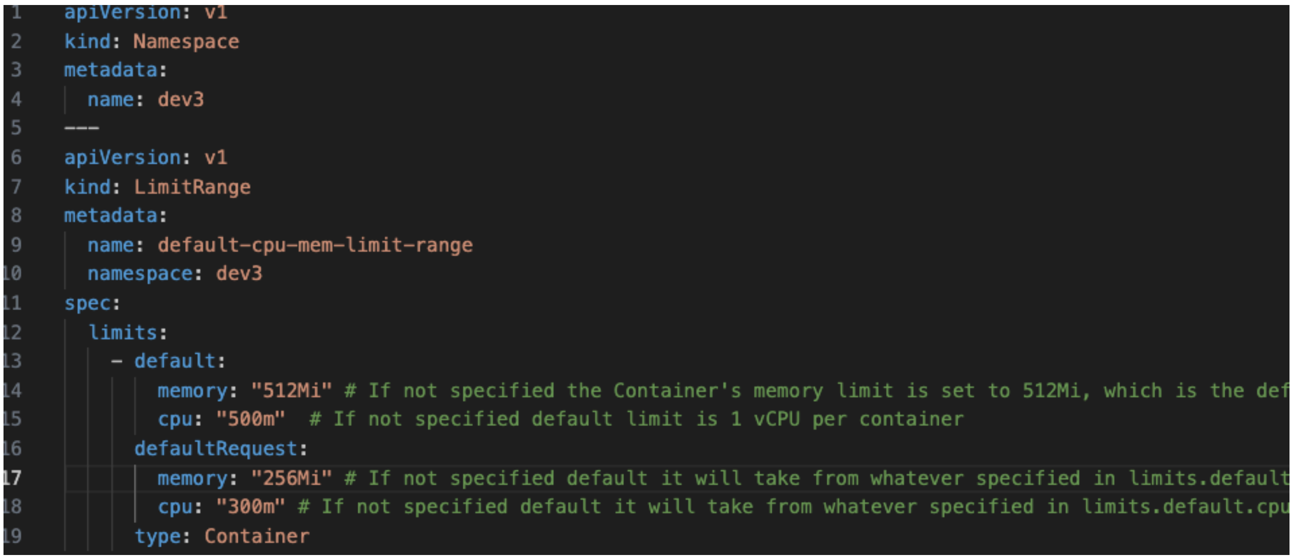
```
kubectl apply -k "github.com/kubernetes-sigs/aws-ebs-csi-driver/deploy/kubernetes/overlays/stable/?ref=master"
```



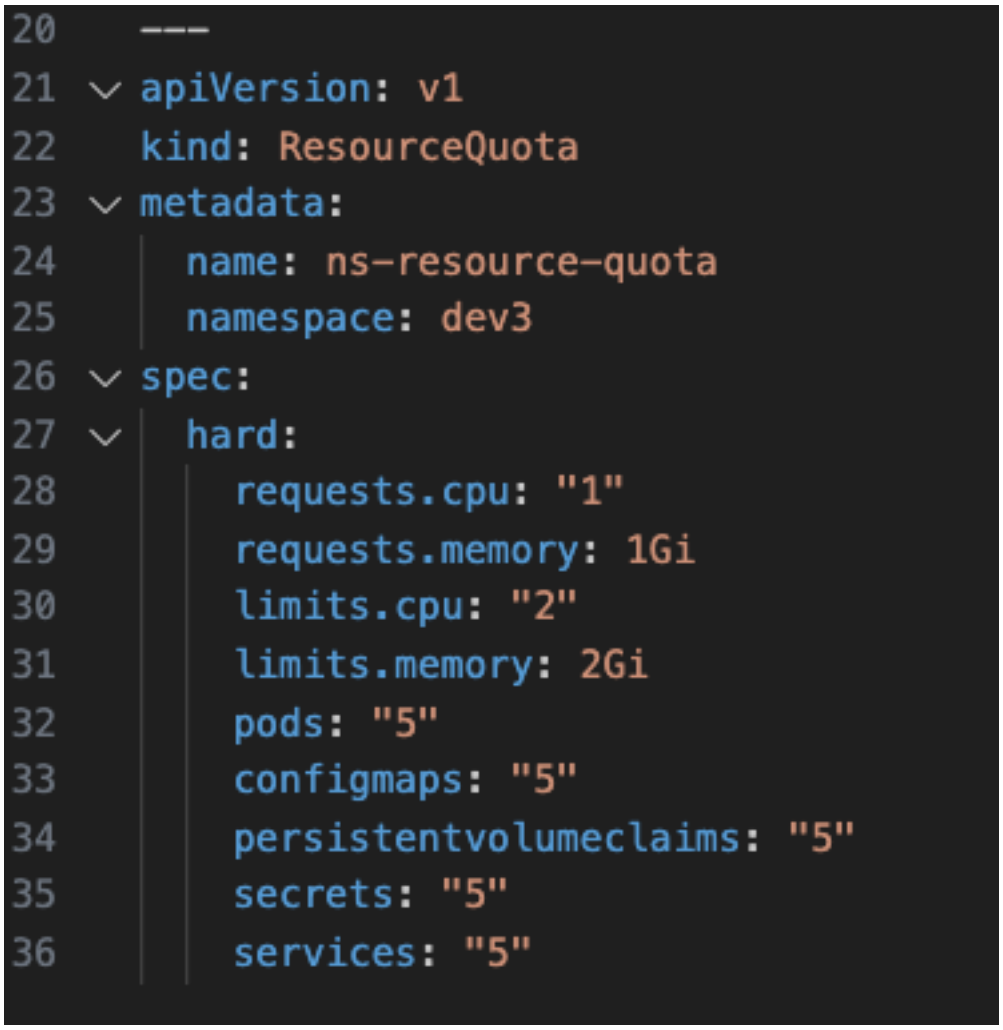
```
❯ kubectl get pods -n kube-system
NAME                                   READY   STATUS    RESTARTS   AGE
aws-node-dldhq                         2/2     Running   0          90m
aws-node-p4ghs                         2/2     Running   0          90m
coredns-d9b6d6c7d-hq9wp                1/1     Running   0          102m
coredns-d9b6d6c7d-szt7x                1/1     Running   0          102m
ebs-csi-controller-66744fbc59-pxx7w    6/6     Running   0          48m
ebs-csi-controller-66744fbc59-rd8f5    6/6     Running   0          48m
ebs-csi-node-97wwz                     3/3     Running   0          48m
ebs-csi-node-lxllz                     3/3     Running   0          48m
kube-proxy-452vr                       1/1     Running   0          90m
kube-proxy-4kc8c                       1/1     Running   0          90m
```

Check the ebs csi driver pods installed in kube-system namespace

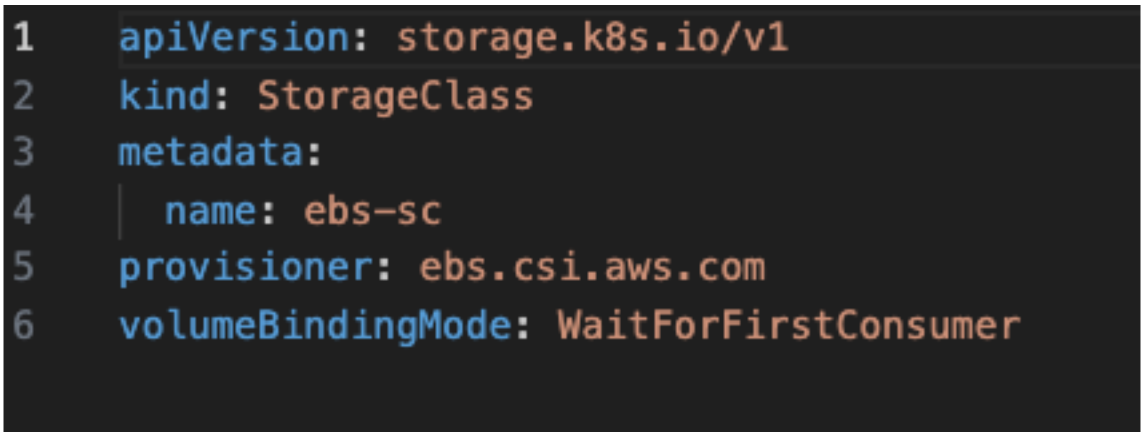
Kube manifests:



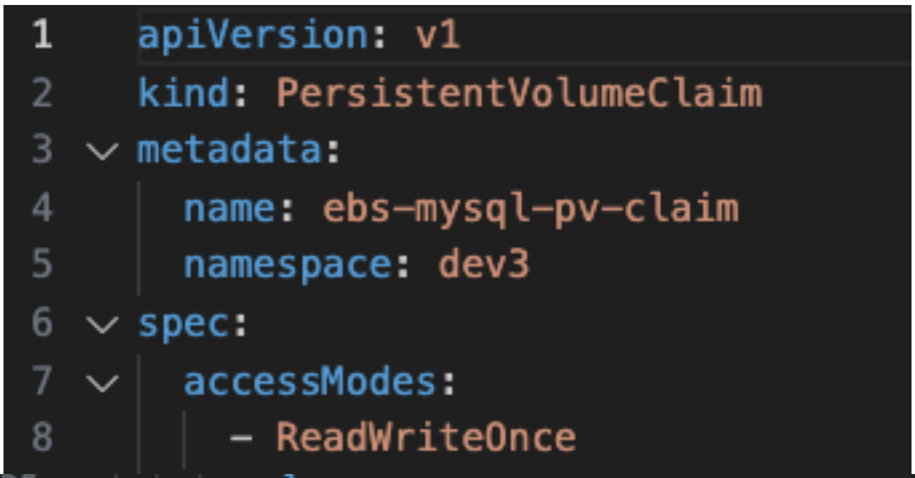
```yaml
apiVersion: v1
kind: Namespace
metadata:
  name: dev3
---
apiVersion: v1
kind: LimitRange
metadata:
  name: default-cpu-mem-limit-range
  namespace: dev3
spec:
  limits:
    - default:
        memory: "512Mi" # If not specified the Container's memory limit is set to 512Mi, which is the def
        cpu: "500m"  # If not specified default limit is 1 vCPU per container
      defaultRequest:
        memory: "256Mi" # If not specified default it will take from whatever specified in limits.default
        cpu: "300m" # If not specified default it will take from whatever specified in limits.default.cpu
      type: Container
```



