**ConfigMap**

In Kubernetes, a ConfigMap is an API object used to store configuration data that can be consumed by pods or other resources in the cluster.

It provides a way to decouple configuration settings from the containerized applications, making it easier to manage and update configuration data without modifying the application code or container images.

ConfigMaps **store key-value pairs** or provide the ability to mount configuration files as data.

This can include environment variables, command- line arguments, configuration files, or any other type of configuration data required by your application



**Secrets**

In Kubernetes, a Secret is an API object used to store sensitive information, such as passwords, tokens, or SSH keys.

Secrets provide a way to securely store and manage sensitive data within a cluster.

Secrets can be used by applications and pods to access sensitive information without exposing it in plain text.

They are typically used to store data that needs to be passed to containers securely, such as database credentials, API keys, or TLS certificates.

Kubernetes Secrets are stored in the cluster's etcd datastore, encrypted at rest. They can be accessed by pods or other Kubernetes objects securely.

Secrets can be mounted as files or exposed as environment variables within a container.

**There are different types of secrets in Kubernetes:**

**Opaque**: The most common type of Secret. It allows you to store arbitrary key-value pairs as base64-encoded strings. It is suitable for storing general-purpose sensitive information.

**Docker-registry**: Used for storing credentials to authenticate with a private Docker registry. It includes the server, username, password, and email fields.

**TLS**: Used for storing TLS certificates and private keys. It includes the tls.crt and tls.key fields.

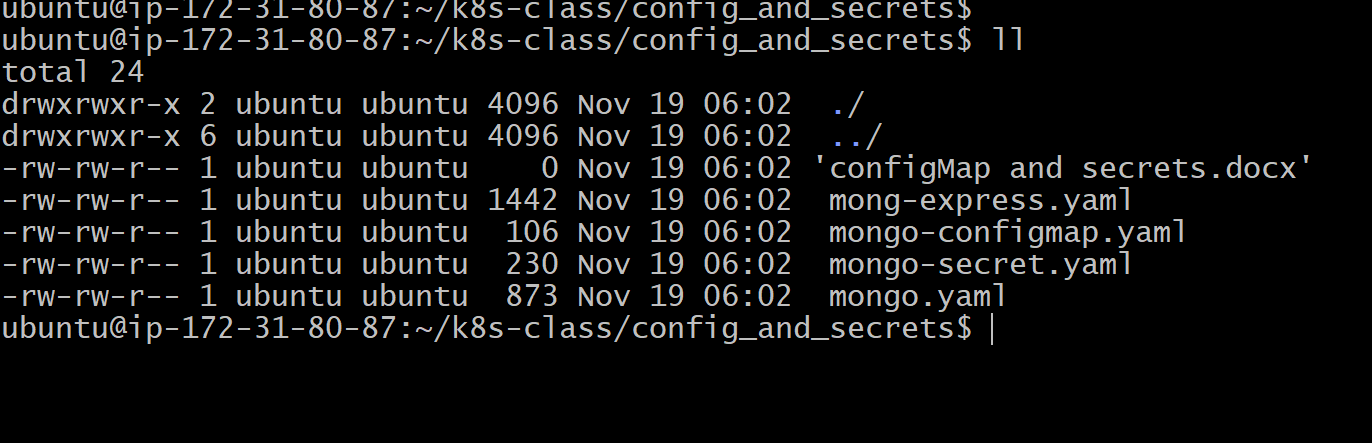
**Service Account**: Automatically created Secrets that provide credentials for accessing the Kubernetes API. They are associated with service accounts and allow pods to authenticate with the API server.

