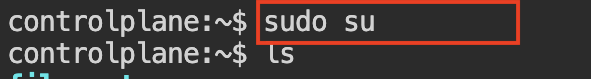
**Secret-yaml-files**

**Step 1:**

**Get into it and perform the commands.**

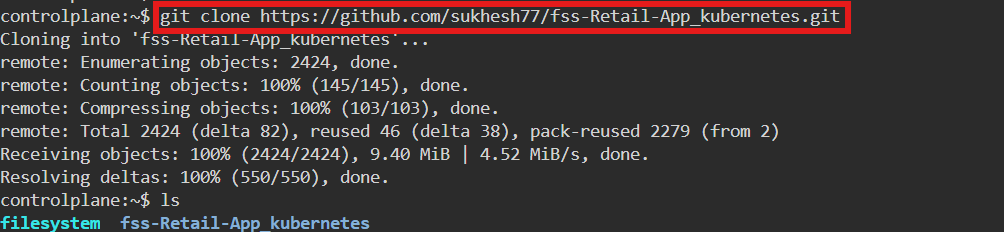
**Step 2:**

**Become Root User**

Cmd: **sudo su**

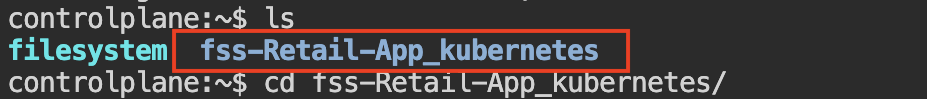
**Step 3:**

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

Cmd : **git clone https://github.com/sukhesh77/fss-Retail-App\_kubernetes.git**

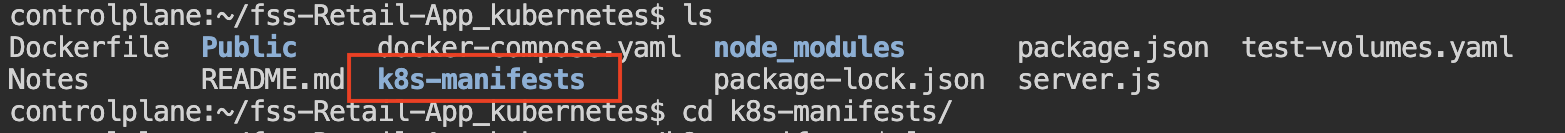
1. Then go into the directory

**Step 4:** Verify the Cloned Repository

Cmd: **cd <directory name>**

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

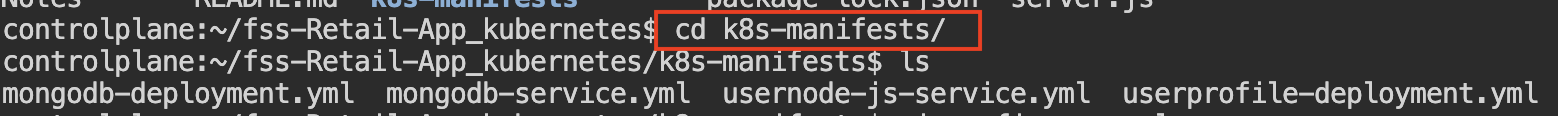
Cmd: **ls**

**Step 5:** Go into the Repository Directory

1. Then go into the k8-manifest directory

Cmd: **cd <directory name>**

**Step 6&7:** Check for k8-manifest Directory,go into k8-manifest directory

 **Step 8:** Create ConfigMap File

Cmd: **vi configmap.yaml**

**apiVersion: v1**

**kind: ConfigMap**

**metadata:**

**name: retail-app-config**

**namespace: sukhesh-ns**

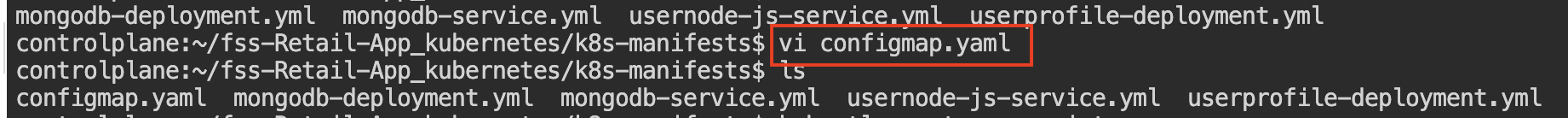
**data:**

**MONGODB\_URI: "mongodb://mongodb:27017/myDatabase"**

**SESSION\_SECRET: "1234"**

**PORT: "3130"**

**MONGO\_INITDB\_DATABASE: "myDatabase"**



**Step 9:** Then create a name space give your name

Cmd: **kubectl create ns sukhesh-ns**