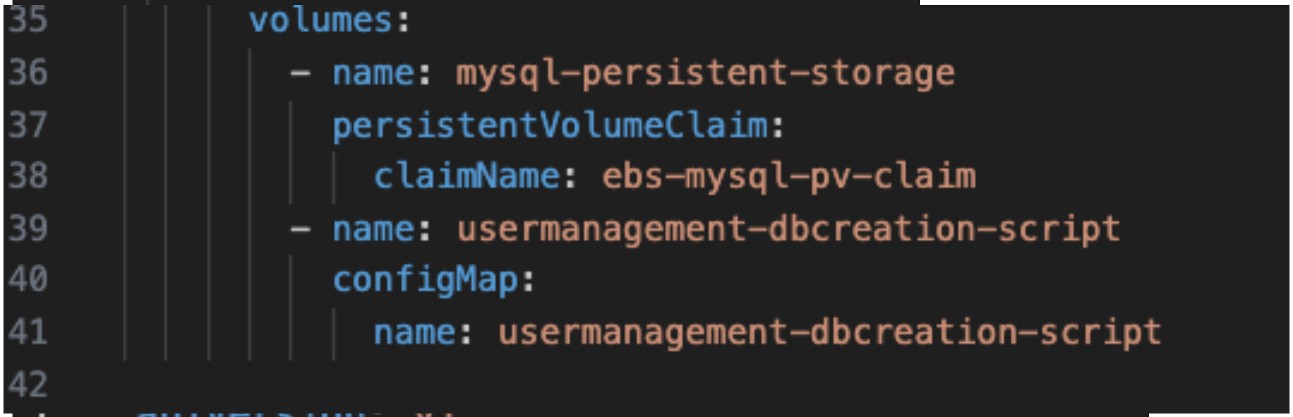
```yaml
---
apiVersion: v1
kind: ResourceQuota
metadata:
  name: ns-resource-quota
  namespace: dev3
spec:
  hard:
    requests.cpu: "1"
    requests.memory: 1Gi
    limits.cpu: "2"
    limits.memory: 2Gi
    pods: "5"
    configmaps: "5"
    persistentvolumeclaims: "5"
    secrets: "5"
    services: "5"
```



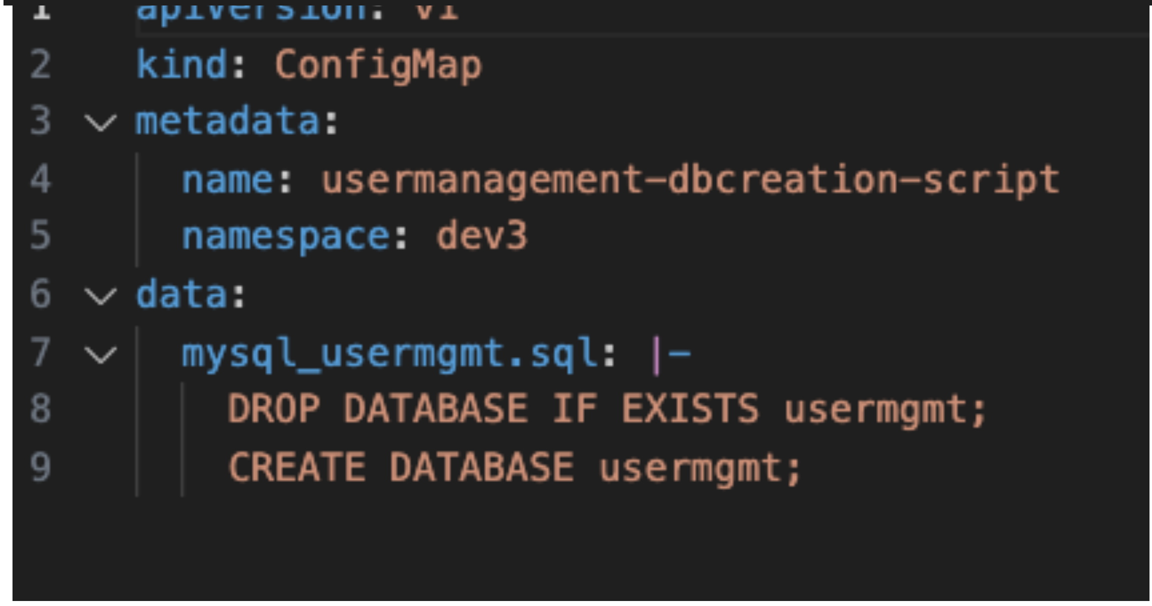
```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ebs-sc
provisioner: ebs.csi.aws.com
volumeBindingMode: WaitForFirstConsumer
```



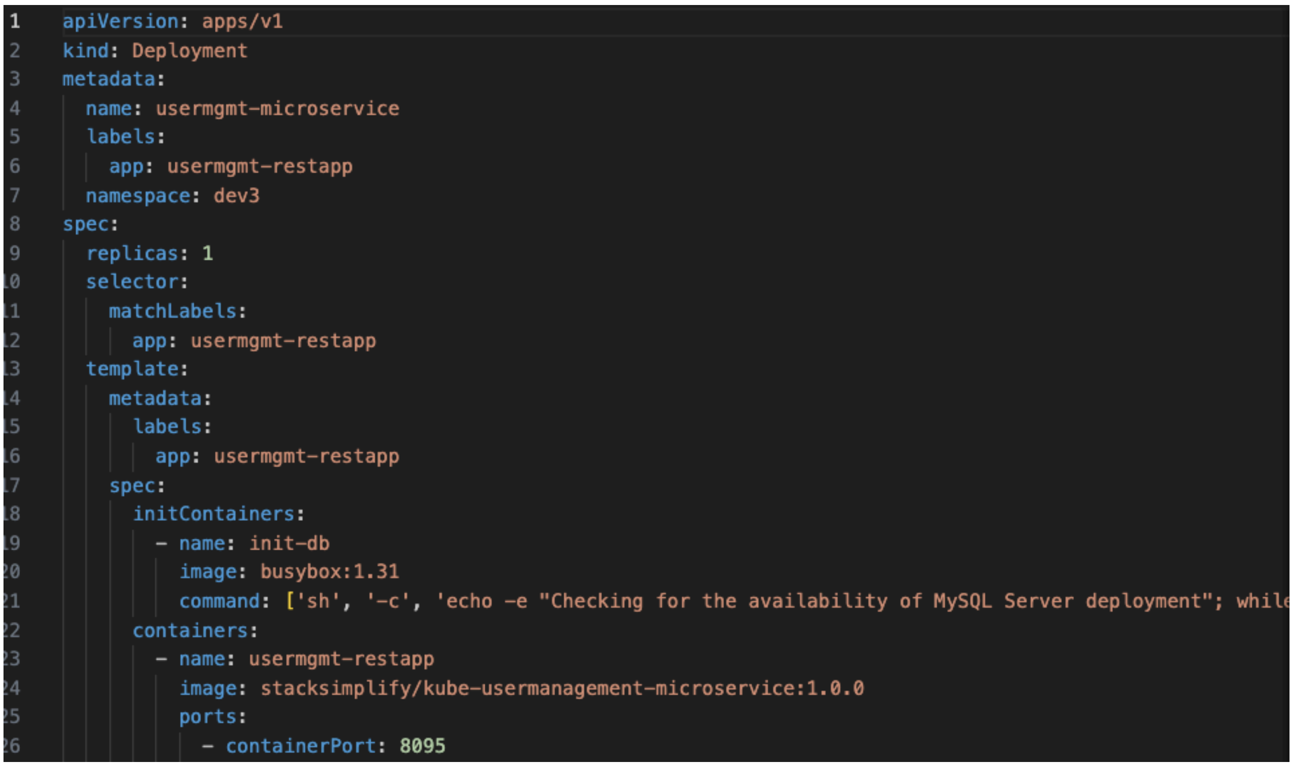
```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: ebs-mysql-pv-claim
  namespace: dev3
spec:
  accessModes:
    - ReadWriteOnce
```



