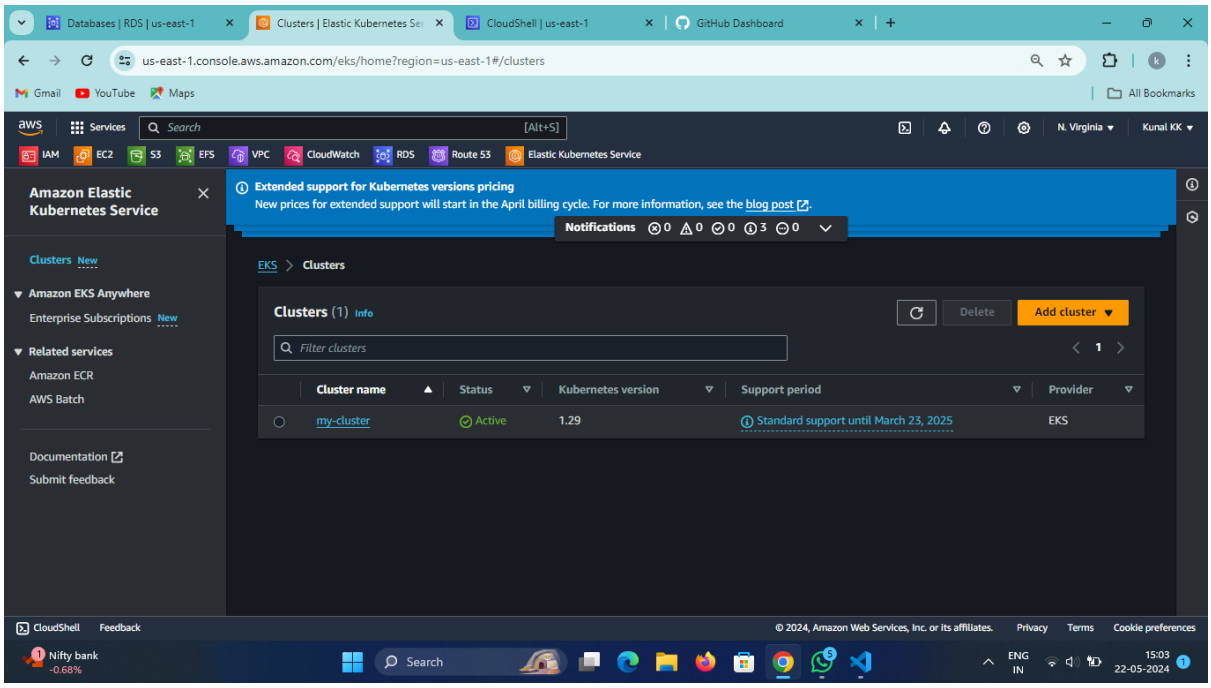
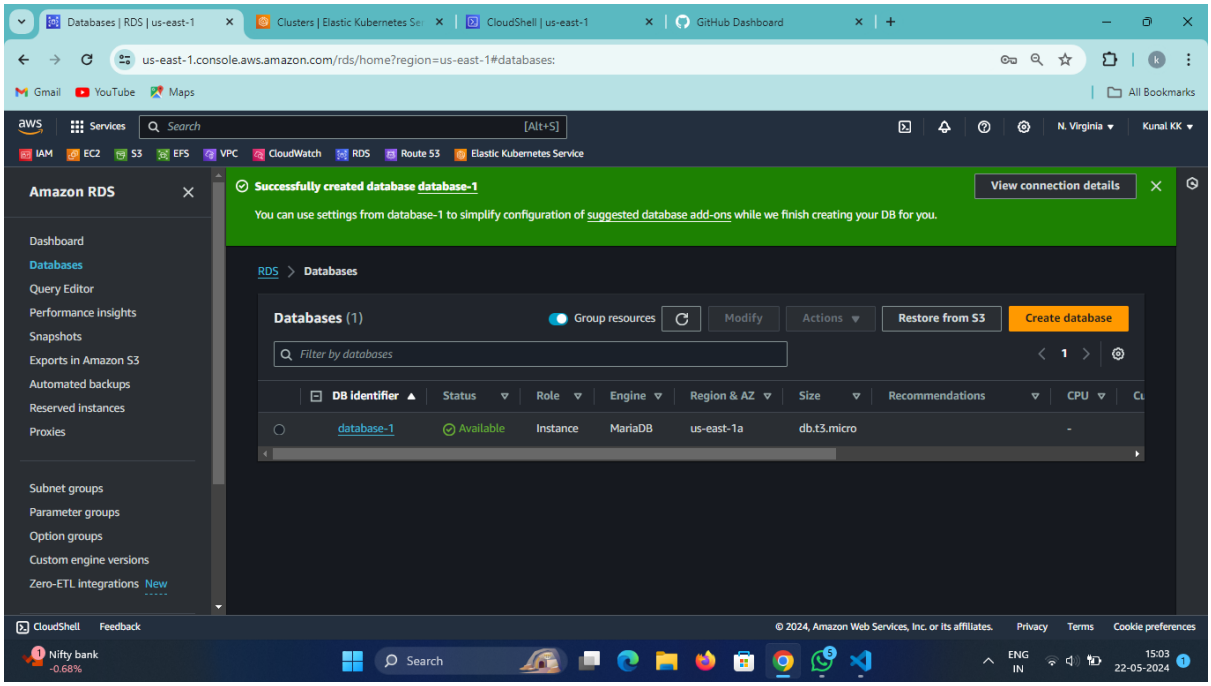
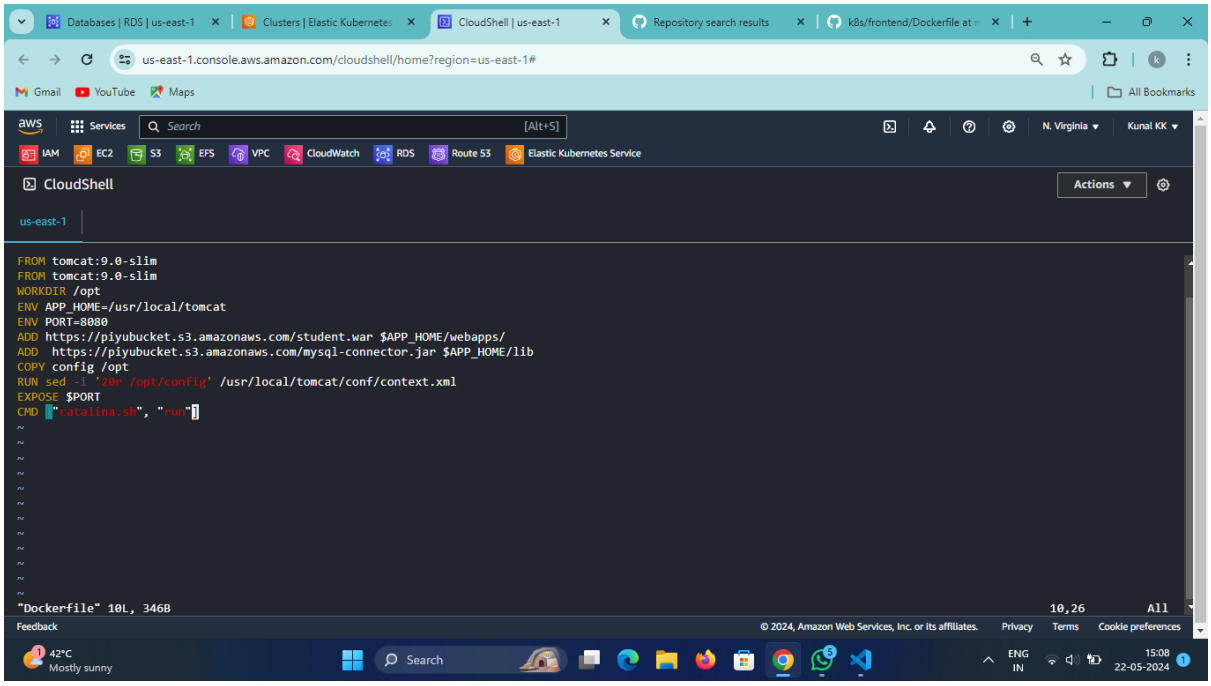
KUBERNETES 3 TIER PROJECT USING DB INSTANCE

