Document on secret-yaml-files:

1. Get into the killercoda site and perform the commands.
2. Start with sudo su command to become as a root use

   Cmd: **sudo su**

```
controlplane:~$ sudo su
controlplane:~$ ls
```

3. Then clone into the given git repo and get ready
4. After cloning check the cloned repo by ls

   Cmd : **git clone <url of git>**

```
controlplane:~$ git clone https://github.com/Msocial123/fss-Retail-App_kubernetes.git
Cloning into 'fss-Retail-App_kubernetes'...
remote: Enumerating objects: 2432, done.
remote: Counting objects: 100% (142/142), done.
remote: Compressing objects: 100% (102/102), done.
remote: Total 2432 (delta 79), reused 50 (delta 36), pack-reused 2290 (from 2)
Receiving objects: 100% (2432/2432), 9.40 MiB | 8.50 MiB/s, done.
Resolving deltas: 100% (551/551), done.
```

5. Then go into the directory

   Cmd: **cd <directory name>**

```
controlplane:~$ ls
filesystem  fss-Retail-App_kubernetes
controlplane:~$ cd fss-Retail-App_kubernetes/
```

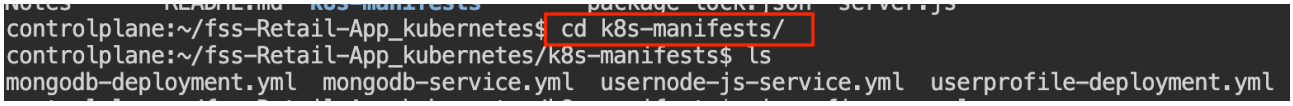
6. Then ls and check for the k8-manifest directory

   Cmd: **ls**

```
controlplane:~/fss-Retail-App_kubernetes$ ls
Dockerfile  Public     docker-compose.yaml  node_modules       package.json  test-volumes.yaml
Notes       README.md  k8s-manifests        package-lock.json  server.js
controlplane:~/fss-Retail-App_kubernetes$ cd k8s-manifests/
```

7. Then go into the k8-manifest directory

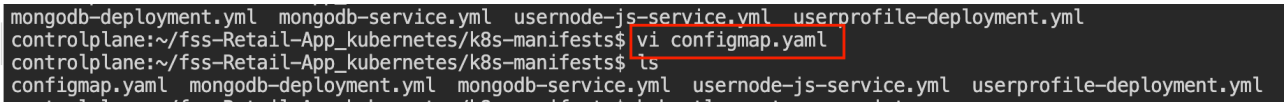
Cmd: **cd <directory name>**



8. Then create a file

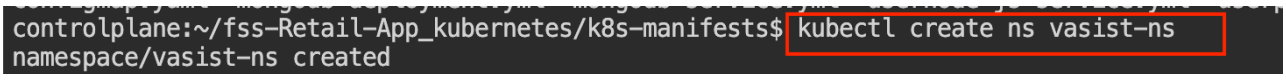
Cmd: **vi configmap.yaml**

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: retail-app-config
  namespace: vasist-ns
data:
  MONGODB_URI: "mongodb://mongodb:27017/myDatabase"
  SESSION_SECRET: "1234"
  PORT: "3130"
  MONGO_INITDB_DATABASE: "myDatabase"
```



9. Then create a name space give your name

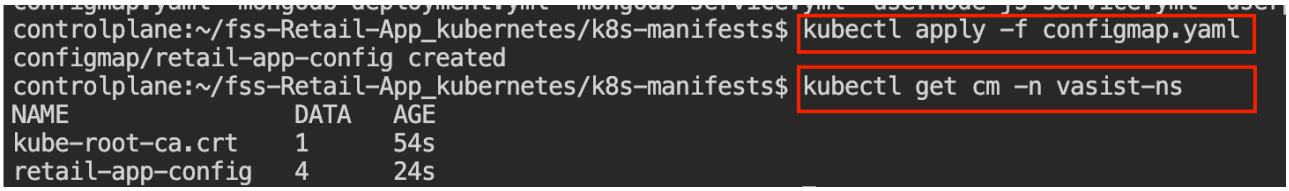
Cmd: **kubectl create ns vasist-ns**



10. Then apply the file and check for the running data

Cmd: **kubectl apply -f configmap.yaml (**this cmd is used to apply the yaml file**)**

Kubectl get cm -n vasist-ns(this is cmd is used check all the data applied or not**)**



11. Change the file name and remove the data present in the file and add the new file data

Cmd: **mv userprofile-deployment.yml retail-app-deployment.yaml**(this cmd is used to change the name of the file)

> retail-app-deployment.yaml(this cmd is used remove the data present inside the file)

vi retail-app-deployment.yaml (this cmd is used to add the yaml file into the file**)**

Yaml file:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: retail-mongodb
  namespace: murali-ns
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mongodb
  template:
    metadata:
      labels:
        app: mongodb
    spec:
      containers:
      - name: mongodb
        image: mongo:latest
        ports:
        - containerPort: 27017
        env:
        - name: MONGO_INITDB_DATABASE
          valueFrom:
            configMapKeyRef:
```