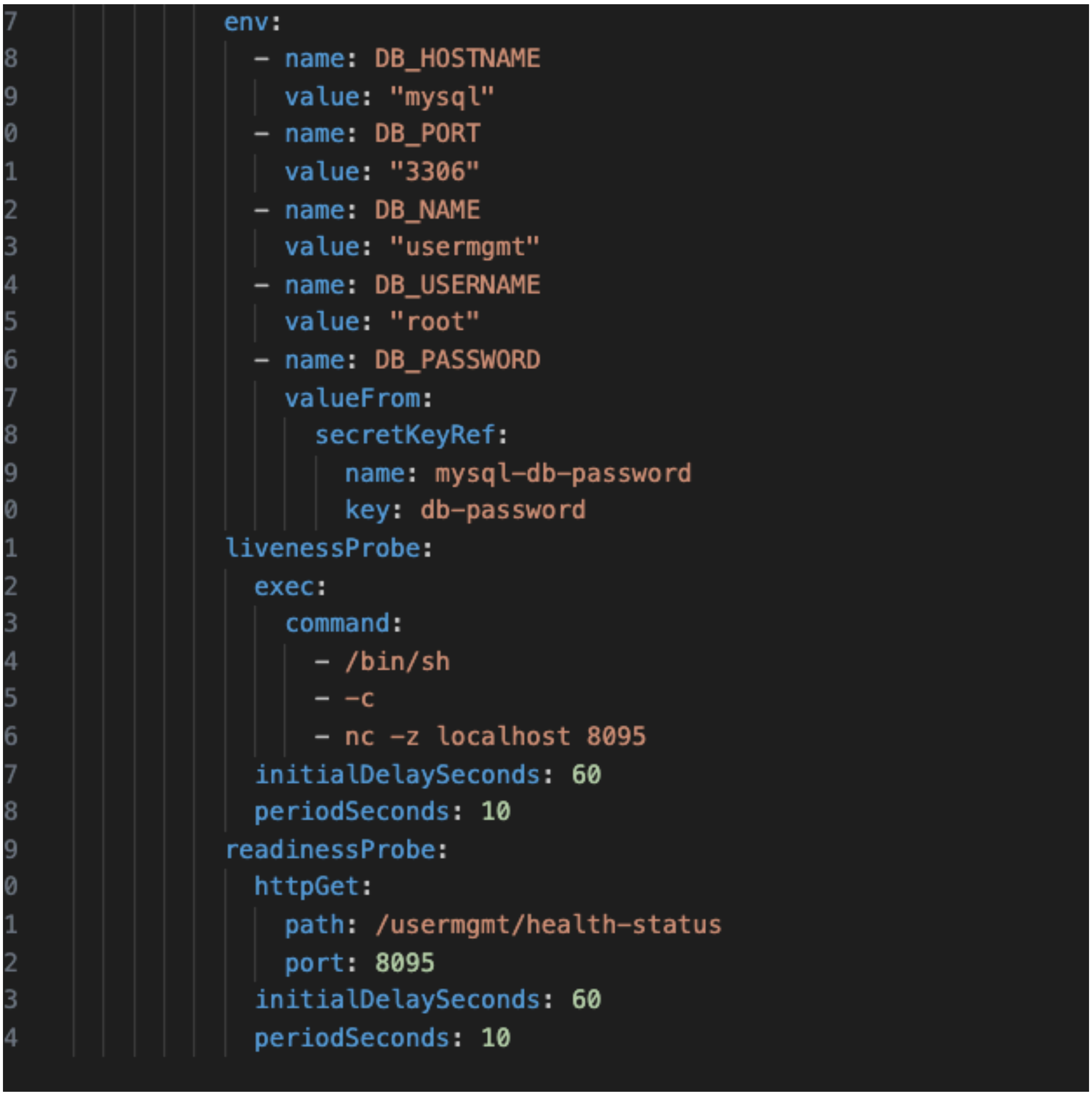
```yaml
      volumes:
        - name: mysql-persistent-storage
          persistentVolumeClaim:
            claimName: ebs-mysql-pv-claim
        - name: usermanagement-dbcreation-script
          configMap:
            name: usermanagement-dbcreation-script
```



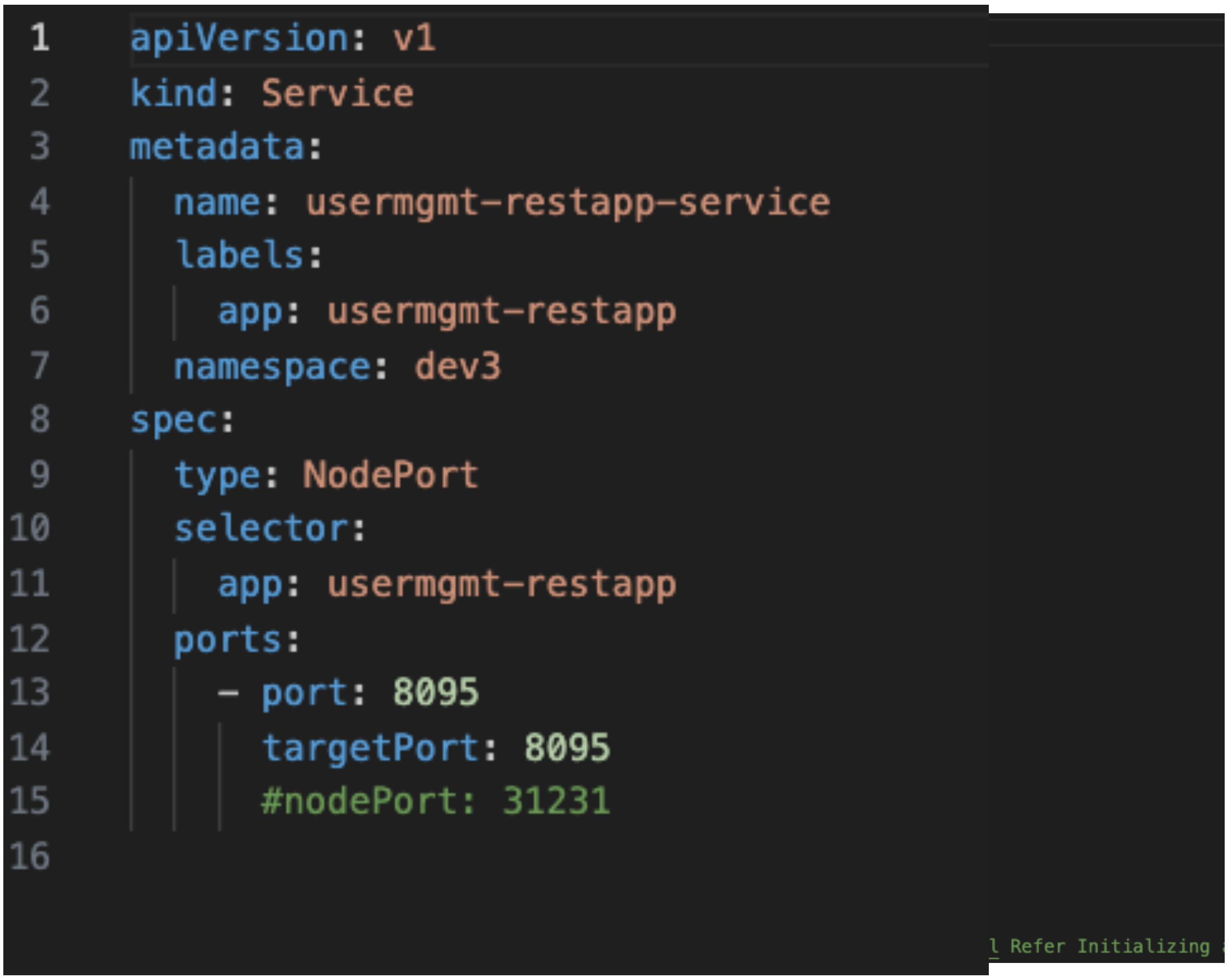
```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: usermanagement-dbcreation-script
  namespace: dev3
data:
  mysql_usermgmt.sql: |-
    DROP DATABASE IF EXISTS usermgmt;
    CREATE DATABASE usermgmt;
```