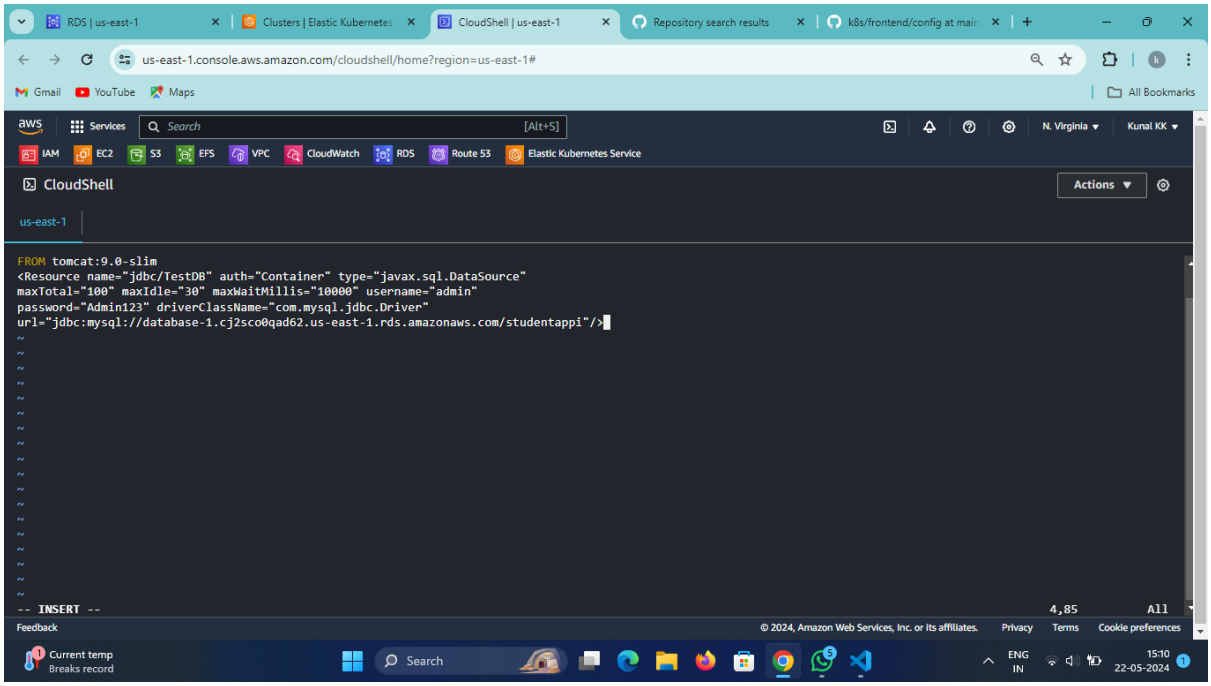
1. First create a cluster.
2. Then create an DB instance in RDS.



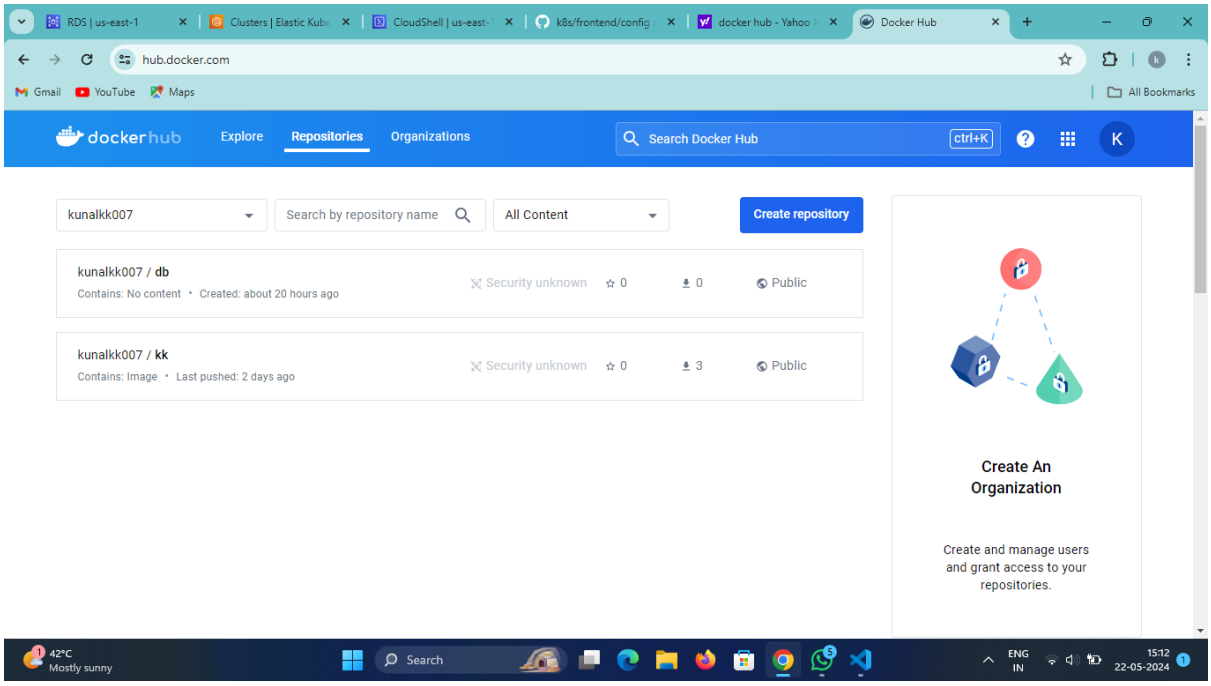


3. Then go on cloudshell and create a Dockerfile and Config file for frontend.
4. Just insert the endpoint of DB instance in Config file.