name: retail-app-config
key: MONGO_INITDB_DATABASE

```
controlplane:~/fss-Retail-App_kubernetes/k8s-manifests$ mv userprofile-deployment.yml retail-app-deployment.yaml
controlplane:~/fss-Retail-App_kubernetes/k8s-manifests$ ls
configmap.yaml  mongodb-deployment.yml  mongodb-service.yml  retail-app-deployment.yaml  usernode-js-service.yml
controlplane:~/fss-Retail-App_kubernetes/k8s-manifests$ > retail-app-deployment.yaml
controlplane:~/fss-Retail-App_kubernetes/k8s-manifests$ vi retail-app-deployment.yaml
controlplane:~/fss-Retail-App_kubernetes/k8s-manifests$ ls
configmap.yaml  mongodb-deployment.yml  mongodb-service.yml  retail-app-deployment.yaml  usernode-js-service.yml
```

12. Change the file name and change the namespace

Cmd : **mv usernode-js-service.yml retail-app-svc.yaml**(this cmd is used to change the name of the file)

```
controlplane:~/fss-Retail-App_kubernetes/k8s-manifests$ ls
configmap.yaml  mongodb-deployment.yml  mongodb-service.yml  retail-app-deployment.yaml  usernode-js-service.yml
controlplane:~/fss-Retail-App_kubernetes/k8s-manifests$ mv usernode-js-service.yml retail-app-svc.yaml
controlplane:~/fss-Retail-App_kubernetes/k8s-manifests$ vi retail-app-svc.yaml
```

13. Change the file name and remove the data present in the file and add the new file data

Cmd :
mv mongodb-deployment.yml retail-mongodb-deployment.yaml (this cmd is used to change the name of the file)
> retail-mongodb-deployment.yaml(this cmd is used remove the data present inside the file)

vi retail-mongodb-deployment.yaml(this cmd is used to add the yaml file into the file)

Yaml file:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: retail-mongodb
  namespace: murali-ns
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mongodb
  template:
    metadata:
      labels:
        app: mongodb
    spec:
```

```
containers:
- name: mongodb
  image: mongo:latest
  ports:
  - containerPort: 27017
  env:
  - name: MONGO_INITDB_DATABASE
    valueFrom:
      configMapKeyRef:
        name: retail-app-config
        key: MONGO_INITDB_DATABASE
```

```
controlplane:~/fss-Retail-App_kubernetes/k8s-manifests$ mv mongodb-deployment.yml retail-mongodb-deployment.yaml
controlplane:~/fss-Retail-App_kubernetes/k8s-manifests$ mv mongodb-deployment.yml retail-mongodb-deployment.yamlls
mv: cannot stat 'mongodb-deployment.yml': No such file or directory
controlplane:~/fss-Retail-App_kubernetes/k8s-manifests$ ls
configmap.yaml  mongodb-service.yml  retail-app-deployment.yaml  retail-app-svc.yaml  retail-mongodb-deployment.yaml
controlplane:~/fss-Retail-App_kubernetes/k8s-manifests$ > retail-mongodb-deployment.yaml
controlplane:~/fss-Retail-App_kubernetes/k8s-manifests$ vi retail-mongodb-deployment.yaml
```

14. Change the file name and remove the data present in the file and add the new file data

Cmd: **mv mongodb-service.yml retail-mongodb-svc.yaml**(this cmd is used to change the name of the file)