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: usermgmt-microservice
  labels:
    app: usermgmt-restapp
  namespace: dev3
spec:
  replicas: 1
  selector:
    matchLabels:
      app: usermgmt-restapp
  template:
    metadata:
      labels:
        app: usermgmt-restapp
    spec:
      initContainers:
        - name: init-db
          image: busybox:1.31
          command: ['sh', '-c', 'echo -e "Checking for the availability of MySQL Server deployment"; while
      containers:
        - name: usermgmt-restapp
          image: stacksimplify/kube-usermanagement-microservice:1.0.0
          ports:
            - containerPort: 8095
```

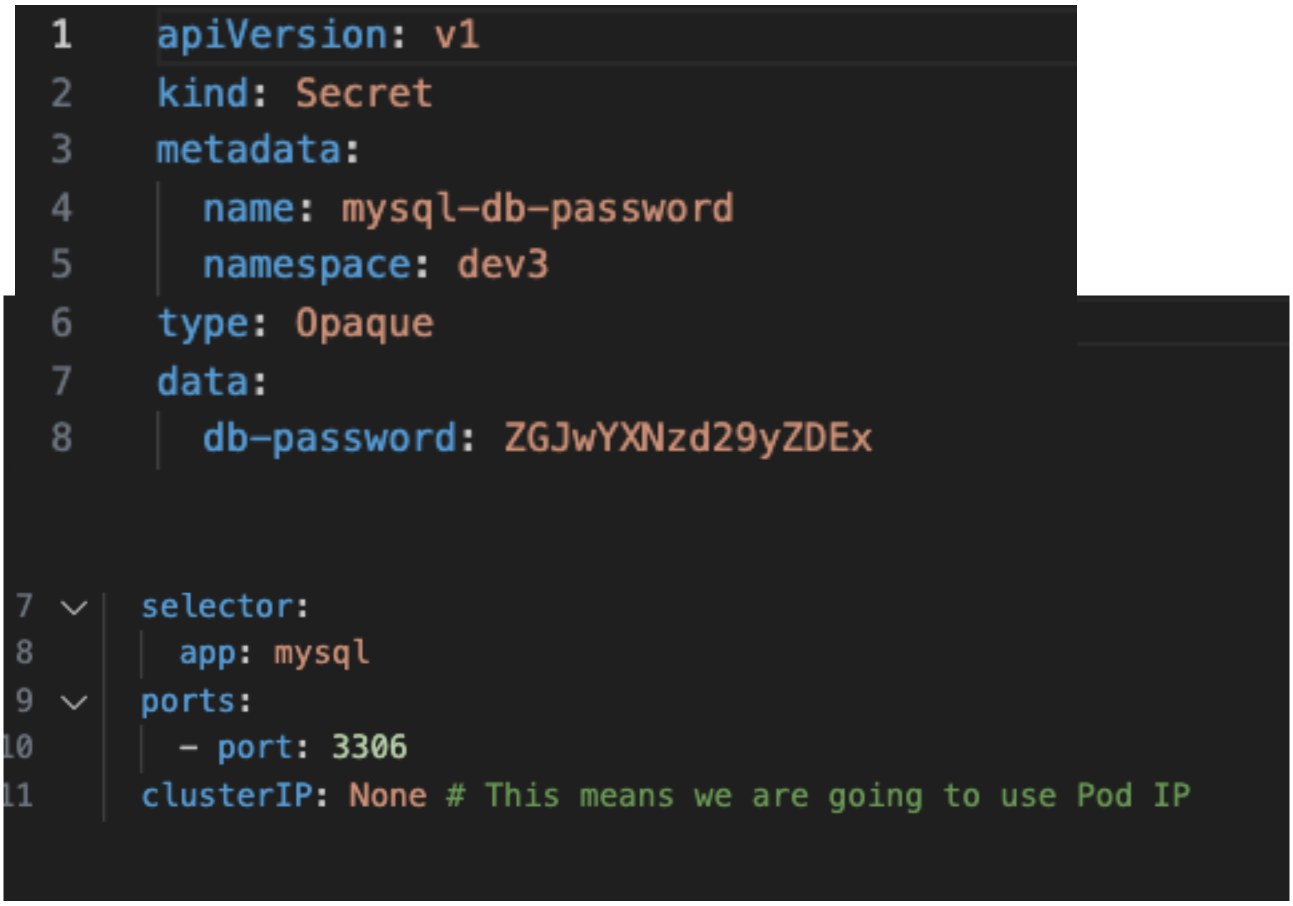


```yaml
          env:
            - name: DB_HOSTNAME
              value: "mysql"
            - name: DB_PORT
              value: "3306"
            - name: DB_NAME
              value: "usermgmt"
            - name: DB_USERNAME
              value: "root"
            - name: DB_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: mysql-db-password
                  key: db-password
          livenessProbe:
            exec:
              command:
                - /bin/sh
                - -c
                - nc -z localhost 8095
            initialDelaySeconds: 60
            periodSeconds: 10
          readinessProbe:
            httpGet:
              path: /usermgmt/health-status
              port: 8095
            initialDelaySeconds: 60
            periodSeconds: 10
```



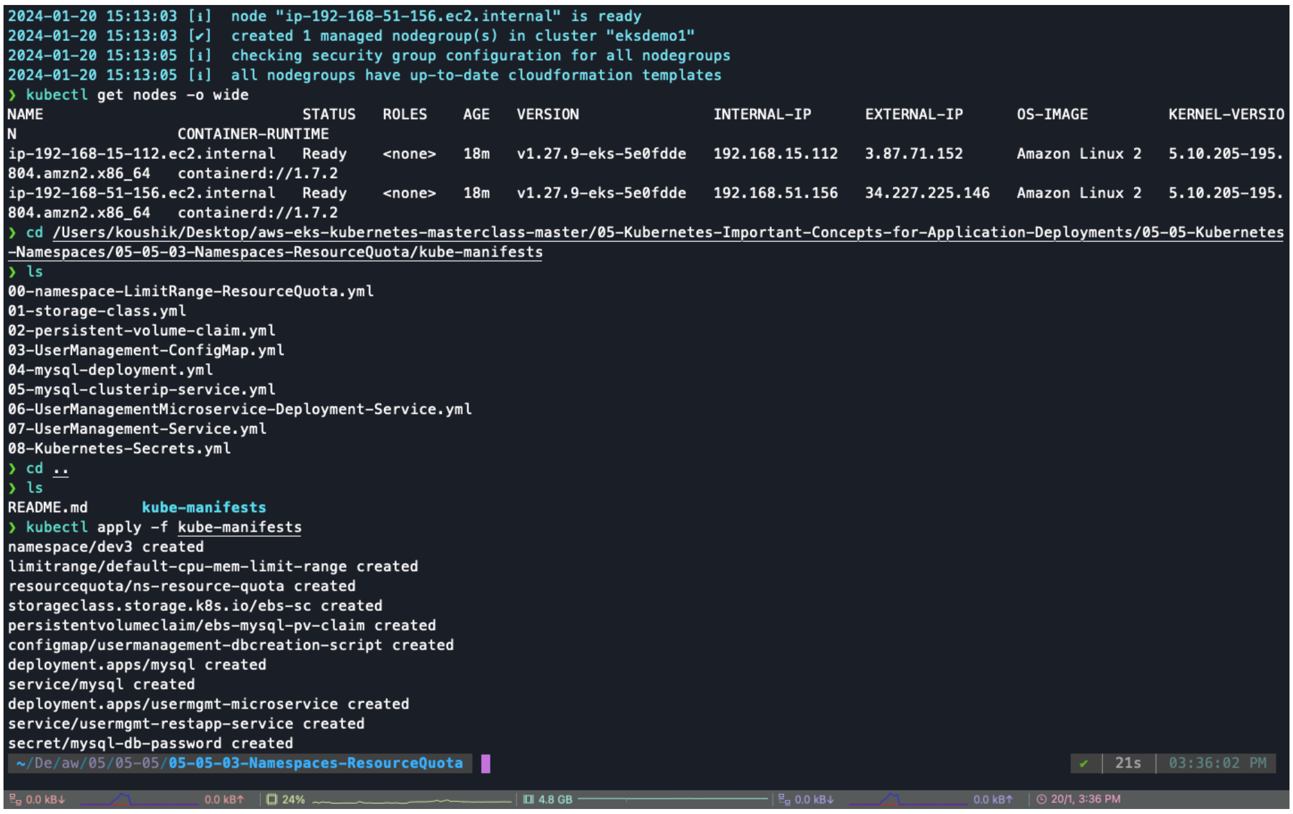
```yaml
apiVersion: v1
kind: Service
metadata:
  name: usermgmt-restapp-service
  labels:
    app: usermgmt-restapp
  namespace: dev3
spec:
  type: NodePort
  selector:
    app: usermgmt-restapp
  ports:
    - port: 8095
      targetPort: 8095
      #nodePort: 31231

```

l Refer Initializing

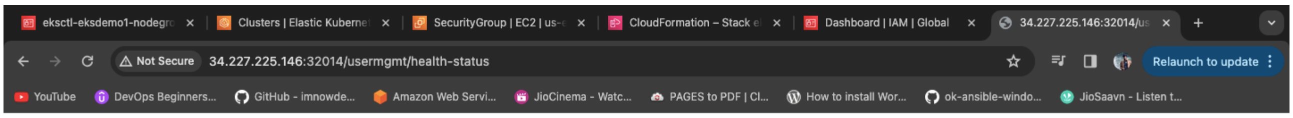


```yaml
apiVersion: v1
kind: Secret
metadata:
  name: mysql-db-password
  namespace: dev3
type: Opaque
data:
  db-password: ZGJwYXNzd29yZDEx
```

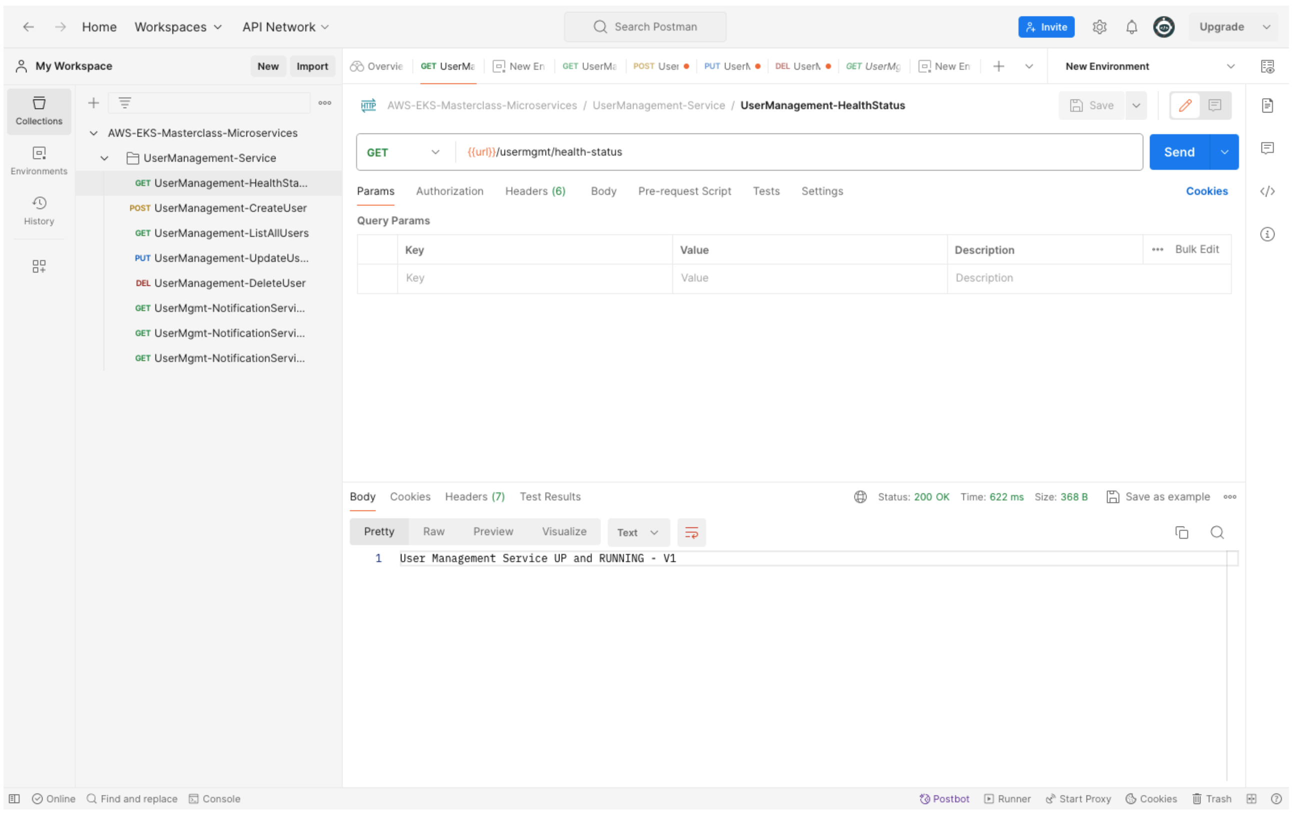
```yaml
  selector:
    app: mysql
  ports:
    - port: 3306
  clusterIP: None # This means we are going to use Pod IP
```