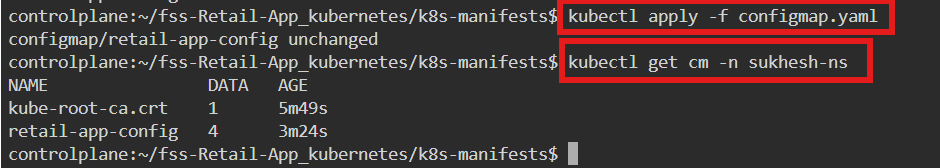


**Step 10:** Then apply the file and check for the running data

Cmd: **kubectl apply -f configmap.yaml**

* This cmd is used to apply the yaml file

**kubectl get cm -n sukhesh-ns**

* This is cmd is used check all the data applied or not
* 

**Step 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: sukhesh-ns

spec:

replicas: 1

selector:

matchLabels:

app: mongodb

template:

metadata:

labels:

app: mongodb

spec:

containers:

- name: mongodb

image: mongo:latest

ports:

- containerPort: 27017

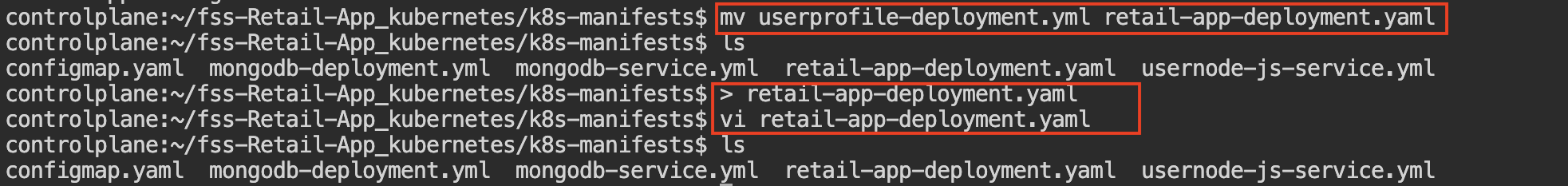
env:

- name: MONGO\_INITDB\_DATABASE

valueFrom:

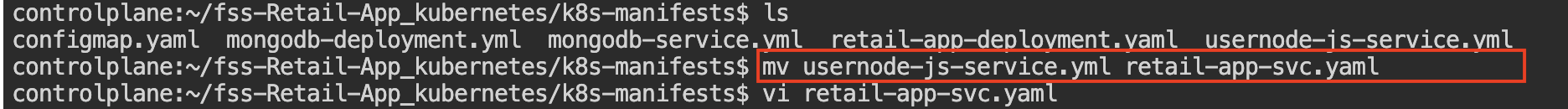
configMapKeyRef:

name: retail-app-config

key: MONGO\_INITDB\_DATABASE

**Step 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

**Step 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: sukhesh-ns

spec:

replicas: 1

selector:

matchLabels:

app: mongodb

template:

metadata:

labels:

app: mongodb

spec:

containers:

- name: mongodb

image: mongo:latest

ports:

- containerPort: 27017

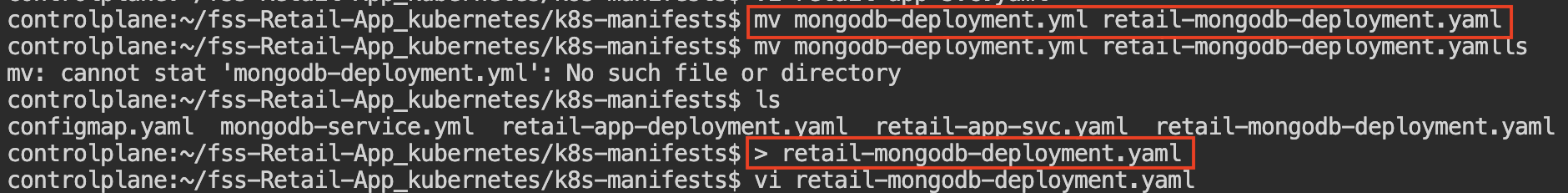
env:

- name: MONGO\_INITDB\_DATABASE

valueFrom:

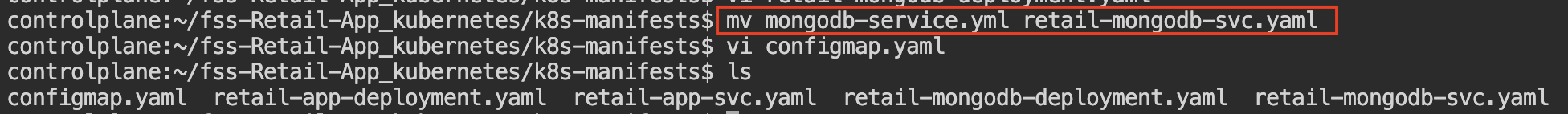
configMapKeyRef:

name: retail-app-config

key: MONGO\_INITDB\_DATABASE

**Step 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

**Step 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 sukhesh-ns**

* Used to check all the running data



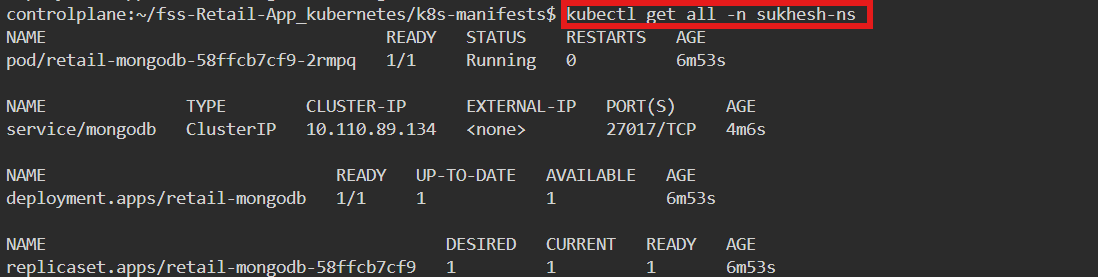
**Step 16:**Then apply the file and check for running data