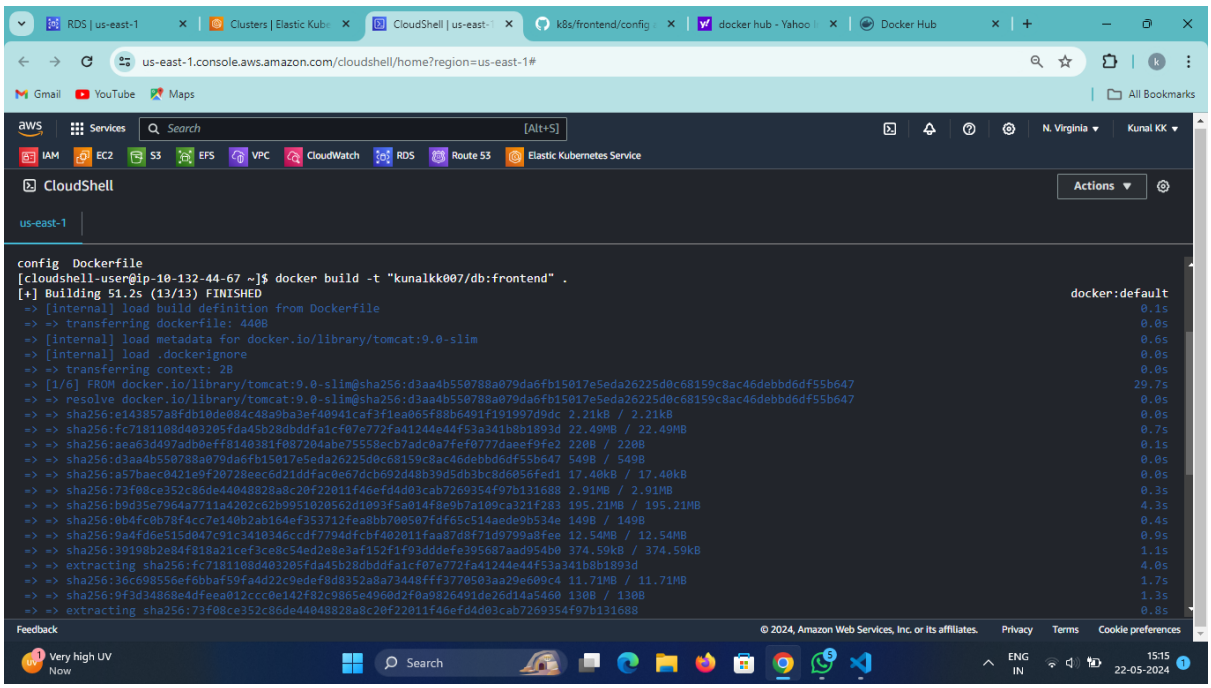




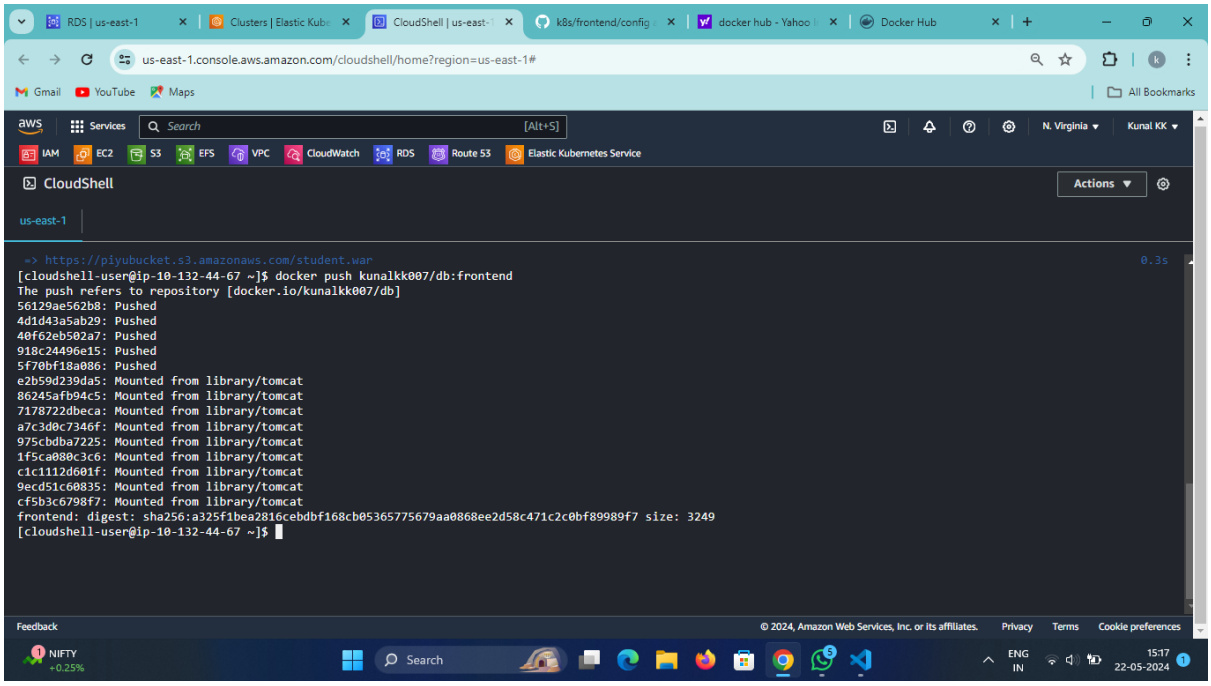
5. After that open docker hub and create a repository.



6. create the docker containers using Dockerfile using following command.
   # docker build -t "kunalkk007/db:frontend" .