**Configmap and Secrets DEMO**

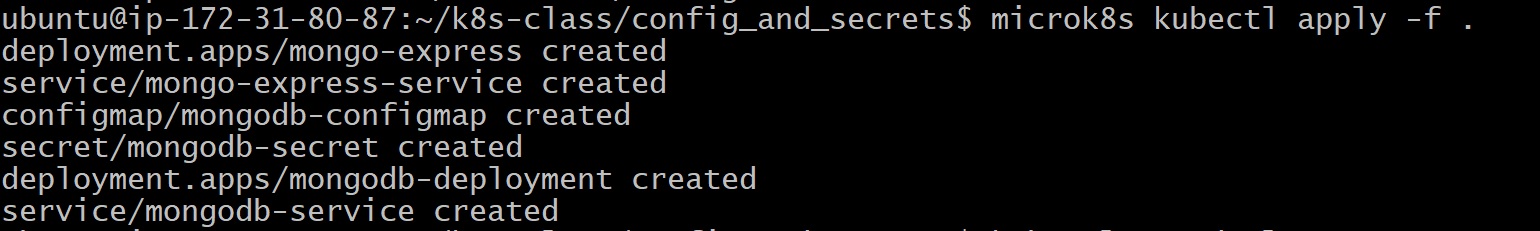
1.Install microk8s and do setup

Git clone repo

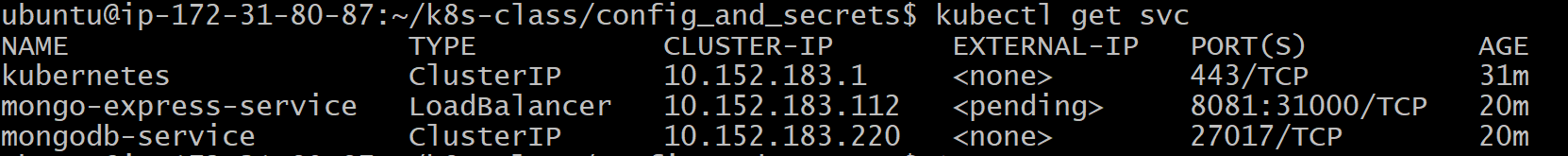
https://github.com/vickydevo/k8s-class.git