```
controlplane:~/fss-Retail-App_kubernetes/k8s-manifests$ mv mongodb-service.yml retail-mongodb-svc.yaml
controlplane:~/fss-Retail-App_kubernetes/k8s-manifests$ vi configmap.yaml
controlplane:~/fss-Retail-App_kubernetes/k8s-manifests$ ls
configmap.yaml  retail-app-deployment.yaml  retail-app-svc.yaml  retail-mongodb-deployment.yaml  retail-mongodb-svc.yaml
```

15. Then apply the file and check for running data

Cmd: **kubectl apply -f configmap.yaml**(this cmd is used to apply the yaml file)

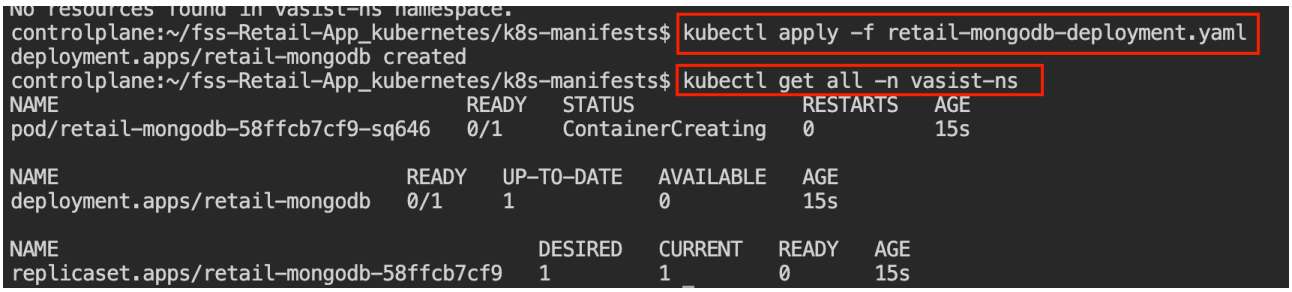
kubectl get all -n vasist-ns(used to check all the running data)

```
controlplane:~/fss-Retail-App_kubernetes/k8s-manifests$ kubectl apply -f configmap.yaml
configmap/retail-app-config created
controlplane:~/fss-Retail-App_kubernetes/k8s-manifests$ kubectl get cm -n vasist-ns
NAME               DATA   AGE
kube-root-ca.crt   1      54s
retail-app-config  4      24s
```

16. Then apply the file and check for running data

Cmd: **kubectl apply -f retail-mongodb-deployment.yaml**(this cmd is used to apply the yaml file)

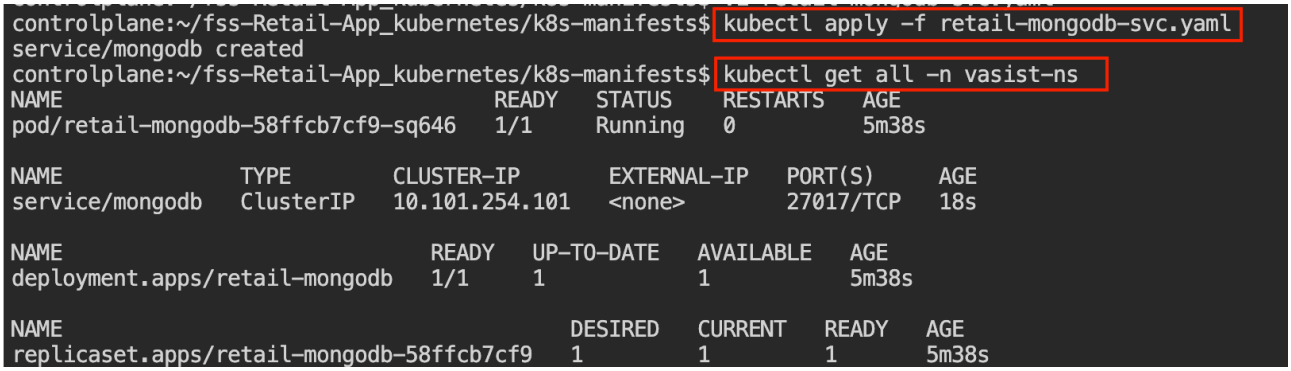
kubectl get all -n murali-ns(this cmd is used to check all the status and data)



17. Then apply the file and check for running data

Cmd: **kubectl apply -f retail-mongodb-svc.yaml**(this cmd is used to apply the yaml file)

kubectl get all -n vasist-ns(used to get all the running status of the data)



18. Then apply the file and check for running data

Cmd: **kubectl apply -f retail-mongodb-deployment.yaml**(this cmd is used to apply the yaml file)

kubectl get all -n vasist-ns(used to get all status of data and all the working report)