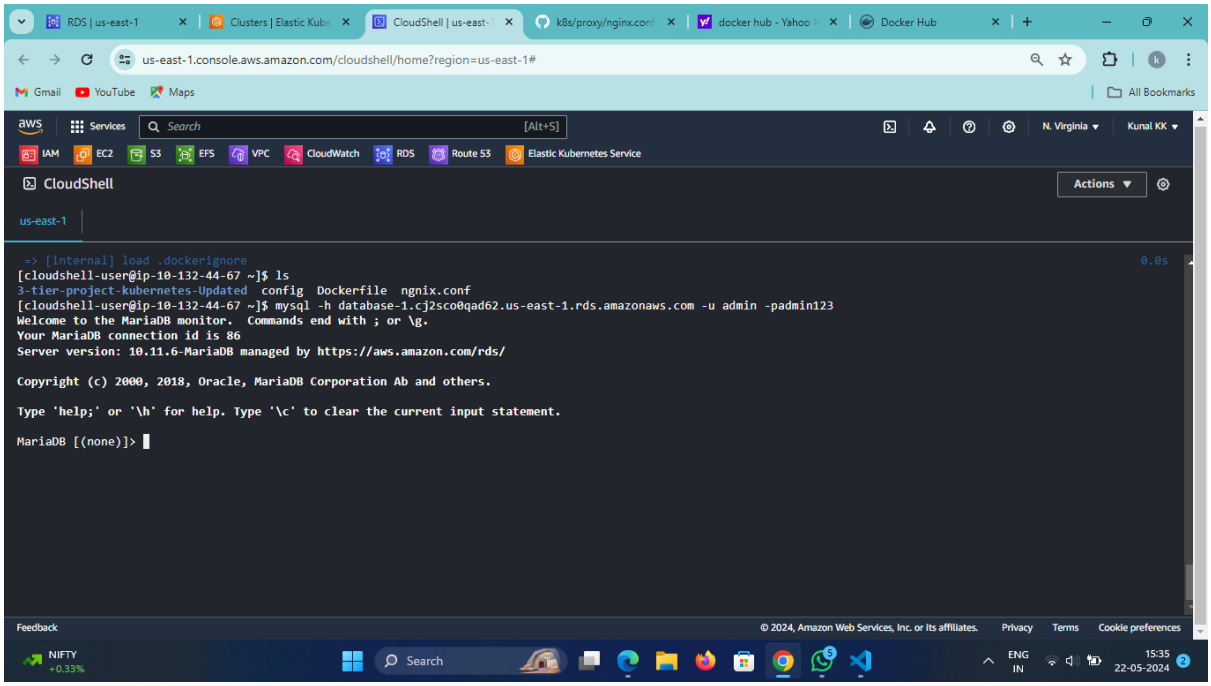


7. Upload Docker containers in Docker hub
8. Note: we need to configure username and password with docker login command…
   # docker push kunalkk007/db:frontend

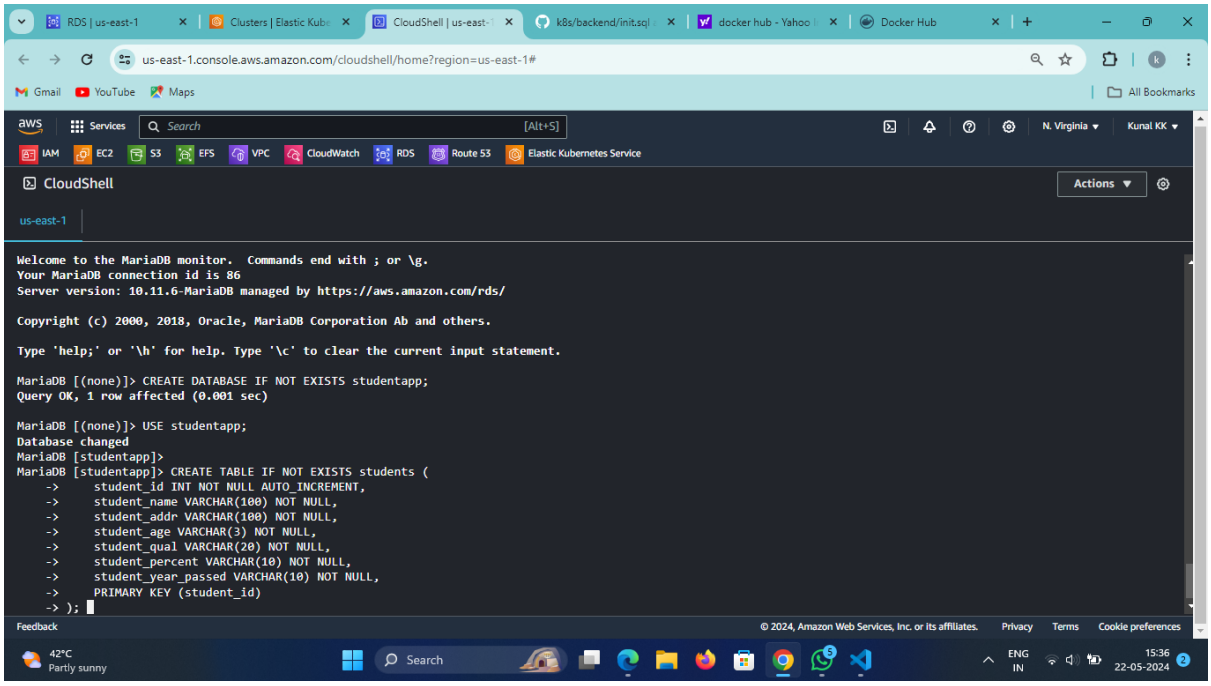


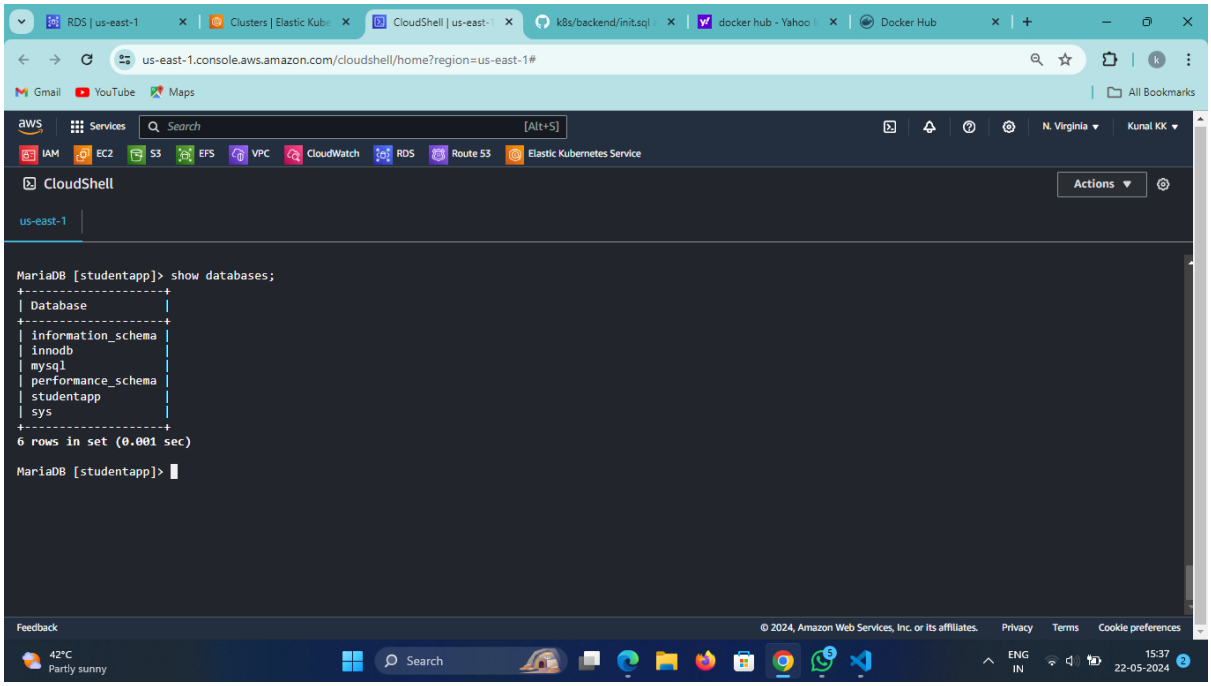
9. Create Schema in Amazon RDS database…
   - Use below commands for connecting with RDS database

