**9. Test User Management Microservice UMS using Postman**

--- **note** - in this lecture, we are going to test our user management microservice using postman.

**Test User Management Microservice using Postman**

**Download Postman client**

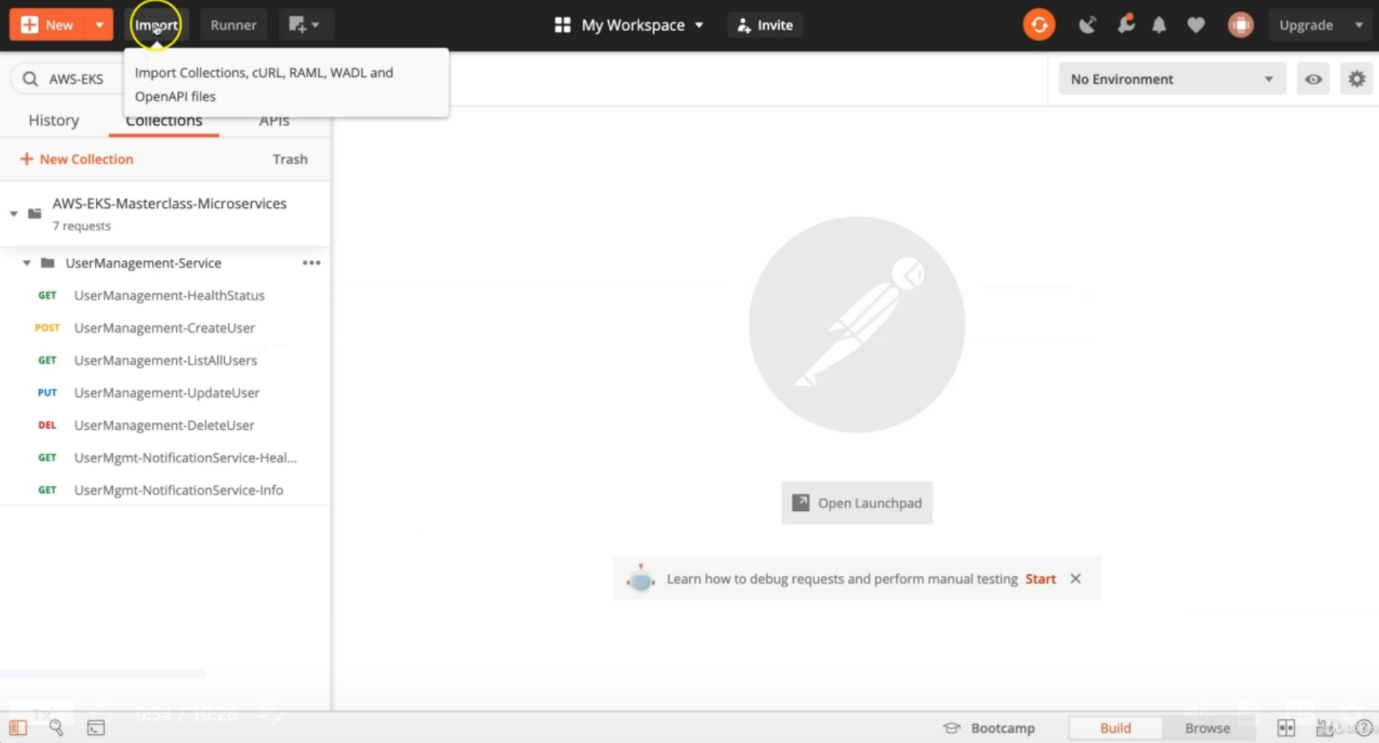
--- <https://www.postman.com/downloads/> - from here you can download postman client on your local instance.

--- **Import Project to Postman** - Import the postman project AWS-EKS-Masterclass-Microservices.postman\_collection.json present in folder 04-03-UserManagement-MicroService-with-MySQLDB

**Create Environment in postman**

--- Reference - <https://github.com/stacksimplify/aws-eks-kubernetes-masterclass/blob/master/04-EKS-Storage-with-EBS-ElasticBlockStore/04-03-UserManagement-MicroService-with-MySQLDB/AWS-EKS-Masterclass-Microservices.postman_collection.json>

**Import file in postman**



--- **note** – import the above-mentioned file to postman.