```
2024-01-20 15:13:03 [ℹ]  node "ip-192-168-51-156.ec2.internal" is ready
2024-01-20 15:13:03 [✔]  created 1 managed nodegroup(s) in cluster "eksdemo1"
2024-01-20 15:13:05 [ℹ]  checking security group configuration for all nodegroups
2024-01-20 15:13:05 [ℹ]  all nodegroups have up-to-date cloudformation templates
❯ kubectl get nodes -o wide
NAME                              STATUS   ROLES    AGE   VERSION               INTERNAL-IP      EXTERNAL-IP      OS-IMAGE         KERNEL-VERSIO
N                   CONTAINER-RUNTIME
ip-192-168-15-112.ec2.internal   Ready    <none>   18m   v1.27.9-eks-5e0fdde   192.168.15.112   3.87.71.152      Amazon Linux 2   5.10.205-195.
804.amzn2.x86_64   containerd://1.7.2
ip-192-168-51-156.ec2.internal   Ready    <none>   18m   v1.27.9-eks-5e0fdde   192.168.51.156   34.227.225.146   Amazon Linux 2   5.10.205-195.
804.amzn2.x86_64   containerd://1.7.2
❯ cd /Users/koushik/Desktop/aws-eks-kubernetes-masterclass-master/05-Kubernetes-Important-Concepts-for-Application-Deployments/05-05-Kubernetes
-Namespaces/05-05-03-Namespaces-ResourceQuota/kube-manifests
❯ ls
00-namespace-LimitRange-ResourceQuota.yml
01-storage-class.yml
02-persistent-volume-claim.yml
03-UserManagement-ConfigMap.yml
04-mysql-deployment.yml
05-mysql-clusterip-service.yml
06-UserManagementMicroservice-Deployment-Service.yml
07-UserManagement-Service.yml
08-Kubernetes-Secrets.yml
❯ cd ..
❯ ls
README.md       kube-manifests
❯ kubectl apply -f kube-manifests
namespace/dev3 created
limitrange/default-cpu-mem-limit-range created
resourcequota/ns-resource-quota created
storageclass.storage.k8s.io/ebs-sc created
persistentvolumeclaim/ebs-mysql-pv-claim created
configmap/usermanagement-dbcreation-script created
deployment.apps/mysql created
service/mysql created
deployment.apps/usermgmt-microservice created
service/usermgmt-restapp-service created
secret/mysql-db-password created
 ~/De/aw/05/05-05/05-05-03-Namespaces-ResourceQuota                                                                    ✔ │ 21s │ 03:36:02 PM
```



User Management Service UP and RUNNING - V1



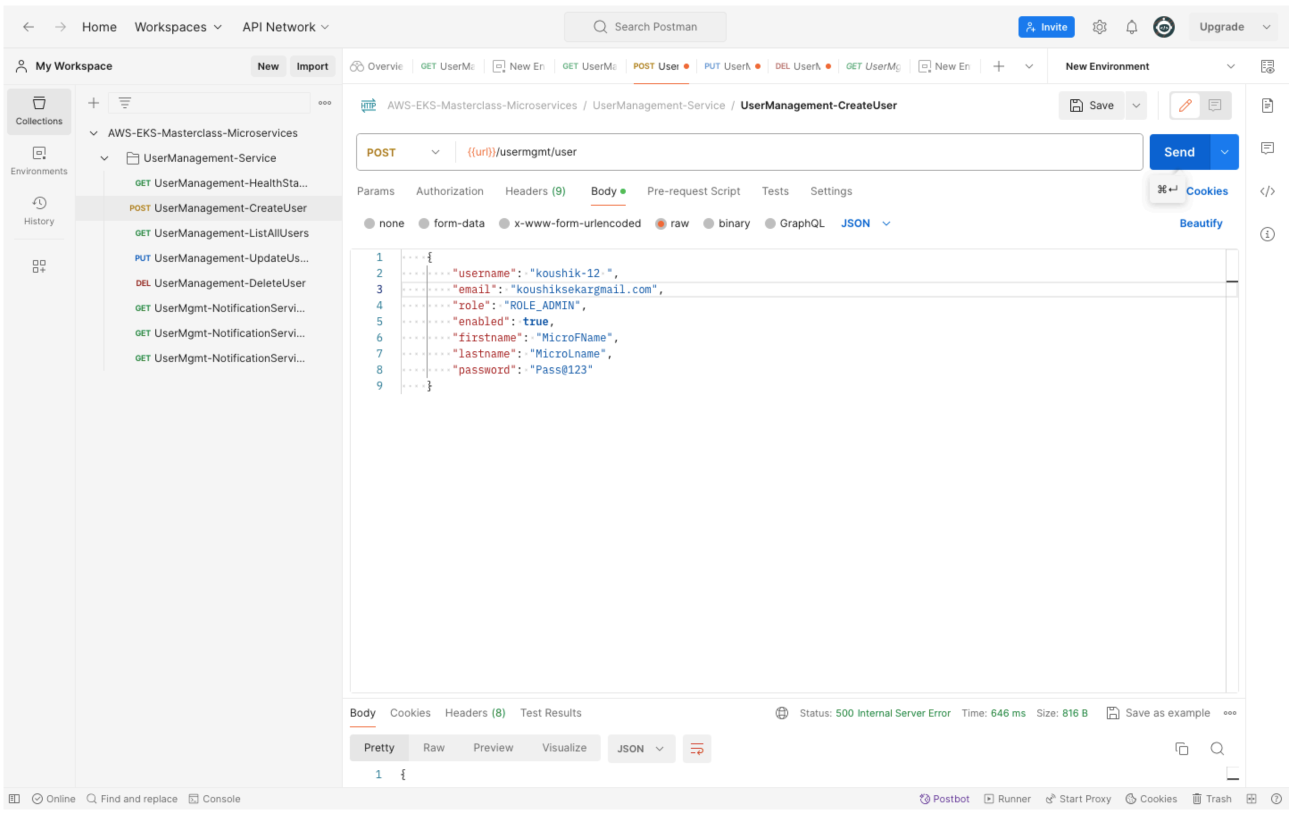
AWS-EKS-Masterclass-Microservices
- UserManagement-Service
  - GET UserManagement-HealthSta...
  - POST UserManagement-CreateUser
  - GET UserManagement-ListAllUsers
  - PUT UserManagement-UpdateUs...
  - DEL UserManagement-DeleteUser
  - GET UserMgmt-NotificationServi...
  - GET UserMgmt-NotificationServi...
  - GET UserMgmt-NotificationServi...

AWS-EKS-Masterclass-Microservices / UserManagement-Service / UserManagement-HealthStatus

GET {{url}}/usermgmt/health-status

Body | Cookies | Headers (7) | Test Results — Status: 200 OK Time: 622 ms Size: 368 B