Cmd: **kubectl apply -f retail-mongodb-deployment.yaml**

* This cmd is used to apply the yaml file

**kubectl get all -n sukhesh-ns**

* This cmd is used to check all the status and data

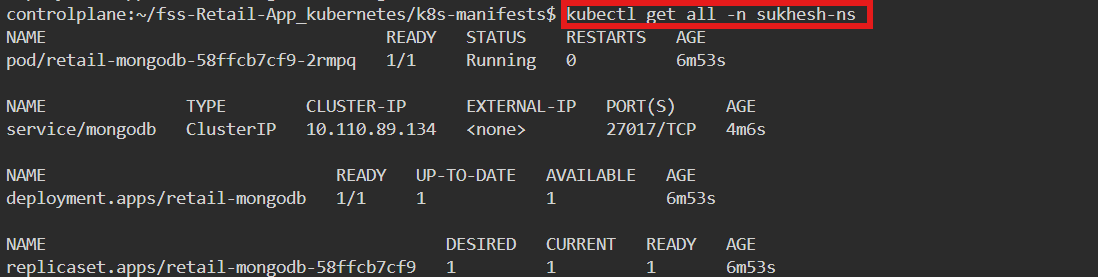


**Step 17:** Then apply the file and check for running data

Cmd: **kubectl apply -f retail-mongodb-svc.yaml**

* This cmd is used to apply the yaml file

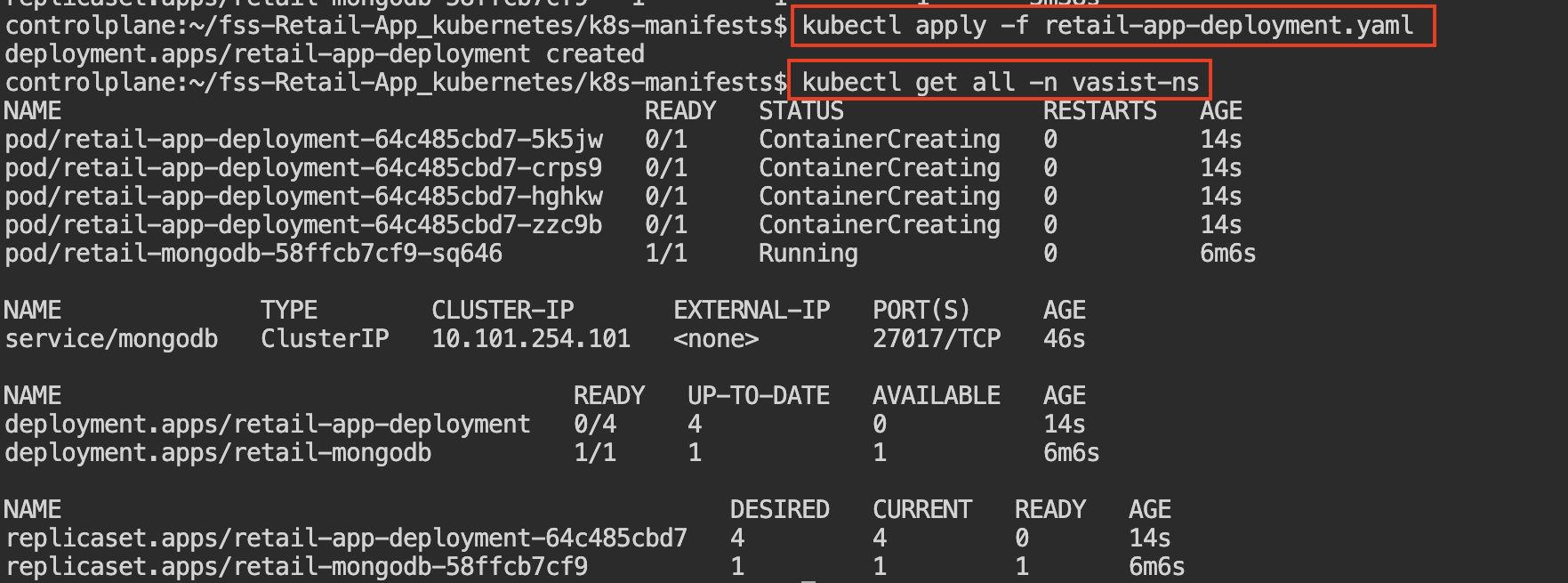
**kubectl get all -n sukhesh-ns**

* Used to get all the running status of the data

**Step 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 sukhesh-ns**

* Used to get all status of data and all the working report

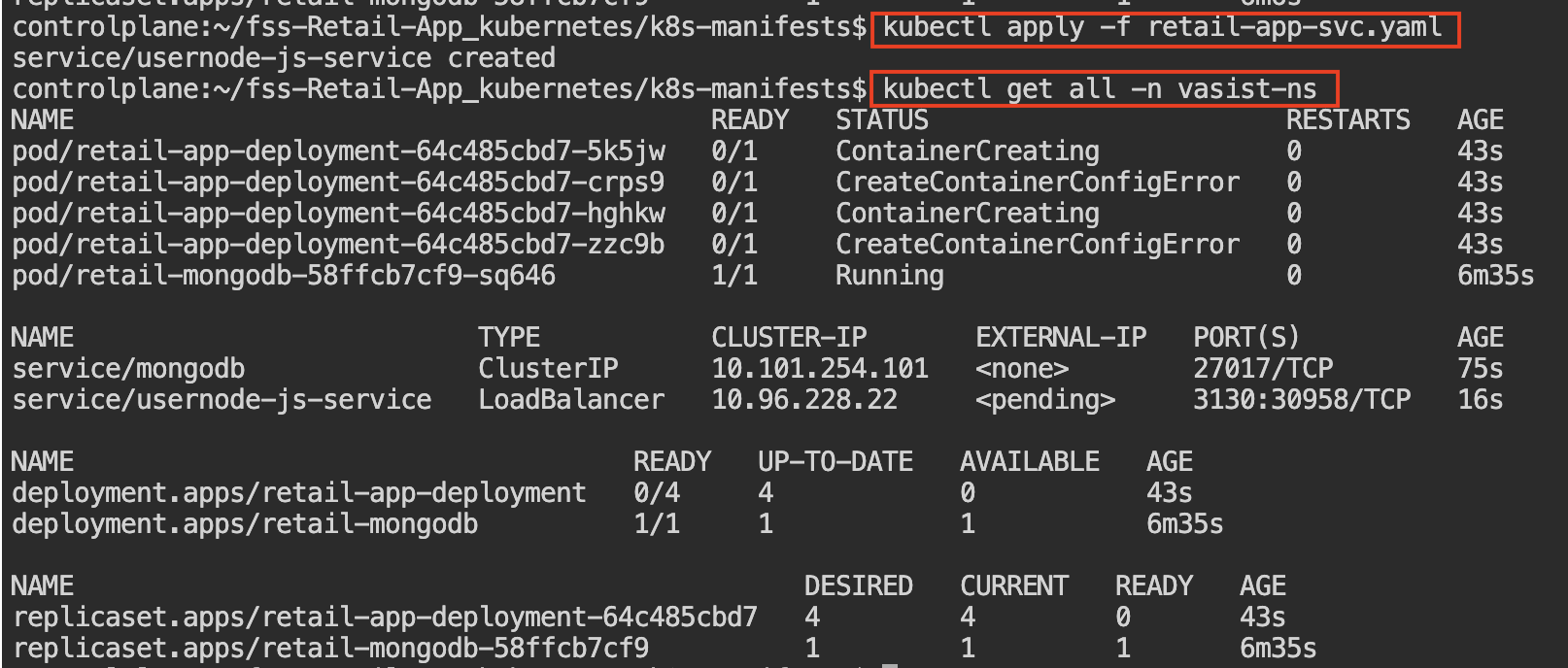
**Step 19:** Then apply the file and check for running data