**UserManagement-HealthService**

--- Go to Settings -> Click on Add

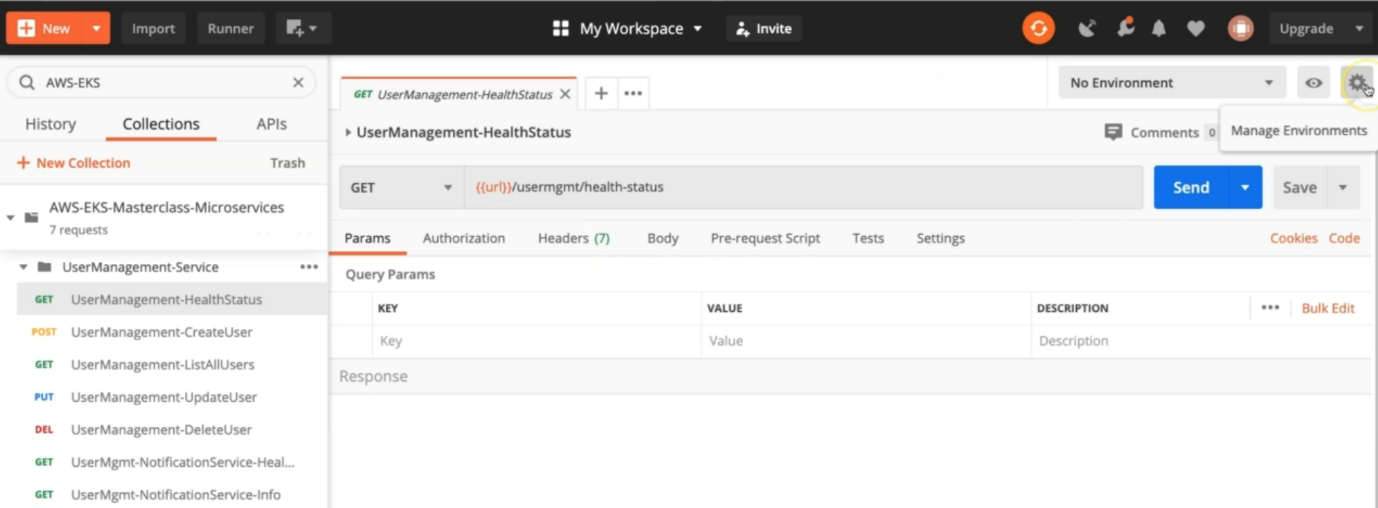
Environment Name: UMS-NodePort

Variable: url

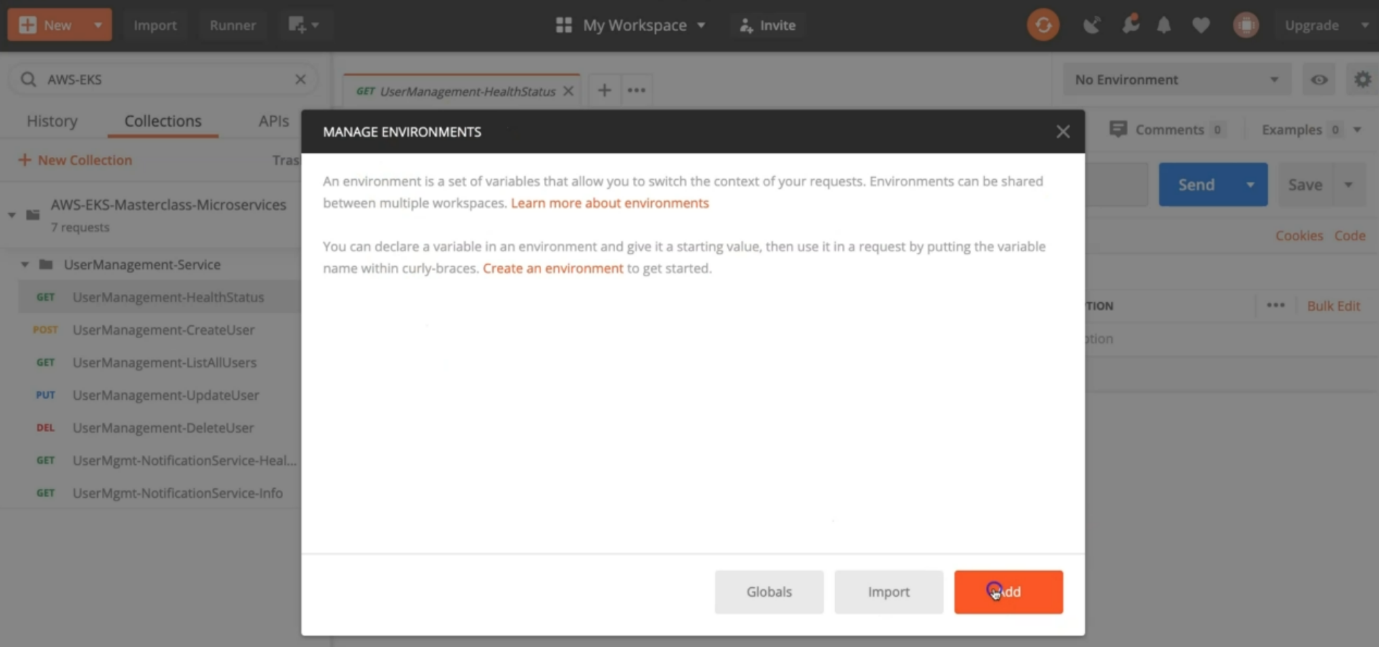
Initial Value: http://WorkerNode-Public-IP:31231