```
User Management Service UP and RUNNING - V1
```



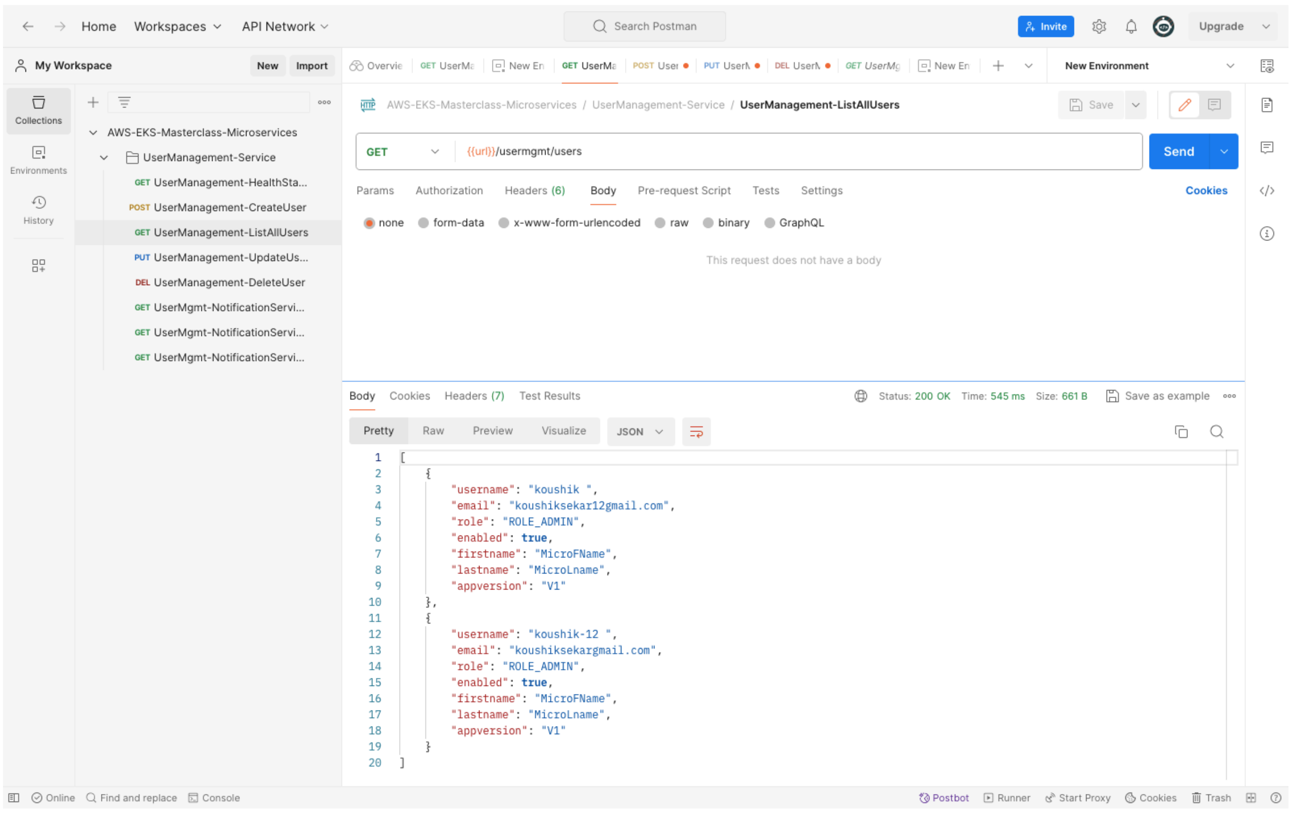
AWS-EKS-Masterclass-Microservices / UserManagement-Service / UserManagement-CreateUser

POST {{url}}/usermgmt/user

```
{
    "username": "koushik-12 ",
    "email": "koushiksekargmail.com",
    "role": "ROLE_ADMIN",
    "enabled": true,
    "firstname": "MicroFName",
    "lastname": "MicroLname",
    "password": "Pass@123"
}
```

Body | Cookies | Headers (8) | Test Results — Status: 500 Internal Server Error Time: 646 ms Size: 816 B

```
{
```



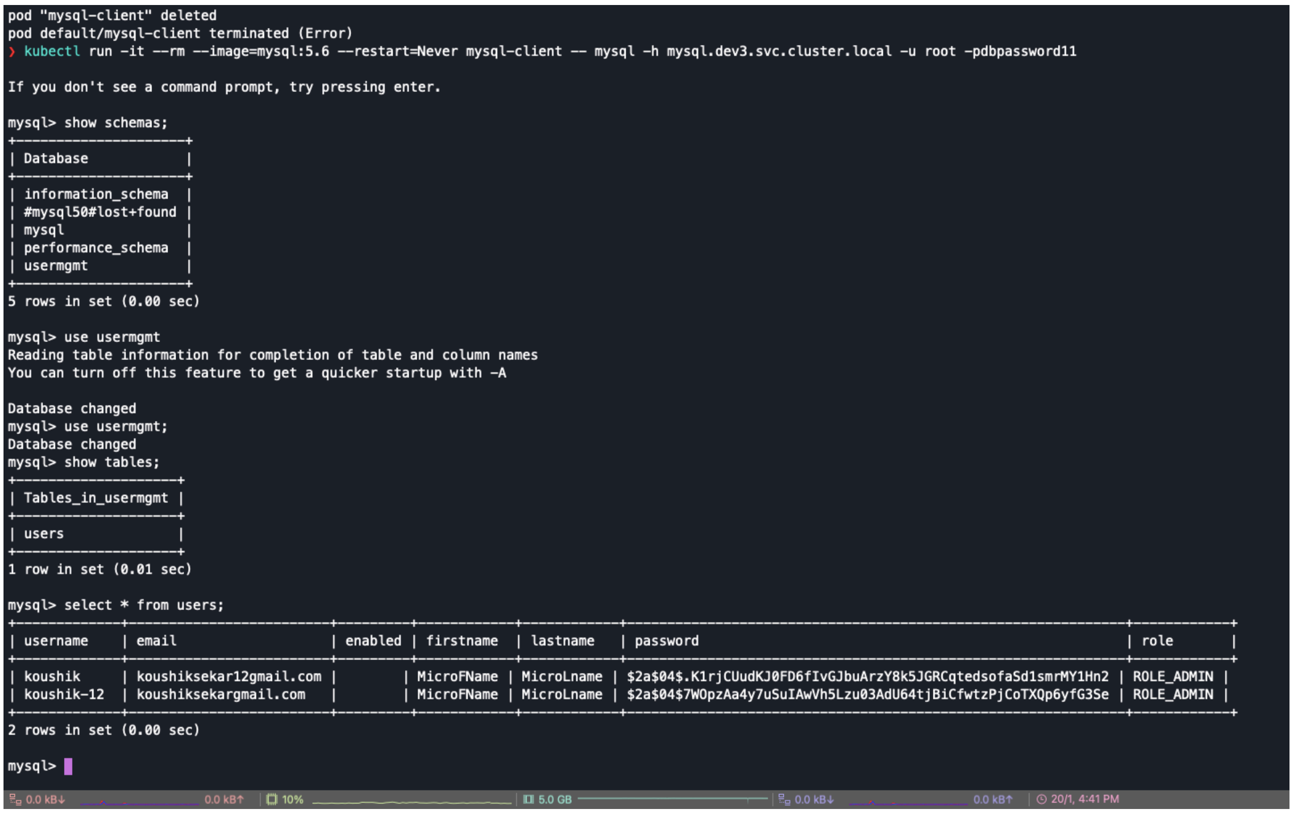
AWS-EKS-Masterclass-Microservices / UserManagement-Service / UserManagement-ListAllUsers

GET {{url}}/usermgmt/users

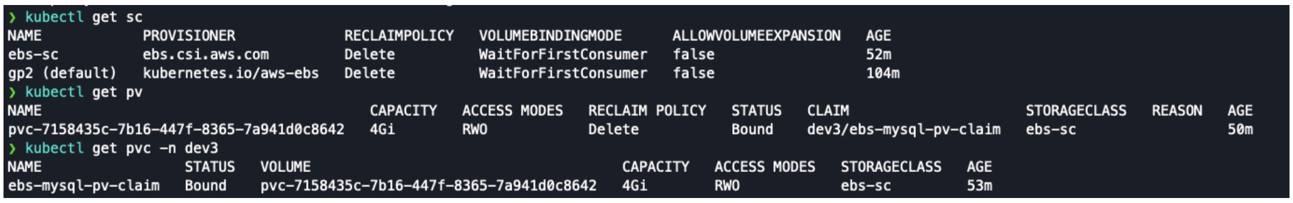
This request does not have a body

Body Cookies Headers (7) Test Results   Status: 200 OK Time: 545 ms Size: 661 B

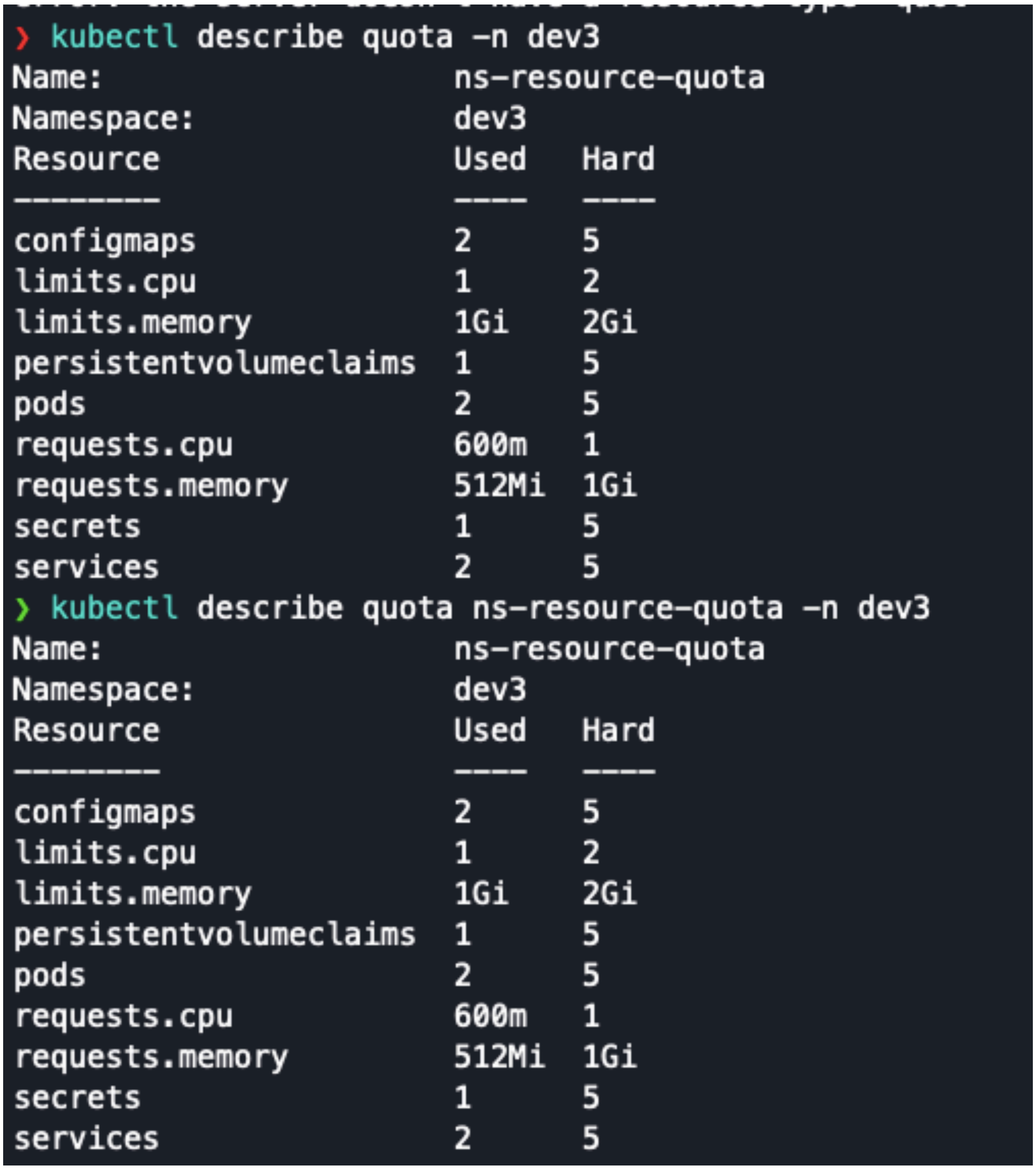
```
[
    {
        "username": "koushik ",
        "email": "koushiksekar12gmail.com",
        "role": "ROLE_ADMIN",
        "enabled": true,
        "firstname": "MicroFName",
        "lastname": "MicroLname",
        "appversion": "V1"
    },
    {
        "username": "koushik-12 ",
        "email": "koushiksekargmail.com",
        "role": "ROLE_ADMIN",
        "enabled": true,
        "firstname": "MicroFName",
        "lastname": "MicroLname",
        "appversion": "V1"
    }
]
```



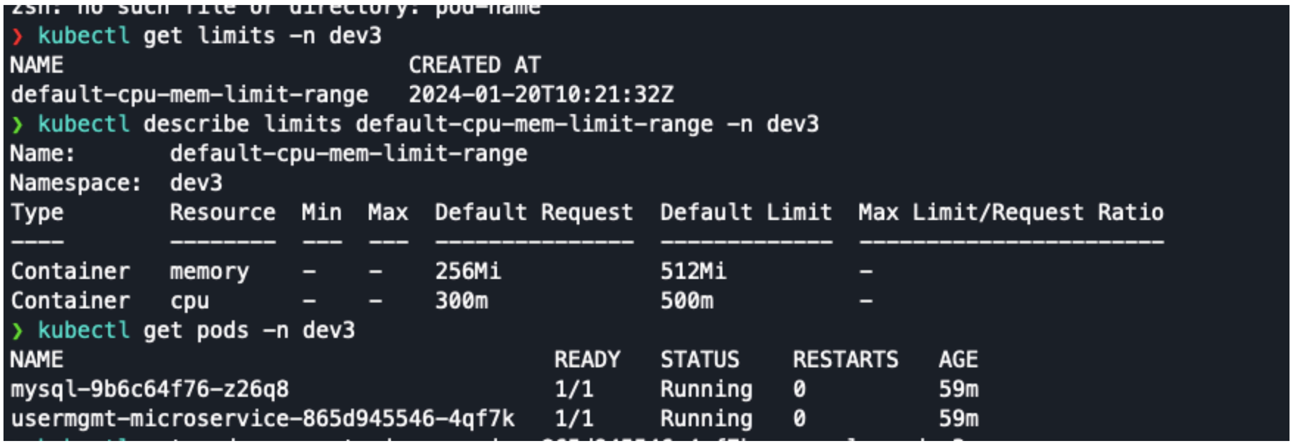
```
pod "mysql-client" deleted
pod default/mysql-client terminated (Error)
❯ kubectl run -it --rm --image=mysql:5.6 --restart=Never mysql-client -- mysql -h mysql.dev3.svc.cluster.local -u root -pdbpassword11