```
controlplane:~/fss-Retail-App_kubernetes/k8s-manifests$ kubectl apply -f retail-app-deployment.yaml
deployment.apps/retail-app-deployment created
controlplane:~/fss-Retail-App_kubernetes/k8s-manifests$ kubectl get all -n vasist-ns
NAME                                      READY   STATUS              RESTARTS   AGE
pod/retail-app-deployment-64c485cbd7-5k5jw   0/1     ContainerCreating   0          14s
pod/retail-app-deployment-64c485cbd7-crps9   0/1     ContainerCreating   0          14s
pod/retail-app-deployment-64c485cbd7-hghkw   0/1     ContainerCreating   0          14s
pod/retail-app-deployment-64c485cbd7-zzc9b   0/1     ContainerCreating   0          14s
pod/retail-mongodb-58ffcb7cf9-sq646          1/1     Running             0          6m6s

NAME              TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)     AGE
service/mongodb   ClusterIP   10.101.254.101   <none>        27017/TCP   46s

NAME                                    READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/retail-app-deployment   0/4     4            0           14s
deployment.apps/retail-mongodb          1/1     1            1           6m6s

NAME                                               DESIRED   CURRENT   READY   AGE
replicaset.apps/retail-app-deployment-64c485cbd7   4         4         0       14s
replicaset.apps/retail-mongodb-58ffcb7cf9          1         1         1       6m6s
```

19. Then apply the file and check for running data

Cmd: **kubectl apply -f retail-app-svc.yaml**(this cmd is used to apply the yaml file)

kubectl get all -n vasist-ns(used to get all status of data and all the working report)

```
controlplane:~/fss-Retail-App_kubernetes/k8s-manifests$ kubectl apply -f retail-app-svc.yaml
service/usernode-js-service created
controlplane:~/fss-Retail-App_kubernetes/k8s-manifests$ kubectl get all -n vasist-ns
NAME                                      READY   STATUS                       RESTARTS   AGE
pod/retail-app-deployment-64c485cbd7-5k5jw   0/1     ContainerCreating            0          43s
pod/retail-app-deployment-64c485cbd7-crps9   0/1     CreateContainerConfigError   0          43s
pod/retail-app-deployment-64c485cbd7-hghkw   0/1     ContainerCreating            0          43s
pod/retail-app-deployment-64c485cbd7-zzc9b   0/1     CreateContainerConfigError   0          43s
pod/retail-mongodb-58ffcb7cf9-sq646          1/1     Running                      0          6m35s

NAME                          TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
service/mongodb               ClusterIP      10.101.254.101   <none>        27017/TCP        75s
service/usernode-js-service   LoadBalancer   10.96.228.22     <pending>     3130:30958/TCP   16s

NAME                                    READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/retail-app-deployment   0/4     4            0           43s
deployment.apps/retail-mongodb          1/1     1            1           6m35s

NAME                                               DESIRED   CURRENT   READY   AGE
replicaset.apps/retail-app-deployment-64c485cbd7   4         4         0       43s
replicaset.apps/retail-mongodb-58ffcb7cf9          1         1         1       6m35s
```

20. Check the status of the application:

```
pod/retail-app-deployment-68b685c7f6-5l6ps   1/1     Running   0          61s
pod/retail-app-deployment-68b685c7f6-8j64t   1/1     Running   0          61s
pod/retail-app-deployment-68b685c7f6-gl656   1/1     Running   0          61s
pod/retail-app-deployment-68b685c7f6-mrf6l   1/1     Running   0          61s
pod/retail-mongodb-7cd9bb8c49-fw4k5          1/1     Running   0          78s

NAME                        TYPE           CLUSTER-IP       EXTERNAL-IP                                                              PORT(S)          AGE
service/mongodb             ClusterIP      10.100.218.198   <none>                                                                   27017/TCP        71s
service/retail-app-service  LoadBalancer   10.100.152.130   a40f909364f534bef96e01cb855ab9b7-1849244736.ap-south-1.elb.amazonaws.com  3130:30183/TCP   54s

NAME                                  READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/retail-app-deployment 4/4     4            4           61s
deployment.apps/retail-mongodb        1/1     1            1           78s