Current Value: <http://WorkerNode-Public-IP:31231>

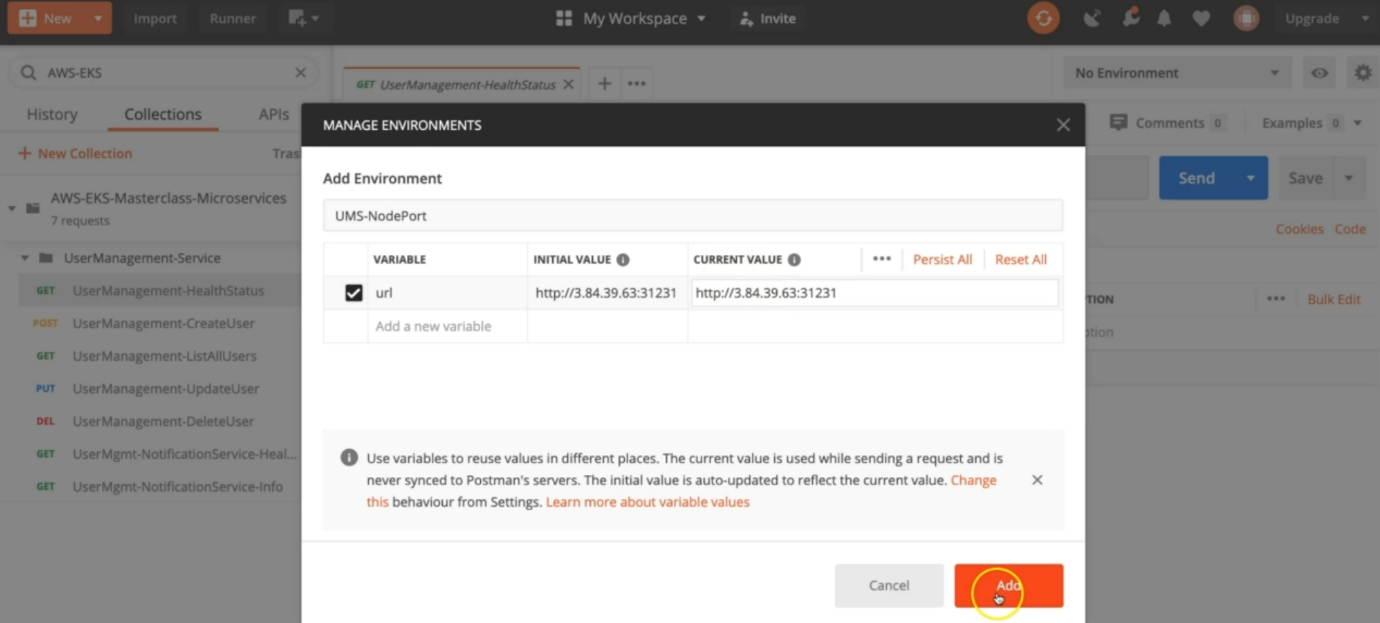
--- Click on Add



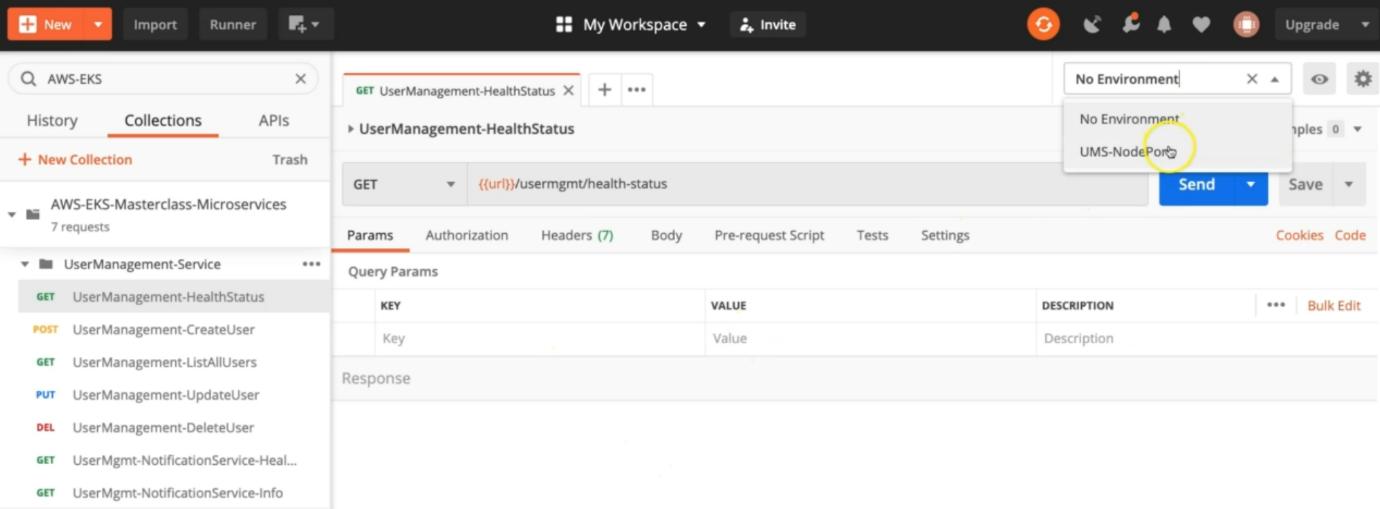