If you don't see a command prompt, try pressing enter.

mysql> show schemas;
+---------------------+
| Database            |
+---------------------+
| information_schema  |
| #mysql50#lost+found |
| mysql               |
| performance_schema  |
| usermgmt            |
+---------------------+
5 rows in set (0.00 sec)

mysql> use usermgmt
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> use usermgmt;
Database changed
mysql> show tables;
+--------------------+
| Tables_in_usermgmt |
+--------------------+
| users              |
+--------------------+
1 row in set (0.01 sec)

mysql> select * from users;
+------------+-------------------------+---------+------------+-----------+--------------------------------------------------------------+------------+
| username   | email                   | enabled | firstname  | lastname  | password                                                     | role       |
+------------+-------------------------+---------+------------+-----------+--------------------------------------------------------------+------------+
| koushik    | koushiksekar12gmail.com |         | MicroFName | MicroLname | $2a$04$.K1rjCUudKJ0FD6fIvGJbuArzY8k5JGRCqtedsofaSd1smrMY1Hn2 | ROLE_ADMIN |
| koushik-12 | koushiksekargmail.com   |         | MicroFName | MicroLname | $2a$04$7WOpzAa4y7uSuIAwVh5Lzu03AdU64tjBiCfwtzPjCoTXQp6yfG3Se | ROLE_ADMIN |
+------------+-------------------------+---------+------------+-----------+--------------------------------------------------------------+------------+
2 rows in set (0.00 sec)

mysql> 
```



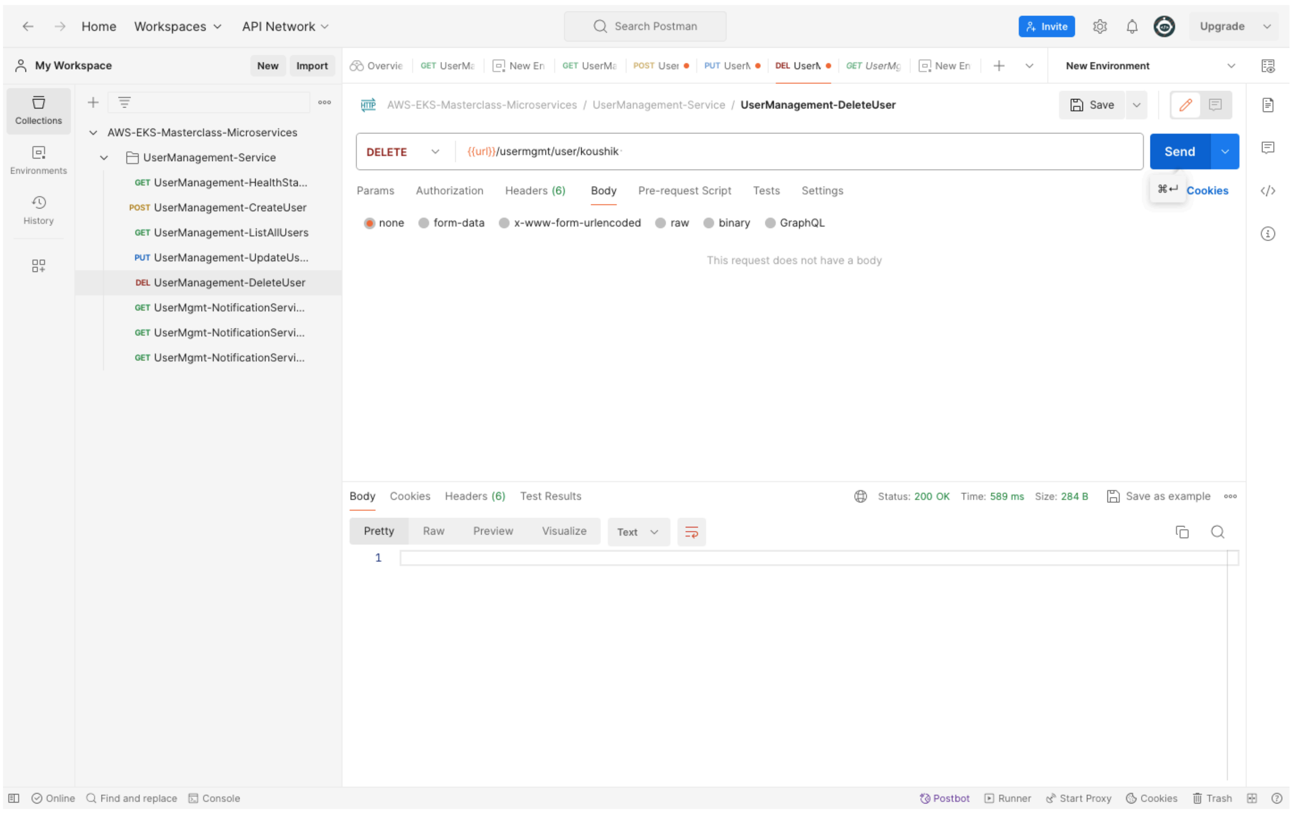
```
❯ kubectl get sc
NAME            PROVISIONER             RECLAIMPOLICY   VOLUMEBINDINGMODE      ALLOWVOLUMEEXPANSION   AGE
ebs-sc          ebs.csi.aws.com         Delete          WaitForFirstConsumer   false                  52m
gp2 (default)   kubernetes.io/aws-ebs   Delete          WaitForFirstConsumer   false                  104m
❯ kubectl get pv
NAME                                       CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM                     STORAGECLASS   REASON   AGE
pvc-7158435c-7b16-447f-8365-7a941d0c8642   4Gi        RWO            Delete           Bound    dev3/ebs-mysql-pv-claim   ebs-sc                  50m
❯ kubectl get pvc -n dev3
NAME                 STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
ebs-mysql-pv-claim   Bound    pvc-7158435c-7b16-447f-8365-7a941d0c8642   4Gi        RWO            ebs-sc         53m
```