NAME                                             DESIRED   CURRENT   READY   AGE
replicaset.apps/retail-app-deployment-68b685c7f6 4         4         4       61s
replicaset.apps/retail-mongodb-7cd9bb8c49        1         1         1       78s
```

21. Open a Linux shell inside the MongoDB container and check files inside the container.

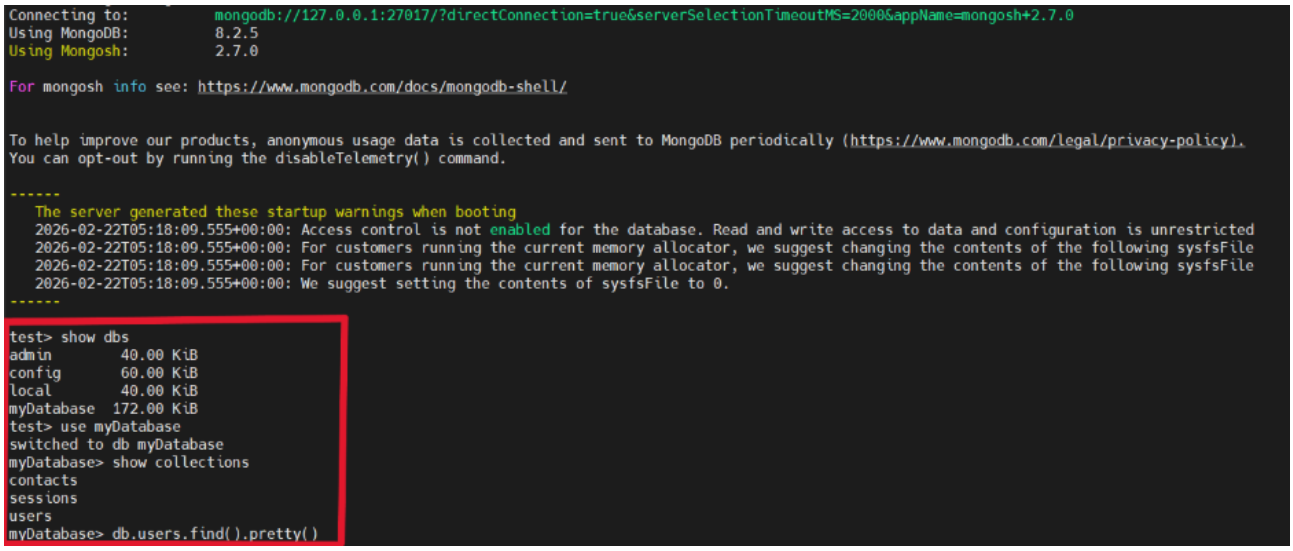
Cmd: **kubectl exec -it retail-mongodb-7cd9bb8c49-fw4k5 -n vasist-ns -- /bin/sh**