--- for this we need to go to the setting here.



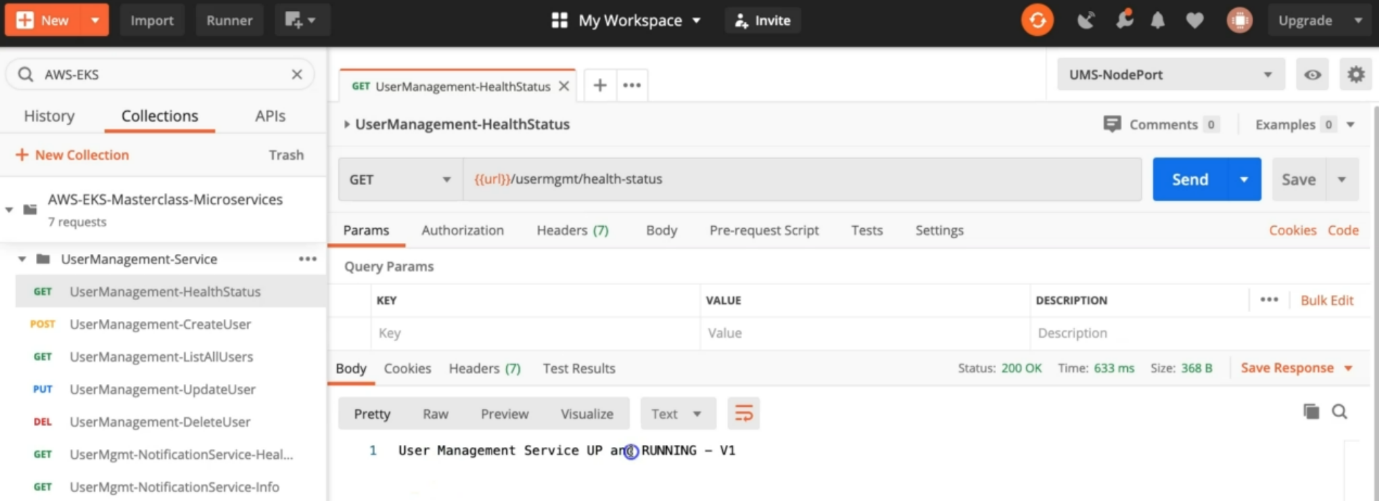
--- click on add option.