DEPLOY all yaml files



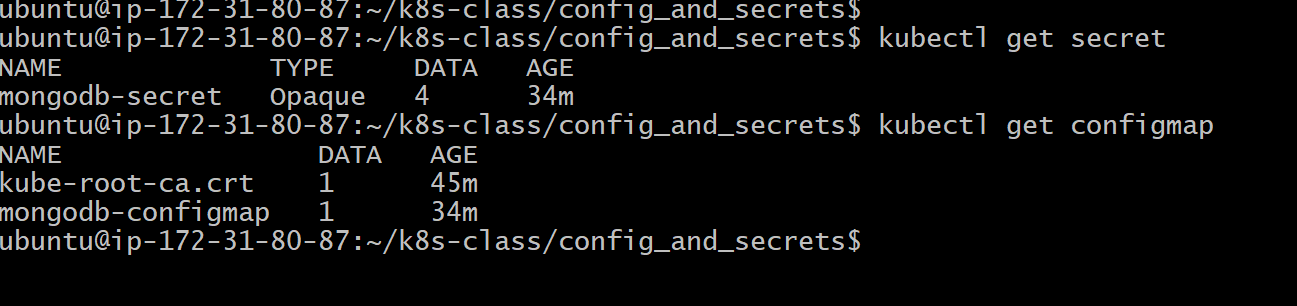
Check service



Check config and secrets

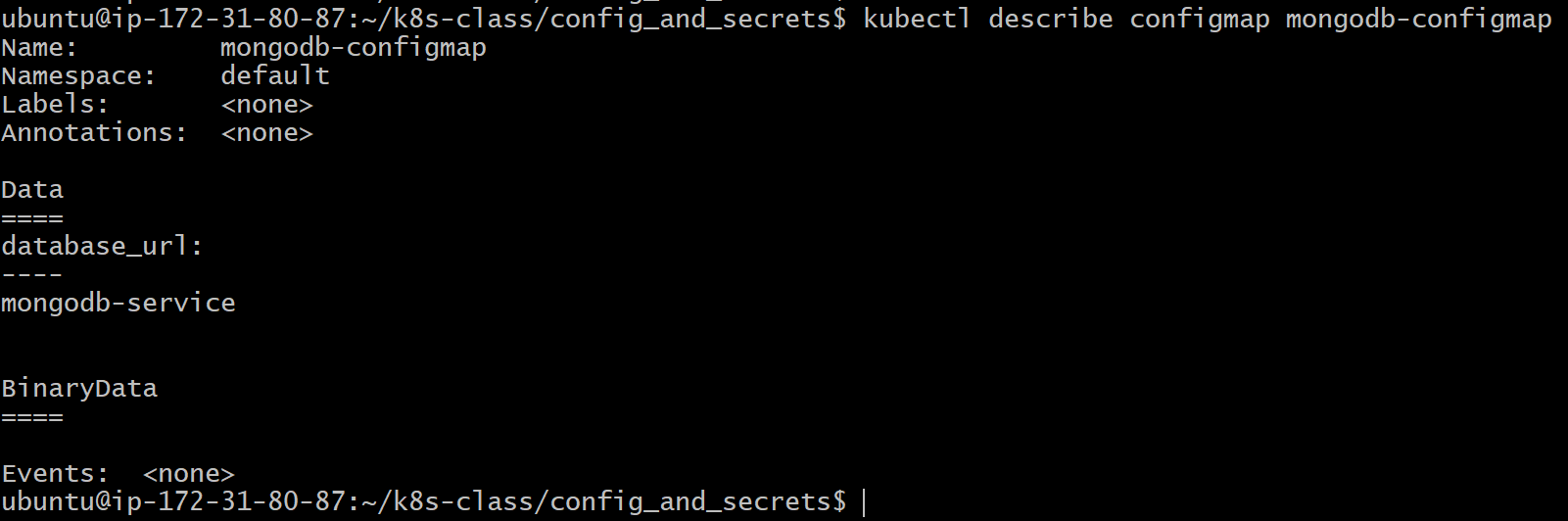
kubectl get secret

kubectl get configmap

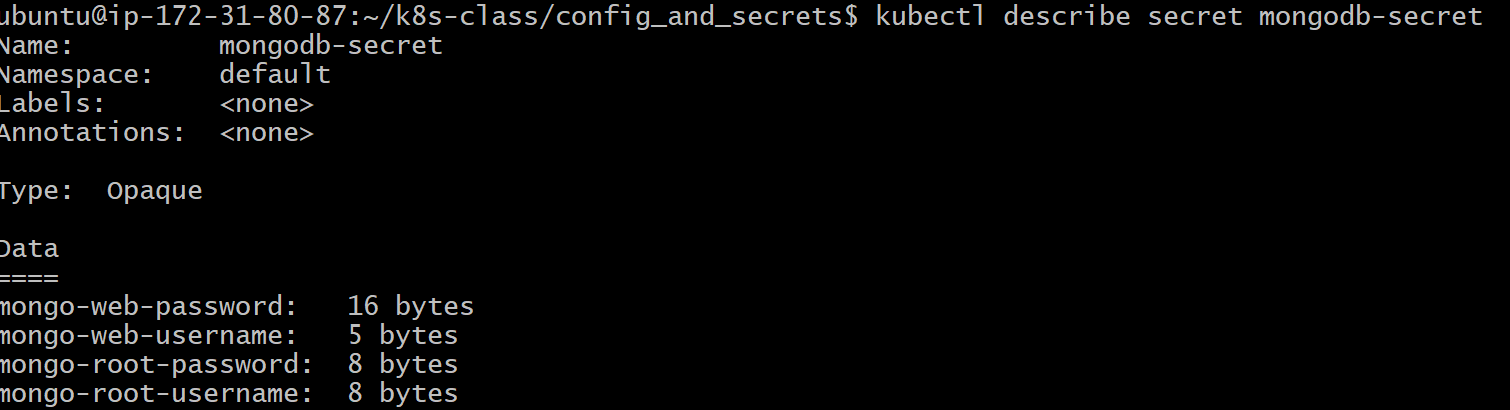


Describe config and secret

kubectl describe cm mongodb-configmap

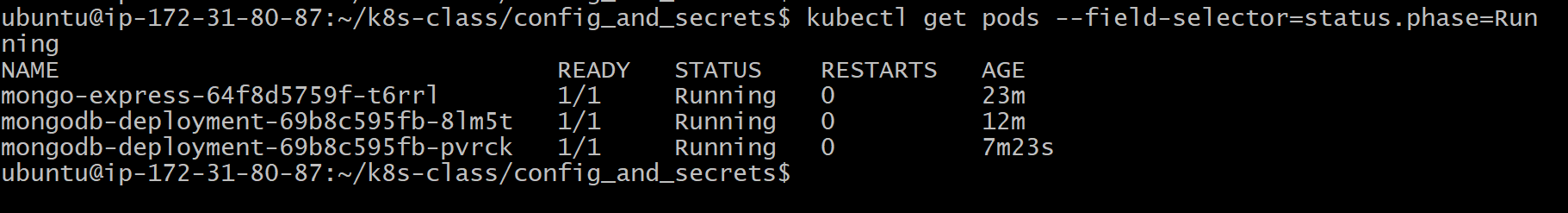