 # mysql –h <endpoint_of_dbinstanc> -u admin -padmin1234

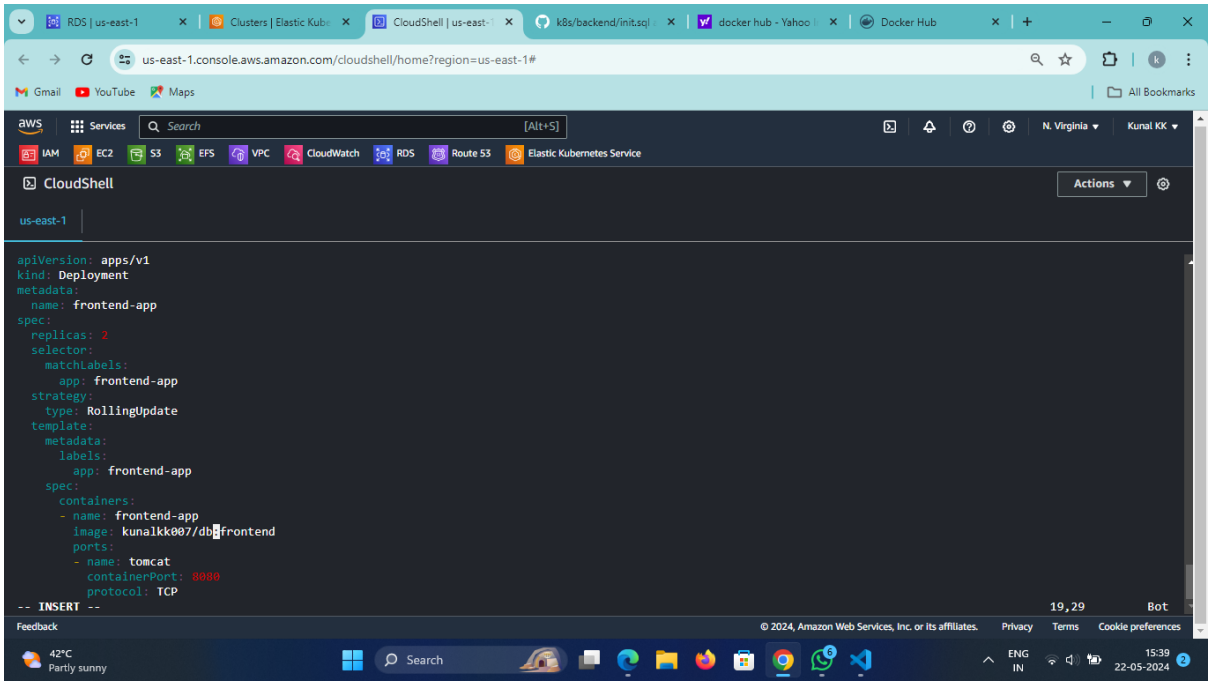


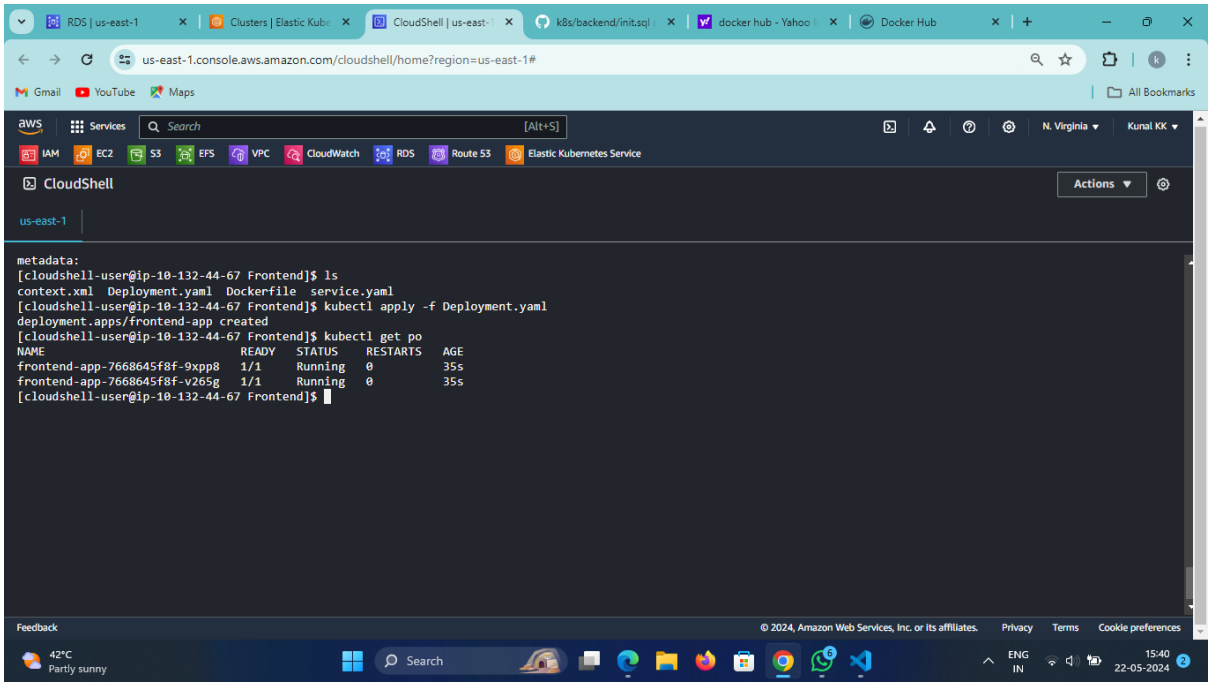
10. Add schema in database.





11. Successfully created studentapp database.
12. Write a Deployment.yaml files for Deploying application in Kubernetes.
13. Deployment.yaml File for Frontend.