--- after giving the details, click on add.

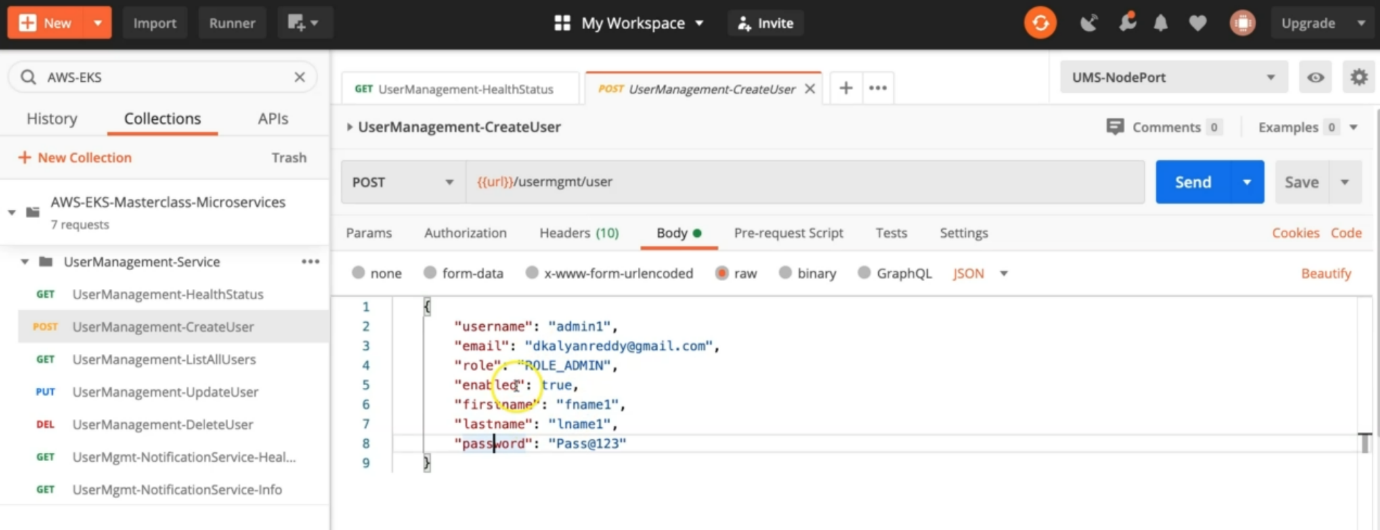


--- select the UMS-Nodeport and click on send option.



--- **note** – the request is sent to respective container present inside of kubernetes cluster.