```
❯ kubectl describe quota -n dev3
Name:                   ns-resource-quota
Namespace:              dev3
Resource                Used   Hard
--------                ----   ----
configmaps              2      5
limits.cpu              1      2
limits.memory           1Gi    2Gi
persistentvolumeclaims  1      5
pods                    2      5
requests.cpu            600m   1
requests.memory         512Mi  1Gi
secrets                 1      5
services                2      5
❯ kubectl describe quota ns-resource-quota -n dev3
Name:                   ns-resource-quota
Namespace:              dev3
Resource                Used   Hard
--------                ----   ----
configmaps              2      5
limits.cpu              1      2
limits.memory           1Gi    2Gi
persistentvolumeclaims  1      5
pods                    2      5
requests.cpu            600m   1
requests.memory         512Mi  1Gi
secrets                 1      5
services                2      5
```



```
zsh: no such file or directory: pod-name
❯ kubectl get limits -n dev3
NAME                           CREATED AT
default-cpu-mem-limit-range    2024-01-20T10:21:32Z
❯ kubectl describe limits default-cpu-mem-limit-range -n dev3
Name:       default-cpu-mem-limit-range
Namespace:  dev3
Type        Resource  Min  Max  Default Request  Default Limit  Max Limit/Request Ratio
----        --------  ---  ---  ---------------  -------------  -----------------------
Container   memory    -    -    256Mi            512Mi          -
Container   cpu       -    -    300m             500m           -
❯ kubectl get pods -n dev3
NAME                                     READY   STATUS    RESTARTS   AGE
mysql-9b6c64f76-z26q8                    1/1     Running   0          59m
usermgmt-microservice-865d945546-4qf7k   1/1     Running   0          59m
```

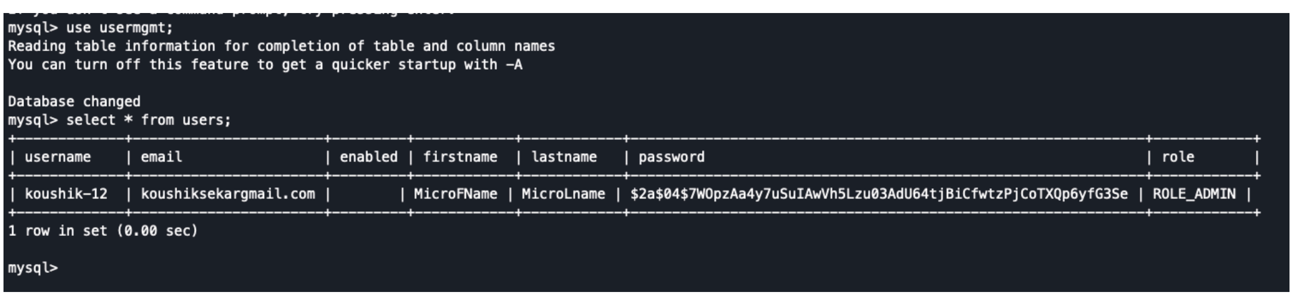


AWS-EKS-Masterclass-Microservices / UserManagement-Service / UserManagement-DeleteUser

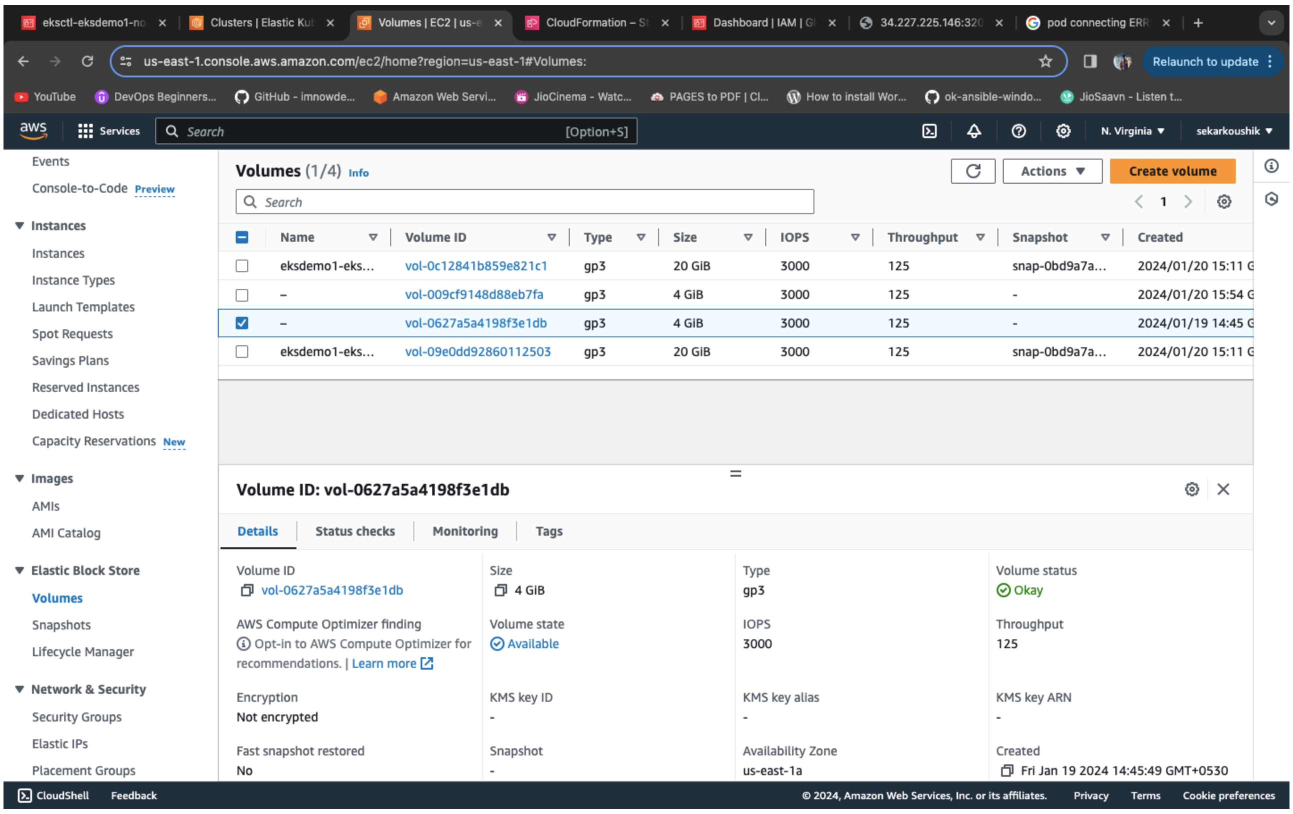
DELETE {{url}}/usermgmt/user/koushik

This request does not have a body

Body Cookies Headers (6) Test Results — Status: 200 OK Time: 589 ms Size: 284 B



```
mysql> use usermgmt;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from users;
+------------+---------------------+---------+------------+-----------+--------------------------------------------------------------+------------+
| username   | email               | enabled | firstname  | lastname  | password                                                     | role       |
+------------+---------------------+---------+------------+-----------+--------------------------------------------------------------+------------+
| koushik-12 | koushiksekargmail.com |       | MicroFName | MicroLname | $2a$04$7WOpzAa4y7uSuIAwVh5Lzu03AdU64tjBiCfwtzPjCoTXQp6yfG3Se | ROLE_ADMIN |
+------------+---------------------+---------+------------+-----------+--------------------------------------------------------------+------------+
1 row in set (0.00 sec)

mysql>
```



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Volumes:

Volumes (1/4) Info

	Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot	Created
☐	eksdemo1-eks...	vol-0c12841b859e821c1	gp3	20 GiB	3000	125	snap-0bd9a7a...	2024/01/20 15:11 G
☐	–	vol-009cf9148d88eb7fa	gp3	4 GiB	3000	125	-	2024/01/20 15:54 G
✓	–	vol-0627a5a4198f3e1db	gp3	4 GiB	3000	125	-	2024/01/19 14:45 G
☐	eksdemo1-eks...	vol-09e0dd92860112503	gp3	20 GiB	3000	125	snap-0bd9a7a...	2024/01/20 15:11 G

Volume ID: vol-0627a5a4198f3e1db

Details | Status checks | Monitoring | Tags

Volume ID vol-0627a5a4198f3e1db	Size 4 GiB	Type gp3	Volume status Okay
AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. | Learn more	Volume state Available	IOPS 3000	Throughput 125
Encryption Not encrypted	KMS key ID -	KMS key alias -	KMS key ARN -
Fast snapshot restored No	Snapshot -	Availability Zone us-east-1a	Created Fri Jan 19 2024 14:45:49 GMT+0530