```
# env
KUBERNETES_SERVICE_PORT=443
KUBERNETES_PORT=tcp://10.100.0.1:443
MONGODB_PORT_27017_TCP=tcp://10.100.218.198:27017
HOSTNAME=retail-mongodb-7cd9bb8c49-fw4k5
MONGO_INITDB_DATABASE=myDatabase
HOME=/data/db
GLIBC_TUNABLES=glibc.pthread.rseq=0
MONGODB_SERVICE_HOST=10.100.218.198
TERM=xterm
MONGO_PACKAGE=mongodb-org
KUBERNETES_PORT_443_TCP_ADDR=10.100.0.1
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
MONGO_MAJOR=8.2
KUBERNETES_PORT_443_TCP_PORT=443
KUBERNETES_PORT_443_TCP_PROTO=tcp
MONGODB_PORT=tcp://10.100.218.198:27017
MONGODB_SERVICE_PORT=27017
MONGODB_PORT_27017_TCP_ADDR=10.100.218.198
JSYAML_VERSION=3.13.1
KUBERNETES_SERVICE_PORT_HTTPS=443
MONGODB_PORT_27017_TCP_PORT=27017
KUBERNETES_PORT_443_TCP=tcp://10.100.0.1:443
GOSU_VERSION=1.19
MONGO_REPO=repo.mongodb.org
MONGODB_PORT_27017_TCP_PROTO=tcp
KUBERNETES_SERVICE_HOST=10.100.0.1
PWD=/
JSYAML_CHECKSUM=662e32319bdd378e91f6757 8e56a34954b0a2e33aca11d70ab9f4826af24b941