kubectl describe secret mongodb-secret



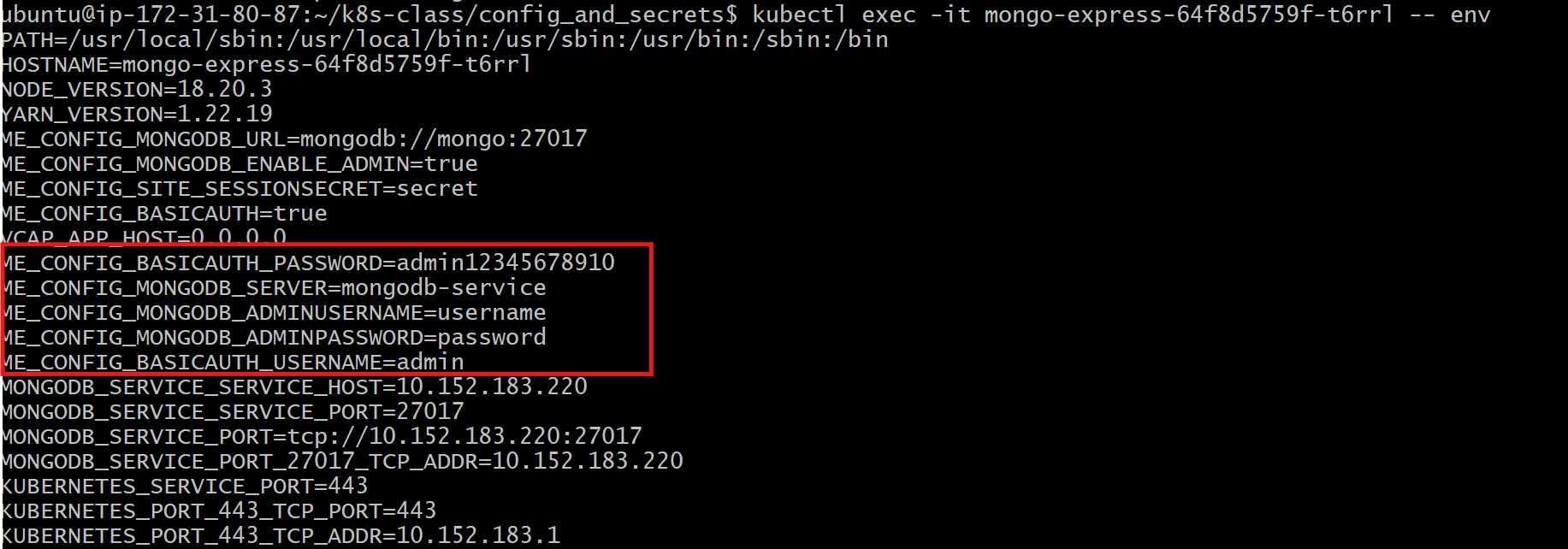
Check the pods running

kubectl get pods --field-selector=status.phase=Running



Connect to pods using kubectl exec to get the environmental variables

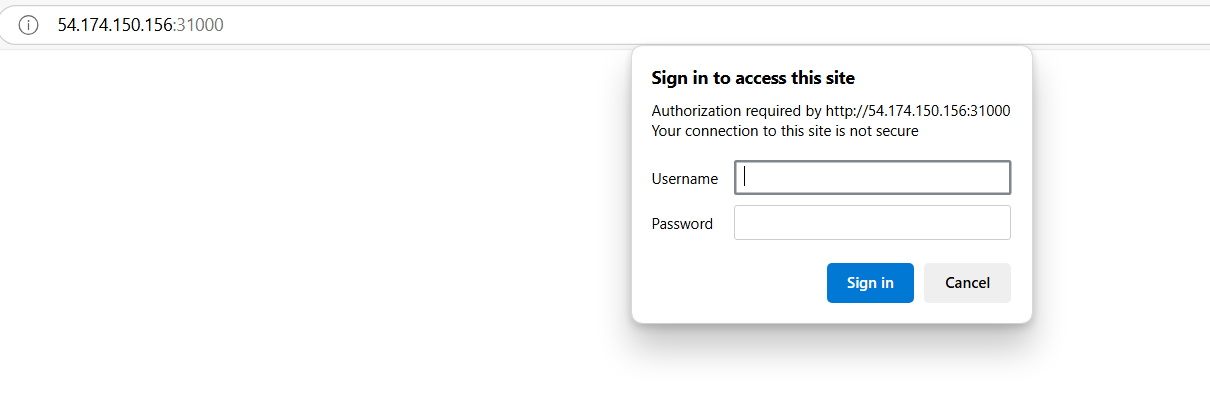
kubectl exec -it mongo-express-64f8d5759f-t6rrl -- env