Cmd: **kubectl apply -f retail-app-svc.yaml**

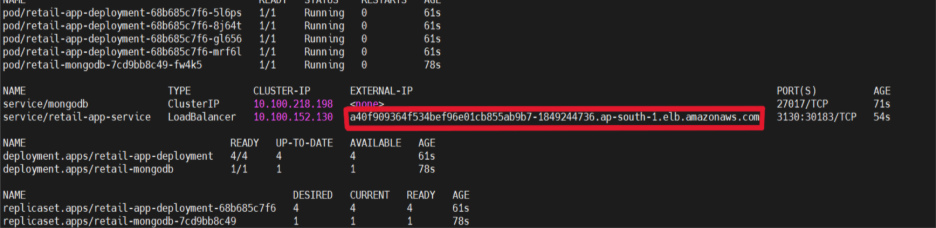
* This cmd is used to apply the yaml file

**kubectl get all -n sukhesh-ns**

* Used to get all status of data and all the working report

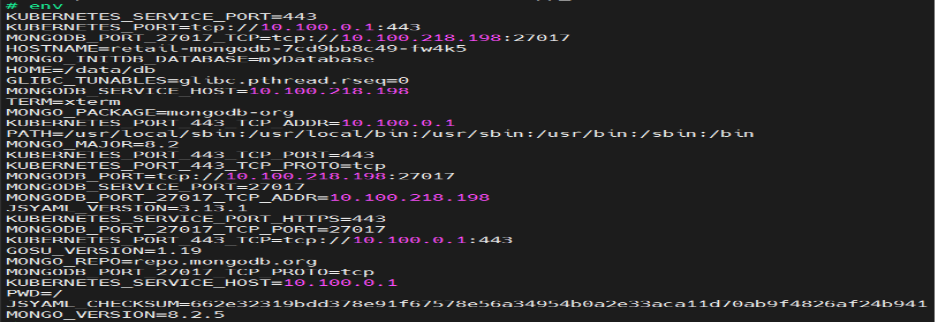


**Step 20:** Check the status of the application:



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