**UserManagement-CreateUser**



    {

        "username": "admin1",

        "email": "dkalyanreddy@gmail.com",

        "role": "ROLE\_ADMIN",

        "enabled": true,

        "firstname": "fname1",

        "lastname": "lname1",

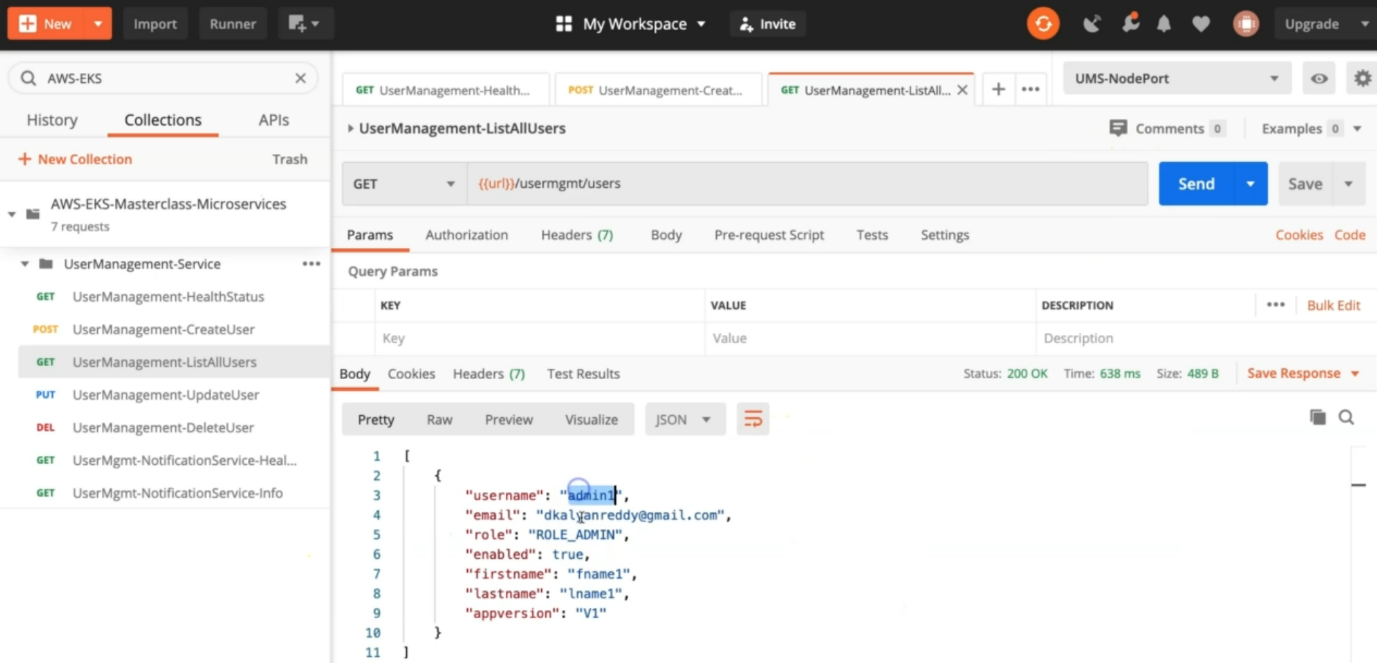
        "password": "Pass@123"

    }

--- **note** – in the UserManagement-CreateUser body, there is block of code. What this block of code means that using this block of code. We will send an api request for creating user in mysql db.

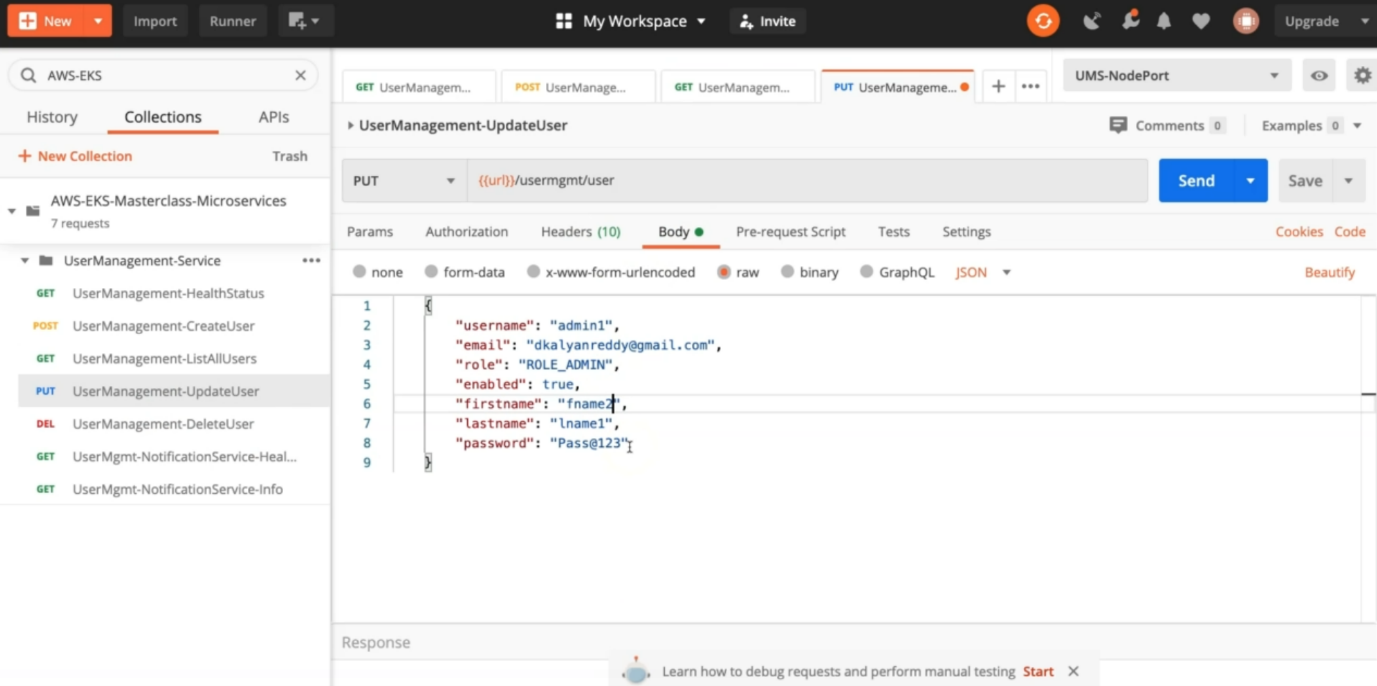
--- click on send option. Option api request sent to the mysql db, a user will be created in the mysql db.