Access the application from outer world (browser)

http://<public\_IP>:31000

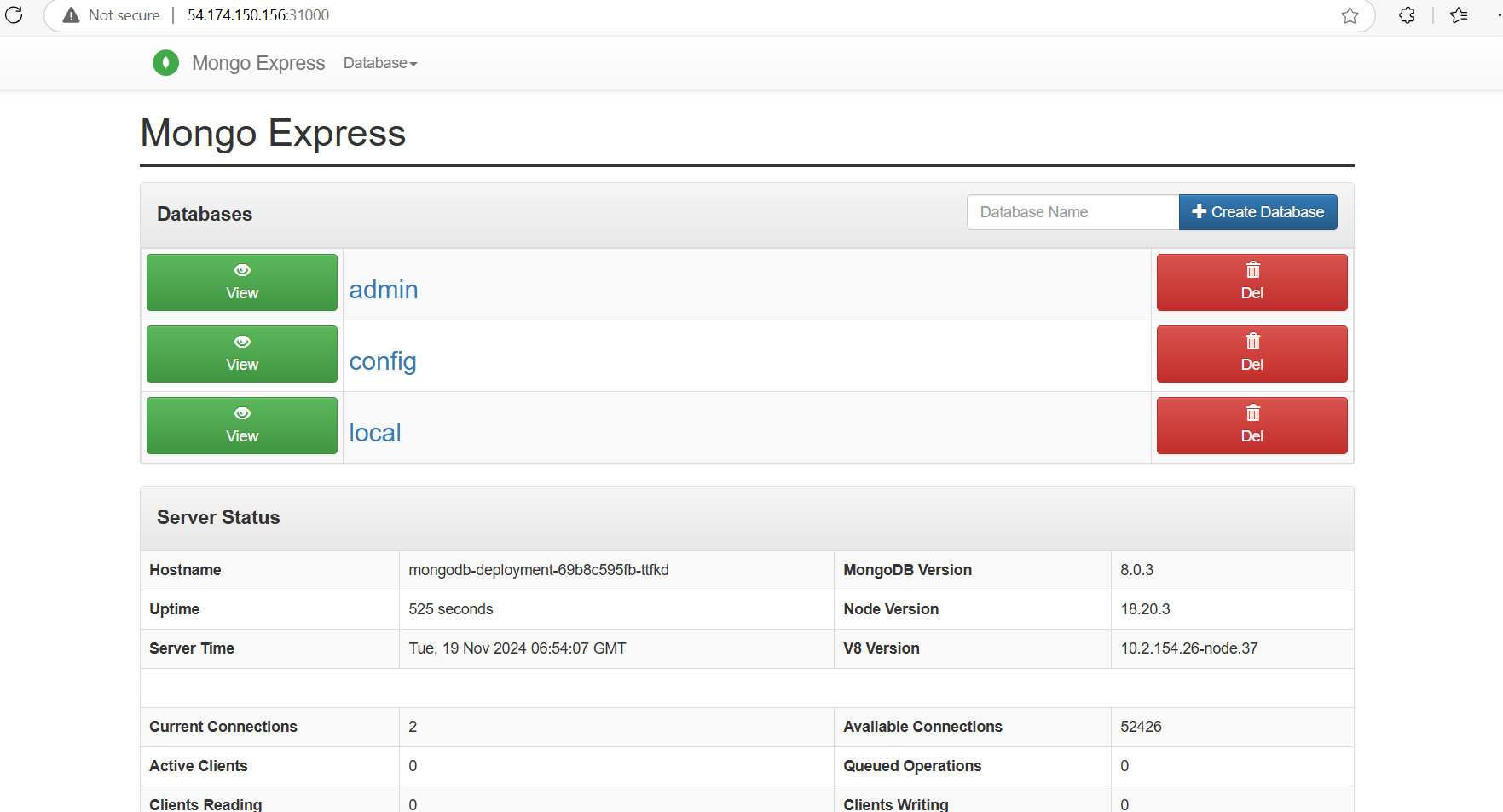
it prompt for username and password



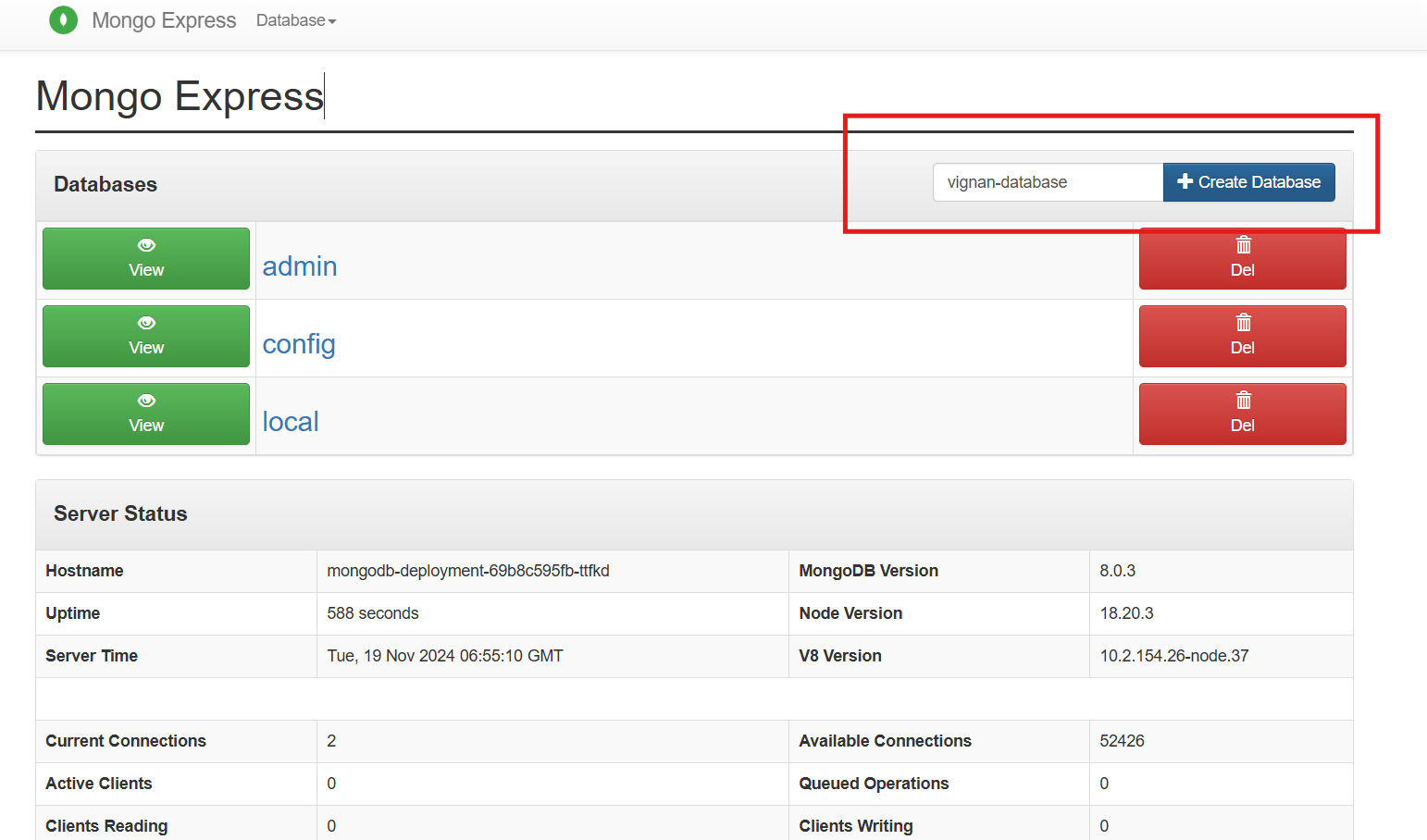
**Provide username and password by taking them from env of pod**

**THEN**

Mongo express application will be accessible



Add new database with your name from UI



New datadase will appear