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



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

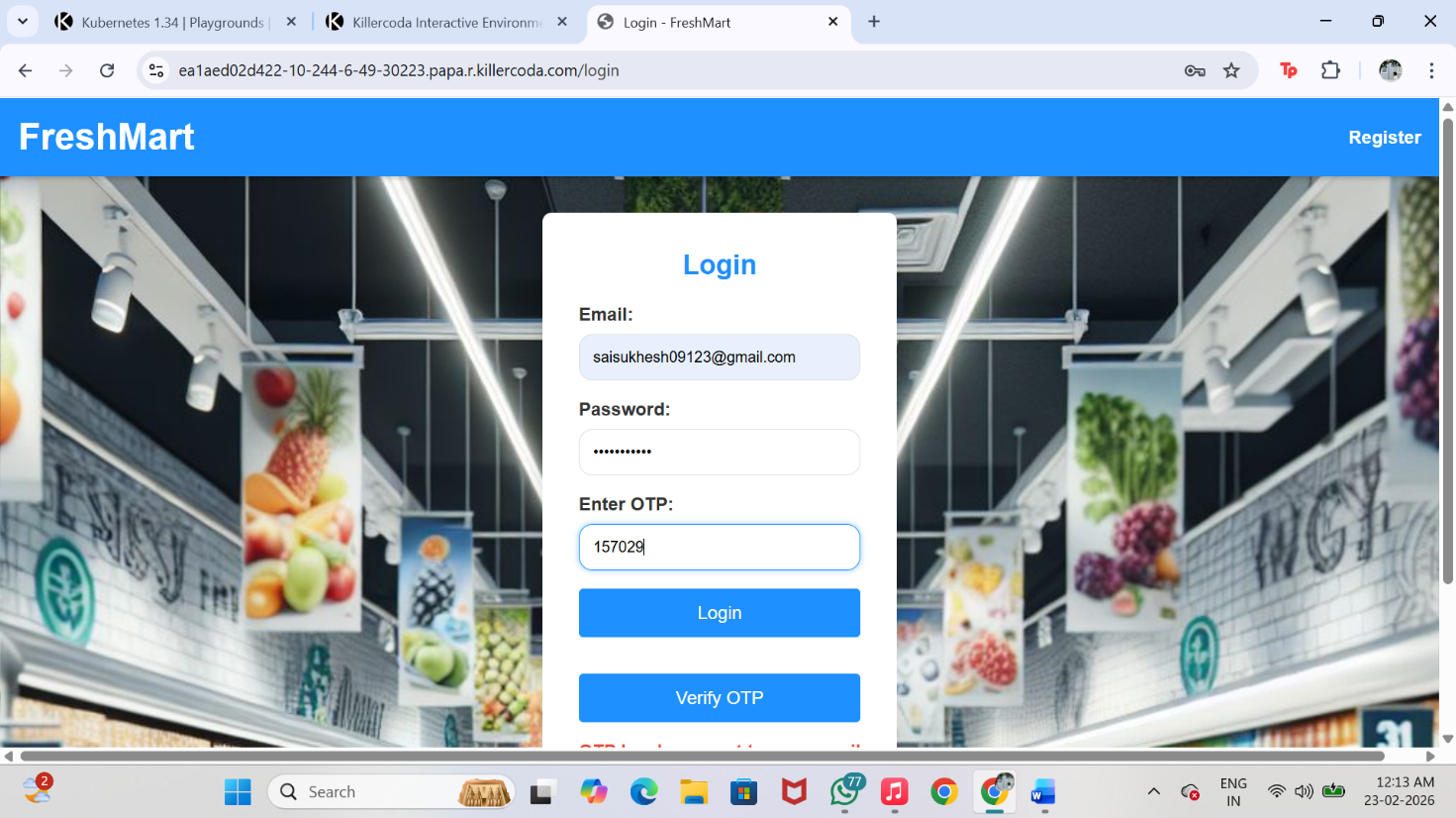
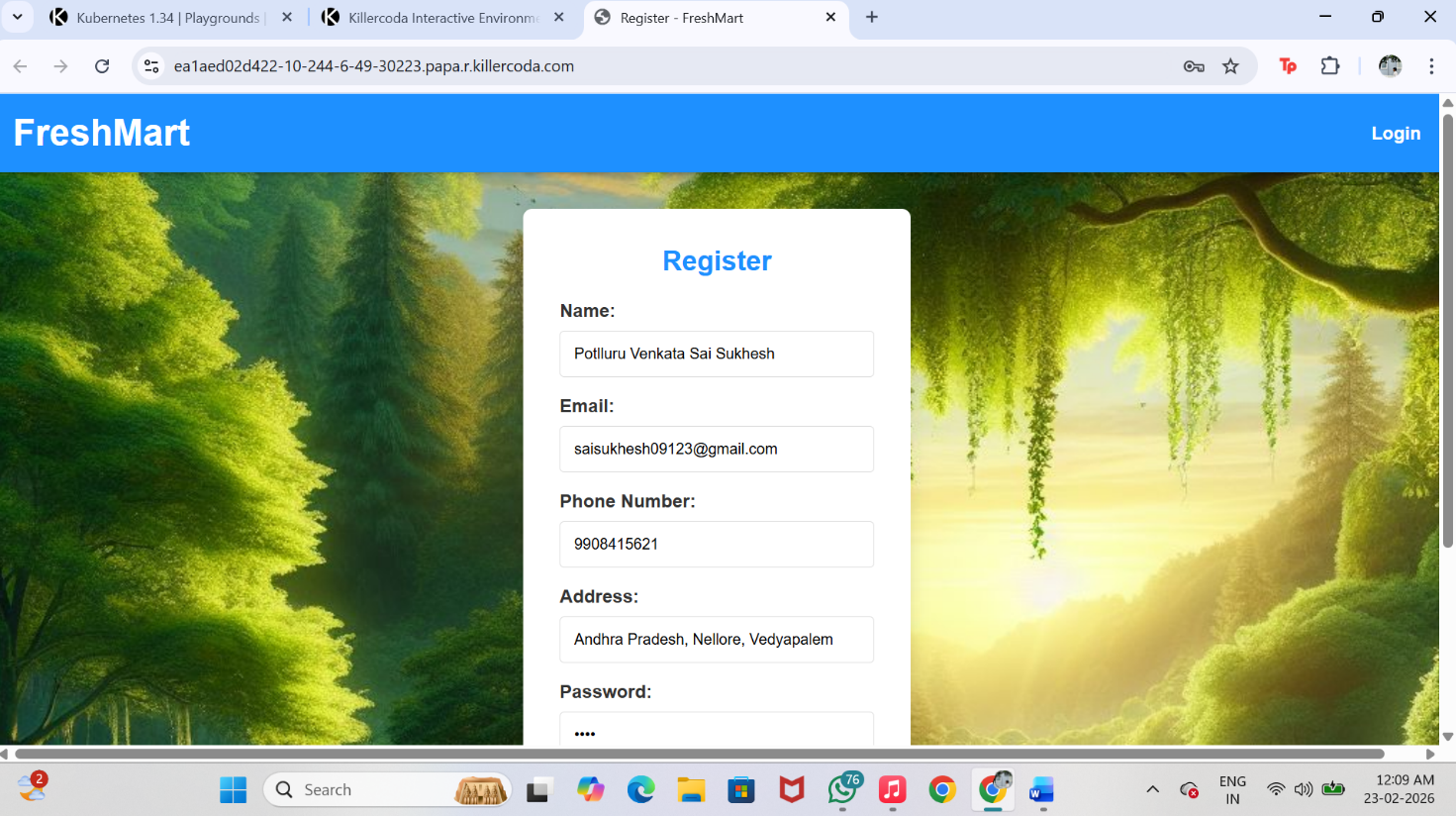
**Cmd:** **kubectl exec -it retail-mongodb-7cd9bb8c49-fw4k5 -n sukhesh-ns – mongosh**