14. Write service.yaml files for Exposing The application over Internet
    Service.yaml file for frontend.

apiVersion: v1

kind: Service

metadata:

name: frontend-service

spec:

selector:

app: frontend-app

ports:

- **name: http**
  targetPort: 8080
  port: 80

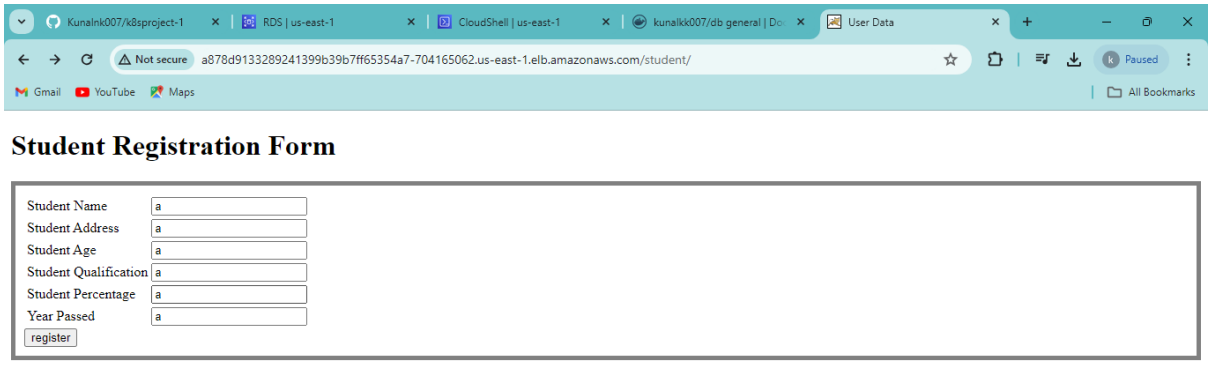
type: ClusterIP

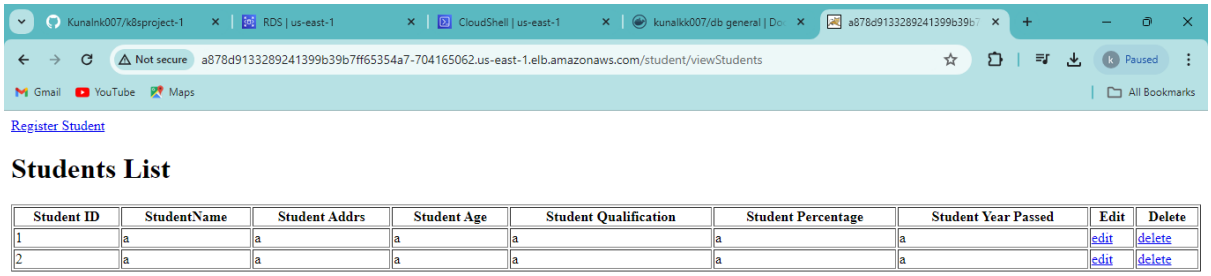
```yaml
apiVersion: v1
kind: Service
metadata:
  name: frontend-service
spec:
  selector:
    app: frontend-app
  ports:
  - name: http
    targetPort: 8080
    port: 80
  type: ClusterIP
```

15. Deploy Deployment.yaml and Service.yaml files using below commands…
    # kubectl apply -f deployment.yaml (for frontend)
    # kubectl apply -f service.yaml (for frontend)

Note:- before Accessing the website with load-balancer we need to edit security group of following…..

1) Edit security group of RDS database (add port 3306 in inbound rule)
2) Edit security group of Load balancer (add port 80 in inbound rule)