**UserManagement-ListAllUsers**



--- now you can see that admin1 user got created along with other details.

**UserManagement-UpdateUser**



--- **note** - you want to update the first name of admin1 then you can do this way.

    {

        "username": "admin1",

        "email": "dkalyanreddy@gmail.com",

        "role": "ROLE\_ADMIN",

        "enabled": true,

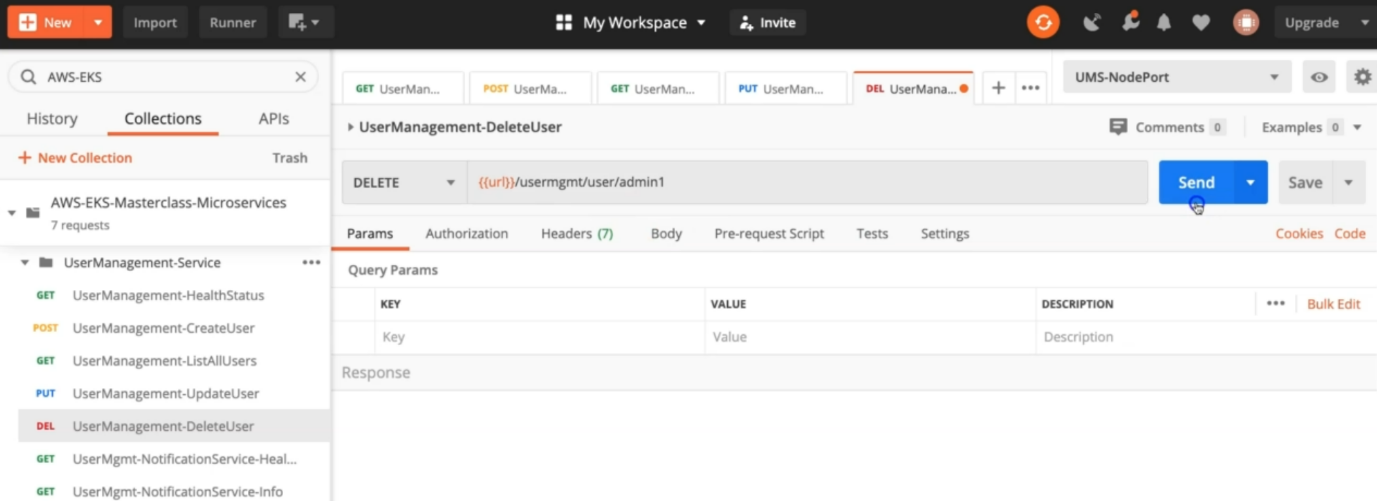
        "firstname": "fname2",

        "lastname": "lname2",

        "password": "Pass@123"

    }

**UserManagement-DeleteUser**



**Verify Users in MySQL Database**

**# Connect to MYSQL Database**

--- **kubectl run -it --rm --image=mysql:5.6 --restart=Never mysql-client -- mysql -h mysql -pdbpassword11**

**# Verify usermgmt schema got created which we provided in ConfigMap**

--- mysql> show schemas; - what are the databases available.

--- mysql> use usermgmt; -

--- mysql> show tables;

--- mysql> select \* from users; - it will shows the users, which are created in mysql db.

**Clean-Up**

**Delete all k8s objects created as part of this section**

**# Delete All**

--- **kubectl delete -f kube-manifests/**

**# List Pods**

--- **kubectl get pods**

**# Verify sc, pvc, pv**

--- **kubectl get sc,pvc,pv**