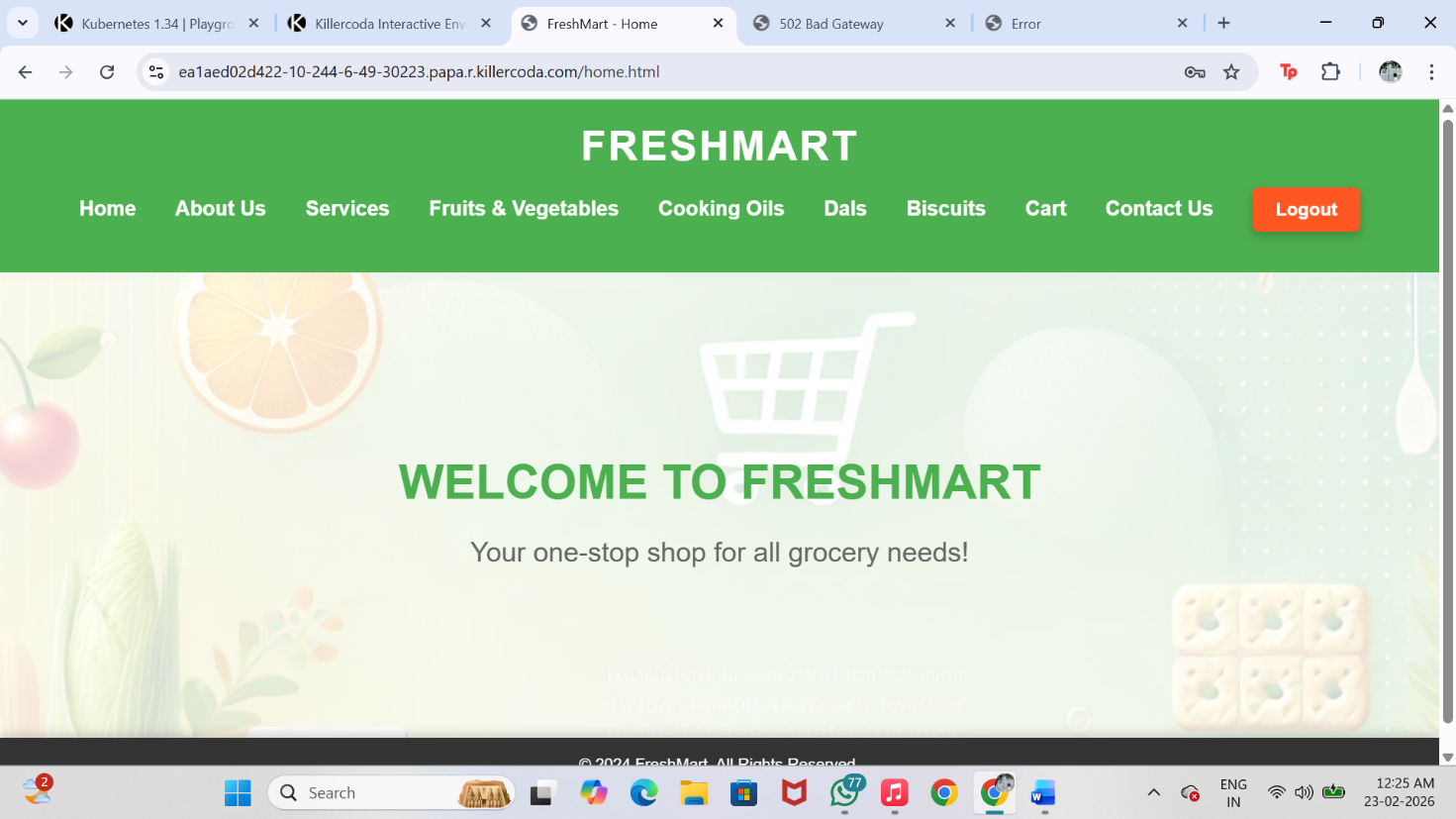
* To view, use the databases and shows all documents in the products collection.