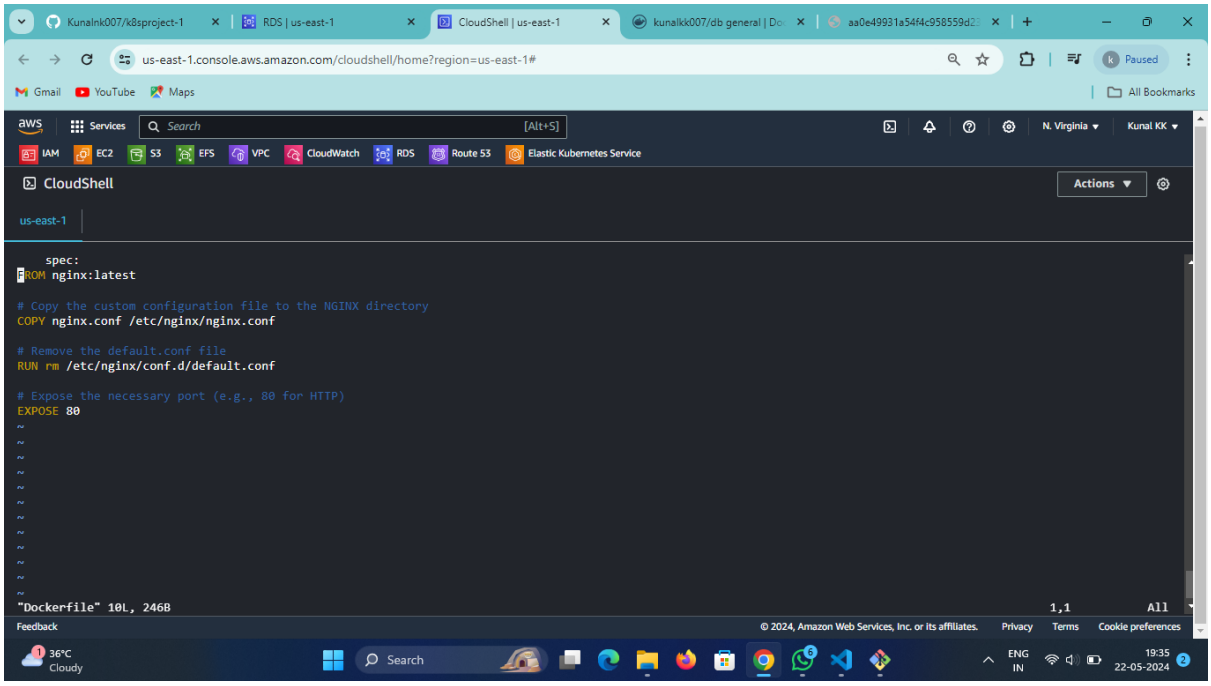


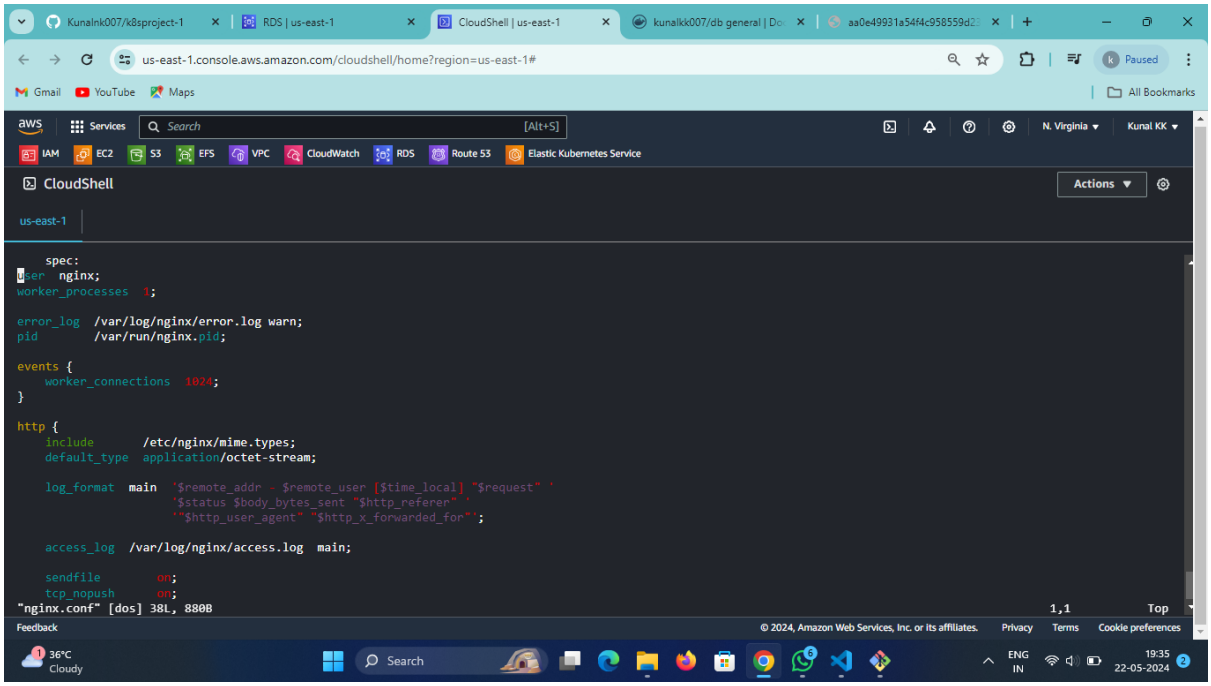


- Here our 2 tier project is completed.

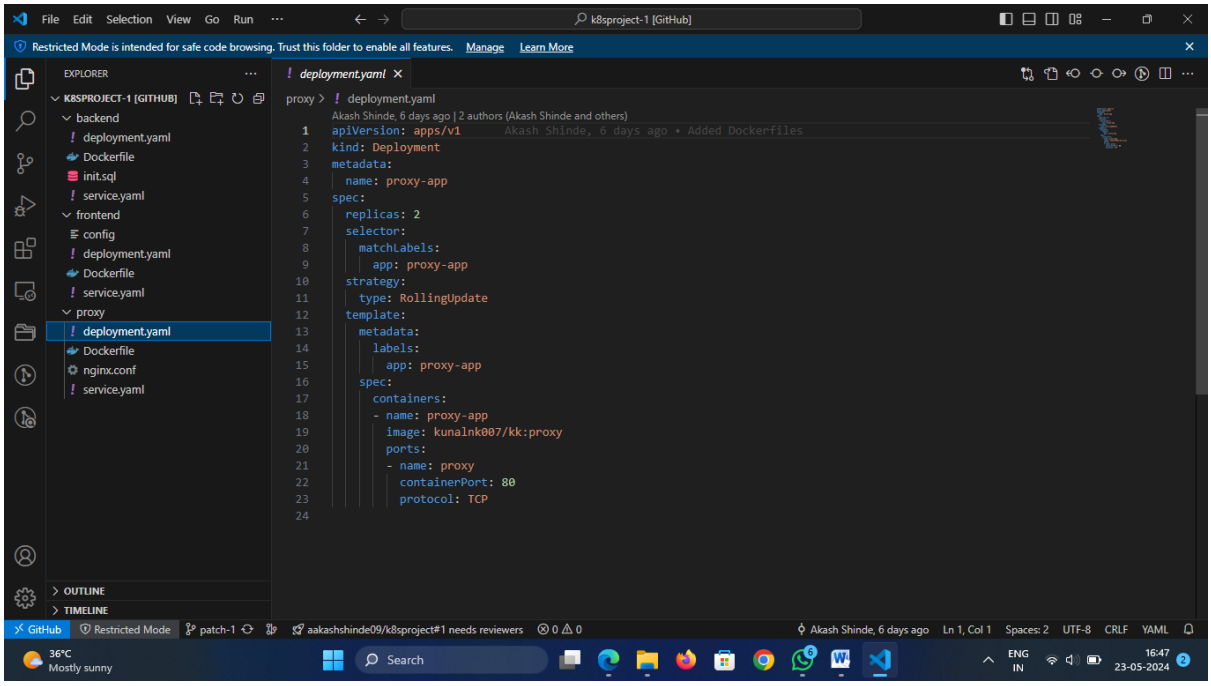
16. Now we will try to pass the proxy.

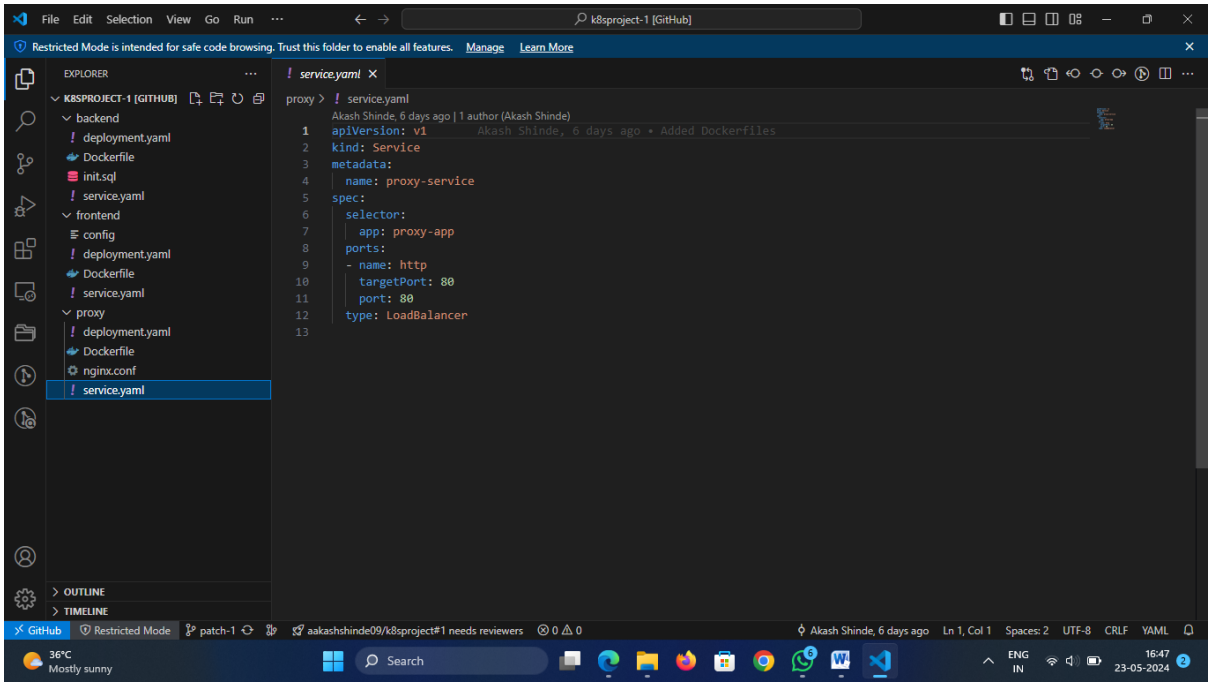
- For that we need a Dockerfile and nginx.conf for our proxy.