MONGO_VERSION=8.2.5
```

22. **Open the MongoDB shell directly:**

Cmd: **kubectl exec -it retail-mongodb-7cd9bb8c49-fw4k5 -n rithika-ns – mongosh**

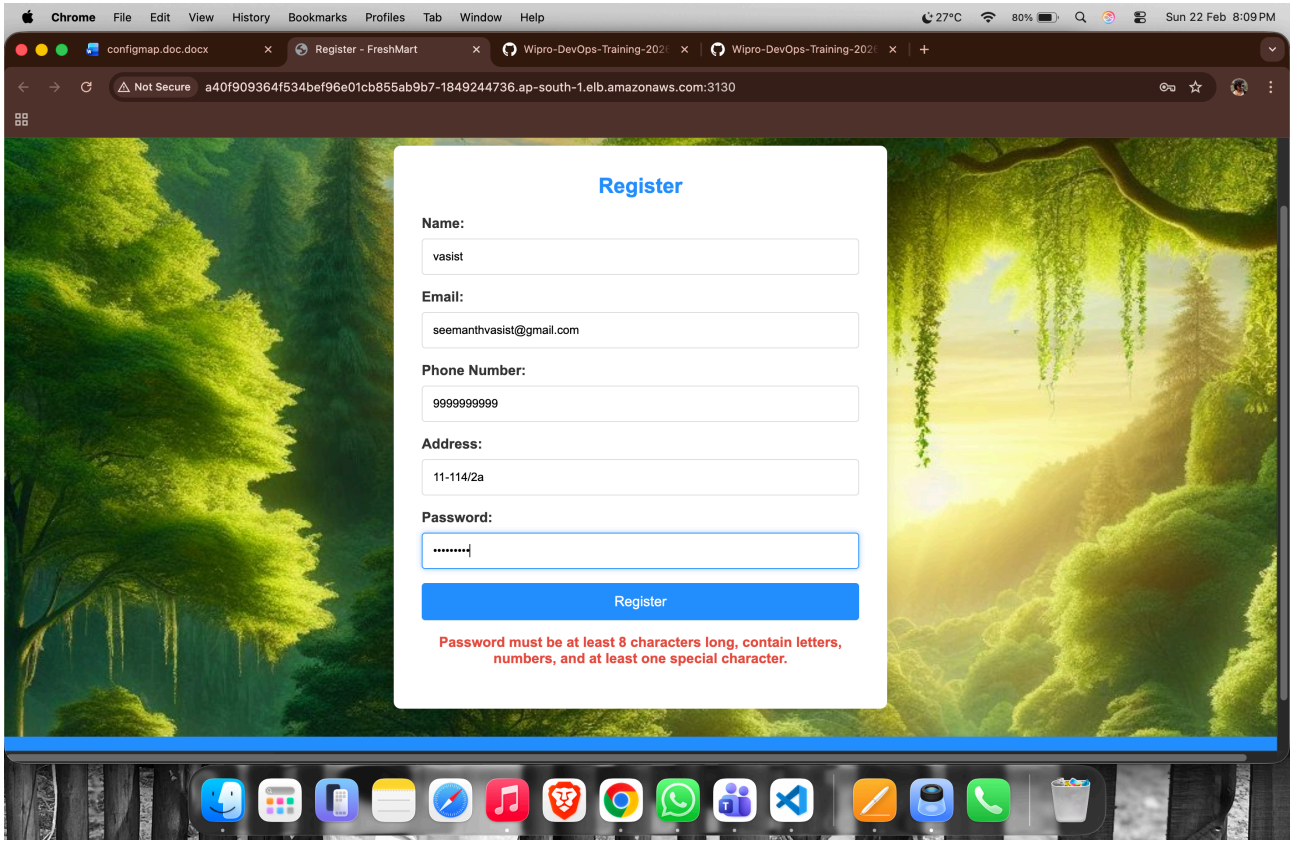
To view ,use the databases and shows all documents in the products collection.

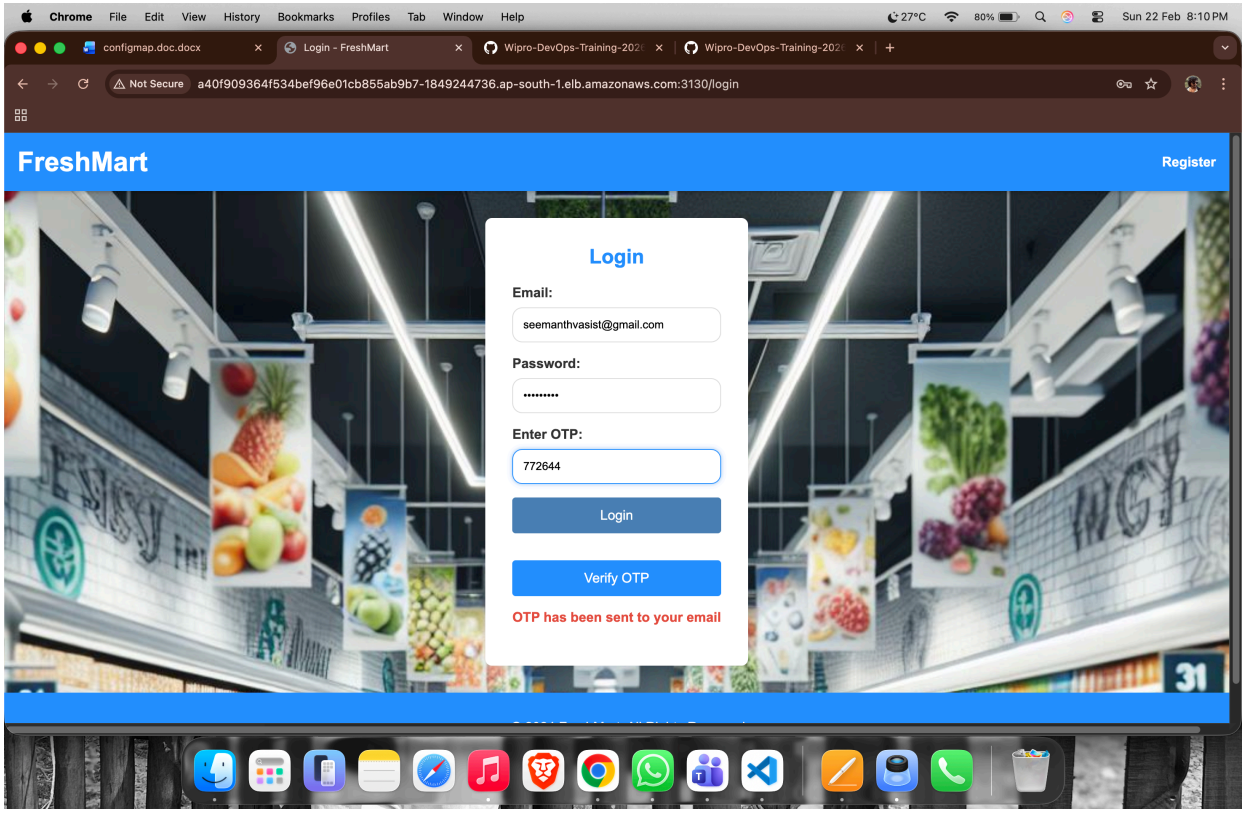


23. **Copy the External IP and paste it in browser:**

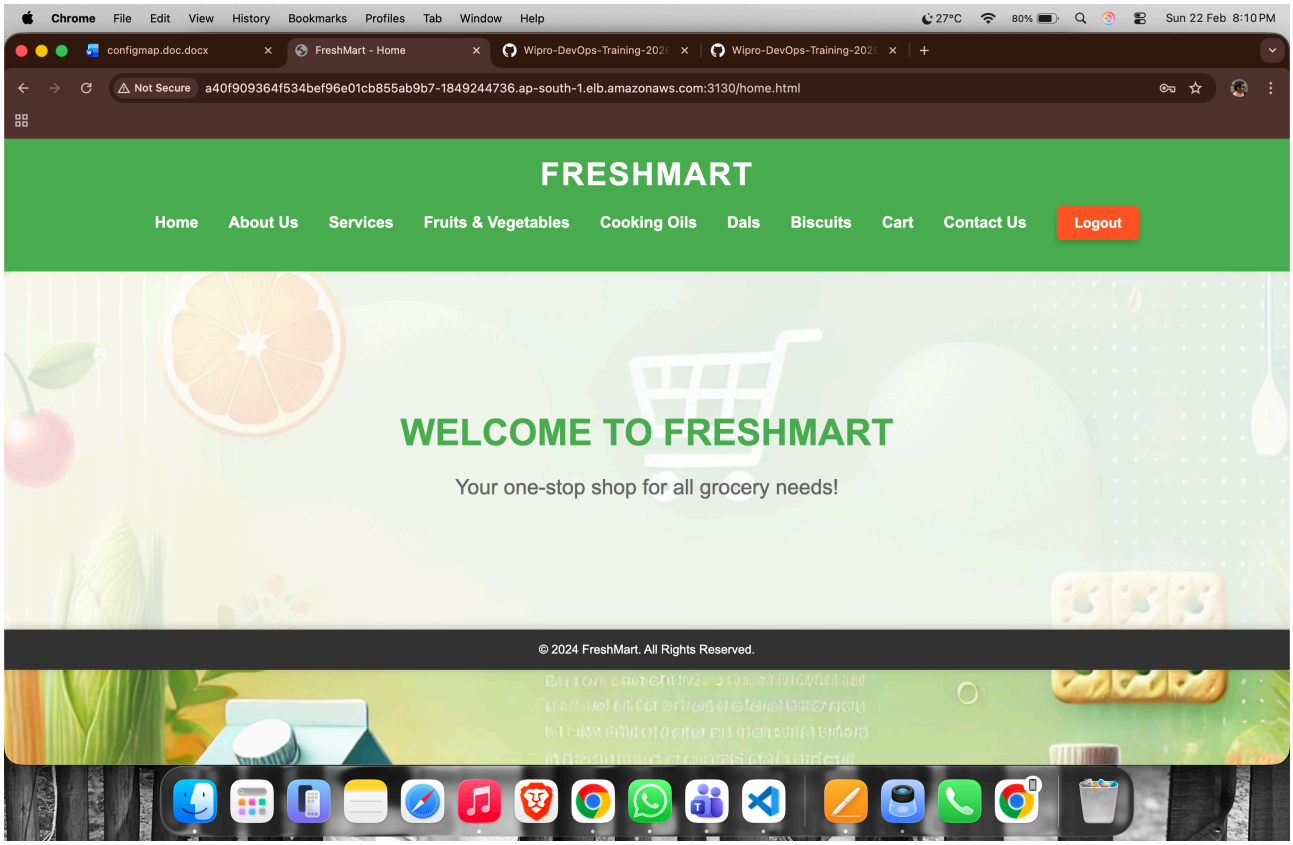
http://a40f909364f534bef96e01cb855ab9b7-1849244736.ap-south-1.elb.amazonaws.com:3130/

It will get us to the home page.





24. The final site output:



SECRET YAML FILE

A Secret is a Kubernetes object used to store sensitive data such as passwords, tokens, API keys, or certificates. The data is stored in Base64-encoded format.

==> **echo -n 'chagantyteja2502@gmail.com' | base64**

- `echo -n` → prints the text without adding a newline
- `|` → sends output to the next command
- `base64` → encodes the input into Base64 format