**Step23 :** **Copy the External IP and paste it in browser:**

<http://a40f768764f534bef96e01cb855ab9b7-1867594736.ap-south-2.elb.amazonaws.com:3130/>

It will get us to the home page.

**Step 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

unknown.png

[chagantyteja2502@gmail.com](mailto:chagantyteja2502@gmail.com)

↓

**Y2hhZ2FudHl0ZWphMjUwMkBnbWFpbC5jb20=**

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

unknown.png

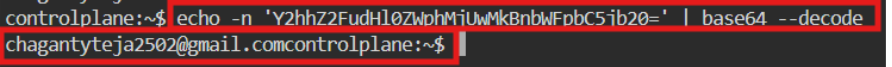
yxoq bjuk rdnt alzp

↓

**eXhvcSBianVrIHJkbnQgYWx6cA==**

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

==> decodes the input



Y2hhZ2FudHl0ZWphMjUwMkBnbWFpbC5jb20=

↓

**chagantyteja2502@gmail.com**

1. **Vi secret.yaml**

Create secret yaml file

unknown.png

apiVersion: v1

kind: Secret

metadata:

name: retail-app-secret

namespace: sukhesh-ns

type: Opaque

data:

EMAIL\_USER: c2Fpc3VraGVzaDA5MTIzQGdtYWlsLmNvbQ==

EMAIL\_PASS: U3VraGVzaEAxMjM=

 2)  **kubectl apply -f secret.yaml**

         Apply the file:

unknown.png

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

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

unknown.png