- Now we need deployment and service.yaml file for Proxy.



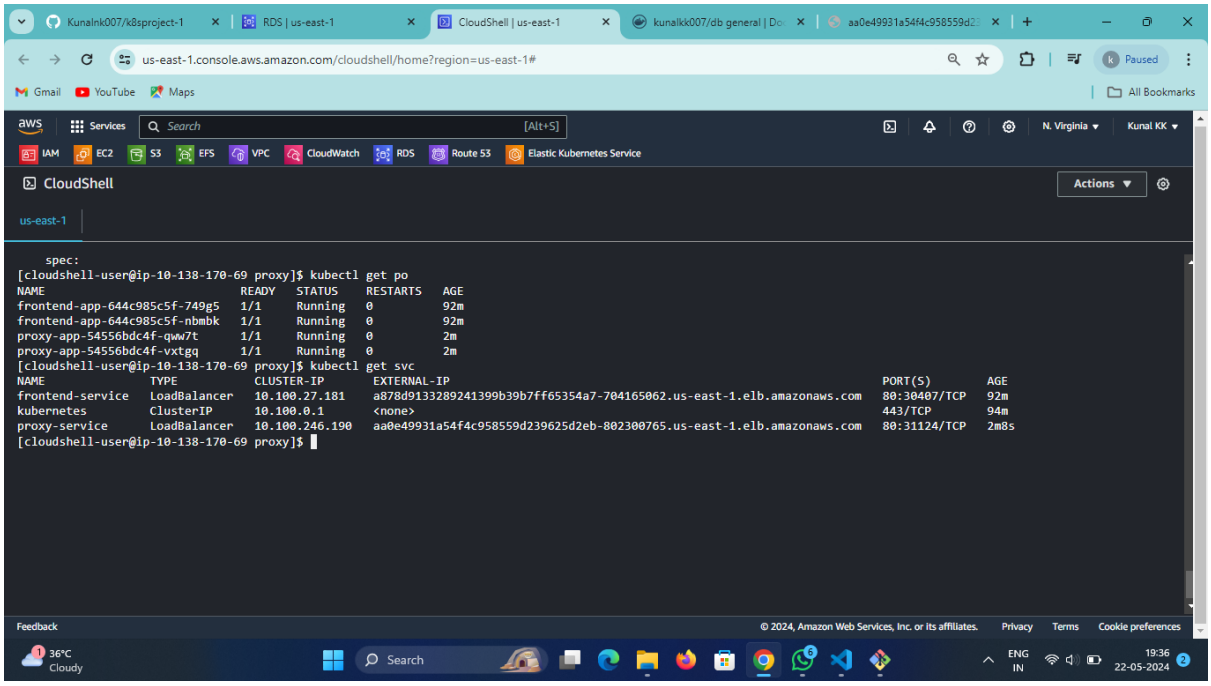


17. Deploy Deployment.yaml and Service.yaml files using below commands…

    # kubectl apply -f deployment.yaml (for proxy)

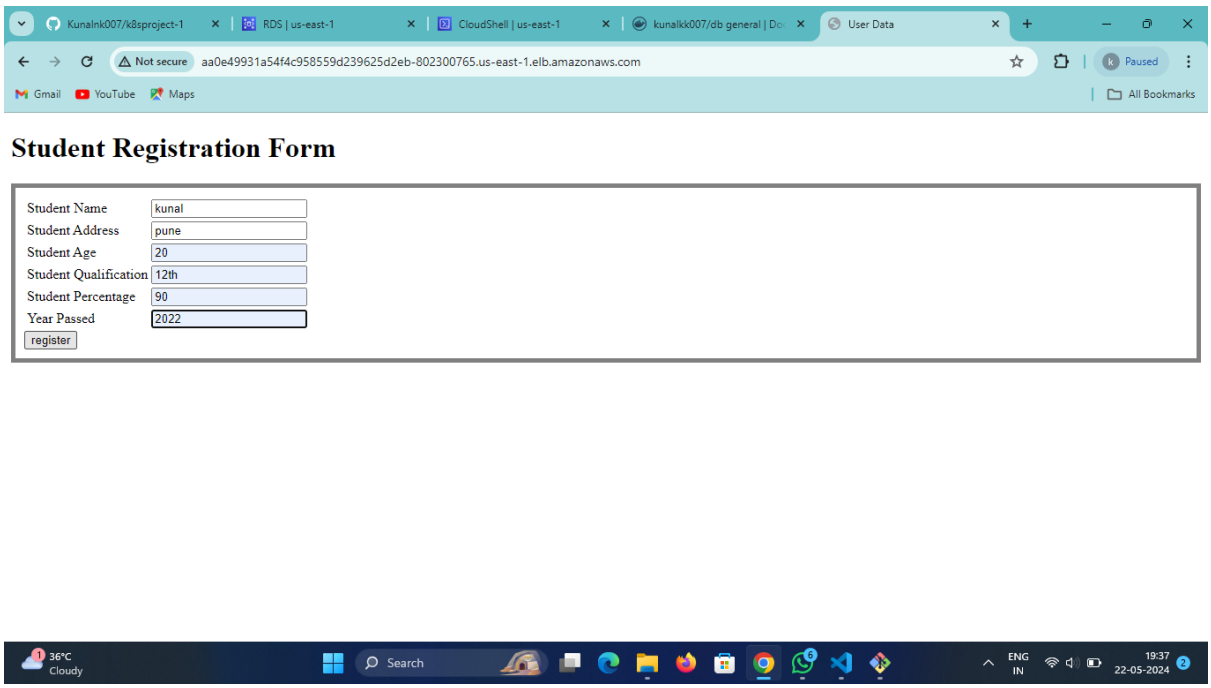
    # kubectl apply -f service.yaml (for proxy)

- Now hit command # kubectl get svc and copy the external ip of proxy-service and hit on web browser.



18. Our application hosted successfully.

- Output-1:-



Output-2:-

Successfully stored the data in RDS.