```
controlplane:~$ echo -n 'chagantyteja2502@gmail.com' | base64
Y2hhZ2FudHl0ZWphMjUwMkBnbWFpbC5jb20=
```

chagantyteja2502@gmail.com

↓

Y2hhZ2FudHl0ZWphMjUwMkBnbWFpbC5jb20=

==> **echo -n 'yxoq bjuk rdnt alzp' | base64**

```
controlplane:~$ echo -n 'yxoq bjuk rdnt alzp' | base64
eXhvcSBianVrIHJkbnQgYWx6cA==
```

yxoq bjuk rdnt alzp

↓

eXhvcSBianVrIHJkbnQgYWx6cA==

==> **echo -n 'Y2hhZ2FudHl0ZWphMjUwMkBnbWFpbC5jb20=' | base64 –decode**

==> decodes the input

```
controlplane:~$ echo -n 'Y2hhZ2FudHl0ZWphMjUwMkBnbWFpbC5jb20=' | base64 --decode
chagantyteja2502@gmail.comcontrolplane:~$
```

Y2hhZ2FudHl0ZWphMjUwMkBnbWFpbC5jb20=

↓

chagantyteja2502@gmail.com

1. **Vi secret.yaml**

Create secret yaml file

```
root@ip-172-31-17-28:/home/ubuntu/fss-Retail-App_kubernetes/k8s-manifests# vi secret.yml
root@ip-172-31-17-28:/home/ubuntu/fss-Retail-App_kubernetes/k8s-manifests# ls
```

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: retail-app-secret
  namespace: vasist-ns
type: Opaque
```

```
data:
  EMAIL_USER: Y2hhZ2FudHl0ZWphMjUwMkBnbWFpbC5jb20=
  EMAIL_PASS: eXhvcSBianVrIHJkbnQgYWx6cA==
```

2) **kubectl apply -f secret.yaml**

Apply the file:

```
root@ip-172-31-17-28:/home/ubuntu/fss-Retail-App_kubernetes/k8s-manifests# kubectl apply -f secret.yml
secret/retail-app-secret created
```

3) **kubectl get secret –n vasist-ns**

- Lists **all Secrets** in the **current namespace**
- Shows secret name, type, and age

```
NAME                TYPE     DATA   AGE
retail-app-secret   Opaque   2      35s
```